Embarcadero ER/Studio Software Architect 1.0

Copyright © 1994-2009 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.

ER/Studio Software Architect Workbench User Guide

Getting Started

Contents

Basic Tutorial

This tutorial provides a step by step walk-through of the Workbench.

The Workbench

When the Workbench is launched, the first thing you see is a dialog that allows you to select where the workspace should be located. The workspace is the directory where your work will be stored. For now, just click OK to pick the default location.

After the workspace location is chosen, a single Workbench window is displayed. A Workbench window offers one or more perspectives. A perspective contains editors and views, such as the Model Navigator. Multiple Workbench windows can be opened simultaneously. Click the arrow labeled **Workbench** in the Welcome view to cause the other views in the perspective to become visible. Note you can get the Welcome view back at any time by selecting **Help > Welcome**.

A shortcut bar appears in the top right corner of the window. This allows you to open new perspectives and switch between ones already open. The name of the active perspective is shown in the title of the window and its item in the shortcut bar is highlighted.

We will switch to the simpler Resource perspective to simplify this tutorial. Select **Window > Open Perspective > Other > Resource**. The Model Navigator, Outline, and Tasks views should now be visible.

Editors and Views

Prior to commencing the Workbench tutorials found in this section, it is important to first be familiar with the various elements of the Workbench. A Workbench consists of:

- perspectives
- views
- editors

A perspective is a group of views and editors in the Workbench window. One or more perspectives can exist in a single Workbench window. Each perspective contains one or more views and editors. Within a window, each perspective may have a different set of views but all perspectives share the same set of editors.

A view is a visual component within the Workbench. It is typically used to navigate a list or hierarchy of information (such as the resources in the Workbench), or display properties for the active editor. Modifications made in a view are saved immediately.

An editor is also a visual component within the Workbench. It is typically used to edit or browse a resource. The visual presentation might be text or a diagram. Typically, editors are launched by clicking on a resource in a view.

Some features are common to both views and editors. We use the term "part" to mean either a view or an editor. Parts can be active or inactive, but only one part can be active at any one time. The active part is the one whose title bar is highlighted. The active part is the target for common operations like cut, copy and paste. The active part also determines the contents of the status line. If an editor tab is not highlighted it indicates the editor is not active, however views may show information based on the last active editor.

Editors

Depending on the type of file that is being edited, the appropriate editor is displayed in the editor area. For example, if a .TXT file is being edited, a text editor is displayed in the editor area. The figure below shows an editor open on the file file1.txt. The name of the file appears in the tab of the editor. An asterisk (*) appearing at the left side of the tab indicates that the editor has unsaved changes. If an attempt is made to close the editor or exit the Workbench with unsaved changes, a prompt to save the editor's changes will appear.

When an editor is active, the Workbench menu bar and toolbar contain operations applicable to the editor. When a view becomes active, the editor operations are disabled. However, certain operations may be appropriate in the context of a view and will remain enabled.

The editors can be stacked in the editor area and individual editors can be activated by clicking the tab for the editor. Editors can also be tiled side-by-side in the editor area so their content can be viewed simultaneously. In the figure below, editors for JanesFile.txt and JanesFile2.txt have been placed above the editor for JanesText.txt. Instructions will be given later in this tutorial explaining how to rearrange views and editors.

If a resource does not have an associated editor, the Workbench will attempt to launch an external editor registered with the platform. These external editors are not tightly integrated with the Workbench and are not embedded in the Workbench's editor area.

Editors can be cycled through using the back and forward arrow buttons in the toolbar. These move through the last mouse selection points and permit moving through several points in a file before moving to another one. Additionally, editors can be cycled by using the Ctrl+F6 accelerator (Command+F6 on the Macintosh). Ctrl+F6 pops up a list of currently open editors. By default, the list will have selected the editor used before the current one, allowing you to easily go back to the previous editor.

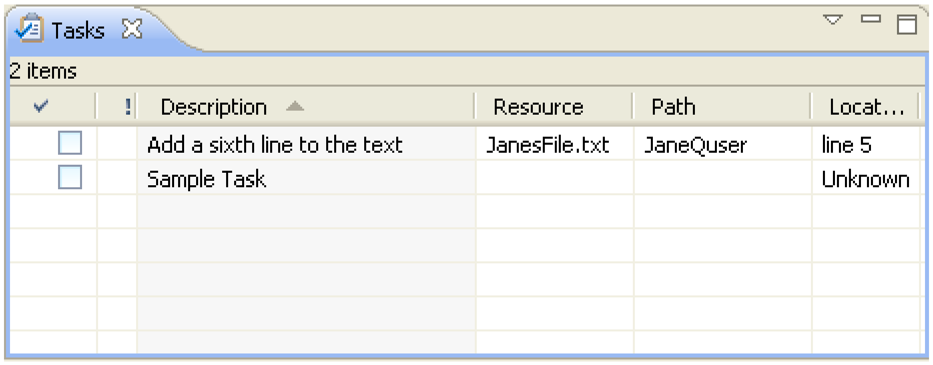
On Windows, if the associated editor is an external editor, the Workbench may attempt to launch the editor in-place as an OLE document editor. For example, editing a DOC file will cause Microsoft Word to be opened in-place within the Workbench if Microsoft Word is installed on the machine. If Microsoft Word has not been installed, Word Pad will open instead.

Views

The primary use of Views is to provide navigation of the information in the Workbench. For example:

- The Bookmarks view displays all bookmarks in the Workbench along with the names of the files with which the bookmarks are associated.
- The Model Navigator view displays the Workbench projects, their folders and files.

A view might appear by itself or stacked with other views in a tabbed notebook.

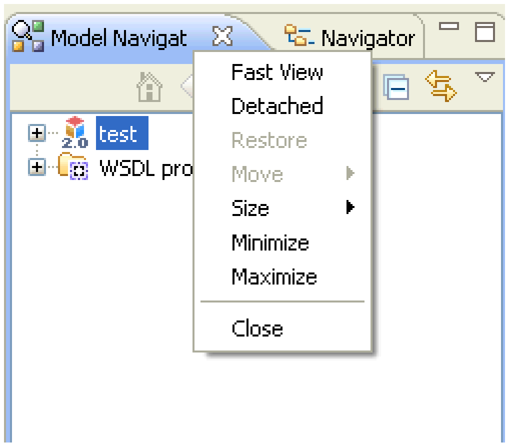


2 items

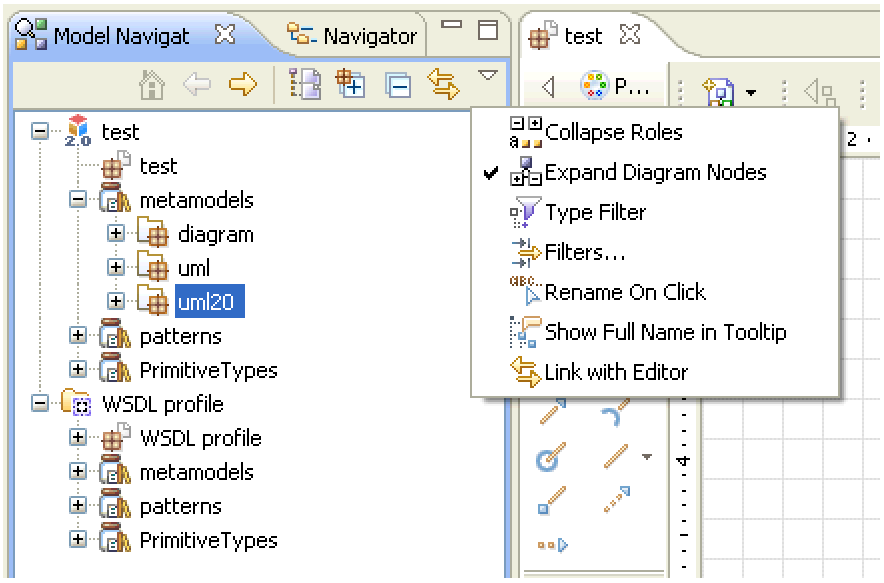
✓	!	Description	Resource	Path	Locat...
☐		Add a sixth line to the text	JanesFile.txt	JaneQuser	line 5
☐		Sample Task			Unknown

To activate a view that is part of a tabbed notebook simply click its tab.

Views have two menus. The first, which is accessed by right clicking on the view's tab, allows the view to be manipulated in much the same manner as the menu associated with the Workbench window.



The second menu, called the "view pull-down menu", is accessed by clicking the down arrow. The view pull-down menu typically contains operations that apply to the entire contents of the view, but not to a specific item shown in the view. Operations for sorting and filtering are commonly found in the view pull-down.



A view can be displayed by selecting it from the **Window > Show View** menu. A perspective determines which views may be required and displays these on the **Show View** sub-menu. Additional views are available by choosing **Other** at the bottom of the **Show View** sub-menu. This is just one of the many features that provide for the creation of a custom work environment.

Through the normal course of using the Workbench you will open, move, resize, and close views. If you'd like to restore the perspective back to its original state, you can select the **Window > Reset Perspective** menu operation.

Closing An Editor

Now that there are a couple of editors open, here's how to close them.

1. Select the JanesFile.txt editor tab.
2. In the text area add a 6th line of text:

   This is a 6th line
3. To close the editor, choose one of the following options:
   - Click the close button ("X") in the tab of the editor.
   - Select **File > Close** from the menu bar.
4. Note the prompt to save the file before the editor is closed.
5. Click OK to save any changes and close the editor.

If the editor was closed using **File > Close**, notice that the option **File > Close All** was also displayed. This is a quick way to close all of the open editors. If **File > Close All** is chosen, a prompt will appear to choose which editors with unsaved changes should be saved.

The preference to close editors automatically can be found in the **General > Editors** preference page. There you can configure the number of editors that can be opened prior to editors being reused and what should occur when all editors have unsaved changes.

Navigating Resources

This section will work with the Model Navigator and Tasks views. These views are initially part of the resource perspective. To experiment with other views, they can be displayed by using the **Window > Show View** menu.

One important view to become familiar with is one of the navigation views, which displays information about the contents of the Workbench and how the resources relate to each other in a hierarchy.

In the Workbench, all resources reside in projects. Projects can contain folders and/or individual files.

Go To

The Go To operation makes it easy to jump to a specific resource in the navigation views.

1. Select one of the navigation views. Its title bar will be highlighted (according to the operating system's color scheme).
2. From the menu bar choose **Navigate > Go To > Resource...**.
3. In the Go To Resource dialog type "JanesF" into the pattern field at the top of the dialog.

   As the filename is typed, the dialog filters the set of possible matches based on what has been entered so far.
4. Select JanesFile.txt from the **Matching items** field and click OK or simply press Enter.

The navigation view will select the file JanesFile.txt.

Files

The projects, folders and files that you create with the Workbench are all stored under a single directory that represents your workspace. The location of the workspace was set in the dialog that first opens when you start the Workbench.

If you have forgotten where that location is, you can find it by selecting **File > Switch Workspace...**. The workspace directory will be displayed in the dialog that appears. **IMPORTANT:** After recording this location, hit **Cancel** to close the dialog, or the Workbench will exit and re-open on whatever workspace was selected.

All of the projects, folders and files that you create with the Workbench are stored as normal directories and files on the machine. This allows the use of other tools when working with the files. Those tools can be completely oblivious to the Workbench. A later section will look at how to work with external editors that are not integrated into the Workbench.

Exporting Files

Files can be exported from the Workbench either by:

- Dragging and dropping to the file system (Windows and Linux GTK only), or
- Copying and pasting to the file system, or
- Using the Export wizard.

Drag and drop or copy and paste

The operating system's file system explorer can be used to export a copy of a folder or file from the Workbench to the file system.

1. Open the operating system's file system explorer.
2. Drag the file JanesFile.txt from one of the navigation view to the file system explorer.

3 Depending on where you are dragging to, you may need to hold down the Ctrl or Shift key while dragging to ensure the file is copied. Look for a small plus sign on the drag cursor to know whether the file is being copied or moved.

4 The export can also be achieved by selecting the file in the Model Navigator and choosing **Edit > Copy**, then pasting it in the file system explorer.

Export wizard

The Export wizard can be used to export from the Workbench to the file system.

1 Select the project JaneQuser in the navigation view.

2 From the popup menu, select **Export**.

3 In the Export wizard, select **File system**, then click **Next**.

4 Expand JaneQuser project, and click on JanesFolder. In the right pane ensure that only JanesINIFile.ini is selected. Notice the folder and project in the left pane now have a grayed checkbox indicating that some, but not all, of their contents will be exported.

The **Select Types** button can be used to filter the types of resources to export.

NOTE: To export JanesINIFile.ini only, simply select it in the navigation view and choose **File > Export**. The Export wizard will automatically ensure it is the only file selected for export.

5 In the **To directory** field, type or browse to select a location in the file system for the exported resources to reside.

If the name of a directory that does not exist is entered the Export wizard will offer to create it once **Finish** is selected.

6 In the **Options** area, options are given to:

- Overwrite existing resources without warning
- Create directory structure for files or Create only selected directories

7 Click **Finish** when done.Importing Files

Importing Files

Files can be imported into the Workbench either by:

- dragging and dropping from the file system, or
- copying and pasting from the file system, or
- Using the Import wizard

Using drag and drop or copy/paste to import files relies on operating system support that is not necessarily available on all platforms. If the platform you are using does not have this support, you can always use the Import wizard.

In this section the two files that were just exported will be imported and placed into the second project JaneQuser2.

Drag and Drop or Copy and Paste

On some platforms, for example on the Macintosh and Microsoft Windows, the operating system's file system browser can be used to copy folders and files from the file system into the Workbench.

NOTE: The resource(s) must be dragged to the exact location in the hierarchy of one of the navigation views where the resources are to reside; they cannot be simply dragged and dropped onto a blank area in the navigation view.

1 Open the operating system's file system explorer.

2 Locate the file JanesFile.txt that was recently exported and drag it to a specific location in one of the navigation views in the Workbench.

When dragging resources into one of the navigation views, the project/folder that the resource is being dropped into will be selected.

Drag the file over JaneQuser2 and release the mouse button.

3 Notice that the file is copied into the Workbench and placed into JaneQuser2.

4 This can also be achieved by copying the file in the file system explorer, then selecting the destination in the navigation view and choosing **Edit > Paste**.

Import Wizard

The Import wizard can be used to copy resources into the Workbench.

1 Select the project JaneQuser.

2 Select **Import** from the popup.

3 In the Import wizard, select **File system**, then click **Next**.

4 In the **From directory** field, type or browse to select the directory containing the file JanesINIFile.ini that was recently exported.

Recent directories that have been imported from are shown on the **From directory** field's combo box.

5 In the right pane check the file JanesINIFile.ini

Checking a folder in the left pane will import its entire contents into the Workbench. A grayed checkbox (as shown below) indicates that only some of the files in the folder will be imported into the Workbench.

The **Filter Types** button can be used to filter the types of files that will be imported.

6 The **Into folder** field should already be filled in with the name of the project (JaneQuser).

7 The destination project or folder can be changed by clicking **Browse**;

Click the **Browse** button and choose the second project JaneQuser2

8 In the **Options** area, options are given to:

- Overwrite existing resources without warning
- Create complete folder structure or Create selected folders only

9 Click **Finish** when done. The file JaneINIFile.ini is now shown in the one of the navigation views in the project JaneQuser2.

Deleting Resources

Now that a few files have been imported into the second project (JaneQuser2), here are instructions on how to delete the project.

1. Select project JaneQuser2 in one of the navigation views.
2. To delete the project do one of the following:
   - From the project's pop-up menu choose **Delete**
   - Press the DEL key
   - Choose **Edit > Delete** from the pull-down menu
3. A prompt will ask for confirmation of the deletion and also what type of deletion should be performed.

   It is possible to:

   - Delete the contents of the project from the file system
   - Delete the project from the workspace but keep its contents in the file system
   - Accept the default (do not delete contents) and click **Yes**.

   NOTE: This is only an option for projects. Files and folders are always deleted from the file system when deleted.

The same steps work for any resource shown in the navigation view.

Working with Other Editors

Instructions have been given explaining how to import and export resources from the Workbench. This section will look at how to edit Workbench resources using the following three approaches:

- External editors launched by the Workbench
- Embedded OLE documents
- External editors launched without the Workbench's knowledge

Before continuing, take a moment and confirm that the Model Navigator contains the following resources:

External Editors

When opening a resource the Workbench first consults its list of registered editors. If no registered editors are found for the resource the Workbench checks with the underlying operating system to determine if it has any editors registered for the particular file type. If an editor is located, the Workbench will automatically launch that editor. This type of editor is referred to as an external editor because it does not show up as an editor tab in the Workbench.

1. Select the file JanesINIFile.ini.
2. Double-click the file in one of the navigation views to launch the external editor.

   If an editor for INI files is not registered with the underlying operating system the Workbench will attempt to use its own default text editor. If this happens, to see an external editor, import another file (see previous sections) that is associated with a third party editor. Double click again on this new file and the selected editor will open in its own window.

The Workbench supports OLE document editors, and some editors provide OLE document support allowing it to be opened in its own window, or embedded inside another window like the Workbench. This will be discussed in more detail in the next section.

Embedded Editors

The Workbench supports OLE document editors.

1. Select JanesFolder in the navigation view.
2. Create the file Jane.doc.
3. Notice how the Workbench automatically opens the OLE document editor in place and integrates its pull-down menu options into the menu bar. There should also be an editor tab for Jane.doc.
4. Make a change to the file.
5. Close the editor by clicking the X on the editor tab; when prompted, save its contents.
6. Reopen the file by double-clicking it in the navigation view.

Editing Files Outside the Workbench

In a previous section an external editor was launched on the file JanesINIFile.ini by double clicking on it in the Model Navigator. The same external editor can also be used by launching it outside of the Workbench.

1. Close any editors that are open on JanesINIFile.ini.
2. In the file system explorer, navigate to the directory where the Workbench was installed and go into the workspace sub-directory.
3. Edit the file JanesINIFile.ini and save it. Do not use the Workbench's Model Navigator to open the editor.
4. Return to the Workbench and in the Model Navigator view, select the project JaneQuser.
5. Choose **Refresh** from the project's context menu. This instructs the Workbench to look for any changes to the project that have been made in the local file system by external tools.
6. Select the file JanesINIFile.ini.
7. For a little variety choose **Open With > Text Editor** from the file's popup menu.
8. Observe that the Workbench's own default text editor is opened.
9. In the default text editor verify that the externally made changes are reflected in the Workbench.

The Workbench stores all of its resources in the local file system. This means the system file explorer can be used to copy files from the Workbench's workspace area even when the Workbench is not running. Resources can also be copied into the workspace directory. Use Refresh to update the Workbench with any changes made outside the Workbench.

Copying, Renaming and Moving

Workbench resources can be copied, moved and renamed using popup menu operations in one of the navigation views. In this section several of the files that have been created will be copied and renamed.

Prior to copying files, some setup is required:

Setup

1. In one of the navigation views, delete the file JanesWordDoc.doc. The navigation view should look like this:
2. Double click on JanesFile.txt and ensure that it contains the following text.

   This is a sample text file
   There is not much else
   we can really say about it other
   than it has five lines of
   text that are rather dull.

3. Close the editor on JanesFile.txt.
4. Select the project JaneQuser. Using the project's pop-up menu create a folder named JanesOtherFolder.

Copying

You can copy JanesFile.txt to the new folder (JanesOtherFolder) using the following steps.

1. Ensure that the setup described in the introduction to this section has been performed.
2. In one of the navigation views, select JanesFile.txt.
3. From the file's context menu, select **Copy** (or Ctrl+C)
4. In one of the navigation views, select JanesOtherFolder as the destination.
5. From the folder's context menu, select **Paste** (or Ctrl+V).

As an alternative to copying files using the copy operation, it is also possible to copy files by holding down the Ctrl key while dragging a file from one folder to another folder. (On the Macintosh, the Option key is used for this.)

Once the file has been copied it can be renamed.

Renaming

Now that JanesFile.txt has been copied from JanesFolder to JanesOtherFolder it is ready to be renamed as something else.

1. In one of the navigation views, select JanesFile.txt in JanesOtherFolder.
2. From the file's context menu, select **Rename**.

The navigation view overlays the file's name with a text field. Type in JanesText.txt and press Enter.

To halt the renaming of a resource, Escape can be pressed to dismiss the text field.

Copy and rename works on folders as well.

1. In one of the navigation views, select the folder JanesOtherFolder.
2. From the folder's context menu choose **Rename**.
3. Once again the navigation view overlays the folder name with an entry field to allow the typing in of a new name. Change the folder name to be JanesSecondFolder.
4. Rename the folder back to its original name (JanesOtherFolder).

Moving

Having copied and renamed several of the resources, now it's time to move some resources around. JanesOtherFolder and its file will be moved to be a sub-folder of the original folder JanesFolder.

1 In the Model Navigator view, select JanesOtherFolder.

2 From the file's context menu, select **Move**.

3 In the Folder Selection dialog choose JanesFolder and click **OK**.

Searching

Text strings and files can be searched for in the Workbench. In this section, **Search** will be used to perform a text search across the resources that are shown in the navigation view. Instruction will also be given on how to use the Search view to work with the results.

Starting a Search

Text strings can be searched for in the Workbench as follows:

1 In the Workbench toolbar, click the search button.

2 In the **Containing text** field, type in: *it*

The combo box for the **Containing text** field also displays a list of recently performed searches to select from.

3 The **Case sensitive** checkbox can be selected or deselected depending on whether or not a case sensitive or insensitive search is to be performed. You can also select the **Regular expression** checkbox to enable more powerful searching capabilities. To see what is available in regular expression mode, you can hit Ctrl-Space over the text field to get content assistance that lists the possibilities. For this example, just check the **Case sensitive** box to search for lowercase "*it*".

4 The kinds of files to include in the search can be specified in the **File name patterns** field. Click **Choose...** to open the Select Types dialog. This dialog provides a quick way to select from a list of registered extensions.

For the moment, the search will be confined to .txt files. Ensure *.txt is checked and click **OK**.

5 In the **Scope** field, specify the files and folders to include in the search. The choices are: the entire workspace, the currently selected resources in the Workbench, or a working set which is a named, customized group of files and folders. Leave the scope as workspace.

6 Use the **Customize** button to choose what kinds of searches are available in the dialog. This setting may be left unchanged.

7 Click **Search**. At this point, the Search view will be made visible, and it will begin to fill in with the results of the search. The stop button in the Search view can be clicked to cancel the search while it is in progress.

8 Observe that the Search view displays:

The next section will describe how to work with the Search view.

The Search View

Now that the search for "it" has been completed, the Search view is visible. The title of the Search view shows that four matches were found.

Within the Search view two files are shown and within each file there were 2 matches found.

1 Click the Show Next Match button to navigate to the first match of the search expression ("it").

  Notice that the file JanesFile.txt is automatically selected and opened in the editor area.
  Click Show Next Match button two more times. Once again the Search view automatically opens the file (JanesText.txt).

2 It is sometimes useful to remove uninteresting matches from the search results. The Search view's popup menu allows you to do this using **Remove Selected Matches** which removes any selected file entries (and all matches in them) from the Search view. Note that this **only** removes the entries in the Search view, it does **not** effect the files themselves. Select JanesFile.txt and choose **Remove Selected Matches** from the popup menu. The Search view now shows only the matches for JanesText.txt

3 Perform a second search for "that" by clicking on the Search button in the Workbench's toolbar. The Search view updates to show the results of the new search.

4 Use the drop down button on the Search view's toolbar to move back and forth between the two search results.

5 In the drop down button choose **'it' - 1 match in workspace**. The Search view switches back to show the original search. On the context menu choose **Search Again** to repeat the initial search. Notice that once again there are four matches.

So far you have seen how to manage your search results and how to switch between different searches. However, it might happen that you do not want the search view to change even if further searches are performed. For this you can **pin** the search view, which causes subsequent searches to be shown in a second Search view.

Tasks and Markers

There are various types of markers including bookmarks, task markers, debugging breakpoints and problems. This section will focus on tasks and the Tasks view.

The Tasks view displays all the tasks in the Workbench. The view displays tasks associated with specific files, specific lines in specific files, as well as generic tasks that are not associated with any specific file.

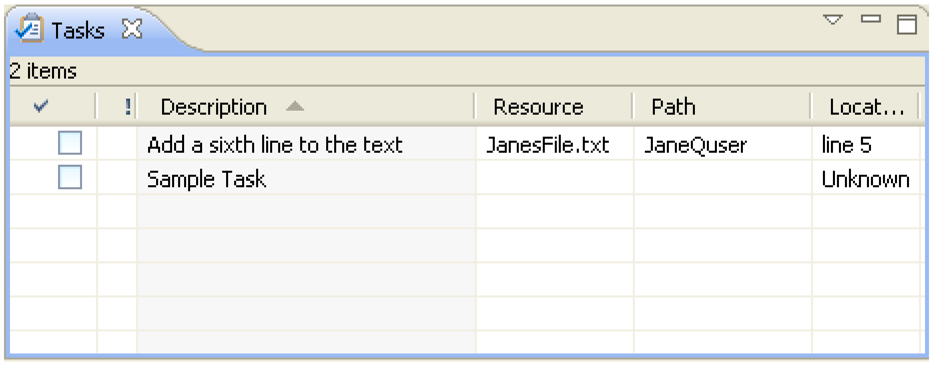
In the figure below "sample task" is a generic task not associated with any specific resource. The second task ("add sixth line to the text") is associated with the file JanesFile.txt.

Unassociated Tasks

Unassociated tasks are not associated with any specific resource. To create an unassociated task:

1 In the Tasks view, click the New Task button. A new task dialog appears.

2 Type a brief description for the task and press Enter. To cancel the dialog while entering the description press Escape. The new task appears in the Tasks view.



✓	!	Description	Resource	Path	Locat...
☐		Add a sixth line to the text	JanesFile.txt	JaneQuser	line 5
☐		Sample Task			Unknown

Associated Tasks

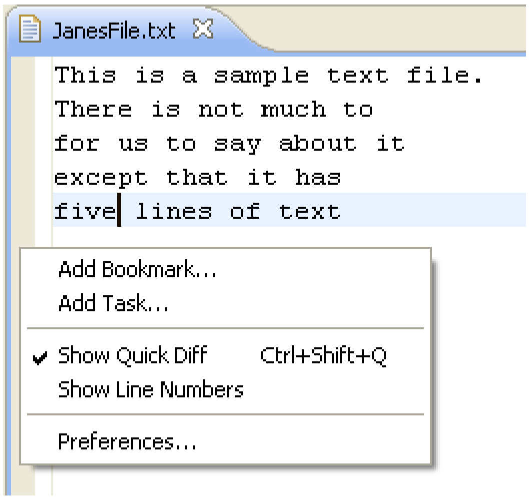
Associated tasks are associated with a specific location in a resource. To associate a task with the JanesFile.txt:

1 Open a text file (JanesFile.txt) by double clicking on it in one of the navigation views.

2 In the editor area directly to the left of any line in the text editor, access the context menu from the marker bar. The marker bar is the vertical bar to the left of the main text area.

3 From the marker bar's context menu, select **Add Task**.

The marker bar displays any marker including bookmarks, task markers (for associated tasks), and/or debugging breakpoints. Various markers can be associated with specific lines in a file by accessing the context menu from the marker bar directly to the left of that line.



```
This is a sample text file.
There is not much to
for us to say about it
except that it has
five lines of text
```

4 In the **Description** field, type a brief description for the task that will be associated with that line in the text file.

5 Click **OK** when done.

6 Notice that a new task marker appears in the marker bar, directly to the left of the line where the task was added. Also, notice that the new task appears in the Tasks view.

7 After the new task has been added, click in the editor on the first line or any other line above the line with which the new task is associated.

8 Add several lines of text to the file at this point.

9. Notice that as lines of text are added above it, the task marker moves down in the marker bar in order to remain with the associated line in the file. The line number in the Tasks view is updated when the file is saved.
10. In the Tasks view, access the context menu for the task that was just created.
11. Select **Mark Completed**.
12. Now select **Delete Completed Tasks** from the marker's context menu.
13. Notice that the task marker disappears from the marker bar and the task is removed from the Tasks view.

Opening Files

The Tasks view provides two approaches for opening the file associated with a task:

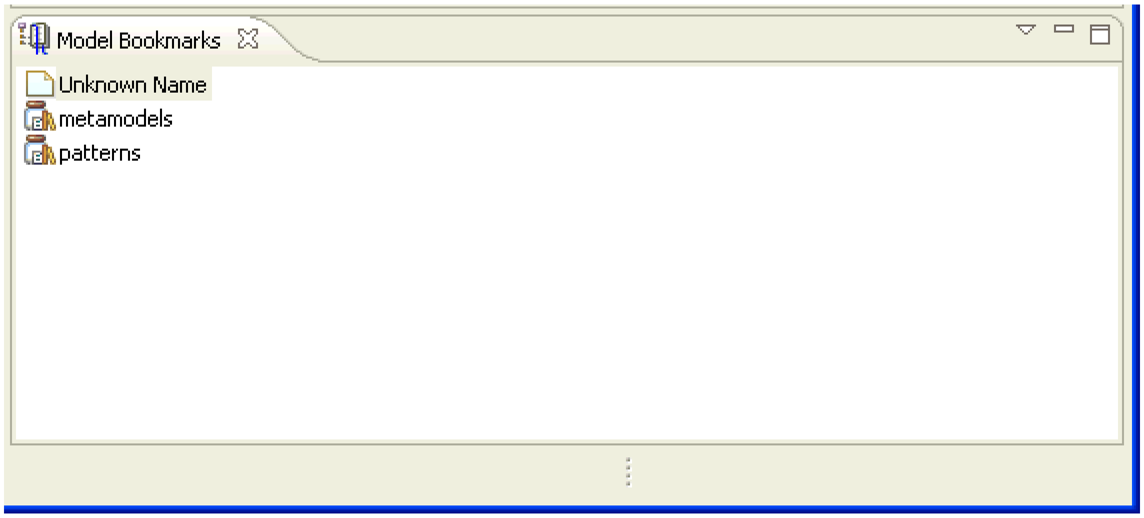
- Select the task, and from the context menu choose **Go To**
- Double click on the task

In both cases the file's editor is opened and the line with which the selected task is associated is highlighted.

Bookmarks

Bookmarks are a simple way to navigate to resources that are used frequently. This section will look at setting and removing bookmarks and viewing them in the Bookmarks view.

The Bookmarks view displays all bookmarks in the Workbench. To show the Bookmarks view select **Window > Show View > Bookmarks**.



Adding and Viewing Bookmarks

The Workbench allows the bookmarking of individual files or locations within a file. This section will demonstrate how to set several bookmarks and how to view them using the Bookmarks view.

1. From the menu bar, select **Window > Show View > Bookmarks**. The Bookmarks view appears in the Workbench.
2. Edit the file JanesFile.txt.

3. Position the cursor over the editor's marker bar next to any line in the file. Then, from the context menu on the marker bar, select **Add Bookmark**.

   When the Add Bookmark dialog opens type in a description for this bookmark. Type in "My Bookmark".

4. Notice that a new bookmark appears in the marker bar.

5. In one of the navigation views select the file JanesText.txt. From the main Workbench menu select **Edit > Add Bookmark**.

   This will bookmark the file using the filename to describe the bookmark. Observe the Bookmarks view now contains two bookmarks.

Using Bookmarks

Now that some bookmarks have been created, instructions will be given on how to get to the files associated with the bookmarks.

1. Close all of the files in the editor area.

2. In the Bookmarks view, double-click on the first bookmark that was created (My Bookmark).

3. Notice that a file editor opens displaying the file with which the bookmark is associated and that the line associated with the bookmark is highlighted.

   NOTE: The Bookmarks view supports an additional way to open the file associated with a selected bookmark, simply select **Go To** from the bookmark's context menu.

In the Bookmarks view, select the associated file in the Model Navigator.

1. In the Bookmarks view, select My Bookmark.

2. From the bookmark's context menu choose **Show in Model Navigator**.

3. Notice that the Model Navigator view is now visible and the file JanesFile.txt is automatically selected. JanesFile.txt is the file My Bookmark was associated with.

Removing Bookmarks

This section will demonstrate how to remove the bookmarks that have been created.

1. In the Bookmarks view, select JanesFile.txt (the second bookmark we created) and do one of the following:

   - Click the Delete button on the view's toolbar.
   - From the bookmark's context menu choose delete.
   - Hit the Delete key on the keyboard.

   Notice that the bookmark is removed from the Bookmarks view.

2. There should be one remaining bookmark. This bookmark is associated with a line in the file JanesFile.txt. There are two other approaches to removing this bookmark.

   - Use **Remove Bookmark** in the marker bar of the JanesFile.txt editor. Remember, **Add Bookmark** was used in the marker bar when the bookmark was first created.
   - Delete the bookmark (as was done above) by using **Delete** from the bookmark's popup menu in the Bookmarks view.

Here is the second approach.

1 Ensure there is an editor open on JanesFile.txt.

  Although the editor doesn't actually need to be open, the editor update will be viewable as the bookmark is deleted.

2 In the Bookmarks view, select JanesFile.txt (the remaining bookmark). Click the Delete button on the view's toolbar. Notice that the bookmark is removed from the Bookmarks view and the JanesFile.txt editor.

Rearranging Views and Editors

This section will explain how to rearrange editors and views to customize the layout of the Workbench.

Setup

Before rearranging the Workbench, a little housekeeping is required.

1 Start by choosing **Window > Reset Perspective** and selecting **OK**. This will reset the current perspective to its original views and layout.

2 Ensure there are editors open for JanesFile.txt and JanesText.txt. Close any other editors.

Drop Cursors

Drop cursors indicate where it is possible to dock views in the Workbench window. Several different drop cursors may be displayed when rearranging views.

⬆	Dock above: If the mouse button is released when a dock above cursor is displayed, the view will appear above the view underneath the cursor.
⬇	Dock below: If the mouse button is released when a dock below cursor is displayed, the view will appear below the view underneath the cursor.
➡	Dock to the right: If the mouse button is released when a dock to the right cursor is displayed, the view will appear to the right of the view underneath the cursor.
⬅	Dock to the left: If the mouse button is released when a dock to the left cursor is displayed, the view will appear to the left of the view underneath the cursor.
🗗	Stack: If the mouse button is released when a stack cursor is displayed, the view will appear as a tab in the same pane as the view underneath the cursor.
🚫	Restricted: If the mouse button is released when a restricted cursor is displayed, the view will not dock there. For example, a view cannot be docked in the editor area.

Rearranging Views

The position of the Model Navigator (or any other) view in the Workbench window can be changed.

1. Click in the title bar of the Model Navigator view and drag the view across the Workbench window. Do not release the mouse button yet.
2. While still dragging the view around on top of the Workbench window, note that various drop cursors appear. These drop cursors (see previous section) indicate where the view will dock in relation to the view or editor area underneath the cursor when the mouse button is released. Notice also that a rectangular highlight is drawn that provides additional feedback on where the view will dock.
3. Dock the view in any position in the Workbench window, and view the results of this action.
4. Click and drag the view's title bar to re-dock the view in another position in the Workbench window. Observe the results of this action.
5. Click and drag the view's title bar outside the Workbench window. Notice that it becomes a "Detached" window (hosted in its own shell).

Finally, drag the Model Navigator view over the Outline view. A stack cursor will be displayed. If the mouse button is released the Model Navigator will be stacked with the Outline view into a tabbed notebook. Please note that you must grab the view tab, **NOT** the window title. This is used only for repositioning the window containing the detached view.

Tiling Editors

The Workbench allows for the creation of two or more sets of editors in the editor area. The editor area can also be resized but views cannot be dragged into the editor area.

1. Open at least two editors in the editor area by double-clicking editable files in one of the navigation views.
2. Click and drag one of the editor's tabs out of the editor area. Do not release the mouse button.
3. Notice that the restricted cursor displays if an attempt is made to drop the editor either on top of any view or outside the Workbench window.
4. Still holding down the mouse button, drag the editor over the editor area and move the cursor along all four edges as well as in the middle of the editor area, on top of another open editor. Notice that along the edges of the editor area the directional arrow drop cursors appear, and in the middle of the editor area the stack drop cursor appears.
5. Dock the editor on a directional arrow drop cursor so that two editors appear in the editor area.
6. Notice that each editor can also be resized as well as the entire editor area to accommodate the editors and views as necessary.
7. It is important to observe the color of an editor tab (in the figure below there are two groups, one above the other)

   blue - indicates that the editor is currently active

   default (gray on Windows XP) - indicates that the editor was the last active editor. If there is an active view, it will be the editor that the active view is currently working with. This is important when working with views like the Outline and Properties that work closely with the editor.
8. Drag and dock the editor somewhere else in the editor area, noting the behavior that results from docking on each kind of drop cursor. Continue to experiment with docking and resizing editors and views until the Workbench has been arranged to satisfaction. The figure below illustrates the layout if one editor is dragged and dropped below another.

Rearranging Tabbed Views

In addition to dragging and dropping views on the Workbench the order of views can also be rearranged within a tabbed notebook.

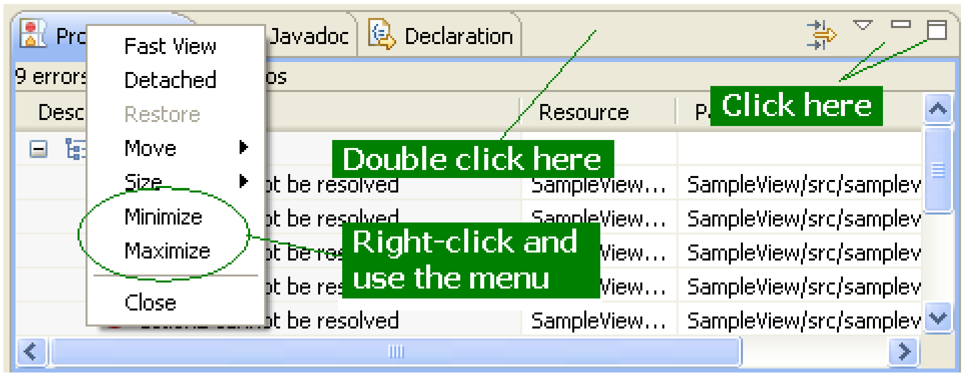
1. Choose **Window > Reset Perspective** to reset the Resource perspective back to its original layout.
2. Click on the Outline title bar and drag it on top of one of the navigation views. The Outline will now be stacked on top of one of the navigation views.
3. Click the tab of the navigation views and drag it to the right of the Outline tab.
4. Once the cursor is to the right of the Outline tab and the cursor is a stack cursor release the mouse button.
5. Observe the Model Navigator tab is now to the right of the Outline tab.

Maximizing and Minimizing

The eclipse presentation provides a rich environment consisting of (in its basic form) an *Editor Area* (containing one or more stacks showing the open editors) surrounded by one or more *View Stacks* (each containing one or more views).These various parts compete for valuable screen real-estate and correctly managing the amount of screen given to each can greatly enhance your productivity within the IDE.

The two most common mechanisms for managing this issue are 'minimize' (i.e. make me use as little space as possible) and 'maximize' (i.e. give me as much space as you can). The eclipse presentation provides a variety of ways to access these operations:

1. Using the minimize and maximize buttons provided on a stack's border
2. Selecting the 'Minimize' or 'Maximize' item on the context (right-click) menu for a stack
3. Double-clicking on a stack
4. Using 'Ctrl + M': this is a key binding for a command that will toggle the currently active part between its 'maximized' and its 'restored' (i.e. normal) states.



Maximize:

It is desirable at times to focus your attention on one particular part to the exclusion of the others. The most popular candidate for this is, of course, maximizing the editor area in order to make as much of the display available for editing as possible (but there are workflows where it would make sense to focus on a view as well).

As of 3.3 the default presentation implements the maximize behavior by minimizing all stacks *except* the one being maximized. This allows the maximized stack to completely occupy the main presentation while still allowing access any open views in your perspective by using the icons in their *Trim Stack* (the area around the edges of the window is called the 'trim').

Also in 3.3 the behavior for managing the editor maximization has been changed to operate on the complete Editor Area (rather than simply maximizing the particular Editor Stack as is the case in the 3.0 and 2.1 presentations). This allows for 'compare' workflows which require the ability to see both files in a split editor area at the same time.

Minimize:

Another way to optimize the use of the screen area is to directly minimize stacks that are of no current interest. As of 3.3 the default presentation minimizing a stack will cause it to be moved into the trim area at the edges of the workbench window, creating a *Trim Stack*. View Stacks get minimized into a trim representation that contains the icons for each view in the stack:

The minimize behavior for the Editor Area is somewhat different; minimizing the Editor Area results in a trim stack containing only a placeholder icon representing the entire editor area rather than icons for each open editor (since in most cases all the icons would be the same, making them essentially useless).

If your particular workflow is such that you need to have more than one element (i.e. having the Editor Area *and* a View Stack in the presentation at the same time) you can still gain additional screen space by minimizing the stacks that aren't of current interest. This will remove them from the main presentation and place them on the outer edge of the workbench window as *Trim Stacks*, allowing more space for the remaining stacks in the presentation.

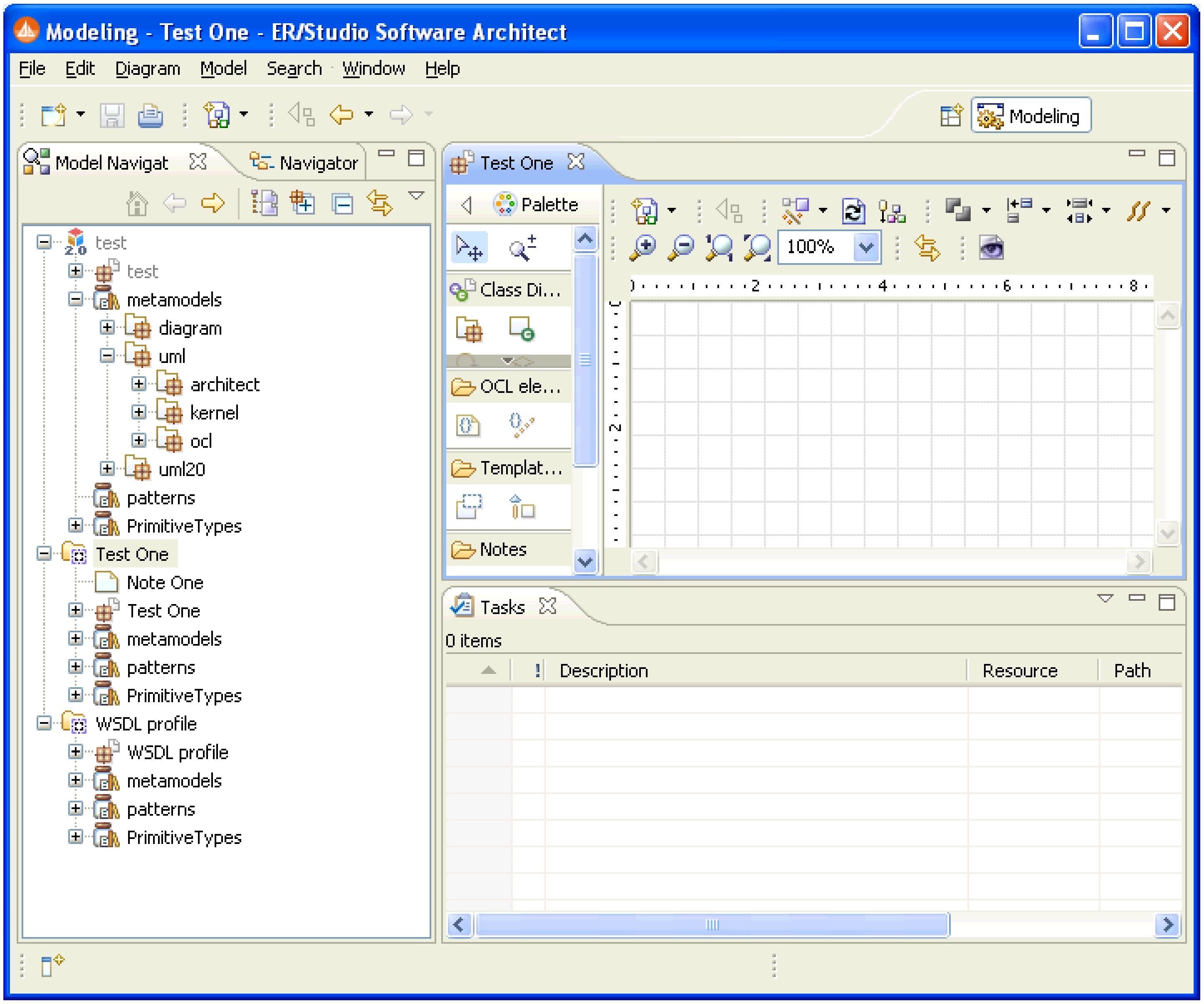
NOTE: There are two ways to end up with a stack in the trim:

- Directly minimizing the stack
- As the result of another stack being maximized

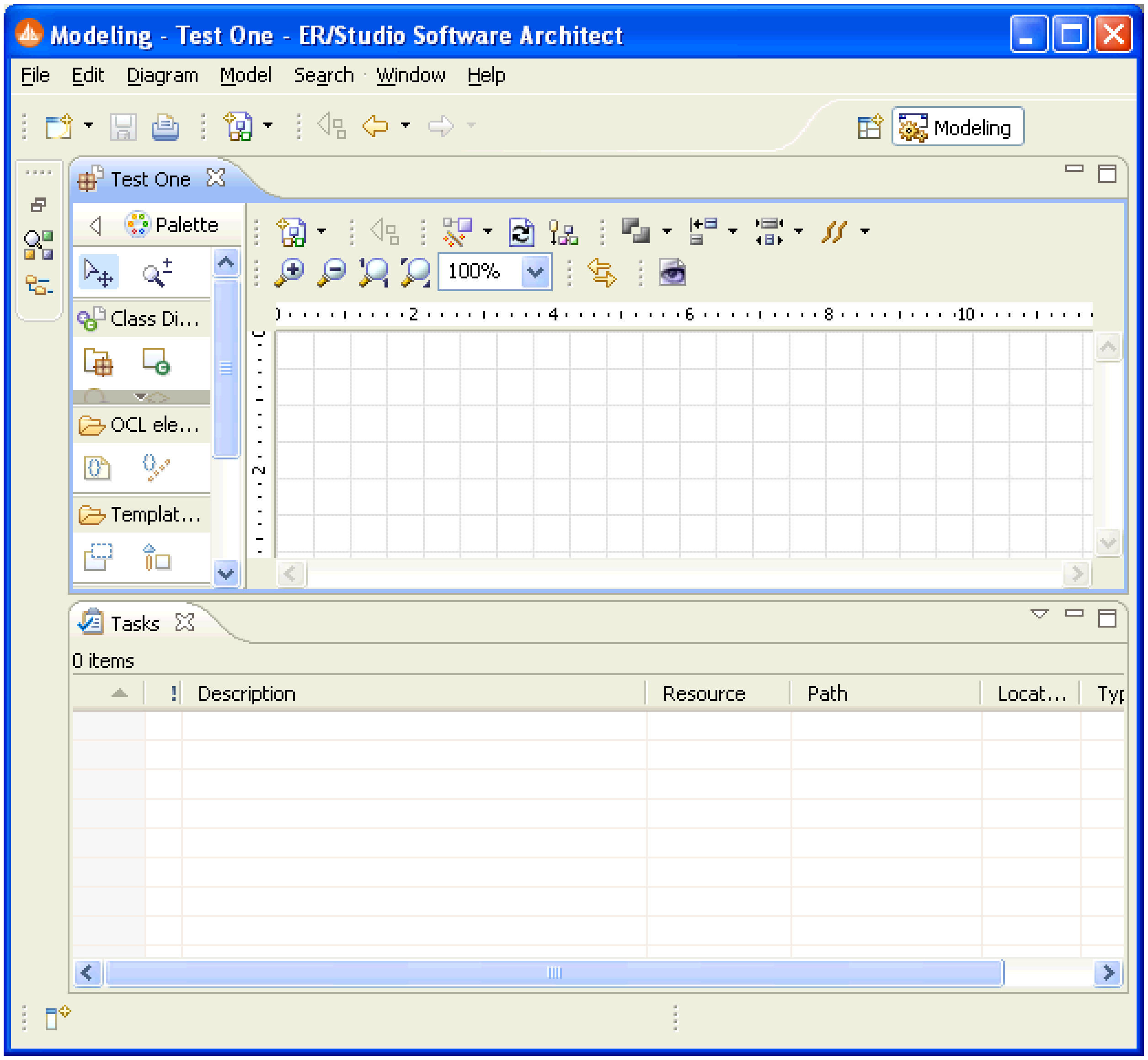
Depending on how the Trim Stack was created its behavior is different; when un-maximizing only those trim stacks that were created during the initial maximize will be restored to the main presentation while stacks that were independently minimized stay that way.

TIP: This difference is important in that it allows you fine grained control over the presentation. While using maximize is a one-click operation it's an 'all or nothing' paradigm (i.e. no other stack is allowed to share the presentation with a maximized stack). While adequate for most tasks you may find yourself wanting to have the presentation show more than stack. In these scenarios don't maximize; minimize all the other stacks *except* the ones you want in the presentation. Once you have it set up you can still subsequently maximize the editor area but the un-maximize will only restore the particular stack(s) that were sharing the presentation, not the ones you've explicitly minimized.

Normal Presentation



Editor Area Maximized



Perspectives

A perspective defines the initial set and layout of views in the Workbench window. One or more perspectives can exist in a single Workbench window.

Perspectives can be opened in one of two ways:

- In the same (existing) Workbench window.
- In a new Workbench window.

Perspectives define visible action sets, which can be changed to customize a perspective. A perspective that is built in this manner can be saved, creating a custom perspective that can be opened again later.

The Workbench window displays one or more perspectives. Each product determines initially what default perspective is displayed, in this example it is the Resource perspective. A perspective consists of views such as the Project Explorer as well as editors for working with resources. More than one Workbench window can be open at any given time.

So far, only the Resource perspective (shown below) has been used in this tutorial. This section will explore how to open and work with other perspectives.

A perspective provides a set of functionality aimed at accomplishing a specific type of task, or working with a specific type of resource.

New Perspectives

There are several ways to open a new perspective within this Workbench window:

- Using the Open Perspective button on the shortcut bar.
- Choosing a perspective from the **Window > Open Perspective** menu.

To open one by using the shortcut bar button:

1. Click on the Open Perspective button.
2. A menu appears showing the same choices as shown on the **Window > Open Perspective** menu. Choose **Other** from the menu.
3. In the Select Perspective dialog choose **Debug** and click **OK**. The Debug perspective is displayed.
4. There are several other interesting things to take note of.
    - The title of the window now indicates that the Debug perspective is in use.
    - The shortcut bar contains several perspectives, the original Resource perspective, the new Debug perspective and a few others. The Debug perspective button is pressed in, indicating that it is the current perspective.
    - To display the full name of the perspective right click the perspective bar and check **Show Text**.
5. In the shortcut bar, click on the Resource perspective button. The Resource perspective is once again the current perspective. Notice that the set of views is different for each of the perspectives.

New Windows

As well as opening a perspective inside the current Workbench window, new perspectives can also be opened in their own window.

By default, new perspectives are opened in the current window. This default behavior can be configured using **Window > Preferences > General > Perspectives**.

Comparing

Workbench allows for the comparison of multiple resources and for the presentation of the results in a special compare editor.

Setup

Before commencing with compare a few files must be created. This will also be a good time to recap some of the basic features that have already been introduced.

1 Start by selecting all of the projects in one of the navigation views and deleting them by using **Delete** on the pop-up menu.

2 Create a new simple project using **File > New > Project**. Be sure to give the project a distinct name by typing unique name as the name of the new project (for example, "JaneQUserCompare"). Do not use spaces or special characters in the project name.

3 Use the project's pop-up menu to create a file called file1.txt.

In the editor for file1.txt type the following lines of text and save the file:

This is line 1.
This is line 2.
This is line 3.
This is line 4.
This is line 5.

4 In one of the navigation views select file1.txt and use Ctrl+C to copy the file.

5 Use Ctrl+V (Paste) to create the copy. In the name conflict dialog which appears, rename the file to file2.txt.

(On the Mac, use Command+C and Command+V.)

There are now two identical files, file1.txt and file2.txt.

Simple Compare

In one of the navigation views select file1.txt and file2.txt choose **Compare With > Each Other** from the context menu.

A dialog will appear indicating that the two files are the same.

Edit file1.txt as follows:

- delete line 1 "*This is line 1."*
- change line 3 to be "This is a much better line 3."
- insert a line 4a (before line 5) that reads "This is line 4a and it is new"

The file (file1.txt) should now read as follows:

This is line 2.
This is a much better line 3.
This is line 4.
This is line 4a and it is new
This is line 5.

Save the contents of the file by choosing **File > Save** (or pressing Ctrl+S).

To compare the files, once again select file1.txt and file2.txt and choose **Compare With > Each Other** in the navigation view context menu.

A special compare editor opens. The next section will explain how to use this compare editor.

Understanding the Comparison

Comparing file1.txt and file2.txt resulted in the following compare editor. The left side shows the contents of file1.txt and the right side shows the contents of file2.txt. The lines connecting the left and right panes indicate the differences between the files.

If more room is needed to look at the comparison, the editor tab can be double clicked to maximize the editor.

The numbered changes on the left side of the difference editor are as follows:

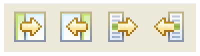
- Starting with the top line (in the left pane) the difference bar (in the area of the blue circle) indicates something is missing from the very top of the left file. Follow the difference band (see #1) to the right file. It contains "This is line 1".
- The next line "This is line 2." is white indicating it matches the right file.
- Moving onto the next line (colored in the background color), see that the left file and right file have different contents for this line (see #2).
- The next line (This is line 4) is once again in white, so it can be skipped.
- The next line exists in the left file but since it is in the background color its difference bar can be followed to the right (see #3) and notice that the right file does not contain the line (see red circle).

Initially the compare editor might seem a bit daunting but when simply working down the left side and focusing on the items marked as gray, and those items missing from the left side, it turns out not to be as tricky as it first seems.

Working With the Comparison

Comparing file1.txt and file2.txt resulted in the following compare editor. This section demonstrates how to use the compare editor to resolve the differences between the two files.

There are two parts to the compare editor's local toolbar. Move to the next or previous change using the right group of local toolbar buttons.



1. Click the **Select Next Change** button. Observe how it selects the next difference.
2. Click the **Select Next Change** button a second time to go to the next change.
3. Click the **Select Previous Change** button.

To merge changes from the left file to the right file and vice versa use the left group of local toolbar buttons. There are four types of merges that can be performed:

- Copy whole document from left to right.
- Copy whole document from right to left.
- Copy current change from left to right.
- Copy current change from right to left.

Typically the copy whole document actions are used when the entire file on either the left or right can just be replaced by the contents of the other file.

The Copy current change buttons allow a single change to be merged.

1. Ensure that the second difference is selected (as shown below):
2. Click **Copy Current Change from Right to Left**. Observe that the selected text from the right file is copied to the left file.

3. Close the compare editor and choose **Yes** to save the changes. Alternatively, save the changes by choosing **File > Save** (Ctrl+S).

Local History

Every time an editable file is saved in the Workbench, the Workbench updates the local history of that file and logs the changes that have been made. The local history of a file can then be accessed and a previously saved copy of the file can be reverted to, as long as the desired state is recent enough in the save history.

1. Create a new file named sampleFile.txt.
2. In the editor for sampleFile.txt modify the file by adding the line "change1" and saving the file.
3. Repeat this by entering a new line "change2" and saving it again.
4. Add a third line "change3" and save it again.
5. Right-click the file in a navigation view (e.g. the Model Navigator) and select Team > Show Local History.

   NOTE: Repository tooling may override the Team > Show Local History operation to provide a unified history for a file. The Eclipse CVS client does override this action. To get the history for a file in a project that is shared with CVS, you will need to right-click the file and select Team > Show History. See Working with CVS History for more details about the CVS history of a file.

6. The History view opens and shows the history for the file.
7. The top entry in the view represents the current contents of the file. The next represents the previous contents and so on. Right-click on the previous entry and select Compare Current with Local to open a Compare editor that displays the differences between the Workbench file and the specific copy of the file selected in the local history.
8. Right-click on the previous entry again and select Get Contents. This replaces the Workbench's copy of sampleFile.txt with the chosen local history item.
9. Observe that the sampleFile.txt editor now contains two lines.

Responsive UI

By default all Eclipse operations run in the user interface thread. Employing the Responsive UI, which allows the threading of sequential code, will enable you to continue working elsewhere in Eclipse. Without the Responsive UI support you would be locked out of performing any other actions when met with a slow operation.

While some operations automatically run in the background (such as auto build), in many cases a dialog will be displayed providing you with the option to run an operation in the background. For example, building a project manually can sometimes take more than a few minutes, during which time you may wish to continue to use other functions in Eclipse.

While the project is being built, select **Run in Background** from the **Building Workspace** dialog and the Responsive UI will allow you to carry on with other tasks in Eclipse.

For information on the status of the action and additional operations that are currently running, click **Details**.

The **Details** panel displays the status information of the operation at hand as well as any additional operations that may be running simultaneously.

The Progress Information dialog also indicates when one operation is being blocked by another.

To have operations running in the background set as the default, select **Window > Preferences > General** and check **Always run in background.**

Exiting the Workbench

Each time the Workbench is exited, the Workbench is automatically saved, including all open perspectives and windows. The next time the Workbench is reopened, it will appear exactly as it was when it was last closed.

To exit the Workbench, select **File > Exit** from the menu bar or close the workbench with the window close button (x). When the latter option is used a prompt will ask if you really wish to exit the workbench.

Note: This dialog also presents an option to turn off the prompting. To turn it back on, select **Window > Preferences** from the main Workbench menu bar, in the Preferences Dialog select **General > Startup and Shutdown** and check Confirm exit when closing last window.

Team CVS Tutorial

This chapter will explain how to use the CVS team capabilities built into the Workbench. The steps in this chapter are team oriented; it will be difficult to complete these tutorials if others are not simultaneously working through this chapter.

You will learn how to work on a project and then commit changes to the repository for others on the team to use. By continuing to work on the project alongside other users, this section will demonstrate how to commit resources that other users may be simultaneously working on, and how to update a workspace with changes others make in the project.

Remember, before getting started, that you need to entice at least one co-worker into working through these steps as well. Another option is to start up two Eclipse instances at once and step through both roles yourself.

Setting up a CVS Repository

A repository is a persistent store that coordinates multi-user access to the resources being developed by a team. The Workbench comes with CVS support built-in. The CVS server is available at http://ximbiot.com/cvs/.

Refer to this site for information on how to install and configure a CVS repository including user access and passwords.

In order to continue with this tutorial, access to a CVS repository is required.

Starting Offline

The team CVS tutorial will start by working offline and creating a simple project. Once the project is created, instructions will be given explaining how to commit it to the repository.

1 Create a new "General" project using **File > New > Project > General > Project**. Use a unique name as the project name (e.g. JanesTeamProject).

2 Create a folder named folder1.

3 Create two text files (.txt) in folder1 called file1.txt and file2.txt. Their contents should be as follows:

   file1.txt

   This is the contents
   of file 1.

   file2.txt

   File2 is a small file
   with simple text.

The project can be continued within this mode but unless the project is committed into the repository, others on the team will be unable to work on the project as well.

Sharing the Project

Now that a project has been created in the workspace you can make it available to other Team members. To do this the following steps will have to occur:

1. Create a CVS location identifying the team's shared CVS repository.
2. Share the project with the CVS location and commit the changes.

Specifying a Repository Location

You must specify an available repository before it is possible to share the project with others.

1. Open the CVS Repository Exploring perspective. The top left view shows all of the CVS repositories that are currently being worked with. As you can see, the view is empty, meaning a repository still needs to be specified.
2. In the context menu of the CVS Repositories view choose **New > Repository Location**.
3. In the CVS Repository Location wizard, the location of the repository and the login information needs to be filled in. You may need assistance from a repository administrator in order to fill in the necessary information.
4. In the **Host** field, type the address of the host (for example, "teamsamples.com").
5. In the **Repository path** field, type the path for the repository at the host address (for example, "/home/cvsroot/repositoryName").
6. In the **User** field, type the user name under which to connect.
7. In the **Password** field, type the password.
8. In the **Connection type** field, select the type of CVS connection for the repository (The default is pserver).
9. Leave **Use Default Port** enabled.
10. By default the **Validate Connection on Finish option** is checked.
11. Click **Finish** when done.

    Since **Validate location on finish** was checked, the wizard will now attempt to validate the information by connecting to the repository. In doing so it may prompt for a password. **Note:** The repository connection is only used to validate the information.
12. Observe that the CVS Repositories view now shows the new repository location.

Repository Locations

Now that a new repository location has been added to the CVS Repositories view, the following will explain what a repository location is or, more importantly, what it isn't.

A repository location is not an actual live connection. Instead, it is a description of where the repository is located. At a later time, when instructions are given explaining how to commit work to the repository or update with work done by others, the Workbench will create a connection based on this location information. Connections will be opened and closed as required when performing CVS team operations, and those connections will be based on the information provided in the repository location.

When disconnected from the network the CVS Repositories view continues to display the list of known repository locations. In addition, the projects themselves will still know the repository location they are associated with.

Sharing a Project

Now that a project has been created and a repository location has been specified, the project can be made available to other team members.

1. In one of the navigation views select the project JanesTeamProject.
2. From the project's context menu choose **Team > Share Project**. If more than one repository provider is installed, select **CVS** and select **Next**.
3. In the sharing wizard page, select the location that was previously created.
4. The next page will ask you the module name to create on the server. Simply use the default value and use the name of the project you are sharing. Click Next.
5. The next page will allow you to see the files that are going to be shared with your team. The arrows with the plus sign show that the files are new outgoing additions. They have never existed on the server yet.
6. Press Finish and you will be prompted to commit the files. Enter a commit comment to describe the commit you are making. You can enter anything you like. Press **Finish** when you are done and the files will be committed.
7. Now you have shared the project and all the files have been committed to the CVS repository. Others can now see your files!

Working with Another User

Instructions were given explaining how to create a project and commit it to the repository. First a repository location was specified. Next the project was shared with that location and the HEAD branch and the resources were added then committed.

The repository currently contains the project and the two files that were committed (file1.txt and file2.txt).

It's time to find that co-worker (let's call him Fred) to work through some steps with you (for the remainder your name is Jane). This section will demonstrate how two people can work on the same project and simultaneously commit changes to the repository. Specifically, the following actions will be taken:

- Fred will import the project to his Workbench
- Fred will make changes to file1.txt and file2.txt.
- You will update these new changes and also add a new file called file3.txt.
- Fred will make more changes to some files while you change the same files. Fred will release his changes first.
- You will have to resolve the conflicts between files that you both changed.

Here is some terminology commonly used to describe changes:

incoming	Changes that are in the repository, which you have not yet updated to.
outgoing	Changes that you are committing
conflict	Both you and the repository have modifications to the same resource.

Checking out a Project

Coworker Fred has several tasks in front of him:

- Fred will import the project that Jane committed to the CVS repository into his Workbench.
- Fred will make changes to file1.txt and file2.txt.
- Fred will synchronize and commit his outgoing changes to the two files.

Fred's first step is to import the project into his workspace as follows:

1. Run the import wizard via the File > Import menu item.
2. Select the Projects from CVS item and press Next.
3. Create a repository location as described when Jane created a repository.
4. On the next page select
5. From the list of available projects, select JanesTeamProject and press Finish. A progress dialog will appear showing the progress of the import operation.
6. Open one of the navigation views and observe that it now includes the project JanesTeamProject. Notice that there are CVS decorators indicating the file revisions, the repository name, and the file types.

Another User Making Changes

Now that Fred has the project in his Workbench he needs to modify several of the files and synchronize with the repository in order to commit them.

- Fred will make changes to file1.txt and file2.txt
- Fred will synchronize and commit his outgoing changes to the two files.

Fred should proceed as follows:

1. Modify file1.txt as follows

   Original contents:

   ```
   This is the contents

   of file 1.
   ```

   New contents (changes shown in bold):

   This is the contents

   Fred-update

   of file 1.

2. Modify file2.txt as follows

   Original contents:

   ```
   File2 is a small file

   with simple text.
   ```

   New contents (changes shown in bold):

   ```
   File2 is a (Fred was here) small file

   with simple text.
   ```

3. Observe that the Model Navigator updates the CVS state of a resource. Notice that the two files which Fred has changed are preceded by ">".

4 To commit his changes to the repository Fred can either:

- Select the two files and choose Team > Synchronize with Repository or,
- Select the project and choose Team > Synchronize with Repository

Often choosing the project is the easiest thing to do. Let's do that. Select the project and choose Team > Synchronize with Repository from its context menu. When you are asked to switch to the Team Synchronizing perspective select Yes.

5 When the Synchronize View opens Fred can browse the changes he made and at the end should commit his changes to file1.txt and file2.txt by selecting JanesTeamProject in the Synchronize View and from the context menu select Commit. You will have to enter a commit comment before committing the changes.

Making Our Own Changes

Fred has made several changes to file1.txt and file2.txt and committed them to the repository. In this section additional changes will be made and then synchronized with the repository. When synchronizing, expect to see the changes made in this section along with changes that have been made by Fred.

1 Add file3.txt as follows:

New contents: (add the following lines to the file)

```
This is the brief contents

of file 3
```

2 Observe that the Model Navigator displays the CVS state of a resources. Notice that the new file added by Jane is preceded by ">".

3 Select the project JanesTeamProject and from the project's context menu, select Team > Synchronize with Repository. When asked to switch perspective select Yes. The Team Synchronizing perspective will open and you will see the files you have changed appear in the Synchronize View. Double click on file1.txt and you should see a compare editor open:

4 There are a couple of other things worth observing. First, the icon next to file1.txt (in the structured compare area) indicates that file1.txt has an incoming change. This means a change was released to the server which you need to take. Looking at file1.txt we can see the text that Fred has added to the file. Also, notice at the bottom of the window, in the status line, there are arrows with numbers beside them. These show the number of files you have incoming, outgoing, and in conflict. The first number beside a file is the revision you have in the workspace, and the other is the revision on the server when you last synchronized.

5 Normally you should update the changes made by others, then test your workspace with the changes, then commit your changes once you are sure that the new changes by others didn't break anything in your workspace.

6 Before deciding to accept Fred's changes you may want to find out why he made the changes. Select file1.txt and from the context menu select Show in History.

The row starting with a * indicates that this is the current revision loaded. In this case you can see the comment made by Fred when he released revision 1.2.

TIP: You can select the Link with Editor and Selection toolbar button in the History view to have the history automatically update when a new editor is opened or when the selection changes. This allows for quick browsing of comments.

7 To update simply select JanesTeamProject in the Synchronize View and from the context menu select Update.

8 The Synchronize View will update to reflect the fact that file1.txt and file2.txt are no longer out-of-sync with the repository. You should only have file3.txt visible now.

9 Next you can commit file3.txt.

Working with Conflicting Changes

There are cases where two users are editing the same files and when the second to commit to the repository tries to commit their changes, the repository won't allow the commit to succeed because of the conflict. Let's simulate this by making Fred and Jane change the same files.

1 In Fred's workspace open one of the navigation views and edit file1.txt. Make the text the following:

```
Fred line 1

This is the contents

Fred-update

of file 1.
```

2 Fred will also change file2.txt with the following change:

```
File2 is a (Fred was here again)

with simple text.
```

3 Fred committed his changes to the repository.

4 Next, at the same time Jane was making changes to file1.txt. She added the following line to the end of the file:

```
This is the contents

Fred-update

of file 1.

Jane was at the end
```

5 And finally, Jane changed file2.txt to the following:

```
File2 is a (Jane was here) small file

with simple text.
```

6 When Jane was finished making changes she synchronized the project and found the following appear in the Synchronize View:

7 Both file1.txt and file2.txt show with a red icon indicating that they have conflicting changes. You can't commit the files until the conflicts are resolved. Click on file1.txt and notice that Fred and Jane made changes to two different parts of the file. In this case, Jane can simply update the file and the lines added by Fred will be merged into Jane's local file. Select file1.txt and from the context menu select Update.

8 Next, double on file2.txt to see the conflict. In this case you can see that both Jane and Fred changed the same line. For this type of conflicting change a regular update can't resolve the conflict. Here you have three options (with the command to use in brackets): accept the changes from Fred (Override and Update), ignore Fred's changes (Mark as Merged), or manually merge the files within the compare editor.

9 For this example, let's say Jane updated file1.txt and selected override and update for file2.txt. After the operations are run the conflicts turn into outgoing changes. Jane can review the changes and commit them.

Replacing

Suppose after further reflection, it becomes apparent to Jane that the revision of file1.txt that was just received is incorrect and an earlier revision is required. A Workbench resource can be replaced with an earlier revision of the resource from the repository. To roll back to an earlier revision:

1 In one of the navigation views select the file1.txt.

2 From the file's context menu, select Replace With > History....

3 The history view will open and display the revisions available for the selected file. Make sure that the history view is in Remote Revisions mode. (If it is not you can put it in Remote mode by clicking on the Remote Revisions button in the History view toolbar.) Select revision **1.3** in the table and from the pop-up menu choose Get Contents.

4 Observe that the navigation view updates to show the CVS state of a resource. Notice that the modified file is preceded by ">" indicating that file1.txt has changed (by replacing it with the earlier version).

5 Now assume that this older revision is not as good as it initially seemed and in fact the revision in the repository is better after all.

Instead of choosing Team > Synchronize with Repository, choose Replace With > Latest from HEAD.

Observe that file1.txt is updated to have the same contents as the latest revision from the repository, and that the leading indicator ">" has been removed since it is now the same revision as the repository.

6 Now that you have a file's revisions displayed in the History View, it is possible to compare revisions directly by selecting the revisions in the table and choosing Compare With Each Other.

Versioning Your Project

Now that the project is complete it is time to version it. While committing the file resources the repository automatically assigned them version numbers, or to be more accurate "revision numbers". As the files were committed the only information supplied was a commit comment.

To version a project proceed as follows:

1 Select the project JanesTeamProject.

2 In one of the navigation views select Team > Tag as Version...

3 When the Tag Resources dialog opens enter the version name *A1* and click OK.

4 Select the CVS Repositories view.

5 Expand **Versions**, and JanesTeamProject. Observe that under JanesTeamProject there is now a version of this project whose version number is *A1*. Looking closely at the contents of folder1 note that the version numbers of the individual files match the version numbers in the Workbench.

A Quick Review

Here are some of the more important but subtle issues associated with working in a repository.

- The project was tagged as a version by versioning the project as it appeared in the Workbench. For this reason it is important to synchronize the project with the repository (that is the HEAD or the branch that is being worked in) prior to versioning it. Otherwise another user may have committed interesting changes to the project which have yet to be updated in the Workbench. By proceeding to version the project without updating, it will be versioned without these changes.
- The repository contains all projects in the repository. Individuals pick the projects they are interested in and check them out into the workspace. From that point on they are synchronizing those projects (only) with respect to the repository.
- The repository represents a large in-progress collection of all known projects. From the repository's perspective, everything in HEAD or on a branch is always open for change.
- The act of versioning a project effectively snapshots it and places it into the Versions section of the repository, however the repository branches are still open for change.
- It is important to first update to changes made to the repository, retest with those changes and the soon to be committed changes and then commit the changes. By first taking the latest changes in the branch, and retesting, it helps to ensure that the changes about to be committed will actually work with the current state of the branch.

- Each project is associated with a specific repository. Different projects can be associated with different repositories that may be on completely different servers.

Working with CVS History

Projects that are managed by CVS now have their file local history and remote history appear together in the same view. To view the consolidated history for a shared file in the History View, perform the following steps:

1 In any navigation view, select a file that is contained in a project that is shared using CVS.

2 Right-click the file in order to access one of the following history related options:

- Team > Show History... which opens the History View with the revisions of the selected file displayed in the table.
- Compare With > History... which does the same as Show History but which launches comparisons when entries in the history table are double-clicked (for more info see CVS History View.
- Replace With > History... which does the same as the comparison operation.

3 The history view will open and display the revisions available for the selected file. You can control what type of revisions are shown in the table by selecting the appropriate toggle button in the History View toolbar. The options are: Local and Remote Revisions, Local Revisions and Remote Revisions.

4 The following operations are available when right-clicking the entries in the History view:

- Open: opens the selected revision
- Compare Current with Revision: compares the selected file with the current revision of the file that resides in the workspace
- Compare with Each Other: compares two selected revisions
- Get Contents: replaces the current workspace file with the contents of the selected revision.

Concepts

Contents

Workbench

The term *Workbench* refers to the desktop development environment. The Workbench aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workspace resources.

Each Workbench window contains one or more perspectives. Perspectives contain views and editors and control what appears in certain menus and tool bars. More than one Workbench window can exist on the desktop at any given time.

Resources

Resources is a collective term for the projects, folders, and files that exist in the Workbench. The navigation views provide a hierarchical view of resources and allows you to open them for editing. Other tools may display and handle these resources differently.

There are three basic types of resources that exist in the Workbench:

Files

> Comparable to files as you see them in the file system.

Folders

> Comparable to directories on a file system. In the Workbench, folders are contained in projects or other folders. Folders can contain files and other folders.

Projects

> Contain folders and files. Projects are used for builds, version management, sharing, and resource organization. Like folders, projects map to directories in the file system. (When you create a project, you specify a location for it in the file system.)

A project is either open or closed. When a project is closed, it cannot be changed in the Workbench. The resources of a closed project will not appear in the Workbench, but the resources still reside on the local file system. Closed projects require less memory. Since they are not examined during builds, closing a project can improve build time.

When a project is open, the structure of the project can be changed and you will see the contents.

Folders and files can be linked to locations in the file system outside of the project's location. These special folders and files are called linked resources.

Different tools that plug into the Workbench use their own specialized types of projects, folders, and files.

Resource Hierarchies

Resources are stored and displayed in the Workbench in hierarchies. Described below are the terms used when referring to resources that are stored and displayed in a hierarchical structure.

Root

> The top level of the Workbench contents (in the file system).

Parent resource

> Any resource that contains another resource. Only projects and folders can be parent resources.

Child resource

> Any resource that is contained within another resource. Only files and folders can be child resources.

Resource hierarchies are displayed in the Model Navigator view, which is one of the default views in the Resource perspective.

Linked Resources

Linked resources are files and folders that are stored in locations in the file system outside of the project's location. These special resources can be used to add files and folders to your project that for some reason must be stored in a certain place outside of your project. For example, a linked folder can be used to store build output separately from your source files.

You can even use linked resources to overlap other resources in the workspace, so resources from one project can appear in another project. If you do want to have overlapping resources in your workspace, do so with caution. Keep in mind that this means changing a resource in one place will cause simultaneous changes in the duplicate resource. Deleting one duplicate resource will delete both!

Deleting a linked resource will *not* cause the corresponding resource in the file system to be deleted. However, deleting child resources of linked folders *will* cause them to be removed from the file system.

Some plug-ins built on top of the Eclipse platform are not compatible with linked resources. If this is the case, you can completely disable the linked resource feature to prevent them from being created in your workspace. Linked resources can be disabled from the **General > Workspace > Linked Resources** preference page. Certain types of projects or team repository providers may also disallow linked resources from being created in some projects.

Path Variables

Path variables specify locations on the file system. The location of linked resources may be specified relative to these path variables. They allow you to avoid references to a fixed location on your file system.

By using a path variable, you can share projects containing linked resources with team members without requiring the exact same directory structure as on your file system.

You can load a project that uses path variables even if you do not currently have all the path variables defined in your workspace. A linked resource that uses a missing path variable is flagged using a special decorator icon. In addition, the **File > Properties > Info** property page and the Properties view (**Window > Show View > Other... > General > Properties**) for a linked resource indicate the variable and whether it is defined or not. A path variable can also specify a location that does not currently exist on the file system. Linked resources that use such a path variable are indicated using the same decorator icon mentioned above.

You can create new path variables and edit and remove existing path variables on the **General > Workspace > Linked Resources** preference page.

Builds

A *build* is a process that derives new resources from existing ones, updates existing resources, or both.

In the Workbench, different *builders* are invoked for different types of projects. Builders usually enforce the constraints of some domain. For example, a Web link builder could update links to files whose name/location changes.

There are two kinds of builds:

- An *incremental build* leverages a previously built state and applies the transforms of the configured builders to the resources that have changed since the previous state was computed (that is, since the last build).
- A clean build discards any problems and previously built state. The next build after a clean will transform all resources according to the domain rules of the configured builders.

Incremental and clean builds can be done over a specific set of projects or the workspace as a whole. Specific files and folders cannot be built. There are two ways that builds can be performed:

- Automatic builds are performed as resources are saved. Automatic builds are always incremental and always operate over the entire workspace. You can configure your preferences (Window > Preferences > General > Workspace) to perform builds automatically on resource modification.
- Manual builds are initiated when you explicitly select a menu item or press the equivalent shortcut key. Manual builds can be either clean or incremental and can operate over collections of projects or the entire workspace.

Local History

A local edit history of a file is maintained when you create or modify a file. Each time you edit and save the file, a copy is saved so that you can replace the current file with a previous edit or even restore a deleted file. You can also compare the contents of all the local edits. Each edit in the local history is uniquely represented by the date and time the file was saved.

Only files have local history; projects and folders do not.

Editors

Most perspectives in the Workbench are comprised of an editor area and one or more views.

You can associate different editors with different types of files. For example, when you open a file for editing by double-clicking it in one of the navigation views, the associated editor opens in the Workbench. If there is no associated editor for a resource, the Workbench attempts to launch an *external editor* outside the Workbench. (On Windows, the Workbench will first attempt to launch the editor in place as an OLE document. This type of editor is referred to as an *embedded editor*. For example, if you have a .doc file in the Workbench and Microsoft Word is registered as the editor for .doc files in your operating system, then opening the file will launch Word as an OLE document within the Workbench editor area. The Workbench menu bar and toolbar will be updated with options for Microsoft Word.)

Any number of editors can be open at once, but only one can be active at a time. The main menu bar and toolbar for the Workbench window contain operations that are applicable to the active editor.

Tabs in the editor area indicate the names of resources that are currently open for editing. An asterisk (*) indicates that an editor has unsaved changes.

By default, editors are stacked in the editor area, but you can choose to tile them in order to view source files simultaneously.

The gray border at the left margin of the editor area may contain icons that flag errors, warnings, or problems detected by the system. Icons also appear if you have created bookmarks, added breakpoints for debugging, or recorded notes in the Tasks view. You can view details for any icons in the left margin of the editor by moving the mouse cursor over them.

External Editors

Sometimes you may need to use an external program to edit a file in the Workbench. This can occur, for example, when the Workbench has no editor for that file type.

The external program will be used as the default editor if that program is registered as the system default editor for that file type and no other editor is registered for that file type in the Workbench. For example, on most systems the default editor for JPEG files is an application for editing or viewing image files. If there is no other editor associated with .jpg or .jpeg files in the Workbench, then opening a JPEG file from the Workbench would cause the file to be opened externally, in the system default editor.

To open an external program that is not the default editor, you can use **Open With > Other...** from the context menu of a file. External programs can also be registered in Eclipse as the associated editor for a given file type. Use the **General > Editors > File Associations** preference page to register editors.

Views

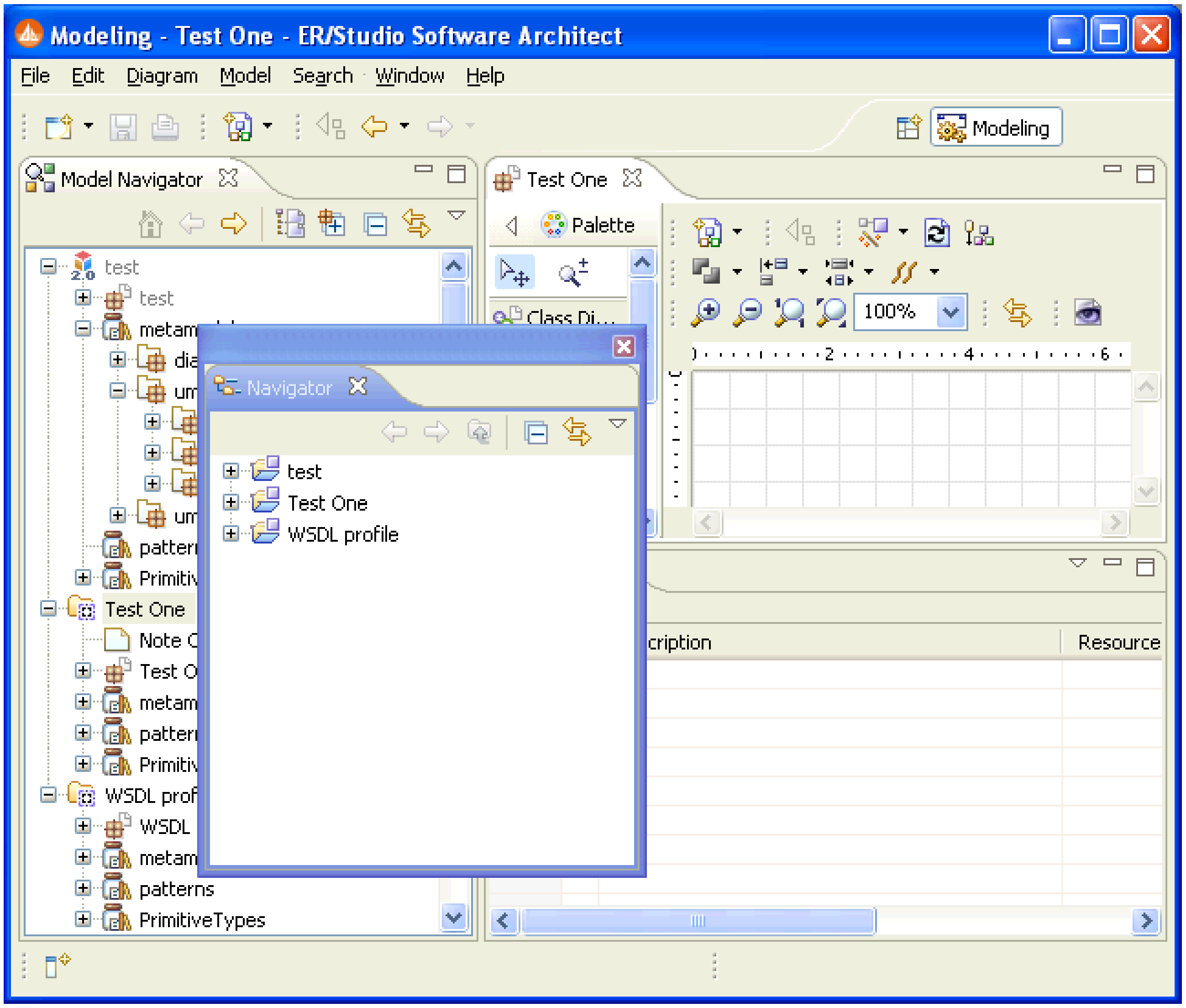
Views support editors and provide alternative presentations as well as ways to navigate the information in your Workbench. For example, the Model Navigator and other navigation views display projects and other resources that you are working with.

Views also have their own menus. To open the menu for a view, click the icon at the left end of the view's title bar. Some views also have their own toolbars. The actions represented by buttons on view toolbars only affect the items within that view.

A view might appear by itself, or stacked with other views in a tabbed notebook. You can change the layout of a perspective by opening and closing views and by docking them in different positions in the Workbench window.

Detached Views

Detached views are views that are shown in a separate window with a smaller trim. They work like other views except they are always shown in front of the Workbench window.



Detached views are views that are shown in a separate window with a smaller trim. They work like other views except they are always shown in front of the Workbench window.

Help VIew

The Help view provides user assistance inside the workbench. The view consists of several pages. Each page presents help in a different fashion. Hyper links at the bottom of the help view allow switching among pages, and clicking any topic will display its contents.

The **Related Topics** page shows description and help topics related to the current workbench context. The **About** section shows context help specific to your current context, and the **Search** link allows you to search for more results based on the name of the view or dialog that has focus. The Related Topics page tracks changes in the workbench and continuously updates displayed information.

The **Contents** page shows the table of contents. It is a hierarchy of all help topics arranged in a tree. The tree branches can be expanded to browse topics that can be displayed using a single mouse click.

The **Search** page allows locating local topics, cheat sheets, welcome content, remote documents, and other documents given a search query. Links to search hits are displayed along with a summary of topic contents. Search scope controls a subset of documentation being searched. Multiple search scopes can be configured, each defining a custom set of resources from among local documentation, additional local search engines or remote engines on the web.

The **Index** page provides an index of keywords that direct the user to specific help topics, similar to indexes found at the back of a book. As you type in the text field, the best match will automatically be highlighted in the list of keywords. Pressing enter or clicking on a keyword will display the given topic.

NOTE: *The index page will only appear when index content is available in the workbench.*

The **Bookmarks** shows topics marked as personal bookmarks.

The **toolbar** of the Help view contains the following buttons. The available buttons depends on the currently displayed page.

Tasks View

You can assign tasks within your project by right clicking marker bar's context menu and selecting **Add Task** or you can add items within the Tasks view by selecting **Add Task**. For example, if you would like to record reminders to follow up on something later, add it to the tasks view. When you add a task, you have the option of associating it with a resource so that you can use the Tasks view to quickly open that resource for editing.

The first column of the Tasks view displays an icon that denotes the type of line item. You can sort and filter line items in the task view, to view only high-priority tasks or tasks associated with a particular resource or group of resources.

By default, the Tasks view is included in the Resource perspective. To add it to the current perspective, click **Window > Show View > Other... > General > Tasks**.

Problems View

As you work with resources in the workbench, various builders may automatically log problems, errors, or warnings in the Problems view. When you double-click the icon for a problem, error, or warning, the editor for the associated resource automatically opens to the relevant line of code.

By default the problems view will group your problems by severity. You can also group them by type or not at all. Certain components will add their own grouping. The grouping can be selected using the **Group By** menu.

The first column of the Problems view displays an icon that denotes the type of line item, the category and the description. Left-click the item to open the file in an editor and highlight the line containing the problem.

You can configure the contents of the Problems view to view only warnings and errors associated with a particular resource or group of resources. This is done using the **Configure Contents** dialog available from the drop down menu. You can add multiple filters to the problems view and enable or disable them as required. Filters can either be additive (any problem that satisfies at least one of the enables filters will be shown) or exclusive (only problems that satisfy all of the filters will be shown) The two most popular filters (All Errors and Warnings on Selection) are provided by default.

Problems can be fixed by selecting **Quick Fix** from the context menu.

To add the Problems view to the current perspective, click **Window > Show View > Other... > General > Problems.**

Properties View

The properties view displays property names and values for a selected item such as a resource.

Toolbar buttons allow you to toggle to display properties by category or to filter advanced properties. Another toolbar button allows you to restore the selected property to its default value.

To see more detailed information about a resource than the Properties view gives you, right-click the resource name in one of the navigation views and select Properties from the pop-up menu.

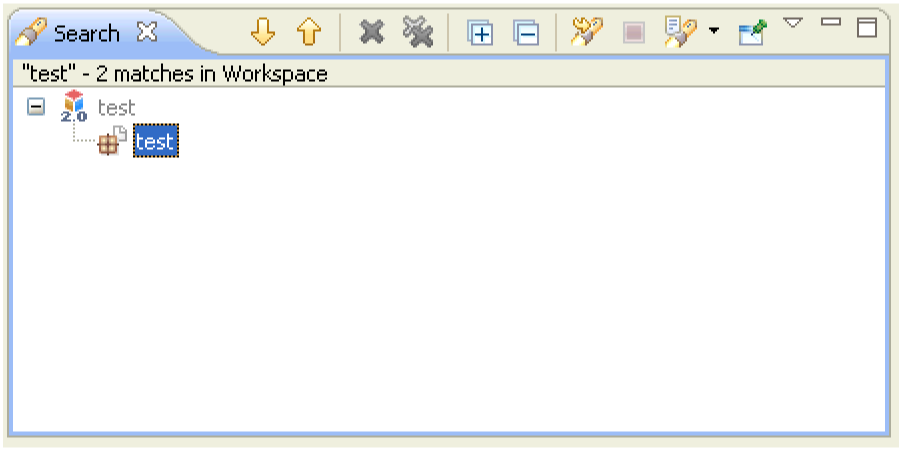
To add the Properties view to the current perspective, click **Window > Show View > Other... > General > Properties**.

Search View

This view displays the results of a search.

Text searches will only search for expressions in files with extensions (file types) specified in the search dialog.

Here is what the Search view looks like:



Toolbar

The toolbar in the Search view contains the following buttons:

Show Next Match

> This command highlights the next match of the search expression in the editor area, opening the file if required.

Show Previous Match

> This command highlights the previous match of the search expression in the editor area, opening the file if required.

Remove Selected Matches

> Removes all highlighted matches from the search results.

Remove All Matches

> Removes all search result form the search view

Expand all

> Expands every tree item in the search view

Collapse all

> Collapses every tree item in the search view

Run the Current Search Again

> This command reruns the current search again, so that removed search results reappear or changes are reflected.

Cancel Current Search

> Cancels the current search

Show Previous Searches

> This command allows you to browse previously conducted searches and repeat a previous search. You can select a previous search from the drop-down menu or clear the search history.

Pin the Search view

> Pinning the search view means that subsequent searches will shown their results in another search view and that the pinned view remains unchanged.

Toolbars

There are five kinds of toolbars in the Workbench.

The *main toolbar*, sometimes called the Workbench toolbar, is displayed at the top of the Workbench window directly beneath the menu bar. The contents of this toolbar change based on the active perspective. Items in the toolbar might be enabled or disabled based on the state of either the active view or editor. Sections of the main toolbar can be rearranged using the mouse.

There are also individual *view toolbars*, which appear in the title bar of a view. Actions in a view's toolbar apply only to the view in which they appear. Some view toolbars include a **Menu** button, shown as an inverted triangle, that contain actions for that view.

A third type of toolbar is the perspective switcher. The perspective switcher allows quick access to perspectives that are currently open. It also has a button that can open new perspectives. The perspective switcher is normally located in the top-right, next to the main toolbar. However, it is also possible to position it below the main toolbar ("top-left"), or to position it vertically on the left-hand side of the workbench ("left"). The name of the perspectives is shown by default, but it is possible to hide the text and show only the icons. To reposition the perspective or hide the text, right-click on it and choose the appropriate item from the context menu.

Minimizing a view stack will also produce a toolbar in the trim at the outer edge of the workbench window (a *Trim Stack*). This bar will contain an icon for each of the views in the stack. Clicking on one of these icons will result in the view being displayed as an overlay onto the existing presentation.

Finally, the fast view bar is a toolbar that contains icons representing the current set of fast views. A fast view is a shortcut to a view that is frequently used; see the section on fast views for more information. The fast view bar appears in the bottom left corner of the workbench by default. However, it is possible to position it on the left or right as well.

In all cases, you can find out what toolbar buttons do by moving your mouse pointer over the button and reading the tooltip that opens. See the list of related reference topics below for a table of all toolbar buttons.

Markers

Markers are objects that may be associated with Workbench resources. There are many uses of markers in the Workbench. The three main uses of markers in the Workbench are:

- Tasks
- Problems

- Bookmarks

Markers are shown in a marker view (Tasks, Problems or Bookmark view) or on the marker bar in the editor area.

The following sections give more detail on each of these marker types.

Tasks

A task marker represents a work item. There are two kinds of tasks:

- Automatically-generated information associated with the resource
- User specified tasks that may or may not be associated with a resource

Both of these task types appear in the Tasks view.

Problems

Problem markers represent invalid states and are categorized as follows.

- **Errors**: Error markers are often used to indicate the source location of syntax or compilation errors.
- **Warnings**: Warning markers indicate the source location of, for example, compilation warnings.
- **Information**: Information markers indicate the source location of information only tasks.

Problems are shown in the Problems view.

Bookmarks

Bookmarks place an anchor either at a specific line in a resource or on the resource itself. They are shown in the Bookmarks view.

Bookmarks

You can place an "anchor" either on a resource within the Workbench, or at a specific line within a file, by creating a bookmark. Then you can use the Bookmarks view to return to those files quickly.

The Bookmarks view (**Window > Show View > Bookmarks**) displays all bookmarks that you have created.

Label Decorations

Label decorations are used to show extra information about an item by modifying its label or icon. They can be used to obtain information about the state of an item without having to look at its properties in the Properties view or open its Properties dialog.

A number of label decorators may be available. To control which decorators are visible, go to the **General > Appearance > Label Decorations** preference page. This preference page provides a selectable list and description of the available decorations.

Help

The Help system lets you browse, search, bookmark, and print help documentation. The documentation is organized into sets of information that are analogous to books. The help system also supplies a text search capability for finding the information you need by search phrase or keyword, and context-sensitive help for finding information to describe the particular function you are working with.

You can interact with help system in the workbench using the **Help view** or in the separate **Help window**. The view and window provide the same information but in different ways.

The Help view

The Help view provides help inside the workbench. You can open the view from the main menu by selecting **Help > Dynamic Help** or **Help > Search**. The view will open showing the Related Topics or Search page, respectively. You can use links at the bottom of the help view to turn to other pages.

The Help window

The Help window provides the same content as the Help view, but in a *separate* window instead of in a view. You can open the window from the main menu by selecting **Help > Help Contents**. The first view shown in the window is called Contents. This view displays the table of contents for the product documentation. Click on one of the links to expand the navigation tree for a set of documentation.

Note that if your browser does not have JavaScript enabled or is an does not support the features required to display the normal help application "basic" help which will be displayed - this can easily be identified because closed books in the table of contents do not have an expander icon.

Context-sensitive help

If you are working through a task and encounter a part of the interface that you do not understand, you can summon context-sensitive help. By default, this will display the Help view and give you some specific information about the view/editor/dialog you are using, and possibly some links to topics for further help.

Context-sensitive help can be accessed by bringing focus to the interface part in question by clicking on it or using the Tab key, and then pressing F1 (Shift+F1 on GTK+, and Help on the Mac). Alternatively, in dialogs you can achieve the same result by pressing the help button in the dialog's button bar.

Team Programming with CVS

In the Concurrent Versions System (CVS) team programming environment, team members do all of their work in their own Workbenches, isolated from others. Eventually they will want to share their work. They do this via a CVS Repository.

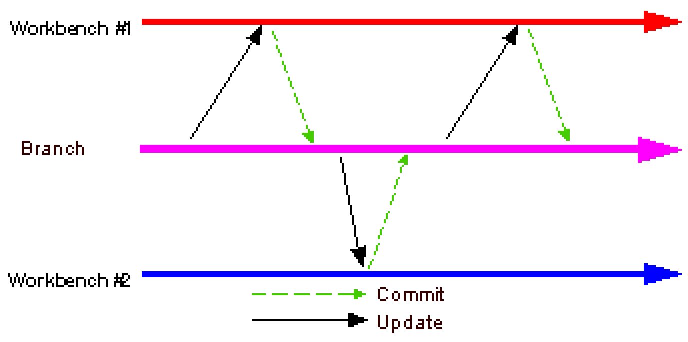
Branches

CVS uses a branch model to support multiple courses of work that are somewhat isolated from each other but still highly interdependent. Branches are where a development team shares and integrates ongoing work. A branch can be thought of as a shared workspace that is updated by team members as they make changes to the project. This model allows individuals to work on a CVS team project, share their work with others as changes are made, and access the work of others as the project evolves. A special branch, referred to as HEAD, represents the main course of work in the repository (HEAD is often referred to as the trunk).

Sharing work

As team members produce new work, they share this work by *committing* those changes to the branch. Similarly, when they wish to get the latest available work, they *update* their local workspaces to the changes on the branch. Thus the branch is constantly changing, moving forward as team members submit new work.

The branch effectively represents the current state of the project. At any point a team member can update their workspaces from the branch and know they are up to date.



Optimistic team model

CVS provides two important features required for working in a team:

- A history of the work submitted by the team
- A way to coordinate and integrate this work

Maintaining history is important so that one can compare the current work against previous drafts, revert to older work that is better, and so on. Coordinating the work is critical so that there exists one definition of the current project state, containing the integrated work of the team. This coordination is provided via the branch model.

An optimistic model is one where any member of the team can make changes to any resource he or she has access to. Because two team members can commit to the branch changes to the same resource, conflicts can occur and must be dealt with. This model is termed *optimistic* because it is assumed that conflicts are rare.

Recommended work flow

Usually resources do not exist in isolation, they typically contain implicit or explicit dependencies on other resources. For example, Web pages have links to other Web pages, and source code has references to artifacts described in other source code resources. No resource is an island.

As resources are committed to the branch, these dependencies can be affected. Ensuring the integrity of the dependencies is important because the branch represents the current project state: at any point a team member could take the branch contents as a basis for new work.

The ideal work flow therefore is one in which the branch integrity is preserved.

Ideal flow enumerated

The ideal work flow proceeds as follows:

1. Start fresh. Before starting work, update the resources in the workspace with the current branch state. If you are sure that you have no local work that you care about, the fastest way to get caught up is to select the projects you are interested in from the branch (or HEAD) and select **Checkout** (or **Replace with > Latest from Repository** if the projects already exist locally). This will overwrite your local resources with those from the branch.
2. Make changes. Work locally in your Workbench, creating new resources, modifying existing ones, saving locally as you go.

3 Synchronize. When you are ready to commit your work, synchronize with the repository.

- Update. Examine incoming changes and add them to your local Workbench. This allows you to determine if there are changes which might affect the integrity of what you are about to commit. Resolve conflicts. Test again, run integrity checkers (for example, check for broken hypertext links, ensure your code compiles, and so on).

Commit. Now that you are confident that your changes are well integrated with the latest branch contents, commit your changes to the branch. To be prudent, you may repeat the previous step if there are new incoming changes.

Of course this is an *ideal* work flow. Under certain conditions you may be confident that the incoming changes do not affect you and choose to commit without updating. However, in general team members should make an effort to follow a flow similar to the above in order to ensure that the branch integrity is not accidentally compromised.

Versions

Resources are versioned in order to capture a snapshot of the current state of the resources at one specific point in time. Resources in CVS are versioned by tagging them with a version label. When a resource is versioned, it means that a non-modifiable copy of it can be retrieved from the repository.

Versioning a project saves the line up of all resource versions in the project. Resources other than projects (files and folders) can be versioned. However, it is more common to version entire projects together as the resources contained in a project are often highly interdependent. Projects can be versioned from the workspace or from the branch (including HEAD) in the CVS Repositories view. The difference between these two approaches is in deciding which child resource versions should be part of the project version.

When tagging a project as a version from the *Workbench*, the base revisions of the files in the Workbench are tagged as belonging to that version. This is the preferred method of versioning a project because you know exactly which file revisions will be in the version. This operation is allowed if you have outgoing changes or uncommitted changes. Uncommitted changes are simply ignored and resources with outgoing changes can still have their base revisions be part of the version. Versioning a project with uncommitted or outgoing changes is handy if you have to split the project at the point where you started making changes to the resources and commit the resources to another branch.

When tagging a project as a version from a *branch* in the CVS Repositories view, you are versioning whatever the latest resource versions are in the branch at that moment in time. You should not version your projects from the branch if you do not know what is committed in the branch. For this reason, versioning from the Workbench is often preferable.

Branches

In CVS, teams share and integrate their ongoing work in *branches*. Think of a branch as a shared work area that can be updated at any time by team members. In this way, individuals can work on a team project, share their work with others on the team, and access the work of others during all stages of the project. The branch effectively represents the current shared state of the project.

Resources can be changed in the Workbench without affecting the branch. Individuals must explicitly provide their changed resources to the branch.

Every CVS repository has at least one branch, referred to as HEAD. Under certain conditions, more than one branch may exist in a repository. For example, one branch may be for ongoing work, and another branch may be for maintenance work.

As you make changes locally in your Workbench, you are working alone. When you are ready to make your local resource changes available to other team members, you'll need to *commit* your work to the branch. All such changes are classified as outgoing changes when you do a synchronization.

Ideally, you should update your local workspace with any changes others have made in a branch before committing to it. This ensures that you have the very latest work from the other team members. After you have updated from the branch, merged any conflicting changes in your local Workbench, and tested your changes locally, you can more easily commit your Workbench's changes to the branch.

When you commit changes to the branch, your changes are copied from your local Workbench to the branch. As a result, these changes are then seen as incoming changes when other developers update from the branch later.

CVS Repositories

A Concurrent Versions System (CVS) repository is a persistent data store that coordinates multi-user access to projects and their contents. Projects in a repository can be of two forms: immutable (a project version) or modifiable (a project in a branch). Communication between the repository and Workbench clients is possible over local or wide area networks. Currently the Workbench supports two internal authentication protocols for CVS: pserver. Authentication protocols provided by external tools can also be used by CVS.

Three-way Comparisons

Three way comparisons show the differences between three different versions of a resource. This feature is most useful when merging resources or when there is a conflict during synchronization. Conflicts occur when two developers add a version from the same branch to their Workbench, then each developer modifies it, then one developer attempts to commit the resource after the other developer has already committed it.

When this situation arises, you can view the differences between three resource versions: the resource in the Workbench, the version of the resource that is committed in the branch, and the *common ancestor* from which the two conflicting versions are based. If a common ancestor cannot be determined, for example because a resource with the same name and path was created and committed by two different developers, the comparison becomes a two-way comparison.

Interpreting compare results

The Synchronize view allows you to view the differences between two or three files. If a common ancestor is available, the sync view performs a three way comparison. It is possible that a common ancestor for two conflicting resource versions cannot be determined, (e.g. a resource with the same name and path is created and committed by two different developers). In this case the compare becomes a regular two way compare.

In a three way compare the Workbench shows you:

- what has been changed in the first child in comparison to the common ancestor.
- what has been changed in the second child in comparison to the common ancestor.

Synchronizing with a CVS Repository

In the CVS team programming environment, there are two distinct processes involved in synchronizing resources: *updating* with the latest changes from a branch and *committing* to the branch.

When you make changes in the Workbench, the resources are saved locally. Eventually you will want to commit your changes to the branch so others can have access to them. Meanwhile, others may have committed changes to the branch. You will want to update your Workbench resources with their changes.

Important: It is preferable to update *before* committing, in case there are conflicts with the resources in your Workbench and the resources currently in the branch.

The synchronize view contains filters to control whether you want to view only *incoming changes* or *outgoing changes*. Incoming changes come from the branch. If accepted, they will update the Workbench resource to the latest version currently committed into the branch. Outgoing changes come from the Workbench. If committed, they will change the branch resources to match those currently present in the Workbench.

Regardless of which mode (filter) you select, the Synchronize view always shows you conflicts that arise when you have locally modified a resource for which a more recent version is available in the branch. In this situation you can choose to do one of three things: update the resource from the branch, commit your version of the resource to the branch, or merge your work with the changes in the branch resource. Typically you will want to merge, as the other two options will result in loss of work.

Watch/Edit

CVS provides a notification scheme which allows a group of developers to know if somebody is working on a given file. This facility is known as *watches*. By setting a *watch* on a file, you can have CVS notify you via email if someone else starts to *edit* this file. This mechanism is notification based only; the file is not locked in any way on the server, and several people are allowed to edit the same file at the same time.

In addition to watch list notification, *edit* on its own is useful for discovering if others are also editing that file. This is because when you edit a file, you will be informed if someone else is already editing it.

Normally with CVS clients, you would need to issue an explicit *edit*. With Team CVS support however, an *edit* is automatically issued by the client when you start to modify a file. In addition, when *editing* a file, Team CVS provides you with the list of people already editing a file. This allows you to find out who is working on a file before you start to edit it.

Accessibility Features in Eclipse

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or those with special needs to use software products successfully. These are the major accessibility features in Eclipse:

- Eclipse uses Microsoft Active Accessibility (MSAA) APIs to render user interface elements accessible to assistive technology.
- You can operate all features using the keyboard instead of the mouse. See the related task.
- You can use screen-reader software such as Freedom Scientific's JAWS ^{TM} and a digital speech synthesizer to hear what is displayed on the screen. You can also use voice recognition software, such as IBM ViaVoice ^{TM} to enter data and to navigate the user interface.
- You can magnify what is displayed on your screen in the graphical views.
- Fonts and colors defined by Eclipse can be set using the General > Appearance > Colors and Fonts preference page. See the related link.

NOTE: The Accessibility features mentioned in this document apply to the Windows operating system

Navigating the User Interface Using the Keyboard

The user interface is navigable using the keyboard. The Tab key is used to iterate through the controls in a particular scope (for example, a dialog or a view and its related icons). To navigate to the main controls for the Workbench window or to tab out of views that use the Tab key (such as editors) use Ctrl+Tab.

Menus

Most menus are assigned mnemonics for each entry which allow you to select them by typing the underlined letter instead of the mouse. You can also select an item by moving through the menus and sub-menus with the arrow keys.

The various menus available can be accessed using the keyboard in the following ways:

- F10 accesses the menus on the main menu bar.
- Shift+F10 pops up the context menu for the current view. (Note: this shortcut is actually dependent on your window manager, but for most people it should be Shift+F10.)
- Ctrl+F10 will open the pull down menu for the current view if there is one. For editors, Ctrl+F10 will open the menu for the marker bar on the left of the editor area.
- Alt+mnemonic will activate the Workbench menu for a particular entry (e.g., Alt+W will bring down the Window menu).
- Microsoft Windows only: Pressing Alt will give focus to the menu bar.

Controls

Mnemonics are assigned to most control labels (e.g., buttons, check boxes, radio buttons, etc.) in dialog boxes, preference pages, and property pages. To access the control associated with a label, use the Alt key along with the letter that is underlined in the label.

Navigation Context

Navigation context is saved for the packages, navigator views, Workbench preferences and properties dialogs. The selected page for the preferences and properties dialog is saved between invocations of the dialog but are not saved between workbench invocations.

Cycling Editors, Views and Perspectives

To switch between editors, views and perspectives, the workbench provides a cycling function that is invoked by Ctrl and a function key. All of these cycling functions recall the last thing selected to allow for rapid cycling back and forth between two items. The cycling functions are

- Ctrl+F6 - Cycle to Editor
- Ctrl+F7 - Cycle to View
- Ctrl+F8 - Cycle to Perspective

Also, Ctrl+E can be used to activate the editor drop-down, and Ctrl+PageUp and Ctrl+PageDown can be used for switching between the open editors.

Accelerators

Many of the actions in Eclipse have an accelerator assigned to them.

Quick Access

To quickly access UI elements such as views, commands, preference pages, and others, you can use the Quick Access dialog, available under **Window > Navigation > Quick Access** and bound to **Ctrl+3** by default. Start typing in the filter field to see matches. For example, to open the help view, type Ctrl+3 followed by "help". One of the first matches will be to open the help view; other matches show commands and preference pages related to help. You can use the arrow keys to select a different match for a given filter string. Press Enter to select the highlighted entry. This will execute the command, or open the view, perspective, or wizard etc.

Help system

You can navigate the help system by keyboard using the following key combinations:

- Pressing Tab inside a frame (page) takes you to the next link, button or topic node.
- To expand/collapse a tree node, press Right/Left arrows.
- To move to the next topic node, press Down arrow or Tab
- To move to the previous topic node, press Up arrow or Shift+Tab
- To display selected topic, press Enter.
- To scroll all the way up or down press Home or End.
- To go back press Alt+Left arrow; to go forward press Alt+Right arrow.
- To go to the next frame, or toolbar press Ctrl+Tab (Ctrl+F6, if using Mozilla, or Mozilla based browser).
- To move to previous frame, press Shift+Ctrl+Tab. (Shift-Ctrl+F6, if using Mozilla, or Mozilla based browser).
- To move to the frame displaying topic content press Alt+K (when using embedded help browser on Windows, or Internet Explorer).
- To move to Contents tab, press Alt+C
- To move to Search Results tab, press Alt+R
- To move between tabs, press Right/Left arrows.
- To switch view, select a tab and press Enter.
- To switch and move to a view, select a tab and press Up arrow.
- To move to the search entry field, press Alt+S
- To print the current page or active frame, press Ctrl+P.
- To find a string in the current page or active frame, press Ctrl+F (when using embedded help browser on Windows, or Internet Explorer).

Most labels of controls on help system pop-up dialogs have mnemonics are assigned to them. To access the control associated with a label, use the Alt key along with the letter that is underlined.

Fonts and Colors in Eclipse

Eclipse uses the fonts and colors provided by the operating system as much as possible. On Windows the platform color and font settings are found on the **Preferences > Colors and Fonts** page. The font used by most widgets in Eclipse is the one set in the Message Box settings of the properties. However, operating systems do not provide enough colors to handle all of the extra information that colors and fonts provide in Eclipse.

Fonts

There are 4 main fonts in use by the Eclipse platform. They are:

Banner Font

Used in PDE editors, welcome pages and in the title area of many wizards. For instance the New Project wizard uses this font for the top title,

Header Font

Used as a section heading. For instance the Welcome page for the Eclipse Platform uses this font for the top title, *Text Font*

Used in text editors.

Dialog Font

Used in dialogs.

These fonts can be set via the **General > Appearance > Colors and Fonts** preference page. As well as these 4 fonts there are several other secondary font settings. These default to the text font. They can be found on the Colors and Fonts preference page:

- **Compare Text Font**
- **Console Text Font**
- **CVS Console Font**
- **Debug Console Font**
- **Detail Pane Text Font**
- **Memory Views Table Font**
- **Part Title Font** (optional: used by some presentations)
- **View Message Font** (optional: used by some presentations)

Colors

Eclipse uses colors as an information enhancement in many places. Whenever possible the operating system color settings are used, but in cases where the operating system settings are not enough, Eclipse defines other colors.

Accessibility and the Windows Color Dialog

For color selection, Eclipse uses a dialog provided by the operating system. On windows, the color selection dialog does not respond properly to assistive technology. When you first get into the dialog, focus is on one of the basic colors, but the dialog provides no indication of this through assistive technology. You can select colors in Eclipse with this dialog in the following way:

1. Select to customize the color of something in Eclipse, for example the color of Error Text in your Workbench Colors and Fonts Basic preferences.
2. In the color selection dialog, tab twice to go from the Basic Color matrix to the Define Custom Colors button and press Enter.

You can now enter the basic colors using an HSL or RGB specification according to the following definitions.

Features

On disk, an Eclipse based product is structured as a collection of *plug-ins*. Each plug-in contains the code that provides some of the product's functionality. The code and other files for a plug-in are installed on the local computer, and get activated automatically as required. A product's plug-ins are grouped together into features. A *feature* is a unit of separately downloadable and installable functionality.

The fundamentally modular nature of the Eclipse platform makes it easy to install additional features and plug-ins into an Eclipse based product, and to update the product's existing features and plug-ins. You can do this using the Eclipse platform's install and update support found in the **Help** menu. Eclipse allows you to discover, download, and install features and plug-ins from special web-based Eclipse software sites.

Large Eclipse based products can organize their features into trees starting from the root feature that represents the entire product. This root feature then includes smaller units of functionality all the way down to leaf features that list one or more plug-ins and fragments. The capability to group features hierarchically allows products to be stacked using a 'Russian doll' approach - a large product can build on top of a smaller one by including it and adding more features.

Some included features may be useful add-ons, but are not vital to the proper functioning of the overall product. Feature providers can elect to mark them as **optional**. Optional features will only be installed if all their required features and plug-ins are available. If not installed right away, optional features can be added at a later date.

The **About** option on the **Help** menu provides information about installed features and plug-ins. The **Check for Updates** and **Install New Software** commands on the **Help** menu provide the ability to update existing features, and to find, download, and install new features.

Tasks

Contents

Working with Perspectives

Perspectives define the initial set and layout of views in the Workbench window. They provide a set of functionality aimed at accomplishing a specific type of task or working with specific types of resources.

See the Related tasks links for more details.

Contents

- Showing and Hiding Menu Items and Toolbar Buttons
- Configuring Perspective Command Groups
- Configuring Perspective Shortcuts
- Saving a User-defined Perspective
- Deleting a User-defined Perspective
- Resetting Perspectives

Switching between Perspectives

Open perspectives are represented by icons on the perspective bar. When you have more than one perspective open, you can switch between them by clicking the icons on the shortcut bar.

Specifying the Default Perspective

The default perspective is indicated in the Select Perspective dialog (accessible via the **Window > Open Perspective > Other...** menu). The pre-defined default perspective is indicated by the word default in brackets following the perspective name, for example, **Resource (default)**.

To change the default perspective:

1. Open the General > Perspectives preference page.
2. Select the perspective that you want to define as the default from the list of available perspectives, and click **Make Default**. The default indicator moves to the perspective that you selected.
3. Click **OK**.

Opening Perspectives

Perspectives provide combinations of views and editors that are suited to performing a particular set of tasks. For example, you would normally open the Debug perspective to debug a Java program.

To open a new perspective:

1. Click the **Open Perspective** button on the shortcut bar on the left side of the Workbench window. (This provides the same function as the **Window > Open Perspective** menu on the menu bar.)
2. To see a complete list of perspectives, select **Other...** from the drop-down menu.
3. Select the perspective that you want to open.

When the perspective opens, the title bar of the window it is in changes to display the name of the perspective. In addition, an icon is added to the shortcut bar, allowing you to quickly switch back to that perspective from other perspectives in the same window.

By default, a perspective will open in the same window. If you would rather it opened in a new window, change the setting in the General > Perspectives preference page.

Changing where Perspectives Open

You can change the default behavior for how perspectives are opened in the Workbench.

1. Open the General > Perspectives preference page.
2. Select either **In the same window** or **In a new window** from the **Open a new perspective** group.
3. Click **OK**.

Showing and Hiding Menu Items and Toolbar Buttons

You can choose to hide menu items and toolbar buttons and then show them again.

Hiding a menu item or toolbar button

To hide a menu item or toolbar button:

1. Switch to the perspective that you want to configure.
2. Select Window > Customize Perspective.
3. Open the **Menu Visibility** or **Tool Bar Visibility** tab.
4. Find the item you want to hide. You can do this two ways:
   - Expand the menu or toolbar hierarchy to find the item you want to hide.
   - Click the **Filter by command group** check box to see a list of command groups which contribute items, and choose the command group the item you wish to hide. Then navigate to the item in the hierarchy in the Structure tree.
5. Hover over the item to get additional information:
   - a description of what the item does
   - the name of the command group which contributes the item (click the link in this item to switch to the **Command Groups Availability** tab with the appropriate command group selected).
   - any key bindings associated with the command the item performs (click the link in this item to open the **Keys** page of the **Preferences** dialog with the command selected, if possible).
   - if the item is dynamic, a preview of its current appearance (dynamic items are listed as [Dynamic]).
6. Uncheck the check box next to the item. Uncheck a menu to hide all its children.
7. Click **OK** to cause the changes to take effect.

Using the tooltip which appears over items, you can navigate to the **Command Group Availability** tab and make the entire command group unavailable if you wish to remove all menu items, toolbar buttons and keybindings of all commands contributed by the command group.

Showing a menu item or toolbar button

To show a menu item or toolbar button, you can follow the same instructions as hiding one, except check the check box instead of unchecking it.

If an item you want to make visible is grayed out, this is because the command group which contributes it is not available. You need to make it available before you can show or hide items it contributes. You can do this by hovering over the item, and clicking the command group link in the tooltip which appears.

Configuring Perspective Command Groups

You can choose which command groups are available in a perspective. Command groups contribute menu items, toolbar buttons and key bindings, all of which run commands. Making a command group unavailable in a perspective removes these methods of running commands. Making a command group available makes it possible for these methods to be used, however they can be turned on and off with more fine-grained control via the **Menu Visibility** and **Tool Bar Visibility** tabs of the **Customize Perspective** dialog.

To configure command group availability:

1. Switch to the perspective that you want to configure.
2. Select Window > Customize Perspective.
3. Open the **Command Groups Availability** tab.
4. Select the command group in the **Available command groups** list whose availability you want to edit.
5. Preview the menu items and toolbar buttons which are contributed by the selected command group in the **Menubar details** and **Toolbar details** trees. Hover over items in these trees to view their key bindings.
6. Use the check box next to the command group to make it available or unavailable.
7. Click **OK** to cause the changes to take effect.

Configuring Perspective Shortcuts

You can choose which menu items are available under the **File > New**, **Window > Open Perspective** and **Window > Show View** menus using the **Shortcuts** tab of the **Customize Perspective** dialog. To configure shortcuts:

1. Switch to the perspective that you want to configure.
2. Select Window > Customize Perspective.
3. Open the **Shortcuts** tab.
4. Choose the sub menu whose shortcuts you want to edit.
5. Use the Categories tree to turn groups of related shortcuts on or off, or select categories to work with individual menu items in the Shortcuts list.
6. Click **OK** to cause the changes to take effect.

Saving a User-defined Perspective

If you have modified a perspective by adding, deleting, or moving (docking) views, you can save your changes for future use.

1. Switch to the perspective that you want to save.
2. Click Window > Save Perspective As.
3. Type a new name for the perspective into the **Name** field.
4. Click **OK**.

The name of the new perspective is added to the **Window > Open Perspective** menu.

Deleting a User-defined Perspective

You can delete perspectives that you defined yourself, but not those that are delivered with the Workbench.

1 Open the General > Perspectives preference page.

1 From the **Available perspectives** list, select the one that you want to delete and click **Delete**.

1 Click **OK**.

Resetting Perspectives

To restore a perspective to its original layout:

2 Open the General > Perspectives preference page.

3 From the **Available perspectives** list, select the perspective you want to restore.

4 Click **Reset**.

5 Click **OK**.

Working with Views And Editors

Views and editors are the main visual entities which appear in the Workbench. In any given perspective there is a single editor area, which can contain multiple editors, and a number of surrounding views which provide context.

The Workbench provides a number of operations for working with views and editors. See the Related tasks links for more details.

Opening Views

Perspectives offer pre-defined combinations of views and editors. To open a view that is not included in the current perspective, select **Window > Show View** from the main menu bar.

You can create *fast views* to provide quick access to views that you use often.

Moving and Docking Views

To change the location of a view in the current perspective:

1 Drag the view by its title bar. Do not release the left mouse button yet.

2 As you move the view around the Workbench, the mouse pointer changes to one of the *drop cursors* shown in the table below. The drop cursor indicates where the view will be docked if you release the left mouse button. To see the drop cursor change, drag the view over the left, right, top, or bottom border of another view or editor. You may also drag the view *outside* of the Workbench area to turn it into a "Detached" view.

3 When the view is in the location that you want, relative to the view or editor area underneath the drop cursor, release the left mouse button.

4 (Optional) If you want to save your changes, select **Window > Save Perspective As...** from the main menu bar.

5 Note that a group of stacked views can be dragged using the empty space to the right of the view tabs.

You can also move a view by using the pop-up menu for the view. (Left-click on the icon at the left end of the view's title bar, or right-click anywhere else in the view's title bar). As well as moving the view this menu will provide shortcut options for turning a view into either a "Fast" or "Detached" view.

Rearranging Tabbed Views

In addition to dragging and dropping (docking) views inside the Workbench, you can rearrange the order of views within a tabbed notebook.

1. Click on the tab of the view that you want to move and drag it to where you want it. A stack symbol appears as you drag the view across other view tabs.
2. Release the mouse button when you have the view tab in the desired location. The view that you selected is now moved.

Creating Fast Views

Fast views are hidden views that can be quickly opened and closed. They work like other views except they do not take up space in your Workbench window.

To create a fast view:

1. Click the title bar of the view that you want. Hold the mouse button down.
2. Drag the view to the Fast View bar and release the mouse button. By default the shortcut bar is located in the lower left corner of the window.

Alternatively, you can click on the button located on the left side of the Fast View bar which will present you with a selection of views. Choosing one of these views will add it to the Fast View bar immediately.

The icon for the view that you dragged now appears on the shortcut bar. You can look at the view by clicking its icon on the shortcut bar. As soon as you click somewhere else outside the view, it is hidden again.

To restore the view to its original location (and remove it from the Fast View bar), toggle the fast view item in the view button's context menu.

You can also create and restore fast views by selecting **Fast View** from the context menu that opens when you click the icon at the left side of the view's title bar.

Detaching Views

Detached views are views that are shown in a separate window with a smaller trim. They work like other views except that they are always shown in front of the Workbench window.

To detach a view:

1. If the Workbench window is maximized, resize it so that it does not fill the entire screen.
2. Click the title bar of the view that you want to detach. Hold the mouse button down.
3. Drag the view to the outside of the Workbench window and release the mouse button.
4. To restore the view to be shown inside of the Workbench window, drag it into the Workbench window.
5. You can also detach a view by selecting **Detached** from the context menu that opens when you click the icon at the left side of the view's title bar.

Opening File for Editing

You can launch an editor for a given file in several ways.

- By right-clicking the file in one of the navigation views and then selecting **Open** from the pop-up menu.
- By double-clicking the file in one of the navigation views.
- By double-clicking a bookmark that is associated with that file, in the Bookmarks view.
- By double-clicking an error or warning, or task record that is associated with that file, in the Problems view.

All of the above alternatives open the file in the default editor for that type of file. To open it in a different editor, select **Open With** from the file's pop-up menu.

Associating Editors with File Types

To associate editors with various file types in the Workbench:

1. Open the **General > Editors > File Associations** preference page.
2. Select the file type from the **File types** list, or click **Add** to add a type that is not already on the list.
3. In the **Associated editors** list, select the editor that you want to associate with that file type. To add an editor to the list:
   - Click **Add**. The Editor Selection dialog box opens.
   - Select **Internal Editors** or **External Programs**, depending on whether the editor that you want was built for the Workbench or runs outside the Workbench.
   - If you select **External Programs**, you can click the Browse button to browse the file system.
   - Select the editor from the list and click **OK**.
4. Click **OK** to finish associating the editor with the selected file type.

When you associate an internal editor with a file type, that editor opens in the editor area of the Workbench. For example, if you double-click a file in the Model Navigator or an entry in the Bookmarks or Tasks view it opens in the editor area.

Editors that support OLE document mode can also run in the editor area of the Workbench.

TIP: You can choose to override your default editor selections by selecting **Open With** from the pop-up menu for any resource in one of the navigation views.

Editing Files outside the Workbench

To edit a Workbench resource outside the Workbench:

1. Navigate in the file system to the Workbench's installation directory. Go into the workspace directory and open the file that you want to edit with the external editor.
2. Edit the file as needed. Save and close it as usual.
3. *Important*: Go back to the Workbench, right-click the edited file in one of the navigation views, and select **Refresh** from the pop-up menu. The Workbench will perform any necessary build or update operations to process the changes that you made outside the Workbench.

   TIP: If you work with external editors regularly, you may want to enable auto-refresh. This can be done by opening the **General > Workspace** preference page, and checking the **Refresh automatically** option. When this option is enabled, any external changes will be automatically discovered by the Workbench. Depending on the platform this may not happen immediately

Tiling Editors

The Workbench allows you to have multiple files open in multiple editors. Unlike views, editors cannot be dragged outside the Workbench to create new windows. However, you can tile editor sessions within the editor area, in order to view source files side by side.

1. With two or more files open in the editor area, select one of the editor tabs.
2. Holding down the left mouse button, drag that editor over the left, right, top or bottom border of the editor area. Notice that the mouse pointer changes to a "drop cursor" that indicates where the editor session will be moved when you release the mouse button.
3. (Optional) Drag the borders of the editor area or each editor, to resize as desired.

This is a similar operation to moving and docking views inside the Workbench, except that all editor sessions must be contained within the editor area.

Maximizing and Minimizing the Eclipse Presentation

The eclipse presentation provides a rich environment consisting of (in its basic form) an *Editor Area* (containing one or more stacks showing the open editors) surrounded by one or more *View Stacks* (each containing one or more views).These various parts compete for valuable screen real-estate and correctly managing the amount of screen given to each can greatly enhance your productivity within the IDE.

The two most common mechanisms for managing this issue are 'minimize' (i.e. make me use as little space as possible) and 'maximize' (i.e. give me as much space as you can). The eclipse presentation provides a variety of ways to access these operations:

1. Using the minimize and maximize buttons provided on a stack's border
2. Selecting the 'Minimize' or 'Maximize' item on the context (right-click) menu for a stack
3. Double-clicking on a stack
4. Using 'Ctrl + M': this is a key binding for a command that will toggle the currently active part between its 'maximized' and its 'restored' (i.e. normal) states.

Maximize:

It is desirable at times to focus your attention on one particular part to the exclusion of the others. The most popular candidate for this is, of course, maximizing the editor area in order to make as much of the display available for editing as possible (but there are workflows where it would make sense to focus on a view as well).

As of 3.3 the default presentation implements the maximize behavior by minimizing all stacks *except* the one being maximized. This allows the maximized stack to completely occupy the main presentation while still allowing access any open views in your perspective by using the icons in their *Trim Stack* (the area around the edges of the window is called the 'trim').

Also in 3.3 the behavior for managing the editor maximization has been changed to operate on the complete Editor Area (rather than simply maximizing the particular Editor Stack as is the case in the 3.0 and 2.1 presentations). This allows for 'compare' workflows which require the ability to see both files in a split editor area at the same time.

Minimize:

Another way to optimize the use of the screen area is to directly minimize stacks that are of no current interest. As of 3.3 the default presentation minimizing a stack will cause it to be moved into the trim area at the edges of the workbench window, creating a *Trim Stack*. View Stacks get minimized into a trim representation that contains the icons for each view in the stack:

The minimize behavior for the Editor Area is somewhat different; minimizing the Editor Area results in a trim stack containing only a placeholder icon representing the entire editor area rather than icons for each open editor (since in most cases all the icons would be the same, making them essentially useless).

If your particular workflow is such that you need to have more than one element (i.e. having the Editor Area *and* a View Stack in the presentation at the same time) you can still gain additional screen space by minimizing the stacks that aren't of current interest. This will remove them from the main presentation and place them on the outer edge of the workbench window as *Trim Stacks*, allowing more space for the remaining stacks in the presentation.

NOTE: There are two ways to end up with a stack in the trim:

- Directly minimizing the stack
- As the result of another stack being maximized

Depending on how the Trim Stack was created its behavior is different; when un-maximizing only those trim stacks that were created during the initial maximize will be restored to the main presentation while stacks that were independently minimized stay that way.

TIP: This difference is important in that it allows you fine grained control over the presentation. While using maximize is a one-click operation it's an 'all or nothing' paradigm (i.e. no other stack is allowed to share the presentation with a maximized stack). While adequate for most tasks you may find yourself wanting to have the presentation show more than stack. In these scenarios don't maximize; minimize all the other stacks *except* the ones you want in the presentation. Once you have it set up you can still subsequently maximize the editor area but the un-maximize will only restore the particular stack(s) that were sharing the presentation, not the ones you've explicitly minimized.

Customizing the Workbench

Many aspects of the appearance and behavior of the Workbench can be customized to suit your individual needs. For example, you can:

- Rearrange where items appear in the main toolbar.
- Change the key bindings used by editors.
- Change the fonts and colors which are used.

Customizing Welcome

The welcome appearance can be customized via the "customize page" button above the welcome page. This opens a customize dialog which allows you to select one of the pre-defined themes, which affects the overall look of the welcome. You can also select which pages will be displayed, and the visibility, layout, and priority of the items within each page.

Rearranging the Main Toolbar

You can rearrange sections of the main toolbar. Toolbar sections are divided by a thin vertical line.

1. Make sure the toolbar is unlocked. The toolbar is unlocked if it has thick vertical bars next to the thin vertical toolbar dividers.

   If it is locked, unlock the toolbar by right clicking the toolbar and selecting the **Lock the Toolbars** menu item.
2. Grab the section of the toolbar you want to rearrange by moving the mouse over the thick vertical line on the left side of the desired segment. The mouse cursor changes its shape to indicate that you can click to move the toolbar section.
3. Click and hold the left mouse button to grab the toolbar section.
4. Move the section left and right or up and down. Release the mouse button to place it in the new location.
5. To prevent accidental changes to the toolbar lock it again by right clicking the toolbar and selecting the **Lock the Toolbars** menu item.

Changing the Key Bindings

The function of the keyboard can be extensively customized in Eclipse.

Use the **General > Keys** preference page to assign key sequences to many of the commands in Eclipse.

Changing the Fonts and Colors

By default, the Workbench uses the fonts and colors provided by the operating system. However, there are a number of ways that this behavior can be customized.

Fonts

The Workbench lets you directly configure the following fonts:

Banner Font

> Used in PDE editors, welcome pages and in the title area of many wizards. For instance the New Project wizard uses this font for the top title.

Dialog Font

> Used for widgets in dialogs.

Header Font

> Used as a section heading. For instance the Welcome page for the Eclipse Platform uses this font for the top title.

Text Font

> Used in text editors.

CVS Console Font

> Used in the CVS console.

Ignored Resource Font

> Used to display resources that are ignored from CVS.

Outgoing Change Font

> Used to display outgoing changes in CVS.

Console Font (defaults to text font)

> Used by the debug console.

Detail Pane Text Font (defaults to text font)

> Used in the detail panes of debug views.

Memory View Table Font (defaults to text font)

> Used in the table of the memory view.

Properties File Editor Text Font (defaults to text font)

> Used by Properties File editors.

Compare Text Font (defaults to text font)

> Used by textual compare/merge tools.

Part Title Font (defaults to properties file editor text font)

> Used for view and editor titles. **Note:** It is recommended that this font not be bold or italic because the workbench will use bold and italic versions of this font to display progress.

View Message Font (defaults to properties file editor text font)

> Used for messages in the view title bar (if present).

To change these fonts:

1. Open the **General > Appearance > Colors and Fonts** preference page.
2. Select the font you want to change.
3. Click **Change**.
4. Use the dialog which opens to select a font.
5. Click **OK**.

NOTE: You can also click **Use System Font** to set the font to a reasonable value chosen by the operating system. For example, on Windows this will use the font selected in the Display Properties control panel.

Plug-ins that use other fonts may also provide preference entries to allow them to be customized.

In addition to the above, some text is always displayed in the system font. For example, the navigator tree always does this. To change the font used in these areas, you can use the configuration tools provided by the operating system (for example, the Display Properties control panel on Windows, or the .Xdefaults file in Motif).

Colors

To set the colors used by the Workbench to display error text and hyperlink text:

1. Open the **General > Appearance > Colors and Fonts** preference page.
2. Select the color you want to change in the tree view and click the color bar on the right.
3. Use the dialog which opens to select a color.
4. Click **OK**.

Plug-ins that use other colors may also provide preference entries to allow them to be customized. For example, the searching support provides a preference for controlling the color used to display potential matches (see the **Foreground color for potential matches** item on the **General > Search** preference page).

In general, the Workbench uses the colors that are chosen by the operating system. To change these colors you can use the configuration tools provided by the system (for example, the Display Properties control panel on Windows, or the .Xdefaults file in Motif).

Changing the Placement of the Tabs

You can change the placement of the tabs. The tabs for stacked views or editors can appear at the top or bottom of the area which contains them.

1. Open the **General > Appearance** preference page.
2. Select from the choices displayed in the **Editor tab positions** group or **View tab positions** group to control whether you want the tabs at the top or the bottom.
3. Click **Apply** or **OK**.
4. The tabs will immediately move to their new locations

Controlling Single and Double Click Behavior

You can control how the Workbench responds to single and double clicks. To do this:

1. Open the **General** preference page.
2. Select the behavior you want to change from the **Open mode** group.
3. Click **OK**.

The effect of these selections varies by view. For example, in one of the navigation views:

Double click

Will cause a single click on a resource to select it, and a double click to open it in an editor.

Single click

Will cause a single click on a resource to both select it and immediately open an editor on it.

The check boxes under the **Single click** radio button further refine the single click behavior. Checking **Select on hover** will cause the resource to be selected if you hover over it with the mouse. Checking **Open when using arrow keys** will cause the resource to be opened if you use the arrow keys to navigate to it.

Important and Exporting Preferences

Preference files can be both imported to and exported from the Workbench allowing you to easily share individual or group preferences.

The Import wizard can be used to import preferences from the local file system to the Workbench.

To import a preference file:

1. Select **File > Import**.
2. In the Import wizard select **General > Preferences** and click **Next**.
3. Click **Browse...** and locate the Preferences file on the local file system.

4 Select one of the following:
   - **Import all** to accept all of the preferences defined in the file.
   - **Choose specific preferences to import** to select only specified preferences defined in the file.

5 Click **Finish**.

The Export wizard can be used to export preferences from the Workbench to the local file system.

To export a preference file:

1 Select **File > Export**.

2 In the Export wizard select **General > Preferences** and click **Next**.

3 Select one of the following:
   - **Export all** to add all of the preferences to the file.
   - **Choose specific preferences to export** to add only specified preferences to the file.

4 Click **Browse...** and locate the preferences file on the local file system.

5 Click **Finish**

NOTE: If no changes have been made to the original preference settings the preferences file will be empty.

Working with Projects, Folders and Files

There are three different types of resources in the workbench: projects, folders, and files. Projects are the largest structural unit used by the Workbench. Projects contain folders and files, and they can be opened, closed, or built. Folders can contain other folders and files. The Workbench provides a number of mechanisms for working with projects, folders and files. See the related tasks section for more details.

Folders and files can be linked to locations in the file system outside of the project's location. These special folders and files are called linked resources.

Creating a Project

To create a project:

1 On the main menu bar, click **File > New Project**. The New Project wizard opens.

2 Select a category from the left column and then select the type of project to create from the right column. To assist in locating a particular wizard, the text field can be used to show only the wizards that match the entered text. Click **Next**.

3 In the **Project name** field, type a name for your new project.

4 (Optional) The project that you create will map to a directory structure in the file system. The default file system location is displayed in the Location field. If you want to create the project and its contained resources in a different location, clear the **Use default location** checkbox and specify the new location.

5 If you want the new project to be dependent on one or more other projects, click **Next** and select the projects to be referenced.

6 Click **Finish**. The new project is listed in one of the navigation views.

TIP: The **General > Perspectives >** preference page allows you to specify the perspective behavior when a new project is created.

Closing Projects

When a project is closed, it can no longer be changed in the Workbench and its resources no longer appear in the Workbench, but they do still reside on the local file system. Closed projects require less memory. Also, since they are not examined during builds, closing a project can improve build time.

To close a project:

1. Select the project in one of the navigation views.
2. Click **Close Project** on the pop-up menu.

To re-open the project:

1. Select the project in one of the navigation views.
2. Click **Open Project** on the pop-up menu.

Deleting Projects

To delete a project and remove its contents from the file system:

1. Select the project in one of the navigation views.
2. Click **Delete** on the pop-up menu.
3. In the dialog which opens select **Also delete contents under...**.
4. Click **Yes**.

To delete a project from the workspace without removing its contents from the file system:

1. Select the project in one of the navigation views.
2. Click **Delete** on the pop-up menu.
3. In the dialog which opens select **Do not delete contents**.
4. Click **Yes**.

Creating a Folder

To create a new folder:

1. In one of the navigation views, right-click the project or folder where you want to create the new folder.
2. From the pop-up menu, select **New > Folder.**
3. Enter the name of the new folder and click **Finish**.

Creating a File

To create a file:

1. In one of the navigation views, right-click the project or folder where you want to create the new file.
2. From the pop-up menu, select **New > File**.
3. Specify the name of the file, including the file extension (for example, newfile.txt).
4. Click **Finish**.

5 The file opens in the editor associated with its type

Creating Linked Resources

Folders and files can be linked to locations in the file system outside of the project's location. These special folders and files are called linked resources.

To create a linked folder:

1 In one of the navigation views, right-click the project or folder where you want to create the linked folder.

2 From the pop-up menu, select **New > Folder**.

3 Specify the name of the folder as it will appear in the workbench. This name can be different from the name of the folder in the file system.

4 Click **Advanced**.

5 Check **Link to folder in the file system**.

6 Enter a file system path, or click **Browse** to select a folder in the file system.

7 Click **Finish**.

To create a linked file, follow the same steps as above, except choose **New > File** instead of **New > Folder** in the context menu.

Linked resource locations can also be specified relative to a variable. This makes it easier to share projects containing linked resources with other team members, since it avoids hard-coded absolute file system paths that may vary from one machine to the next.

To define a linked resource relative to a path variable, do the following after step 5 above:

1 Click the **Variables** button.

2 In the resulting dialog, select an existing path variable or create a new one.

3 If the chosen variable defines the exact path of the linked resource, click **OK**. Otherwise, click **Extend** to specify a file or folder below the location described by the path variable, then click **OK**.

4 Click **Finish**.

TIP: The **General > Workspace > Linked Resources** preference page also allows you to define path variables.

Note that, once you create a linked resource you will not be able to change the link target path that you entered in step 6. or 8. above.

Moving Resources

You can move resources from one Workbench location to another (for example, from one project to another project).

1 In one of the navigation views, select the resources that you want to move.

2 From the pop-up menu, select **Move**.

3 In the Folder Selection window, select the project or folder where you want to move the resources to and click **OK**.

TIP: You can also move resources by dragging them from the original location to the new location in one of the navigation views.

Moving a linked resource always moves the link and not the resource that it is linked to. E.g., when moving a linked folder the folder contents is not moved on the file system.

Copying Resources

You can copy resources from one Workbench location to another (for example, from one project to another project).

1 In one of the navigation views, select the resources that you want to copy.

2 From the pop-up menu, select **Copy**.

3 In one of the navigation views, select the project or folder where you want to copy the resources to.

4 From the pop-up menu, select **Paste**.

> **TIP:** You can also copy resources by holding down the Ctrl key and dragging them from the original location to the new location in one of the navigation views.

Copying a linked resource always copies the link and not the resource that it is linked to. E.g., when copying a linked folder the folder contents is not copied on the file system. Instead, a new linked folder is created linking to the same location on the file system.

Renaming Resources

You can rename Workbench resources using the Rename command on the context menu in one of the navigation views.

1 In one of the navigation views, right-click the resource that you want to rename.

2 From the pop-up menu, select **Rename**.

3 Type the new name for the resource.

4 Hit the return key.

Deleting Resources

To delete a resource from the Workbench:

1 In one of the navigation views, select the resources that you want to delete. (Hold down the Ctrl key to select more than one resource.)

2 Press the Delete key.

3 Click **Yes** in the dialog which appears.

> **TIP:** You can achieve the same result by selecting **Delete** from the pop-up menu.

Deleting resources from the workspace will also delete the corresponding files and/or folders from the local file system. The only exception is linked resources, which are not removed from the file system when you delete the link in the workspace. However, deleting child resources of linked folders *will* delete those resources from the local file system.

In most cases it is possible to undo a resource delete, which will restore deleted files from the local history. The files can also be restored manually from the history. See the related tasks for more details.

Viewing Resources Properties

To display the various properties of a Workbench resource:

1. Right-click the resource in one of the navigation views.
2. Select **Properties** from the pop-up menu.

The kinds of properties that are displayed depend on the specific resource you have selected and the features and plug-in that are installed in the Workbench.

Tip: When the resource is selected in one of the navigation views you can also see basic properties for a resource by clicking **Window > Show View > Other... > General > Properties**.

Navigating and Finding Resources

The Workbench provides a number of mechanisms for navigating and finding resources. See the Related tasks links for more details.

Finding a Resource Quickly

To navigate to a particular resource in a view such as the Model Navigator view:

1. From **Navigate** menu, select **Go To > Resource**.
2. In the window that opens, start typing the name of the resource in the **Pattern** field. As you type the file name, the system offers a list of possible matches, based on what you have entered so far.
3. Select the resource that you want from the **Matching Resources** list, and click **OK**.

The view displays the selected resource.

If you want to open a particular resource in an editor rather than select it in a view, you can use the **Navigate > Open Resource** action. This action presents the same dialog as the **Go To > Resource** action, but immediately opens the matching resource for editing.

You can also do a contextual search for character strings contained within files in the Workbench. See the links to related tasks below.

Searching for Files

The **Go To > Resource** action in the **Navigate** menu allows you to quickly find a resource in the Workbench by searching resource *names*.

You can also do more complex searches for files in the Workbench. For example, to find all files that end with .xml:

1. On the main toolbar, click the **Search** button.
2. Type `.xml` into the **File name patterns** field and leave the **Containing text** field empty.
   (You can use the pull-down list to select `.xml` if it had been previously entered.)
3. Finish entering your search options, for example to scope the search to specified working sets, and click **Search**.
4. The Search view displays the results of your search. Right-click on any item in the Search view to open a pop-up menu that allows you to remove items from the list, copy search results to the clipboard, or rerun the search. To open one of the listed files, double-click it or select **Go to File** from its pop-up menu.

If you close the Search view, you can return to it later by selecting **Window > Show View > Other... > General > Search**.

Searching for Text Within a File

The **Go To** selection in the pop-up menu for one of the navigation views allows you to quickly find a resource in the Workbench by searching resource *names*.

You can also do contextual searches for information that is contained *inside* files in the Workbench. To find all files that contain a particular string of characters:

1. On the main toolbar, click the **Search** button.
2. Type your search string in the **Containing Text** field, or use the pull-down list to select a previously entered search expression.
   Use `\` as an escape character for search strings that contain the special characters `*`, `?`, or `\`, (for example: `d:\\directory\\filename.ext`).
3. Finish entering your search options, (for example, to scope the search to specified file types), and click **Search**.
4. The Search view displays the results of your search. Right-click on any item in the Search view to open a pop-up menu that allows you to remove items from the list, copy search results to the clipboard, or rerun the search. To open one of the listed files, double-click it or select **Go to File** from its pop-up menu.

If you close the Search view, you can return to it later by selecting **Window > Show View > Other... > General > Search**.

Problems, Bookmarks, Tasks, and Other Markers

Markers are objects that may be associated with Workbench resources. There are many uses of markers in the Workbench, including providing support for bookmarking resources or locations within resources, tracking ongoing tasks, or displaying error messages. See the related tasks section for more details.

Creating a Bookmark within a File

The Workbench allows you to create bookmarks in files that you edit so that you can quickly reopen those files from the Bookmarks view **(Window > Show View > Other... > General > Bookmarks)**.

1. With the file open in an editor, right-click in the gray border at the left of the editor area, next to the line of code or text that you want to bookmark.
2. Select **Add Bookmark** from the pop-up menu.
3. Notice that an icon for the bookmark now appears in the left border of the editor area. A line is also added to the Bookmarks view.)

You can reopen the file for editing at any time by double-clicking the bookmark in the Bookmarks view.

Creating a Bookmark for an Entire File

You can bookmark individual files in the Workbench, in order to open them quickly from the Bookmarks view (**Window > Show View > Other... > General > Bookmarks**) later.

1. In one of the navigation views, select the file that you want to add to your list of bookmarks.
2. From the main Workbench menu select **Edit > Add Bookmark**.

At any time, you can open the bookmarked file for editing by double-clicking the bookmark in the Bookmarks view.

You can also create bookmarks for specific lines of text or source code or within a file. See the list of related topics below.

Deleting a Bookmark

To delete any bookmark:

1 Open the Bookmarks view (**Window > Show View > Other... > General > Bookmarks).**

2 Right-click the bookmark that you want to delete and select **Delete** from the pop-up menu.

If you have added bookmarks for specific lines within a source file, you can also delete them while editing the file by right-clicking the bookmark icon in the editor area.

Adding Line Items in the Tasks View

The Tasks view contains line items for system-generated problems, warnings, and errors. You can add your own entries to the table to build a list of to-do items, or tasks.

1 On the toolbar in the Tasks view, click the **New Task** button. The **New Task** dialog will open.

2 Type a brief description for the task in the **Description** field. The new task is assigned default priority and completed values. These values may also be modified within the **New Task** dialog.

3 Press **OK**.

Associating a Task with a Resource

You can associate tasks with an editable resource, for instance to remind yourself to update a line of source code later.

1 In one of the navigation views, double-click the resource with which you wish to associate the new task. The resource opens in the editor area.

2 Right-click in the gray border at the left of the editor area, beside the line of text or source code against which you want to log the new task.

3 On the pop-up menu, select **Add Task**.

4 When prompted, enter a brief description of the task.

A new task icon appears in the border of the editor area, to the left of the line where you added the task. When you move the mouse pointer over the marker, the description of the task is displayed as a tooltip. The task is also added to the Tasks view. You can delete a task either by right-clicking its icon in the editor area and selecting **Remove Task**, or by pressing the Delete key in the Tasks view.

Deleting Tasks

You can delete associated tasks from the gray border at the left of the editor area.

1 In the marker bar in the editor area, locate the task marker that you want to delete.

2 From the marker's pop-up menu, select **Remove Task**.

3 The task marker disappears and the task is removed from the Tasks view.

You can also delete one or more line items from the Tasks view by pressing the Delete key or by clicking the **Delete** button on the Tasks view toolbar, or from the pop-up menu.

TIP: To delete all completed tasks, right-click in the Tasks view and select **Delete Completed Tasks** from the pop-up menu.

Filtering the Tasks and Problems Views

You can filter the tasks or problems that are displayed in the Tasks or Problems views. For example, you might wish to see only problems that have been logged by the Workbench, or tasks that you have logged as reminders to yourself. You can filter items according to which resource or group of resources they are associated with, by text string within the Description field, by problem severity, by task priority, or by task status.

1. On the toolbar of the Tasks or Problems view, click **Filter**.
2. Select the radio buttons and check boxes that correspond to your filtering objectives.
3. Click **OK**.

Comparing Resources

When a comparison is performed, comparison editors appear in the editor area. The differences between files are highlighted in the comparison editors, allowing you to browse and copy changes between the compared resources.

To compare resources:

1. Select one or more resources in one of the navigation views.
2. From the resource's pop-up menu, select **Compare With**.

A tool for viewing differences is opened in the editor area.

Here are some of the comparisons that you can perform.

Compare With > Latest from HEAD

compares the selected resource with the version of the same resource that is currently committed to the active branch. This option is available for CVS. This, or a similar option, may or may not be available for other repository types.

Compare With > Another Branch or Version

compares the selected resource with a version that has been committed to the repository or to the latest version in a particular branch. This requires you to choose a version of the resource or a branch from a list. This option is available for CVS. This, or a similar option, may or may not be available for other repository types.

Compare With > Each Other

compares two or three selected resources with each other

Compare With > History...

compares the selected resource with a version of the same resource that was committed to the active branch. This action will open the History view in *compare mode*. You will then be able to choose which revision you wish to compare by double clicking on a revision in the History view. This option is available for CVS. This, or a similar option, may or may not be available for other repository types.

Compare With > Local History

compares the selected resource to one that is in the local history, which is maintained when you save changes. (*Tip:* to configure the size and depth of your local history, use the **General > Workspace > Local History** preference page.)

TIP: You can customize the behavior of the comparison editor with the **General > Compare/Patch** preference page. You can select to "synchronize scrolling" of the pane contents, as well as setting options for showing conflicts with other team members and changing the default font

Setting Preferences for Comparing Files

When you select to compare or synchronize two or more resources in the Workbench, one or more comparison editors usually open. To customize how these editors behave:

1. Open the **General > Compare/Patch** preference page.
2. Set your preferences and click **OK**.

Understanding the Comparison

If you compare two files (file1.txt and file2.txt), the following results are shown in the compare editor. The left side shows the contents of file1.txt and the right side shows the contents of file2.txt. The lines connecting the left and right panes indicate the differences between the files.

You can double click on the editor tab to maximize the editor.

Merging Changes in the Compare Editor

The toolbar buttons in the compare editor allows you to merge changes from the left file to the right file and vice versa. There are four types of merges you can perform:

- Copy All from Left to Right
- Copy All from Right to Left
- Copy Current Change from Left to Right
- Copy Current Change from Right to Left

The **Copy All from Left to Right** and **Copy All from Right to Left** actions completely replace the contents of one resource with the other resource.

To merge a single change:

1. Select the highlighted difference that you want to merge.
2. Depending on what you want to do, click either the **Copy Current Change from Right to Left** or the **Copy Current Change from Left to Right** toolbar button. The selected text is copied from one file to the other.
3. Right click to get the resource's pop-up menu, and select **Save**.

Working with Local History

A local edit history of a file is maintained when you create or modify a file. Each time you edit and save the file, a copy is saved so that you can replace the current file with a previous edit or even restore a deleted file. You can also compare the contents of all the local edits. Each edit in the local history is uniquely represented by the date and time the file was saved. See the Related tasks links for more details.

Comparing Resources with the Local History

To compare an unmanaged Workbench resource with a state in the local history:

1. In one of the navigation views, select the resource that you want to compare with a local history state.
2. From the resource's pop-up menu, select **Compare With** > **Local History**. The Compare with Local History page opens.

3 Select a state in the **Local History** list. The Text Compare editor opens.

4 Click the **Select Next Change** and **Select Previous Change** buttons to browse the changes made between the state in the local history and the Workbench resource.

5 Click **OK** when you are finished.

To compare a Workbench resource that is managed by CVS with a state in the local history:

1 In one of the navigation views, select the resource that you want to compare with a local history state.

2 From the resource's pop-up menu, select **Compare With** > **History...**. The History View opens.

3 Put the History View into *Local Revisions* mode by clicking on the Local Revisions toolbar button in the History View. This filters out all revisions except for Local History revisions.

4 Select a revision from the History View. The Text Compare editor opens.

5 Click the **Select Next Change** and **Select Previous Change** buttons to browse the changes made between the state in the local history and the Workbench resource.

6 Click **OK** when you are finished.

Replacing a Resource with Local History

To replace an unmanaged Workbench resource with a state in the local history:

1 In one of the navigation views, select the resource that you want to replace with a local history state.

2 From the resources pop-up menu, select **Replace with** > **Local History**. The Replace from Local History page opens.

3 Select a state from the **Local History** list. The Text Compare editor opens.

4 Click the **Select Next Change** and **Select Previous Change** buttons to browse through the changes made between states in the local history and the Workbench resource.

5 Select the state you want to replace, and click **Replace**.

To replace a Workbench resource managed by CVS with a state in the local history:

1 In one of the navigation views, select the resource that you want to replace with a local history state.

2 From the resources pop-up menu, select **Replace with** > **History...**. The History View opens.

3 Put the History View into *Local Revision* mode by clicking on the Local Revision toolbar button. This will display only Local History revisions in the History View.

4 Select the revision you want to replace the current revision and select **Get Contents** from the context menu.

TIP: You can configure your general preferences to specify how many days to keep files, or how many entries per file you want to keep, or the maximum file size for files to be kept with the **General > Workspace > Local History** preference page.

Restoring Deleted Resources from Local History

To restore a deleted Workbench resource with a state from the local history:

1 In one of the navigation views, select the folder or project into which you want to restore a local history state.

2 From the resource's pop-up menu, select **Restore from Local History...**. The Restore From Local History dialog opens showing all files that were previously contained in the selected folder or project and all of their sub-folders.

3 Check the files that you want to restore

4 If you don't want to restore just the last state of a file you can select any other state of the file from the **Local History** list on the right hand side of the dialog. The bottom pane of the dialog shows the contents of the state.

5 If you are done with all files click **Restore**.

TIP: You can configure your Workbench preferences to specify how many days to keep files, or how many entries per file you want to keep, or the maximum file size for files to be kept with the **General > Workspace > Local History** preference page.

Setting Local History Preferences

To indicate the level of local history that should be kept for each resource in the Workbench:

1 Open the **General > Workspace > Local History** preference page.

2 In the **Days to keep files** field, type the number of days that you want to keep records for any one Workbench resource. For example, if you type 7, then a history of saved states from the last seven days will be kept.

3 In the **Maximum entries per file** field, type the number of states to keep for any one Workbench resource. Note that when you exceed the number of entries per file, the oldest changes are discarded to make room for the newer changes.

4 In the **Maximum file size (MB)** field, type the maximum file size (in MB) of a resource for which a local history should be kept. If the size of the resource exceeds the maximum amount of file size allocated, no local history is kept for that resource.

5 Click **OK** to set your preferences and close the Local History Preferences page.

Importing

You can import files into the Workbench in several ways, depending on your operating system:

- By using the import wizard
- By dragging files or folders from the file system to one of the navigation views
- By copying files or folders from the file system and pasting them into one of the navigation views

Importing Existing Projects

You can use the Import Wizard to import an existing project into workspace.

1 From the main menu bar, select **File > Import....** The Import wizard opens.

2 Select **General > Existing Project into Workspace** and click **Next**.

3 Choose either **Select root directory** or **Select archive file** and click the associated **Browse** to locate the directory or file containing the projects.

4 Under **Projects** select the project or projects which you would like to import.

5 Click **Finish** to start the import.

Importing Resources from the File System

You can use the Import Wizard to import resources from the local file system into an existing project.

1 From the main menu bar, select **File > Import....** The Import wizard opens.

2 Select **General > File System** and click **Next**.

3 Click the **Browse** button on the next page of the wizard to select the directories from which you would like to add the resources.

4 In the import selection panes, use the following methods to select exactly the resources you want to add:

- Expand the hierarchies in the left pane and select or clear the check boxes that represent the folders in the selected directory. Then in the right pane, select or clear check boxes for individual files.
- Click **Filter Types** to filter the current selection for files of a specific type.
- Click **Select All** to select all resources in the directory, then go through and deselect the ones that you do not want to add.
- Click **Deselect All** to deselect all resources in the directory, then go through and choose individual resources to add.

5 Specify the Workbench project or folder that will be the import destination.

6 When you have finished specifying your import options, click **Finish.**

TIP: You can also import folders and files by dragging them from the file system and dropping them into one of the navigation views, or by copying and pasting.

Importing Resources from an Archive File

You can use the Import wizard to extract files from an archive file into the Workbench.

1 From the main menu bar, select **File > Import....** The Import wizard opens.

2 Select **General > Archive File** and click **Next**.

3 Click the **Browse** button on the next page of the wizard, to select the archive files that contain the files you want to extract and import into the Workbench.

4 In the import selection panes, use the following methods to select exactly the resources you want to add:

- Expand the hierarchies in the left pane and select or clear the check boxes that represent the folders in the selected directory. Then in the right pane, select or clear check boxes for individual files.
- Click **Filter Types** to filter the current selection for files of a specific type.
- Click **Select All** to select all resources in the directory, then go through and deselect the ones that you do not want to add.
- Click **Deselect All** to deselect all resources in the directory, then go through and choose individual resources to add.

5 Specify the Workbench project or folder that will be the import destination.

6 When you have finished specifying your import options. click **Finish.**

Exporting

You can export files from the Workbench in several ways, depending on your operating system:

- By using the Export wizard

- .By dragging files or folders from one of the navigation views to the file system.
- By copying files or folders from one of the navigation views and pasting them into the file system.

Exporting Resources to the file System

You can use the Export wizard to export resources from the Workbench to the file system.

1. In one of the navigation views, select the resources that you want to export.
2. From the main menu bar, select **File > Export....** The Export wizard opens.
3. Select **General > File System** and click **Next**.
4. By default, the resources that you selected will be exported, along with all their children. Optionally, use the check boxes in the left and right panes to select the set of resources to export, and use push buttons such as **Select Types** to filter the types of files that you want to export.
5. Click the **Browse** button on the next page of the wizard, to select the directory you would like to export the resources to.
6. Specify the directory in the file system that will be the export destination.
7. Click **Finish**.

   TIP: You can also export folders and files by dragging them from one of the navigation views to the file system and dropping them in the file system, or by copy and paste.

Importing Resources from an Archive File

You can use the Export wizard to export resources from the Workbench to an archive file in the file system.

1. In one of the navigation views, select the resources that you want to export.
2. From the main menu bar, select **File > Export...**. The Export wizard opens.
3. Select **General > Archive File** and click **Next**.
4. By default, the resources that you selected will be exported along with their children. Optionally, use the check boxes in the left and right panes to select the set of resources to export, and use push buttons such as **Select Types** to filter the types of files that you want to export.
5. Specify the path and name of the archive file into which you want to export the selected resources.
6. Click **Finish.**

Accessing Help

The workbench provides a number of ways to access help information, including context-sensitive help and online documentation. See the **Related tasks** links below for details.

Navigating Help Topics

You can browse help topics using the <XREF> Help window or <XREF> Help view. Which one you use is simply a matter of preference; the view is well integrated with the workbench and is good for quick help lookups. The Help window, on the other hand, has more real estate and may be preferable for longer reading.

Using the Help window

The Help window is a window separate from the workbench used exclusively for browsing, searching, and printing help. To open the window, select **Help > Help Contents** from the menu. This opens the help window with the Contents view visible, which shows the table of contents.

To navigate the Help window:

1. Find the topic you want to read in the table of contents by clicking to expand the subtopics.
2. Most topics provide a list of links to related topics at the bottom. Follow these links to learn more.
3. Use the **Go Back** and **Go Forward** buttons to navigate back and forth. These behave the same way as in Web browsers.
4. Use the **Home** button to return to the help home page as shown above.

Maximizing help frames

The two main frames of the Help window can be maximized to take up the entire window. To maximize a frame, click the **Maximize** button in the toolbar, or double click on the view's title header. To return the view to its original size, click the **Restore** button or double click the title header again.

Printing help

To print a topic from the Help window:

1. Select the topic in the navigation frame.
2. Click the **Print** button in the Help toolbar.
3. Select the desired printer settings, and click **Print**.

Using the Help view

The Help view provides the same features as the Help window, but in the same window as the workbench instead of a separate one.

Capabilities

To show documentation about capabilities that are disabled in the application, select the **Show All Topics** button. When you choose to show all topics in the table of contents, the headings for documentation about any disabled activities are shown in the table of contents and also appear in search results.

Show in table of contents

While reading a topic, you can find out where the topic is located in the table of contents by clicking the **Show in table of contents** button in the toolbar. This will display the table of contents with the topic highlighted.

Searching Help

The help system includes a search engine that can run simple or complex queries on the documentation to help you find the information you are looking for.

To search help:

1. From the main menu, select **Help > Search**
2. Type in the word or phrase for which you want to search
3. Click **GO** or press Enter. The list of results will be displayed below

4 To view the content of a topic in the list of results, click on it

Alternatively, you can search from the Help window using the *Search* field at the top of the window.

Refining the search results in the help view

If the search yields too many results, the information you are looking for may not appear in the top 10 or 15 results. You can then refine the search to reduce the number of results.

To refine a search:

1. Click the **Search Scope** link. to expand search scope section
2. Click on the **Advanced Settings** link. The Search Scope preference dialog will open
3. Select **Local Help** from the list
4. Select the **Search only the following topics** button to narrow down the search scope
5. In the working set content tree, select the topics to which you want to narrow the search
6. Click **OK** to activate the changes and return to search page in the Help view
7. Click **GO** again. The new list of results will appear

Changing the appearance of the results

Two buttons on the toolbar can be used to change the way results are displayed. The **Show result categories** button, when pressed, will cause the results to be grouped by book. The **Show result descriptions** button, when pressed, causes descriptions to show.

Highlighting Search Terms

By default, when a search result is selected, the search terms that were used to find the document will be highlighted. By using the **Highlight Search Terms** toolbar button, you can toggle this feature on and off. This button is available in both the help window and the help view and each will remember the state of the button for displaying subsequent search results.

Query syntax

Follow the following search expression rules for searching local help content:

Unless otherwise stated, there is an implied AND between all search terms. In other words, topics that contain all the search terms will be returned.

- Use OR before optional terms. For example:

  applet OR application

  returns topics that contain the word *applet* or the word *application* (or both).

- Use NOT before terms you want to exclude from search results. For example:

  servlet NOT ejb

  returns topics that contain the word *servlet* and do not contain the word *ejb*. **Note:** NOT only works as a binary operator (that is, "NOT servlet" is not a valid expression).

  Use ? for a single-character wildcard and * for a multi-character wildcard. For example:

  par?

  returns topics that contain *part* or *park*, but not *participate*. On the other hand:

  par*

  returns topics that contain *part*, *park*, *participate*, *pardon*, and so on. **Note:** The search engine does not accept terms with a wild card at first character position.

- Use double quotation marks around terms you want treated as a phrase. For example:

  "creating projects"

  returns topics that contain the entire phrase *creating projects*, and not *creating* or *project* on its own.

- Punctuation acts as term delimiters. For example:

  plugin.xml

  returns hits on topics that contain *plugin.xml*, *plugin*, and *xml*, which is likely broader than you want. If you want to find just those topics containing *plugin.xml*, use double quotes, as in:

  "plugin.xml"

- The search engine ignores character case. For example:

  Workbench

  returns topics that contain 'workbench', 'Workbench', 'WorkBench', and 'WORKBENCH'.

- The following stop words are common English words which will be ignored (not searched for) if they appear in the search expression: a, and, are, as, at, be, but, by, in, into, is, it, no, not, of, on, or, s, such, t, that, the, their, then, there, these, they, to, was, will, with.

- The search engine does "fuzzy" searches and word stemming. If you enter *create*, it will return hits on topics that contain *creates*, *creating*, *creator*, and so on. To prevent search engine from stemming terms, enclose them in double quotes.

Extending the search scope

If you cannot locate information in the local help, you can extend search scope to remote info-center or search engines.

To enable search engines:

1. Click the **Search Scope** link. to expand search scope section. The list of search engine is displayed.
2. Select the ones that contain information you are looking for.

In addition to search engines provided, you may define additional search engines.

To define a new search engine:

1. Click the **Search Scope** link. to expand search scope section.
2. Click the **Advanced Settings** link. Search Scope preference dialog opens.

3. Click **New**.
4. Select the search engine type.
5. Click **OK**.
6. Provide a name and a description
7. Select engine specific settings and scope below. For the remote search engines, accessed using URL, fill in a full URL to query the engine. Use `{expression}` in the place of search expression.

Defining multiple search scopes

By default, changing search scope modifies the search scope named "default". You can define multiple search scope. They will be saved, allowing to quickly change search scope to one of them.

To define a new search scope:

1. Click the current search scope name, beside the **Search Scope** link.Search Scope Sets dialog appears.
2. Click **New**.
3. Type a name, and confirm.
4. Select the newly created search scope.
5. Click **OK**. The new search scope becomes current.

Changes to the search scope affect current search scope.

Search index generation

The first time you search the online help, the help system might initiate an index-generation process. This process builds the indexes for the search engine to use. It may take several minutes, depending on the amount of documentation and whether prebuilt indexes are installed. Results of the search will be available upon completion of the indexing process.

Each time you add or modify the documentation set (for example, when you install a new feature or update an existing one), the index will be updated to reflect the new information set.

Accessing Context-Sensitive Help

To access the context-sensitive help for a given widget:

1. Select the widget by putting focus on it (see the table below).
2. Press the **`F1`** key (**`Shift+F1`** on Linux, **`Help`** on the Mac). Or, in dialogs, click on the **Help** icon in the button bar.

This displays a description of the selected widget, and usually, a list of links to related information, in the help view. If you want more information than what is shown in the description, click a link to one of the related topic, or one of the search results appearing in the dynamic help section in the help view

Help Display Settings

External browser

The Help system can be accessed in Help view, or separate Help window. The window can be the default window with an embedded browser, or a full external browser of your choice.

If the embedded browser is supported on your system, help will use it by default to display help. If you prefer to always use a full external browser, you may select this behavior in the **Help preference page**.

Help Accessibility

The help browser uses your operating system's settings for the font colors, styles, and sizes. Users with visual impairments may wish to change some of these settings to increase the readability of the documentation.

In addition, on Windows platforms using Microsoft Internet Explorer, the help browser uses a component of Internet Explorer to display documentation, so changes you make to its display settings also affect the help display. To change the help browser's font and color settings:

1. Open Microsoft Internet Explorer.
2. Select **Tools > Internet Options**.
3. On the **General** page, click the **Colors**, **Fonts**, or **Accessibility** button.
4. Set the formatting options you desire.
5. Optionally, you can specify a cascading style sheet (CSS) to apply to the content.
6. Click **OK** and exit Internet Explorer.
7. Restart the Workbench. Open the Help perspective and browse the documentation to see the changes.

   NOTE: The help system also uses the Icon font setting on the Display Properties **Appearance** tab.

For more information about creating a CSS, consult a CSS reference. The W3 Consortium (www.w3.org) has an extensive collection of information about CSS and links to valuable resources.

Working in the Team Environment with CVS

The Workbench provides tools to manage, share and synchronize resources. The CVS standard is supported by default. See the Related tasks links below for more details.

Working with a CVS Repository

In the CVS team programming environment, team members do all of their work in their own Workbenches, isolated from others. Eventually they will want to share their work. They do this via a CVS repository. The Workbench provides a number of mechanisms which support working with CVS repositories.

Creating a CVS Repository Location

Prerequisite: A CVS server must already be configured on the host machine to create a valid repository location in the Workbench.

To create a new repository location:

1. Open the CVS Repositories view by selecting **Window > Show View > Other... > CVS > CVS Repositories**. Alternatively, the CVS Repositories view is also shown in the CVS Repository Exploring perspective.
2. On the toolbar, click on **Add CVS Repository** (or from the context menu of the CVS Repositories View, select **New > Repository Location**). The Add CVS Repository wizard opens.

3 Enter the information required to identify and connect to the repository location:

In the **Host** field, type the address of the host. (For example: `mymachine.com`).

In the **Repository path** field, type the path to the repository on the host (for example `/home/repo, d:/repo.`)

In the **User** field, type the user name under which you want to connect to the repository.

In the **Password** field, type the password for the above user name.

From the **Connection Type** list, select the authentication protocol of the CVS server. There are three connection methods that come with the Eclipse CVS client:

- **pserver** - a CVS specific connection method.

ext - the CVS ext connection method which uses an external tool such as SSH to connect to the repository. The tool used by ext is configured in the **Team > CVS > EXT Connection Method** preference page.

If the host uses a custom port, enable **Use Port** and enter the port number.

4 *Optional:* Select **Validate Connection on Finish** if you want to authenticate the specified user to the specified host when you close this wizard. (If you do not select this option, the user name will be authenticated later, when you try to access the contents of the repository.)

5 *Optional:* Select **Save Password** if you want to save the password between sessions, so you do not have to enter the password again the next time you start Eclipse. To persist login credentials in a safe, encrypted form Secure Storage is used.

6 Click **Finish**. The repository location is created.

Password Management

When creating a repository location, an option is provided at the bottom of the wizard to save the password between sessions.

In previous releases, passwords were automatically saved between sessions whereas now you have full control over this behavior. To persist login credentials in a safe, encrypted form Secure Storage is now used.

You can manage the list of passwords that have been saved via the **General > Security > Secure Storage** preference page. The page displays the currently saved passwords and allows you to clear or delete them all.

ADiscarding a CVS Repository Location

NOTE: You cannot discard a location if you have projects in the Workbench that are shared with that location.

To discard a CVS repository location:

1 Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.

2 In the CVS Repositories view, select the repository location that you want to delete from the Workbench.

Select **Discard Location** from the pop-up menu or press the DELETE key. Press **Yes** when prompted.

Refreshing the CVS Repositories View

There are two ways to refresh the CVS Repositories view in order to see the current connection status of all repositories:

- Click the **Refresh View** button on the view's toolbar

- Press F5 on the keyboard

NOTE: Refreshing the CVS Repositories view may change the resources displayed by the view but does not change the branch and version tags known to the view. Refreshing the tags is discussed in Discovering branch and version tags

Discovering Branch and Version Tags

Prerequisite: Before you can discover the branch and version tags for a repository, you must have a valid repository location in the CVS Repositories view.

To discover existing branch tags for multiple folders in a single repository location:

1 Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.

2 In the CVS Repositories view, select a repository location.

From the pop-up menu for the CVS Repositories view, select **Refresh Branches**. The Refresh Branches dialog will open (depending on the repository location selected listed remote projects can differ):

1 In the folder list, select the folders whose tags you would like to refresh. Alternatively, at the bottom of the dialog, you can choose a working set and all remote folders whose names match the name of a project in the working set will become checked.

2 Click **Finish** to discovered the tags for the checked projects. These tags will be added to the repositories view.

NOTE: This operation may be long running especially if the number of selected folders is large.

TIP: If the above operation does not result in the discovery of any tags, it is probable that the folder in the repository does not have a .project file in it. The .project file is the default file used by the tag discovery process. The file can be changed for individual projects using **Configure Branches and Versions** (see below).

To discover existing branch and version tags for a single project folder:

1 Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.

2 In the CVS Repositories view, expand the repository location.

3 Expand **HEAD** and select the root folder whose tags are to be discovered.

4 From the pop-up menu for the CVS Repositories view, select **Configure Branches and Versions**. The Tag Configuration dialog similar to the one below will open.

5 In the top pane on the left (titled *Browse files for tags*), expand one or more projects to find a file that is known to have tags that should be added to the CVS Repositories view.

6 Select a file in the left top pane. If the file selected has tags defined on it, they will be displayed in the right top pane (titled *New tags found in selected files*).

7 Check or uncheck the tags in the right pane as desired.

8 Click **Add Checked Tags** to add the checked tags to the list of remembered tags.

9 Click **OK** to add the remembered tags to the selected repository location or project.

NOTE: You can also set the auto-refresh files by selecting a file in the files pane and clicking **Add Selected Files** in the lower right hand portion of the dialog.

Changing the properties of a CVS repository location

You can change the connection properties of an existing repository location.

To change the properties of a CVS repository location:

1. Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.
2. In the CVS Repositories view, select the repository location whose properties you want to change.
3. Select **Properties** from the pop-up menu. The properties dialog will open.
4. Select the **CVS** tab.
5. Change one or more of the location's properties.
6. Click **OK** to change the properties.

   NOTE: The properties will be changed for all projects shared with the selected repository location

Changing the encoding of a CVS Repository Location

You can change encoding of an existing repository location. This encoding will be used to translate file paths and commit messages from the client encoding to the server encoding and vice versa. The encoding does not affect the contents of file. You may use the editor encoding (available from the **General > Content Types** preference page) or resource encodings (available from the **Properties > Resource** page of resources) to configure the Workbench to use the proper file encodings.

To change the server encoding of a CVS repository location:

1. Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.
2. In the CVS Repositories view, select the repository location whose properties you want to change.
3. Select **Properties** from the pop-up menu. The properties dialog will open.
4. Select the **Server Encoding** tab.
5. Choose to set the encoding and either choose or type in the desired encoding.
6. Click **OK** to change the properties.

Setting the content type of a file extension

CVS repositories distinguish between files that contain ASCII data and those that do not. For ASCII files, additional helpful functionality is available. This includes:

- Proper end of line conversion between the client and the server. This ensures that a Windows user and a UNIX user can work on the same file without introducing incompatibilities.
- Auto merging of conflicts. If a file contains both incoming and outgoing changes but none on the same line as another, then the file may be automatically merged when updating.

Certain file types invariably contain either ASCII data or binary data. For example, *.txt files usually contain ASCII data, and *.exe files usually contain binary data. Eclipse comes with a pre-defined set of file types, which you may add to or modify.

To set the content type associated with a file extension:

1. Open the **Team > File Content** preference page. This page displays a list of file extensions and content type (ASCII or Binary) for file extensions whose content type is known.
2. To add a file extension, click the **Add Extension** button and enter the file extension in the text prompt that appears. Once **OK** is clicked, the extension will be added to the list with a content type of ASCII.

3 To change the content type for an existing file extension, select the file extension entry and click the **Change** button. This will toggle the type from ASCII to Binary or vice versa.

4 To remove a file extension, select the file extension entry and click the **Remove** button.

Working with Projects Shared with CVS

The Workbench provides a number of mechanisms which support working with projects shared via CVS. See the Related tasks links for more details.

Enabling the CVS resource decorations

When enabled, CVS will decorate resources in Workbench views with icon and label decorations that indicate the CVS state of the resource.

To enable the CVS decorations:

1 Open the **General > Appearance > Label Decorations** preference page. This page allows the enabling and disabling of all decorations defined by different plug-ins in the Workbench.

2 Click on the CVS box to enable the CVS decorations.

3 Click **OK**.

NOTE: In some cases, the decorations may not appear immediately due to the fact that they are expensive to compute. In those cases, they are calculated in the background.

To configure the CVS decorations:

1 Open the **Team > CVS > Label Decorations** preference page. This page allows the configuration of how decorations appear on CVS projects, folders and files.

2 To configure the text label decorations click the **Text** tab and then:

- Pick the resource type (file, folder, project) for which you want to change the decoration.
- Click in the decoration string for that resource type at the location where the new decoration element should go.
- Click the **Add Variables** button for that resource type.
- Choose the CVS information variable to add. For instance, for files you could select the *added_flag* which will be inserted in the decoration where you placed the cursor. The string *{added_flag}* will be added to the decoration string which will decorate newly added resources using the added resource label (which can be set in the *Label Decoration for Added* field).

3 Repeat the above for any other decorations to be added.

4 Decorations can be removed by deleting the decoration variable name (i.e. *{added_flag}*) from the decoration string. In a similar manner, spacing can be added.

5 To configure icon decorations, click the **Icon** tab simply enable/disable the icon check boxes.

6 Click **OK** or **Apply** for the new decorations to take effect.

NOTE: Under the **General** tab is an option to disable the deep dirty decoration of folders. This will make decorator determination more efficient but will make it harder to find dirty resources in one of the navigation views. Under the **Synchronize View** tab, the Synchronize view can be configured to show the synchronization state of a resource as part of the text label displayed in the Structure Compare pane.

You can also change the color and font of items based on the item's CVS state. To set the color and font:

1. Open the **Team > CVS > Label Decorations** preference page.
2. Check the box to *Enable font and color decorations*.
3. Next, Open the **General > Appearance > Colors and Fonts** preference page.
4. In the **CVS** category are the various CVS states for which you can set the colors and fonts. Set them as desired.
5. You can return to the **Team > CVS > Label Decorations p**reference page to preview the changes.
6. Click **OK** or **Apply** for the new decorations to take effect.

Sharing a new project using CVS

There are several scenarios that can occur when sharing a project using CVS. The most common scenario is sharing a new project using CVS when the project does not already exist remotely.

To share a new project using CVS:

1. In one of the navigation views, select the project to be shared.
2. Select **Team > Share Project...** from the project's pop-up menu. The Share Project wizard opens.
3. From the list of available repository providers, choose **CVS** and click **Next** to go to the next page. (**Note**: If CVS is not present in the list, it may be disabled. Clicking on the **Show All Wizards** button should make it visible. Also note that if there is only repository provider registered for your workspace, you will automatically bypass this step entirely.)
4. Select the target repository from the list of known repositories or, if the target repository is not in this list, choose to create a new repository location and click **Next**.
5. If entering a new repository location, enter the repository information and click **Next** when completed. (**Note**: this page is the same format as the Creating a CVS repository location wizard.)
6. Either choose to use the name of the local project as the remote project name or enter another name and click **Next**.
7. The final page shows you the resources of the new project as outgoing additions. You can choose to commit or ignore resources from this page. (**Note**: If the project already exists remotely, the page will show conflicts on any files that exist both locally and remotely. Only those files whose contents do not match are shown.)
8. Click **Finish** to share the project with the repository. You will be prompted to commit any resources in the new project that have not yet been committed or ignored. Choosing to do so will run the commit operation in the background.

Consequences for Linked Resources

As mentioned in Linked resources, different providers may handle linked resources differently. For Team CVS, linked resources are allowed but ignored. Specifically, projects which contain linked resources can be shared with CVS, but the linked resources themselves cannot be version controlled.

Project checked out with another CVS tool

It is possible to use a CVS project checked out using another CVS tool into Eclipse. A common concern for those migrating to Eclipse is that they already have their CVS project checked out on their local machine. You *don't* have to check out the contents again to create a project in Eclipse. When such a project is imported into Eclipse, the CVS plug-in may detect that CVS folders exist and auto-share the project. However, there are some instance were this does not occur and the project will need to be shared manually within Eclipse in order to enable the CVS Team menu operations.

To share an CVS project that was already checked out:

1. In one of the navigation views, select the project to be shared.
2. Select **Team > Share Project...** from the project's pop-up menu. The Share Project wizard opens.
3. From the list of available repository providers, choose **CVS** and click **Next** to go to the next page. (**Note**: If CVS is not present in the list, it may be disabled. Clicking on the **Show All Wizards** button should make it visible.)
4. The next page displays the sharing information stored in the CVS folder of the project. Click **Finish** to share the project and enable the CVS Team menu operations.

Repository

Projects can be checked out from a CVS repository into the Workbench using the Checkout wizard or from the CVS Repositories view.

Checking out using the Checkout Wizard

It is available from the Import menu, the New > Project menu and the toolbar of the CVS Repository Exploring perspective. It is also opened when performing a Checkout As from the CVS Repositories view.

1. Launch the Checkout wizard. It is available from the **Import** menu, the **New > Project** menu and the toolbar of the CVS Repository Exploring perspective. It is also opened when performing a Checkout As from the CVS Repositories view. (**Note**: If the CVS capability is disabled, the Checkout wizard may only be available from the **Import** menu).
2. Select the desired repository from the list of known repositories or, if the desired repository is not in this list, choose to create a new repository location and click **Next**.
3. If entering a new repository location, enter the repository information and click **Next** when completed. **(Note**: this page is the same format as the Creating a CVS repository location wizard.)
4. Either select one or more modules from the list of existing modules or type in the name of the module to be checked out. Entering a name is supported for those situations where obtaining the list of existing modules from the server is impractical or unsupported. You can also enter a dot (.) as the module name if the repository is configured to host a single project at its root. Click **Next** when the module or modules to be checked out are specified.
5. *Optional*: Choose whether to check out one or more selected modules as a new projects or into an existing project or folder. If one module is selected and it does not contain a .project file (the Workbench project configuration file), you will also have the option to configure the project using the New Project wizard. Click **Next**.
6. *Optional*: For a single project, you can configure the location of the project to be either the default location or a custom location outside the workspace. For multiple projects, you can configure the parent folder where all the projects should be located. Click **Next**.
7. *Optional*: Select the tag to check out from and click **Finish**.

Checking out from the CVS Repositories view

To check out a project from the CVS repositories view to the Workbench:

1. Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.
2. In the CVS Repositories view, expand the repository location.
3. Expand **HEAD** and select the folders that you want to add as projects to the Workbench. If you are looking for a folder in a particular version:
   - Expand the **Versions** category and find the folder name that you want to add to the Workbench.
   - Expand the folder to reveal its versions.

If you are looking for the latest folder in a branch:

- Expand the **Branches** category.
- Expand the branch that holds the folder that you want to add to the Workbench.

4 From the pop-up menu for the selected folders, select one of the following:

- **Check Out** to check out each of the selected folders as a project in the local workspace with the same name as the folder in the repository.
- **Check Out As...** to check out the selected folders into a custom configured project in the local workspace. *Note:* When multiple folders are selected, this operations only allows the specification of a custom parent location for the new projects.

5 If **Check Out As...** was chosen on a single project, one of two possible dialogs is displayed depending on whether the folder in the repository contains a .project file or not.

- If there is a .project file, the dialog will accept a custom project name and location.
- Otherwise, the dialog will be the New Project wizard which allows full customization of the project.

TIP: Any folder, including non-root folders, can be checked out from a CVS repository.

Checking out a module from a CVS repository

Modules can be defined in the CVSROOT/modules file of a CVS repository. To check out such a module from a CVS repository to the Workbench:

1 Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.

2 In the CVS Repositories view, expand the repository location.

3 Expand **HEAD** and select the modules that you want to add to the Workbench. *Note*: Modules appear as root folders of the repository but are associated with the icon.

4 From the pop-up menu select **Check Out**. This will checkout the module into one or more projects in the local workspace, depending on how the module is defined in the repository.

NOTE: Defined modules can also be checkout out using the Checkout Wizard.

Checking out a folder into an existing project

To check out a folder from a CVS repository into an existing project in the Workbench, you can use either the **Checkout** wizard or **Check Out As** from the CVS Repositories view. Follow these steps to use the Checkout wizard.

1 Launch the Checkout wizard.

2 Select the desired repository from the list of known repositories or, if the desired repository is not in this list, choose to create a new repository location and click **Next**. If entering a new repository location, enter the repository information and click **Next** when completed.

3 Either select one or more modules from the list of existing modules or type in the name of the module to be checked out and click **Next**.

4 Choose whether to check out one or more selected modules into an existing project and click **Next**.

5 On the next page you can provide the following information:

- **Target folder name** (only when checking out a single module): The name of a local folder in which the contents of the remote folder will be placed. *Note*: If the local folder exists, its contents will be deleted and replaced by the contents of the remote folder.
- **Parent of target folder**: The existing local folder into which the target folders will be created.
- **Checkout sub-folders** (only when checking out a single module): The option to indicate whether the sub-folders of the remote folder should be checked out as children of the target folder.

6 (Optional) Select the tag to check out from and click **Finish**.

NOTE: Only folders within non-shared projects or projects shared with the same CVS repository as the selected remote folder are valid targets for the *Checkout Into* operation. Also, if the target project of the operation is an unshared project, the project will be connected to the CVS repository (i.e. the project will become a shared CVS project) but any pre-existing content will be ignored.

Disconnecting a project from CVS

Disconnect a project from CVS to disable the CVS operations that can be performed on the project and it resources and optionally to remove the CVS information (stored in the CVS folders) associated with the project.

To disconnect a project from CVS:

1 In one of the navigation views, select the project to be disconnected.

2 Select **Team > Disconnect** from the project's pop-up menu. The Confirm Disconnect from CVS dialog opens.

3 In the dialog, choose one of:

- **Delete the CVS meta information** - disables the CVS team menu operations and removes the CVS folders and their contents from the file system.
- **Do not delete the CVS meta information** - disables the CVS team menu operations but leaves the CVS meta information.

4 Click **Yes** to disconnect the project

Setting the CVS keyword substitution mode

CVS uses the keyword substitution mode of a file to differentiate binary files from ASCII files and to indicate what type of keyword substitution is to take place when files are committed and checked out.

To set the CVS keyword substitution mode:

1 In one of the navigation views, select the files or containing folders for which a change in keyword substitution mode is desired. **Note**: Ensure that any new files that are to be committed are added to CVS version control as the keyword substitution mode can only be set for files that are already under CVS control.

2 From the pop-up menu, select **Team > Change ASCII/Binary Property**. The Set Keyword Substitution Mode wizard will open. The following dialog was opened was opened on the org.eclipse.jdt.core project:

3. The wizard contains a list of all founds found in the selection. On this page, you can do the following:
   - You can enter a file pattern to filter the list of files.
   - You can change the mode for files individually.
   - You can select multiple files and change the mode for the all using the drop down at the bottom of the page.
   - You can select multiple files and click **Propose** which determines the mode based on your preference settings.
   - You can check the *Only show those files that will be changed* option to see only the files whose mode has been changed in the wizard.
4. When you are done specifying the new modes for any files, enter the commit comment to be associated with any file commits. Files will need to be committed if changing the file type from Binary to ASCII results in a change in the file content due to line terminator adjustments. Click **Finish** to apply the changes.

Filtering CVS Revisions in the History view

Over time, the revision history for a file can grow drastically. As a result, it may be difficult to find the revision or revisions that you are looking for. Using the filtering mechanism of the History View can help.

To set a filter on a CVS managed resource that is currently being displayed in the History View:

1. Select the **Filter...** action from the dropdown menu on the History View toolbar.
2. Enter any author, comment, or date range that you would like the view to display.
3. Click **OK**.

The view will now display only those revisions that match the criteria you entered.

TIP: You can remove the filter at a later time by selecting Remove Filters from the dropdown menu on the History View toolbar.

TIP: You can choose whether the filter should match any or all of the criteria by selecting the appropriate radio button in the dialog.

Changing the sharing of a project

If a project is shared with a CVS repository location, you can change the sharing information so it is shared with a different repository location. The new location must map to the same repository but can have a different connection method and/or user name.

To change the sharing of a project:

1. In one of the navigation views, select the project whose sharing information is to be changed.
2. Select **Properties** from the pop-up menu. The properties dialog will open.
3. Select **CVS** to see the CVS properties page for the project.
4. Click **Change Sharing...**. A repository selection dialog opens containing the compatible repository locations. (**Note**: You can change the sharing to a non-compatible repository location by unchecking the **Show only compatible repository locations** button.)
5. Select the desired location and click **OK** to change the sharing for the project.

Sharing your workspace setup using Project Sets

Your workspace setup may consist of several projects from one or more repositories. Once you have setup your workspace, you can share it with others by exporting a Team Project Set. A project set is a text file that contains a pointer to each of the projects contained in the project set.When a project set is imported, these pointers are used to fetch the projects from the repository. Project sets can include any projects that are mapped to repository tooling that provides support for them, such as CVS. To export a project set:

1. Setup your workspace with all the projects you want to work on by checking them out of CVS or obtaining them in any manner appropriate for the repository tooling you are using.
2. From the **File** menu, choose **Export**. The Export dialog will open.
3. In the Export dialog, choose **Team > Team Project Set** and click **Next**. The dialog will now display all the projects that are eligible for export.
4. Check the projects you wish to include in the project set. Either browse for or type in the name of the file where you wish to save the project set and then click **Finish**.

Now you have created a project set file. You can now share this file with others or use it yourself to recreate your workspace. To Import a project set:

1. From the **File** menu, choose **Import**. This will open the import dialog.
2. In the Import dialog, choose **Team > Team Project Set** and click **Next**.
3. Browse for or type in the name of the file containing the project set and click Finish.
4. The projects contained in the project set will be fetched from the repository.

For CVS, you will be prompted to provide the authentication information (user name and password) for the repository as it is not included in the project set.

Specify Repository Information dialog

When the project set contains only partial repository information (not all locations are known) the Specify Repository Information window is displayed offering three options for Repository locations:

1. Use the original location that is displayed in the first row of the Repository location column drop down list.
2. Click **OK** to choose a known repository location from the same drop down list (default).
3. Create a new repository location by clicking the **Create Location** button.

The user can fully configure the repository location, specify a user name, and change the connection method if required.

Synchronizing with the Repository

Working with shared resources requires tracking their state in context of the repository content. This could be done by creating a synchronization that populates the Synchronize view and benefiting following advantages:

- The local modification state of the resources being synchronized is kept up-to-date. This means that, if you modify a file locally that is within the scope of a synchronization that appears in the Synchronize view, the resource will appear automatically in the view, if it is not already there.
- The remote state of all of the resources being synchronized can be refreshed using the Synchronize button in the toolbar of the Synchronize view.
- The remote state of a selection of resources can be refreshed using the Synchronize command from the context menu in the Synchronize view.

Following methods can be used to create a synchronization.

Method 1: Using the context menu

To synchronize resources in the Workbench with those in the repository:

1 In one of the navigation views, select the resources that you want to synchronize.

2 Right click and select **Team > Synchronize with Repository**. The Synchronize view opens.

Method 2: Using the synchronize action

1 From the Team Synchronizing perspective select the **Synchronize...** action from the Synchronize button drop down.

2 Select **CVS** from the list of possible synchronization types and click **Next**.
Note: If there are no other repository providers registered with the Workbench, then you will bypass this screen altogether.

3 Select the resource scope for the synchronize by either selecting Workspace, Selected Resources or Working Set. Then select **Finish.**

4 The Synchronize view will open.

NOTE: The synchronize action is not enabled by default in other perspectives. You can enable the action to appear in your current perspective by selecting **Window > Customize Perspective.** Then click on the **Commands** tab and check off **Team**.

TIP: You can use method 3 below to avoid loosing the created synchronization.

Method 3: Using a pinned CVS Workspace Synchronization in the Synchronize view

Once you have a CVS workspace synchronization in the Synchronize view, you can pin it. This will prevent it from being replaced by the next CVS workspace synchronization that is launched using one of the previous 2 methods. Here are some of the advantages of using a pinned synchronization.

- You can remove any not pinned synchronization to make the pinned one visible. To do it select **Remove Current Synchronization** from the Synchronize view toolbar's menu.
- You can schedule the refreshing of the remote state to happen at particular intervals (each hour, for example)

The implications of this is that you can see you outgoing resources without re-fetching the remote state from the server (a potentially long running operation). Also, the fetching of the remote state is run in the background so you can do other things (e.g. inspect changes) while the remote state is fetched.

From within the synchronize view

Use the toolbar buttons to switch modes for this view. There are four modes:

- Incoming mode - shows incoming changes only (resources in the repository that differ from what is in the Workbench).
- Outgoing mode - shows outgoing changes only (resources modified in the Workbench).
- Incoming/Outgoing mode - shows both incoming and outgoing changes.
- Conflicts mode - shows only conflicting resources.

NOTE: It is possible that someone has committed a new revision of your file since you started working on it. This will result in a *conflict*, and care must be taken to resolve this. For this reason, conflicts are shown in all modes of the Synchronize view.

Important: It is preferable to update resources in the Workbench first, resolve any conflicts that exist by merging, then commit Workbench resources to the repository

Updating

While you are working on a project in the Workbench, other members of your team may be committing changes to the copy of the project in the repository. To get these changes, you may "update" your Workbench to match the state of the branch. The changes you will see will be specific to the branch that your Workbench project is configured to share. You control when you choose to update.

The update command can be issued from two places: the **Team > Update** menu, or the **Synchronize** view. In order to understand the difference between these two commands, it is important to know about the three different kinds of incoming changes.

- A *non-conflicting* change occurs when a file has been changed remotely but has not been modified locally.
- An *auto-mergeable conflicting* change occurs when an ASCII file has been changed both remotely and locally (i.e. has non-committed local changes) but the changes are on different lines.
- A *non-auto-mergeable conflicting* change occurs when one or more of the same lines of an ASCII file or when a binary file has been changed both remotely and locally (binary files are never auto-mergeable).

When you select **Team > Update**, the contents of the local resources will be updated with incoming changes of all of the above three types. You can specify what the update behavior should be in the **Team > CVS > Update/Merge** preference page. The choices are:

- **Preview all incoming changes before Updating**: All changes will be displayed in either the Sync View or a dialog (depending on your settings). You can then merge each change in line by line, or update all non-conflicting changes at once and then deal with the remaining conflicts.
- **Update all non-conflicting changes and then preview the remaining changes**: All non-conflicting incoming changes will be merged in automatically and any remaining conflicts will be displayed either in the Sync View (*default*) or in a dialog. You can specify where to display conflicts from the Update/Merge preference page.
- **Never preview and use CVS text markup to indicate conflicts (*default*)**: This option will automatically merge all changes in without any user interaction. Conflicting changes will be merged in using the CVS text markup:

  ```
  <<<<<<< original file revision
  [original code]
  = = = = = = =
  [incoming code]
  >>>>>>> incoming file revision
  ```

  You will then have to go in to each file that contains a merge conflict and edit the file to the desired final state.

It is often desirable to know what incoming changes there are before updating any local resources. These issues are addressed by the Synchronize view.

To open the Synchronize view in incoming mode:

1. In one of the navigation views, select the resources which you want to update.
2. From the pop-up menu for the selected resources, select **Team > Synchronize with Repository**. The Synchronize view will open.
3. On the toolbar of the Synchronize View, click the **Incoming mode** button to filter out any modified Workbench resources (outgoing changes) that you may have.

In incoming mode, you will see changes that have been committed to the branch since you last updated. The view will indicate the type of each incoming change. There are two update commands (available from the context menu of any resource in the view) to deal with the different types of conflicts: **Update** and **Override and Update**. When you select the **Update** command in the Synchronize view, all selected incoming and auto-mergeable conflicting changes are processed while conflicts that are not auto-mergeable will not be updated (any files that have been successfully processed are removed from the view). The **Override and Update** command operates on conflicts and will replace the local resources with the remote contents. This "replace" behavior is rarely what is desired. An alternative is described later.

To update non-conflicting and auto-mergeable files:

1 The Structure Compare pane at the top of the Synchronize view contains the hierarchy of resources with incoming changes.

2 Select all conflicting files and choose **Update** from the pop-up menu. This will update the selected resources that are either incoming changes or auto-mergeable conflicts and remove them from the view. Conflicts whose contents are not auto-mergeable will be left in the view.

If your local Workbench contains any outgoing changes that are not auto-mergeable with incoming changes from the branch, then, instead of performing an **Override and Update**, you can merge the differences into your Workbench manually, as follows:

1 In the Structure Compare pane, if there is a conflict in the resource list (represented by red arrows), open it (either by double-clicking or selecting **Open in Compare Editor** from the context menu).

2 In the Text Compare area of the compare editor, local Workbench data is represented on the left, and repository branch data is represented on the right. Examine the differences between the two.

3 Use the text compare area to merge any changes. You can copy changes from the repository revision of the file to the Workbench copy of the file and save the merged Workbench file (using the pop-up menu in the left pane).

4 Once you are completed merging the remote changes into a local file, choose **Mark as Merged** from the pop-up menu in the Synchronize view. This will mark the local file as having been updated and allow your changes to be committed.

NOTE: The repository contents are not changed when you update. When you accept incoming changes, these changes are applied to your Workbench. The repository is only changed when you commit your outgoing changes.

TIP: In the Synchronize view, selecting an ancestor of a set of incoming changes will perform the operation on all the appropriate children. For instance, selecting the top-most folder and choosing **Update** will process all incoming and auto-mergeable conflicting changes and leave all other incoming changes unprocessed.

CAUTION: The behavior of the **Override and Update** command described above only applies to the incoming mode of the Synchronize view. In the **Incoming/Outgoing mode** of the view, the behavior for incoming changes and conflicts is the same but the command will revert outgoing changes to whatever the repository contents are. Exercise great caution if using this command in Incoming/Outgoing mode.

Committing

You can commit Workbench resources that you have modified to the repository so that other team members can see your work. Only those changes committed on that branch will be visible to others working on that branch. The commit command can be issued from two places: the **Team > Commit** menu, or the **Synchronize** view.

Committing changes using Team > Commit

1 In one of the navigation views, select the resources that you want to commit.

2 Right-click on the resources and select **Team > Commit** from the pop-up menu.

3 If there are new files whose file types cannot be determined automatically, the first page of the Commit dialog will present the unknown types and allow you to set them appropriately to either ASCII or binary. Click **Next** to continue.

4 On the Comment page, provide a comment for your changes (for example, `Fixed the spelling mistakes`).

You can preview files that are about to be committed from the Comment page. If any of the files are known to be conflicting changes, the commit will not be allowed. If there are no known conflicting changes the commit will be allowed but there could still be conflicting changes on the server (i.e. conflicting changes on the server become known to the client during a synchronize operation). If there are conflicting changes on any files that are committed, the operation will fail. If this occurs, you must either perform an update or use the Synchronize view to resolve the conflicts. It is considered a more ideal workflow to always update before committing in order to ensure that you have the latest state of the repository before committing more changes.

If one or more of the resources being committed are new and not yet added to CVS control, they will be added automatically unless they are explicitly removed by choosing **Remove from View** from the context menu.

Committing changes in the Synchronize view

1 In one of the navigation views, select the resources that you want to commit.

2 Right-click to open the pop-up menu and select **Team > Synchronize with Repository**. The Synchronize view will open.

3 On the toolbar of the Synchronize view, select the **Outgoing mode** button to show any modified Workbench resources (outgoing changes) that you may have.

4 If there are conflicts (red arrows), resolve them. To do so open them in a Compare editor and use the text compare area to merge resources with conflicts. You can copy changes from the repository revision of the file to the Workbench revision of the file and save the merged Workbench resource. Once all the conflicts in the Structure Compare area have been resolved, perform a **Mark as Merged** on the resource in the Synchronize view to make the change an outgoing change and you are ready to commit.

5 In the Structure Compare pane, right-click the top of the hierarchy that you want to commit, and select **Commit** from the pop-up menu.

6 In the Commit Comment dialog box, provide a comment for your changes (for example, `Fixed the spelling mistakes`). Again, if there are new files of an unknown type, you will be asked to specify what type they should be.

TIP: You can commit files that are in conflict by performing an **Override and Commit**. This will commit the Workbench copy of the resource into the repository and thus remove any of the incoming changes.

CAUTION: The behavior of the **Override and Commit** command described above only applies to the outgoing mode of the Synchronize view. In the **Incoming/Outgoing mode** of the view, the behavior for outgoing changes and conflicts is the same but the command will revert incoming changes to whatever the local Workbench contents are. Exercise great caution if using this command in incoming/outgoing mode.

Versioning

Resources can be versioned in order to capture a snapshot of their current state at one specific point in time. See the Related tasks links for more details.

Creating a version of a project

Prerequisite: The project that you want to version must first be in the workspace.

TIP: It is often preferable to synchronize resources with the CVS repository before versioning. This will ensure that there are no outstanding incoming or outgoing changes that might be accidentally excluded from the version.

To make a version of a project:

1 In one of the navigation views, select the project that you want to version.

2 From the project's context menu, select **Team > Tag as Version**.

3 Enter the name of the version you wish to create.

4 Click **OK** to close the Tag Resources dialog and create the version.

TIP: You can browse existing versions of your project by clicking on the **Details** button in the Tag Resources dialog. Clicking **Refresh from Repository** should populate the Existing Versions list. It is sometimes helpful to see existing versions before naming your new version.

Versioning projects in the repository

You can version a project in the repository without adding the project to the Workbench.

1 Switch to the CVS Repository Exploring perspective or add the CVS Repositories view to the current perspective.

2 In the CVS Repositories view, expand a repository location and find the project you wish to version.

3 Right-click on the project and select **Tag as Version**.

4 Enter the name of the version you wish to create.

5 Click **OK** to close the dialog and create the version.

TIP: You can browse existing versions of your project by clicking on the **Details** button in the Tag Resources dialog. Clicking **Refresh from Repository** should populate the Existing Versions list. It is sometimes helpful to see existing versions before naming your new version.

TIP: If the version tag you wish to apply already exists on some of the resources to be versioned, you can perform a **Tag with Existing** instead. This operation will move an existing tag to the selected resources and can be used to move both version and branch tags. If the tag you wish to apply is not show in the list, you can click **Refresh from Repository**. If this doesn't find the tag, you can click **Configure Tags** which opens a dialog that allows you to search for tags on specific files in the repository.

Comparing Resources with Repository Versions

To compare Workbench resources with those in the repository:

1 Select a resource in one of the navigation views.

2 From the resource's pop-up menu, select one of the following menu items:

- **Compare With > Latest From <Branch Name>** or **<Version>**- Compare the Workbench resource with the latest resources currently committed to the branch that the local project is shared with or if the local project is checked out as a version then compare against the version. A compare editor will be opened. This editor will display a tree of all resource differences between the workspace resource and the latest resources in the repository.

- **Compare With > Another Branch or Version** - Compare the Workbench resource with a specific version or branch that you select in the repository. A compare editor will be opened. The editor will display a tree of all resource differences between the workspace resource and the resources in the specified version or branch.

- **Compare With > History...** - (available only on a file) Compare the Workbench file with another revision of the file. Based on your preference settings, either the History view or a dialog will open. The History view (or dialog) will contain a table of all available revisions of the selected file. Selecting one of the revisions will show the differences between the workspace revision and the selected revision.

  TIP: When comparing with a branch or version, you can specify a particular date instead of a version or branch tag. This can be done by right clicking on the **Dates** category in the tag selection list and choosing **Add Date**. A Date Tag dialog is displayed that allows you to specify a date and optionally a time. After you click **Ok**, the date tag will appear in the tag selection list.

Working with Patches

Patches allow developers to share work without storing it in a repository. This is helpful when a developer wants to contribute to a project that is shared through a repository but does not have write access to the repository. In this situation, the developer can create a patch and either e-mail it to a developer who does have write access or attach it to a bug in the bug reporting system used by the project, depending on the process defined by the project. A developer that does have write access can then apply the patch to the project and commit the changes.

To create a patch from a CVS project:

1. Select the resource that contains the modifications to be included in the patch. You can select resources that reside in multiple projects and at any level as the Create Patch wizard, when run in its default mode, knows how to create a multi-project patch. The patch should also be applied to the same file revisions that it is generated on so steps should be taken to ensure that the patch is applied to the same resource line-up (the easiest way to do this is to create the patch on top of a version).
2. From the popup menu, select Team > Create Patch.... The Create Patch wizard will open.
3. Choose where the patch should be saved:
   - *Save to Clipboard* - this will place the patch on the clipboard so it can be pasted into a text editor such as an e-mail program.
   - *Save to File System* - this will place the patch in the specified file in the local file system
   - *Save in Workspace* - this will place the patch in the specified file inside one of the existing workbench projects.

   For small patches it may be reasonable to transfer the patch using the clipboard but in most cases the local file system in the best option to use.
4. You can preview and fine tune your patch selection by expanding the Resources tree presented in the Changes pane. Only those elements which are checked will be included in the patch. Click **Next** to configure how the patch is generated.
5. Choose how to configure the patch:
   - *Diff output format* - Allows the choice of several common diff output formats. *Unified* is the format used by many patch application tools including Eclipse. The format used in Context and Unified diffs allows to apply a patch, even though the line number mentioned for the hunk (patch terminology) is incorrect. In this case, when applying the patch, the algorithm scans both forwards and backwards for a set of lines matching the context given for the hunk.
   - *Patch Root* - Allows you to specify at what level your patch is rooted at. The choices are *Workspace*, *Project* and *Selection*.

     Workspace allows you to include resources from multiple projects in your patch and is the *default* option. Workspace patches can be applied to any resource in the workspace - they contain enough information to allow the Apply Patch wizard to figure out which resources need to be patched.

Project patches are rooted at the project level - which means they can only contain resources from one project and must be applied to the same project.

Selection patches are rooted at whatever the selected resource is and must be applied to the same resource.

6 Click **Finish**.

7 Transfer the patch as appropriate for the project being patched.

To apply a patch:

1 Select the resource that the patch was generated on. This resource should contain the same file revisions as the line-up on which the patch was generated.

2 From the pop-up menu, select **Team > Apply Patch...**. The Resource Patcher wizard will open.

3 Indicate where the patch is to be found:

- *File* - the patch is in a file on the local file system. Either type in the full path to the file or use the **Browse...** button to find the file.
- *Clipboard* - the patch is on the clipboard. **Warning:** It is safer to use a file based patch. Line endings may not be handled properly if the clipboard is used and the patch was generated on a different platform (i.e. Linux vs. Windows).

4. *Workspace* - the patch has been saved somewhere in the workspace.

Click **Next** to see the effect of applying the patch.

5 **Optional Step**: *this step only applies if you have a patch rooted at the project or selection level - workspace rooted patches will automatically proceed to the Patch Preview Page.* In the Patch Target Specification page, you should select the resource that is to act as the root of the patch.

6 The patch preview page shows whether the patch could be successfully applied to files in your workspace. The top pane shows the list of changes contained in your patch. There are two types of entries in the top pane: file changes and unmatched patch segments (known as 'hunk' in patch terminology).

- If one of more of the patch segments can be automatically applied to the file, the file will be shown with an incoming change indicator. You can inspect the change by double-clicking on the file.
- If one or more hunks cannot be automatically applied, the hunk entries will appear as children of the file in the top pane and a red exclamation mark indicates that there is a problem with a patch or hunk. You can inspect the hunk by double-clicking on it. You can then manually apply the hunk from the right pane to the file on the left. Saving from the left pane will update the parent file entry in the top pane but will not modify the file on disk. The file on disk is only modified when Finish is pressed. In order to apply the full patch successfully you will have to eliminate the problems (red exclamation marks) either by manually applying the patch segments, by excluding the patch segment from the operation by selecting **Remove** from the context menu or by tweaking the options on this wizard page (see 'Options' below).

7 If all is well, click **Finish** to apply the patch. The workspace will now contain outgoing changes for each file modified by the patch.

Options for applying a patch

For getting successful matches of your patch file you have the following options:

1 Go back to the first page of the Apply Patch wizard and select the correct resource to which the patch should be applied.

2 If a common prefix of the path names stored in the patch file doesn't match the path names in you current workspace, you can *Ignore leading path name segments*.

3. Use the *Ignore whitespace* button to make the matching process independent from whitespace differences between the patch file and files in your workspace.
4. Adjust the *Fuzz factor* (patch terminology). This factor indicates the amount of hunk context lines to ignore. If the Fuzz factor value is zero all lines need to match. If the fuzz factor is greater than zero, for example 2, the first two context lines before the hunk and the last two context lines after the hunk are ignored. Click **Guess** to calculate the fuzz factor that allows the most hunks to be matched.
5. Use the *Reverse patch* option for patch files that already have been applied to your workspace. This option is also useful to undo or redo a patch.
6. For Workspace patches, you can select another project in your workspace to apply the patch changes to. To do this, select a project in the top pane and select **Move** from the context menu. This will launch a dialog with a list of all available projects in your workspace. Select a project and click **OK**; the patch changes will be applied to your selected project.

Use the *Show Excluded* option to show the hunks which you have already selected to not include in the patch.

Replacing Resources in the Workbench

To replace Workbench resources with versions in the repository:

1. Select a resource in one of the navigation views.
2. From the resource's pop-up menu, select one of the following menu items:

   Replace With > Latest From <Branch Name> or **<Version>** - replace the Workbench resource with the latest resources currently committed to the branch that the local project is shared with or if the local project is checked out as a version then replace with the same version.

   Team > Revert to Base - replace the Workbench resource with the last checked out revision. Deleted files are restored.

   Replace With > Another Branch or Version - replace the Workbench resource with a specific version or branch that you select in the repository. When you select this option, another window opens, so that you can browse through the branch and version tags in the repository and select the one that you want.

   Replace With > History... - (available only on a file) replace the Workbench file with another revision of the file. Based on your preference settings, either the History view or a dialog will open. The History view (or dialog) will contain a table of all available revisions of the selected file. Selecting one of the revisions will show the differences between the workspace revision and the selected revision. To replace the workspace revision, you should select the revision that you want and choose 'Get Contents' from the context menu. This will replace the contents of the workspace file with those of the selected revision.

   Team > Switch to Another Branch or Version - this update will replace Workbench resources with those on the tag specified in the Update dialog. The behavior differs from the above replace operations in that uncommitted modifications in the Workbench are not replaced but instead are "moved" to the resources from the selected tag. For more details on the specifics of this operation, see the CVS documentation at http://ximbiot.com/cvs/relating to the use of the *-r* option with *cvs update*.

   TIP: When replacing with a branch or version or when updating, you can specify a particular date instead of a a version or branch tag. This can be done by right clicking on the **Dates** category in the tag selection list and choosing **Add Date**. A Date Tag dialog is displayed that allows you to specify a date and optionally a time. After you click **Ok**, the date tag will appear in the tag selection list.

Finding out Who’s Working on What: Watch/Edit

CVS provides a notification scheme which allows you to know if someone is modifying a file that you care about. This facility is known as *watches*. By setting a *watch* on a file, you can have CVS notify you via email (or other) if someone else starts to *edit* this file.

There are two parts to CVS watches: *watch*, and *edit*. The first, *watch*, is how you specify which files you wish to be notified about. The second, *edit*, is how you inform the CVS server (and thus others) that you are about to modify a file.

Edit is useful on its own without ever setting up any watches and lots of people work this way. This is because when you edit a file, you will be told immediately if someone else is already editing that file. Since most people just want to know up front that they may have to merge their changes on commit, *edit* on its own is sufficient for most. Another advantage to using just *edit* is that it doesn't require any administrative changes to the server, where as *watch* does. All that *watches* gives above this is the email notification that some file you are watching is being modified.

For these reasons, *edit* is supported natively by Team CVS where as *watch* isn't.

Setting up Watches

As mentioned, you can't set watches in Team CVS. If you are interested in doing this, you should consult your cvs documentation. In brief though, this is what's involved:

1 First, you or your CVS administrator will need to modify the CVSROOT/notify file. Consult the CVS documentation on watches for details on how to configure this file.

2 Next, you will need to perform a command line "cvs watch add <filename>" for each file that you wish to watch. If <filename> is a directory name, then all files within that directory will be watched.

Setting up a Project for Watch/Edit

Watches and editing are optional in CVS. To use this facility, you must turn on this option in the **Team > CVS > Watch/Edit** preference page. Select "Configure projects to use Watch/Edit on checkout", accept the preference dialog, and then checkout your project. All the files in the project will be checked out read-only. This tells the CVS client which files are being edited by you and which aren't (writable files are being edited). If you've already checked out the project before you turned on this option, you can either check it out again or enable the "Use Watch/Edit for this project" option on the project's CVS properties page. Either of these operations will make the files in the project read-only.

Editing

Although typical CVS clients require you to perform an explicit edit, Team CVS automatically issues an edit as soon as you start to modify a file. This support is built deep into Eclipse, so typing in a text editor, performing refactoring, etc., will all issue a CVS edit for you. You can also perform an explicit edit via the **Team > Edit** context menu on a resource.

When an *edit* is issued, you will be informed immediately if someone is already editing that file. In addition, everyone who is *watching* that file will be notified by the CVS server via email etc. Since watches simply give you email notification, *edit* without ever setting up watch lists is still a useful (and popular) workflow.

If you prefer, you can turn off automatic issuing of edits. This means you will need to manually perform a Team > Edit for each file you are working on. To use this work mode, open the **Team > CVS > Watch/Edit** preference page and enable "Edit the file without informing the server".

Finally, you can see the list of editors of a file at any time by selecting **Team > Show Editors** from the context menu of that file.

Unediting

Just as you can tell CVS that you are editing a file, there also needs to be a way of telling CVS that you are no longer editing that file. This is referred to as *unedit*. This way, if someone checks the editors list for a file, they'll know if someone is still working on that file. This happens in one of two ways:

- When you commit an edited file, an unedit is automatically issued.
- If you discover that you want to back out of the changes to a file, you can explicit unedit the file. In addition to notifying the server, an explicit unedit will revert the file to its base (i.e. the workspace will then contain the copy of the file before you started modifying it).

Determining who last modified a line with the Annotate command

Let's say you have found some code that you don't understand on line 65 of a file. Who do you ask about it? Well you could start by looking at the resource history for the file, but that won't tell you who changed that particular line. This is why the Annotate command is useful. Sometimes jokingly referred to as the blame, it allows you to pick any ASCII file (see note on binary files) and get a listing of who changed what line.

The Team > Show Annotation action is available from the following places: History View, Repository Explorer, Synchronize View, the Project and Package Explorers, and text editor context menus. The action works in a Quick Diff flavor and offers the following features:

- Displays the annotations on a local file right in the same editor.
- Clicking on an annotation or clicking on any line in the editor will prompt the History View to show the corresponding revision information.
- Doesn't lose your place in the editor when displaying the annotation; simply adds the annotations to the annotation bar of the editor that you are currently using.
- To turn off annotations, you have to select Revisions > Hide Revision Information from the annotation bar context menu.

Only works with text files

The annotate command will only work with files that are marked as ASCII in the CVS repository. Also, the command will open a text file to show the changes even if the associated editor in the workbench is a non-text editor. For example, if you run annotate on a plugin.xml file a simple text editor will be opened instead of the full PDE editor.

Quick Diff: CVS Team Settings

Instead of using a compare editor, which will show changes between 2 or 3 files by showing each file side-by-side, you can enable quick diff support and see the changes within the text editor (e.g. any text editor based on the Eclipse text editor). This feature can be enabled via the General > Editors > Text Editors > Quick Diff preference page. You should select the Latest CVS Revision as the reference source. This will annotate the text file with diffs against the latest revision in CVS. Here are the following scenarios that are useful:

1. Open a file and make changes to it. You will see the difference annotations marking the changes. Then if you run Team > Replace with latest. The annotations are removed and the file is clean.
2. Open a file and make changes to it. You will see the difference annotations marking the changes. Then if you run Team > Commit the annotations are removed and the file is clean.
3. If you synchronize the file with the server and a new revision is found on the server, the editor will update showing the incoming changes.

If you enable showing the differences in the overview ruler, you can, at a glance, get an idea of how many changes you have made to a file since your last commit. The differencing happens in a background thread to minimize the impact on actual editing of the file.

NOTE: this feature requires an active connection to your CVS server so that remote contents can be fetched when an editor is opened.

Changing CVS Team Settings

You can customize the settings related to a number of CVS team views and operations.

1. Open the **Team** preference page. You will see a number of general Team options. These options can be used by any Eclipse integrated repository. In our case, they are all applicable to CVS.
2. You will also see a **Team > CVS** preference page containing various CVS options.
3. Underneath the **Team > CVS** preference page are several sub-pages. For example, there are pages for **Watch/Edit**, **Console**, **Label Decorations** and **SSH2** among others.

In the Preferences dialog, you can search by keywords such as CVS or SSH2 in order to show only the relevant pages.

Changing CVS Project Settings

You can change CVS project settings in the properties dialog of a project.

- From the context menu of a view showing the project, select Properties
- Click on the CVS category

From this dialog, you can:

- Enable Watch/Edit on the project.
- Specify whether new or absent folders should be fetched when an update is performed.
- Change the repository location to which the project is associated.

Restoring Deleted Files from the Repository

The CVS plug-in allows you to delete files from the CVS repository. If you delete a managed file and commit the deletion to the server, the file is deleted from the current branch or version. Although the file has been deleted from the current branch, the file and its previous revisions are actually still on the server. The CVS plug-in provides a tool to help restore a file to your workspace that has previously been deleted from the CVS repository:

1. Select a folder managed by CVS and from the context menu select **Team > Restore from Repository**.
2. The repository will be searched for deleted files and the list will be shown in a dialog.
3. Select a file in the left most pane to display the revisions of the file that are available in the repository.
4. Select a revision in the right-most pane to view the contents in the bottom text pane.
5. Check the revision that you would like to restore.
6. Click **Finish** once you have checked off a revision for each file you wish to restore.

After restoring the file if you want to actually restore the file to the current branch you have to add then commit the file back into the repository. If the filename does not change, revision history will be maintained for the restored file.

Reverting a branch to a previous version

It is often useful to revert the contents of a branch to those of a specific version. For example, if your current branch contains changes that you no longer want to release you can revert all or a portion of a project to the contents of any version.

1. Checkout into your workspace the contents from the branch that you want to revert.
2. Select **Compare With > Another Branch or Version** on the resource(s) that you want to revert.

3 From the tag selection dialog box select the version to which you want to revert the branch.

4 When the compare editor opens, review the differences that are shown and ensure that they are what you expected.

5 Select the root folder in the compare view and from the context menu select **Override and Update**. After the operation is completed the folder or project you compared against will have exactly the same contents as the remote revision.

6 You can verify this by performing another comparison against the version. (**Note**: That the CVS preference to **Consider file contents in comparisons** in the **Team > CVS > Synchronize/Compare** preference page should be enabled for this to work.)

7 The comparison should report that there are no changes.

Once your workspace contains the new contents, run your tests then commit the changes to the branch.

Moving Version Tags

CAUTION: Although many people would prefer a CVS version to be frozen in time and not be modifiable, version and branch tags in CVS are mutable. As a result many are convinced that modifying a version is bad practice, but there are a couple of scenarios where it actually comes in handy. With that said, please be cautious when moving tags around.

Moving a tag on a single file

Let's say that you have just submitted your build by versioning your project as R1. But you soon after discover that there is a small change to a file that should be made and included in the build. Instead of having to re-version the project you can move the R1 version tag for the modified file.

1 Modify the file(s) Select the file that was modified after R1 was created and from the context menu select **Team > Show in Resource History**.

2 From within the Resource History view select the revision that should be marked with the R1 version.

3 From the context menu select **Tag with Existing...**.

4 Select the R1 version from the dialog box and press OK.

5 The resource history view will be updated to confirm that the version had been moved.

Moving a tag from within the repositories view

Many projects use a well defined version name for their current stable lineup in HEAD. For example, by versioning HEAD with the STABLE tag the build scripts could simply checkout the STABLE version for builds. As the code evolves the STABLE tag is moved regularly to mark the most current stable lineup. The repositories view provides an action for moving an existing tag.

1 Open the Repositories view and select a resource.

2 From the context menu select **Tag with Existing**

3 A tag selection dialog will appear from which you can select the tag to move. If the tag you wish to apply is not show in the list, you can click **Refresh from Repository**. If this doesn't find the tag, you can click **Configure Tags** which opens a dialog that allows you to search for tags on specific files in the repository.

4 Then press **OK** and your tag will be moved. The operation will move an existing tag to the selected resources and can be used to move both version and branch tags.

Running the CVS Command-line client outside of Eclipse

Compatibility

Because the Eclipse CVS plug-in stores its meta information in a format that is compatible with the command-line CVS client you should be able to use a CVS command line client against Eclipse workspace files on disk. The metadata is stored in CVS/ sub-directories but you rarely see them within Eclipse. They are marked as private which causes them to be hidden from view. If you open a (non-Eclipse) file explorer you will see that these directories and their contents appear on the file system.

Don't forget to refresh!

Whenever you use external tools to modify workspace files, you must perform a **Refresh** from within Eclipse to make the workspace aware of the changes. If you get a *resource out of sync* error in Eclipse it is a sign that there are resources in Eclipse that have been modified outside of Eclipse. One solution is to perform a refresh (available from a resource's popup menu) on any resources or projects that where modified outside of Eclipse. There is also a preference to refresh automatically.

Caveats

1. **Deleted folders**

   You may encounter unexpected behavior when using the command-line CVS client in conjunction with deleted folders. Eclipse's CVS support keeps track of deleted folders and their contents so that, on the next synchronization, the Synchronize view can properly report on the changes. This information is kept outside of the CVS meta folder structure. This is because in CVS you normally inform the repository of deletions prior to deleting them locally, which is a different workflow than we like to support in the Synchronization view. Thus it is recommended that you do not use the command-line CVS client while you have pending deletions to commit. In some circumstances it could cause the Synchronize view to display incorrect contents, although it will not cause any lost work.

2. **CVS directories appear in the workbench**

   When you use the command-line CVS the CVS folders can sometimes appear in one of the navigation views. There are some cases where CVS folders are not hidden from the UI as you would expect. For instance, CVS folders will appear if a user imports a CVS project into Eclipse before the CVS plug-in is loaded. To avoid this, open the CVS Repositories view (thus loading the CVS plug-in) before importing CVS projects into Eclipse.

Reference

Contents

Minimizing Data Loss from Crashes

The Workbench periodically saves a snapshot in order to reduce the risk of losing data due to crashes. The default period between saves is 5 minutes, but you can change this from the Workspace preferences page (**Window > Preferences > General > Workspace**). From that page you can also set Eclipse to save all modified resources before a manual build.

Previously created projects and saved editor data are never lost as they are written to disk immediately upon save. However, there is a risk of data loss in these areas:

- Unsaved data in open editors may be lost, depending on the editor implementation.
- Bookmarks and tasks might be lost.
- If a crash occurs during CVS synchronization, the Workbench may be out of sync. You can check by performing the synchronize operation again.

Preferences

Use the **Window > Preferences** dialog pages to set how you want Eclipse to operate.

You can browse the Preferences dialog pages by looking through all the titles in the left pane or search a smaller set of titles by using the filter field at the top of the left pane. The results returned by the filter will match both Preference page titles and keywords such as "appearance". However, to find specific functions you may have to search the online help instead.

The arrow controls in the upper-right of the right pane enable you to navigate through previously viewed pages. To return to a page after viewing several pages, click the drop-down arrow to display a list of your recently viewed preference pages.

The preferences dialog displaying the **General** preferences:

Accessibiltiy

You can change the following preferences on the **General > Editors > Text Editors > Accessibility** preference page:

Option	Description	Default
Use custom caret	Replaces the original *caret* (the marker that indicates where the next character will appear) with a custom caret and shows a different caret for Overwrite and Insert modes.	On
Enable thick caret	Replaces the original caret with a more visible, thicker caret.	On
Use characters to show changes on line number bar	Quick Diff shows the changes in a vertical ruler using colors. Color-blind persons can enable this option to show differences with different characters in the line number ruler.	Off

Annotations

The following preferences can be changed on the **General > Editors > Text Editors > Annotations** preference page.

Option	**Description**
Show in Text as	This option controls whether the selected annotation type is shown in the text. The corresponding text will be underlined with squiggles or highlighted
Show in Overview ruler	This option controls whether the overview ruler on the right side of the text editor is shown.
Show in Vertical ruler	This option controls whether the selected annotation type is shown in the vertical ruler.
Color	This option controls the color for the selected annotation type.

Appearance

You can change the following preferences on the **General > Appearance** preferences page.

Option	**Description**	**Default**
Current presentation	Specify the currently active presentation (look and feel).	Default (Current)
Override presentation settings	Locally override the settings from the current presentation's defaults.	Disabled
Editor tab positions	Specify either top or bottom to indicate where you want tabs for stacked editors to appear.	Top
View tab positions	Specify either top or bottom to indicate where you want tabs for stacked views to appear.	Top
Perspective switcher positions	Specify the location of the perspective switcher bar.	Top Right
Current theme	Specify the currently active theme (color and font set).	Default (current)
Show text on perspective bar	Specify whether labels should be shown in the perspective bar as well as icons.	Enabled
Show traditional style tabs	Specify whether traditional (square) tabs should be used in place of the curved tabs.	Disabled
Enable animations	Enable/disable the feature where views animate to their location when closed or opened.	Enabled

Automatic Updates

The following preferences can be changed on the **Install//Update > Automatic Updates** page:

Option	**Description**	**Default**
Automatically find new updates and notify me	When selected, Update manager will automatically search for update, as defined by the update schedule	Off

Option	**Description**	**Default**
Update Schedule	Look for updates on each startup, or once a day or some day a week, at a predefined time.	On Startup
Download Options	This option allows you to choose between having Eclipse search for updates and notifying you of them once they are available or having Eclipse automatically download new updates and asking you to install them.	Search and notify
When updates are found	Choose to be notified about new updates only once, or to receive reminders until the updates are installed.	Notify once

Available Software Sites

The following preferences can be changed on the **Install//Update > Available Software Sites** page:

Option	Description
Add/Edit/Remove	Add, edit or remove a software site.
Test Connection	Tests to see if it is possible to connect to the selected software site.
Disable/Enable	Allows a software site to be disabled so that the update system will not try to find software on the site when updating or installing software.
Import/Export	Import or export the list of software sites from a file.

Capabilities

The **General > Capabilities** preference page allows you to enable or disable various product components such as plug-in development. By default, the page only shows general categories of behavior. If you would like configure fine-grained capabilities you should use the "Advanced" dialog.

NOTE: Some capability selections have dependencies on other capabilities, disabling a required capability while leaving dependant capabilities enabled will only result in them becoming re-enabled.

When attempting to enable an action after its capability has been disabled or has yet to be enabled in the preferences page, the following **Confirm Enablement** prompt will appear verifying that you do indeed want to enable the required capability. Click **Details** to display a description of the capability.

Colors and Fonts

Many of the fonts and colors and used by eclipse components can be set using the **General > Appearance > Colors and Fonts** preference page.

A tree is used to navigate among and show a short preview of the various colors and fonts. The current face (but not size) of any font is previewed in its label. Colors are previewed in the icon associated with its label. Additionally, some categories (Workbench in particular) provide a more detailed preview of their contributions. This preview is shown below the description area if available.

Font settings can be changed either by selecting the font from the list and clicking **Use System Font** to choose the Operating System font setting or by clicking **Change** to open up a font selection dialog. **Reset** can be used to return to the default value.

Color settings can be changed by clicking **color** to the right of the tree area when a color is selected. **Reset** can be used to return to the default value.

The Colors and Fonts text field can be used to filter the contents. Simply type in an entry and any matching results will remain in the tree view.

Descriptions and previews are provided when the Workbench colors and font settings are selected.

Compare/Patch

The following preferences can be changed on the **General > Compare/Patch** page.

General Options

Option	Description	Default
Open structure compare automatically	This option controls whether a structure compare is automatically performed whenever a content compare is done. Turn this option off if you don't want to see the structural differences.	On
Show structure compare in Outline view when possible	If this option is on, structure compare will be displayed in the Outline view whenever it is possible.	Off
Show additional compare information in the status line	If this option is on, additional information about a change is shown in the status line. Turn this option on if you are interested in additional information about a change.	Off
Ignore white space	This option controls whether or not whitespace change are shown in the compare viewer. Turn this option on if you want to see changes in whitespace.	Off
Automatically save dirty editors before browsing patches	This option controls whether any unsaved changes are automatically saved before a patch is applied. Turn this option on if you want to save changes automatically.	Off
Added/Removed lines	These options control which lines should be counted as added and removed lines when applying a patch. Both options are based on regular expressions.	
Filtered Members	This option allows you to filter members that should be excluded from 'Compare With Each Other'.	

Text Compare Options

Option	Description	Default
Synchronize scrolling between panes in compare viewers	The two comparison viewers will "lock scroll" along with one another in order to keep identical and corresponding portions of the code in each pane side-by-side. Turn this option off if you do not want the compare viewers to lock scroll.	On

Option	Description	Default
Initially show ancestor pane	Sometimes you want to compare two versions of a resource with the previous version from which they were both derived. This is called their *common ancestor*, and it appears in its own comparison pane during a three way compare. Turn this option on if you want the ancestor pane to always appear at the start of a comparison.	Off
Show pseudo conflicts	Displays pseudo conflicts, which occur when two developers make the same change (for example, both add or remove the exact same line of code or comment). Turn this option on if you want pseudo conflicts to appear in compare browsers.	Off
Connect ranges with single line	Controls whether differing ranges are visually connected by a single line or a range delimited by two lines.	On
Highlight individual changes	Controls whether the individual changes inside conflicts are highlighted.	On
When the end/beginning is reached while navigating an element	Use this option to configure what occurs when the end/beginning is reached while navigating an element. **Prompt:** If this option is on and you selected to compare a single element you will be asked whether you want to go to the beginning/end of the element after the end/beginning is reached. If you are comparing two or more elements you will be asked whether you want to go to the beginning/end of the current element or to go to the next/previous element. Moreover, if you choose to remember your decision, this option will be changed to one of the below respectively. **Loop back to the beginning/end:** When this option is on, the selection will me moved back to the beginning/end after you reach the end/beginning of an element. **Go to the next/previous element:** If you are comparing two or more elements and this option is on after you reach the end/beginning of an element the next/previous element will be opened.	Prompt

Content Types

The **General > Content Types** preference page enables you to edit content types and their associated file names and character sets. You can also associate arbitrary file names or file extensions with content types. A *content type* acts as a description of a certain class of files (for instance, XML files). Eclipse uses this description in various scenarios, such as editor look-ups and file comparisons.

To access the **Content Types** preference page, select **Window > Preferences > General > Content Types**.

By selecting a content type in the topmost tree, you can alter the file names and extensions that are associated with it.

Note: Certain items will be marked as "locked". An item is locked if it is one of the associations provided by the plug-in that declares the content type. In other words, you can remove only user-contributed associations.

Adding an association is as simple as clicking **Add...**. A dialog prompts you to enter the file name or extension

In addition to adding and removing file names or extensions, you can also set the default encoding for a given content type. To do this, simply enter the encoding name in the provided field and click **Update**.

CVS

The following sections describe the preferences available in the tab groups of the **Team > CVS** preferences page.

General

On the General tab group of the CVS preference page you can customize several aspects of the CVS Plug-in

Option	Description	Default
Validate server version compatibility on first connection	Use this option to enable a query of the CVS server version on the first connection to determine server compatibility. The server version will be output to the console and if an incompatibility is detected a warning message will be logged when connecting.	Enabled
Confirm move tag on tag operation	Use this option to be prompted when the Move tag option is chosen when tagging.	Enabled
Display detailed protocol output to stdout	Use this option to display the communication trace between the Workbench and a CVS server	Disabled
Refresh tags when comparing or replacing tags	Use this option to have the Compare with and Replace With tag dialogs automatically refresh the known tags by contacting the server	Enabled
Automatically share projects containing CVS meta information	Use this option to have any imported project that was checked out from CVS using a different CVS tool automatically shared with CVS.	Enabled
Use .project project name instead of module name on check out	Use this option to use the project name stored in the .project file as the project name. If the checkout wizard cannot find a .project file, it will fallback to use the module name.	Disabled
Maximum number of files displayed when committing	Use this option to limit the number of files that get displayed in the commit dialog.	1000
Maximum number of comments on history	Use this option to limit the number of comments that get displayed in the commit dialog.	10

CVS Annotate

The following preference can be changed on the **Team > CVS > Annotate** preference page.

Option	Description	Default
Attempt to annotate a binary file	Use this option to configure whether the annotate operation should attempt to annotate a binary file. The choices are: **Yes:** Always attempt to annotate a binary file. **No:** Never attempt to annotate a binary file. **Prompt:** Prompt user whether to attempt to annotate a binary file.	Prompt

CVS Comment Templates

The **Team > CVS > Comment Templates** preference page allows you to create, edit and remove comments that will show up in the comments section of the commit dialog. This can be helpful if you need to enter the same string (or some part of a string) for a comment in all commits.

Option	Description
New	Opens the 'Enter Template Comment' dialog which allows you to enter a new comment.
Edit	Opens the selected comment in the 'Enter Template Comment' dialog.
Remove	Removes the selected comment.

CVS Console

The following preferences can be changed on the **Team > CVS > Console** preference page.

Option	Description	Default
Fixed width	Use this option to fix the width of console lines. Enabling this option allows the width to be specified. The default width is 80.	Disabled
Limit console output	Use this option to limit the number of characters to be buffered by the console. The default buffer size is 500000.	Enabled
Show CVS output in Console view	Use this option to show the output of CVS commands in the Console view. Enabling this option may show useful information but will slow down command operation	Disabled
Console text color settings	Use these options to change the colors for the text shown in CVS Console. Command line text *(black)* Message text *(blue)* Error text *(red)*	

CVS Ext Connection Method

The following preferences can be changed on the **Team > CVS > Ext Connection Method** preference page.

Option	Description	Default
Use external program vs. Use internal connection method	This page allows you to configure the ext connection method to use an external program or another connection method to to connect to a server. The later option is provided to allow custom connection method.	Use external program
Parameters	Use this option to configure the parameters passed to the CVS_RSH program. The default parameter pattern is {host} -l {user}. It can be tailored using the {host}, {user}, {password} and {port} variables.	-l {host} {user}

Option	Description	Default
CVS_SERVER	Use this option to configure the name of the remote CVS server program to run. Change this setting only if the remote CVS server binary name is different than the default.	cvs
Connection type	Use this option to set the connection method to be used for repository locations that use the ext connection method, if the option to use another connection method is enabled.	pserver

CVS Label Decorations

The following preferences can be changed on the **Team > CVS > Label Decorations** preference page.

Option	Description
General	Use the options on this page to configure general preferences about the decorators: **Compute deep outgoing state for folders:** Use this option to configure if the outgoing indicators on folders should be calculated. Disabling this option improves the performance of the decorators because calculating the dirty state for folders requires computing the dirty state for all child resources. (*enabled by default*) **Enable font and color decorations:** Use this option to enable font and color decorations. The colors and fonts used for outgoing changes, ignored resources, etc. can be configured on the General > Appearance > Colors and Fonts preference page
Text	Use the options on this page to configure how CVS information will be added to text labels
Icons	Use the options on this page to configure the which icons can be used as overlays to show CVS specific information in views.

CVS Synchronize/Compare

The following preferences can be changed on the **Team > CVS > Synchronize/Compare** preference page.

Option	Description
Consider file contents in comparisons	Use this option to compare contents for changed files found when comparing CVS resources. Usually, time stamps are used to compare CVS files, and this is by far the fastest method. However, in some cases a more accurate comparison can be achieved by comparing file content. Disabling this option will speed up comparisons but may result in compare entries whose contents are the same. This option only applies to comparisons and merges but not Workspace synchronizations.
Show revision comparisons in dialog	Use this option to show revision comparisons in a dialog instead of a compare editor.
Automatically enable change set grouping in CVS synchronizations	Use the option to have change sets enabled by default in those CVS synchronizations that support them.

Option	Description
Allow models to participate in synchronizations	Use this option to display model elements in the sync view. If this element is disabled, only resources will be shown in the sync view.
Open a compare editor when comparing a single file	Use this this option to open a compare editor when comparing a single file. If disabled, the results of the comparison will be displayed in the Synchronize view.

CVS Update/Merge

The **Team > CVS > Update/Merge** preferences page contains options which affect the update/merge workflows.

Option	Description	Default
When performing an update	Use this option to select what the workflow should be for performing an update. The options are: **Preview all incoming changes before updating**: All incoming changes are displayed in either the synchronize view or a dialog. You must explicitly select which resources to update and deal with conflicts. **Update all non-conflicting changes and preview remaining updates**: All non-conflicting resources are updated automatically and a list of conflicts (if applicable) are presented to you in either the synchronize view or a dialog. **Never preview and use CVS text markup to indicate conflicts**: All non-conflicting resources are updated automatically and conflicts are dealt with by adding the conflicting code to the file and using CVS text markup to delineate the boundaries of the conflict. You must then go through each file that had a conflict, look for the markup, and manually merge the contents.	Never preview and use CVS text markup to indicate conflicts
When an update preview is required	Use this option to configure how to preview incoming changes. Changes can be previewed in a dialog or in the Synchronize view.	Preview updates in the Synchronize view.

CVS Watch/Edit

The following preferences can be changed on the **Team > CVS > Watch/Edit** preference page.

Option	Description	Default
Configure projects to use Watch/edit on checkout	Use this option to indicate that files checked out from the repository should be made read-only.	Disabled
Enable temporary watches on edit	Use this option to receive a notification whenever a user starts to edit a file that you are currently editing.	Disabled

Option	Description	Default
When read-only files are modified in an editor	Use this option to configure what occurs when a read-only file is modified in an open editor or by another tool. The options are: **Send a cvs edit notification to the server:** Issues a cvs edit notification to the server before making the file writable. If other editors exist on the file, you will be prompted and may continue or cancel. **Send a cvs edit notification to the server in the background:** Issues a cvs edit notification to the server in the background after making the file writable. This allows you to keep typing uninterrupted while the edit notification is sent to the server. **Edit the file without informing the server:** Makes the file read-only without notifying the server.	Send a cvs edit notification to the server in the background
Before a CVS edit notification is sent to the server	Use this option to configure what occurs when a cvs edit notification needs to be send to the server. The options are: **Always Prompt:** Always prompt for confirmation. **Only prompt if there are other editors:** Shows the user the list of current editors and allows you to confirm or cancel the edit. **Never Prompt:** Send the edit notification without prompting	Only prompt if there are other editors
Update edited files	Use this option to configure what occurs before editing a file. The options are: **Always update before editing:** Always update a file before editing. **Prompt to update if out of date:** Prompt for confirmation of update only if a file is out of date. **Never update:** Never update a file before editing.	Never update

Editors

You can change the following preferences on the **General > Editors** preference page:

Option	Description	Default
Size of recently opened files list	Each file that is opened in an editor is stored in a list of recently used files in the **File** menu. This option controls the number of files that is displayed in that list	4
Show multiple editor tabs	Specifies whether you wish to show multiple editor tabs. If off, editor workbooks have one large tab and all non-visible editors are accessible only from the chevron.	On
Close editors automatically	Specifies whether or not to re-use editors in the Workbench. If on, you may specify the number of editors to use before they are recycled (the default is 8). You can also specify if a prompt dialog should be opened or if a new editor should be opened when all editors are "dirty" (have unsaved changes). Once it is turned on, the Pin Editor action is added to the toolbar and editor tab menu. Pinned editors are not recycled.	Off

External Tools

The following preferences can be changed on the **Run/Debug > External Tools** page.

You can configure whether or not you are prompted before external tool project builders are migrated

File Associations

On the **General > Editors > File Associations** preference page, you can add or remove file types recognized by the Workbench. You can also associate editors with file types in the file types list.

File types list

- **Add...**: Adds a new file or file type (extension) to the predefined list. In the resulting New File Type dialog, type the name of a file or a file extension. If you are adding a file extension, you must type either a dot or a "*." before the file type (e.g., ".xml" or "*.xml" as opposed to simply "xml").
- **Remove**: Removes the selected file type from the list

Associated editors list

- **Add...**: Adds a new editor to the list of editors associated with the file type selected above. In the resulting Editor Selection dialog, you can choose an editor to launch either inside the Workbench (internal) or outside the Workbench (external); click **Browse** to locate an editor yourself if the editor you want is not displayed in the list.
- **Remove**: Removes the association between an editor and the file type selected above. **Note:** Any editor that is bound by content type may not be removed from this list. Currently, there is no mechanism available to remove these editors.
- **Default**: Sets the selected editor as the default editor for the file type selected above. The editor moves to the top of the Associated Editors list to indicate that it is the default editor for that file type.

General

General settings for the Workbench. The term *Workbench* refers to the desktop development environment.

Each Workbench window contains one or more perspectives. Perspectives contain views and editors and control what appears in certain menus and tool bars. More than one Workbench window can exist on the desktop at any given time.

The following preferences can be changed on the **General** preference page.

Option	Description	Default
Always run in background	Turn this option on to perform long running operations in the background without blocking you from doing other work.	Off
Keep next/previous part dialog open	If this option is turned on then the editor and view cycle dialogs will remain open when their activation key is let go. Normally the dialog closes as soon as the key combination is release.	Off

Option	Description	Default
Open mode...	You can select one of the following methods for opening resources: Double click - Single clicking on a resource will select it and double clicking on it will open it in an editor. Single click (Select on hover) - Hovering the mouse cursor over the resource will select it and clicking on it once will open it in an editor. Single click (Open when using arrow keys) - Selecting a resource with the arrow keys will open it in an editor. **Note:** Depending on which view has focus, selecting and opening a resource may have different behavior.	Double-click

Help

On the **Help** preferences page, you can indicate how to display help information.

Option	Description	Default
Use external browsers	If embedded web browser is supported on your system, help window uses an embedded help browser to display help contents, whenever possible, and this option is available. Select it, to force help to use external browsers. Use "Web Browser" preference page to select browser to use.	Off
Open window context help	This option allows you to determine whether the window context help will be opened in a dynamic help view or in an infopop.	in a dynamic help view
Open dialog context help	This option allows you to determine whether the dialog context help will be opened in a dynamic help section of help view or in an infopop.	in dialog tray
Open help view documents	This option allows you to determine whether the documents selected in the help view will be opened in-place or in the editor area.	in-place

Help Content

Help topics from remote servers can be included seamlessly into the local help system. Use the **Help > Content** preference page to configure one or more remote server to include content from.

Option	Description
Include help content from a remote infocenter	If checked, this option enables the use of remote help content. The rest of the fields on the page are only enabled if this option is checked.
Add/Edit/Delete	Add, edit or delete a remote data source
View Properties	View the properties for this remote data source
Test Connection	Tests to see if it is possible to connect to this host/port combination

Option	Description
Disable/Enable	Allows a data source to be disabled so the help system will not try to read topics from that source.

Install/Update

The following preferences can be changed on the **Install//Update** page:

Option	Description	Default
Browsing for updates	This option allows you to configure what software versions are shown in the **Available Updates** wizard. You can select to show only the latest version of an update, or all versions of updates that are available.	Show latest
When software selected for an install wizard may not be compatible	This preference allows you to configure how incompatibilities are reported. If you attempt to update, uninstall, or install software that is not compatible with the software you already have installed, you can elect to open a wizard to edit your software selections, or you can just have the problem reported in the error log without opening the wizard. By default you will be prompted about what you want to do in this situation.	Ask me

Keys

The function of the keyboard can be extensively customized in Eclipse using the **General > Keys** preference page. Within Eclipse, key strokes and key sequences are assigned to invoke particular commands.

Key Strokes, Key Sequences, and Key Bindings

A 'key stroke' is the pressing of a key on the keyboard, while optionally holding down one or more of these modifier keys: `Ctrl`, `Alt` (`Option` on the Macintosh), `Shift`, or `Command` (only on the Macintosh.) For example, holding down `Ctrl` then pressing A produces the key stroke `Ctrl+A`. The pressing of the modifier keys themselves do not constitute key strokes.

A 'key sequence' is one or more key strokes. Traditionally, Emacs assigned two or three key stroke key sequences to particular commands. For example, the normal key sequence assigned to `Close All` in emacs is `Ctrl+X Ctrl+C`. To enter this key sequence, one presses the key stroke `Ctrl+X` followed by the key stroke `Ctrl+C`. While Eclipse supports key sequences of arbitrary lengths, it is recommended that keyboard shortcuts be four key strokes in length (or less).

A 'key binding' is the assignment of a key sequence to a command.

Schemes

A 'scheme' is a set of bindings. Eclipse includes two schemes:

- Default
- Emacs (extends Default)

The *Default* scheme contains a general set of bindings, in many cases recognizable as traditional key sequences for well known commands. For instance, `Ctrl+A` is assigned to `Select All`, and `Ctrl+S` is assigned to `Save`.

The *Emacs* scheme contains a set of key bindings familiar to users of Emacs. For instance, `Ctrl+X H` is assigned to `Select All`, and `Ctrl+X S` is assigned to `Save`.

It is important to understand why the *Emacs* scheme says that it 'extends Default'. The *Emacs* scheme is not a complete set of bindings like the *Default* scheme. Rather, it borrows from the *Default* scheme where possible, only defining explicit Emacs-style bindings where they vary from the *Default* scheme. Generally, only well known commands like `Select All`, `Save`, etc. have specific Emacs key sequences associated with them.

Choose the scheme you are most comfortable with by changing the 'Scheme' setting on the keys preference page. If you choose the *Default* scheme, all *Emacs* bindings are ignored. If you choose the *Emacs* scheme, explicit Emacs-style key sequence assignments take precedence over any conflicting assignments in the *Default* scheme.

Contexts

Key bindings can vary based on the current context of Eclipse.

Sometimes the active part might be a file editor, for instance, where a different set of key sequence assignments may be more appropriate than if the active part was an html file editor. This context is usually determined by the active part, but it can be influenced by the active window or dialog as well. If the active part does not choose a particular context, the workbench will set the active context to *In Windows*.

Eclipse includes a number of different contexts. Some examples are:

- In Dialogs and Windows
- In Windows (extends In Dialogs and Windows)
- In Dialogs (extends In Dialogs and Windows)
- Editing Text (extends In Windows)
- Debugging (extends In Windows)
- In Console

Much like configurations, contexts can extend other contexts.

Note: It is not recommended to promote a key binding to a context which it extends. For example, it is not recommended to move an *Editing Text* key binding to the *In Dialogs and Windows* context. This may have unexpected results.

It is possible for some key bindings to work in dialogs. Those key bindings are assigned to the *In Dialogs and Windows* context. One example of such a key binding is the key binding for "cut". It is possible to change these key bindings. For example, it is possible to have Ctrl+X as cut in dialogs, but Ctrl+W as cut in windows.

Platform and Locale

Key bindings also vary by platform and locale. On the Macintosh platform, `Command+S` is assigned to `Save`, instead of the usual `Ctrl+S`. On Chinese locales (zh), `Alt+/` is assigned to `Content Assist`, instead of the usual `Ctrl+Space`.

The current platform and locale is determined when Eclipse starts, and does not vary over the course of an Eclipse instance.

Customizing Key bindings

With multi-stroke key sequences, schemes, and contexts, there are a lot of things to keep in mind when customizing key bindings. To make things easier, all key customization is done on the **General > Keys** preference page.

Label Decorations

Label Decorations allow additional information to be displayed in an item's label and icon.

The **General > Appearance > Label Decorations** preference page provides a description of each decoration and allows the selection of which decorations are visible

Linked Resources

The **General > Workspace > Linked Resources** preference page is used when working with linked **resources**. The preference **Enable linked resources** is used to globally enable or disable the linked resource feature for the entire workspace. By default, linked resources are enabled. If you disable linked resources, then you will not be able to create any new linked resources, or import existing projects that contain linked resources.

Not all versions of the workbench support linked resources and recognize them as such. You may not want to use linked resources if you plan to share your workspace data with other users. Disable this preference if they will not be able to work with linked resources.

The remainder of this page is for defining **patth variables** that are used when creating linked resources. Use the **New** button to define new variables, the **Edit** button to change the value of an existing variable, and the **Remove** button to get rid of an existing variable. Note if you change a path variable that is currently in use, you will need to perform a local refresh on those projects to "discover" what is different in the file system. You can refresh a resource by opening the one of the navigation views' context menu for that resource and selecting **Refresh**. It is not recommended that you remove a path variable that is currently in use.

Local History

The following preferences can be changed on the **General > Workspace > Local History** page.

Option	Description	Default
Days to keep files	Indicates for how many days you want to maintain changes in the local history. History state older than this value will be lost.	7 days
Maximum entries per File	Indicates how many history states per file you want to maintain in the local history. If you exceed this value, you will lose older history to make room for new history.	50 entries
Maximum file size (MB)	Indicates the maximum size of individual states in the history store. If a file is over this size, it will not be stored.	1 MB

Network Connections

The following sections describe the preferences available on the **General > Network Connections** preference page.

Option	Description	Default
Active Provider	Specifies the settings profile to be used when opening connections. Choosing the **Direct** provider causes all the connections to be opened without the use of a proxy server. Selecting **Manual** causes settings defined in Eclipse to be used. On some platforms there is also a **Native** provider available, selecting this one causes settings that were discovered in the OS to be used.	Native (if present) Manual (otherwise
Proxy entries	The table displays entries that are available for all providers. Checkboxes in the first column of the table indicate entries to be used for the currently selected provider.	
Proxy bypass	Use this table to specify, either by name or pattern, which hosts should not use any proxy. A direct connection will always be used for matching hosts. Check boxes in the first column of the table indicate entries to be used for the currently selected provider.	

Perspectives

On the **General > Perspectives** preference page, you can manage the various perspectives defined in the Workbench.

Option	Description	Default
Open a new perspective	Use this option to set what happens when you open a new perspective. Do you want the perspective opened within the current Workbench window or opened in a new window?	In the same window
Open a new view	Use this option to specify what happens when a new view is opened. It is either opened to its default position within the current perspective or it is opened as a fast view and docked to the side of the current perspective.	Within the perspective
New project options	Use this option to specify the perspective behavior when a new project is created. You can set it to switch the current perspective to be the one associated with the project type and open the perspective in the same Workbench window as the current one, switch the perspective and open it in a new Workbench window, or not to switch perspectives at all.	Open perspective in the same window

Quick Diff

The following preferences can be changed on the General > Editors > Text Editors > Quick Diff preference page.

Option	Description	Default
Enable quick diff	This option will enable or disable the quick diff option.	On
Show differences in overview ruler	This option will show differences in the overview ruler.	Off

Option	Description	Default
Colors - Changes	This option controls the color of changes.	
Colors - Additions	This option controls the color of additions.	
Colors - Deletions	This option controls the color of deletions.	
Use this reference source	This option sets which reference to use as the base for generating quick diff comparisons. Options are: **Version on Disk:** Current file is compared against the last saved version on disk. **Latest CVS Revision:** Current file is compared against the latest CVS revision for the file.	Version on Disk

Search

The General > Search preference page allows you to set preferences for searches.

Option	Description	Default
Reuse editors to show matches	This option allows you to keep using the same editor for search results to reduce the number of open editors.	On
Bring Search view to front after search	This option will display the search view at the front after performing a search	On
Ignore potential matches	Select this option if you only want to see exact matches.	Off
Emphasize potential matches	This option allows you to highlight potential matches in the Search view. If the Search engine isn't 100% sure about the match then it is considered a potential match.	On
Foreground color for potential matches	This option allows you to select the foreground color for potential matches.	
Default perspective for the Search view	This option allows you to define which perspective should be brought to the front when there are new search results.	Non

Secure Storage

The **Secure Storage** preference page is used to manage storage of encrypted information such as passwords. Typically you will have no reason to alter the preferences on this page. The options here are mostly for troubleshooting, and, to a lesser degree, for system administrators and power users.

Password options

The **Password** tab combines functionality related to the master password lifecycle and password providers.

The **Clear Passwords** button clears cached master passwords from memory. This is analogous to logging out of the secure storage. Note that some password providers obtain credentials from the operating system automatically. To prevent them from doing so, you'll need to log out from the operating system account.

The **Master password providers** section contains a list of currently available password providers. By default, the enabled provider with the highest priority is used to encrypt data added to secure storage. The priority range is from 0 to 10, with 10 being the highest. A password provider can be disabled it if malfunctions, or if you prefer a lower priority password provider.

Note that data can only be decrypted by the same provider that encrypted the data. This means that changes to the list of the password providers affect only new entries. The password provider for existing entries can only be overwritten by the application storing the data.

By default all password providers are enabled.

Each password provider that has been used at least once will have a master password associated with it. Use the **Change Password...** button can to change the master password of the selected password provider.

The **Recover Password...** button opens the password recovery dialog. Use this option if you have forgotten the master password and have configured password recovery questions. The button will be disabled if the password recovery setup was cancelled when the master password was created. Note that the answers for the password recovery questions have to be entered exactly as they were specified during the password recovery setup. Answers are case-sensitive and white space inside answers are significant.

Contents options

The **Contents** tab displays contents of the default secure storage.

Secure storage is organized as a tree where nodes represent context of the information and values are associated with each node. Selecting a node in the tree will display a table of values associated with that node. Values stored in a non-encrypted form will be displayed; the encrypted values will be shown as "**********".

At the bottom of this tab, you will find the actual file location used to persist secure storage data.

To force changes to the contents of secure storage to be saved, click **Save**.

To delete stored data to recover from an error or to reflect a change in the setup, click **Delete**. This will delete **all** of the contents of secure storage. In some cases, other parts of the application may depend on the contents of secure storage that you deleted. To avoid unexpected errors, it is highly recommended to restart the application after secure storage has been deleted.

Advanced options

The **Advanced** tab of the preferences page offers some extra tweaks to secure storage.

Changes in the encryption algorithm are only applied to data stored after the change. If you have already created a secure storage it would have to be deleted and re-created to use the newly selected encryption algorithm.

Spelling

The following preferences can be changed on the General > Editors > Text Editors > Spelling preference page.

Option	Description	Default
Enable spell checking	This option enables spell checking.	On
Ignore words with digits	This option ignores words with digits when performing spell checking	On
Ignore mixed case words	This option ignores mixed case words when performing spell checking.	On
Ignore sentence capitalization	This option ignores sentence capitalization when performing spell checking.	On
Ignore upper case words	This option ignores upper case words when performing spell checking.	On

Option	Description	Default
Ignore internet addresses	This option ignores internet addresses when performing spell checking.	On
Ignore non-letters at word boundaries	This option ignores non-letters at word boundaries when performing spell checking.	On
Ignore single letters	This option ignores single letters when performing spell checking.	On
Platform dictionary	This option selects a platform dictionary to use.	Default (depends on Platform)
User defined dictionary	This option selects a User defined dictionary to use.	
Maximum number of correction proposals	This option limits the possible corrections displayed to the given value.	20

Startup and Shutdown

The General > Startup and Shutdown preference page allows the selection of plug-ins to be automatically activated during workbench startup.

Normally plug-ins are not activated until they are needed. However some plug-ins may specify that they wish to be activated during startup. This preference page allows the selection of which of these plug-ins will actually be activated during startup.

Option	Description	Default
Refresh workspace on startup	If this option is turned on then the workbench will synchronize its contents with the file system on startup	Off
Confirm exit when closing last window	If this option is turned on then the workbench will ask if you wish to exit when closing the last window	On
Plug-ins activated on startup	This option allows you to select which available plug-ins should be activated on startup.	

Workspaces

The General > Startup and Shutdown > Workspaces preference page allows configuration of the workspace prompting during IDE startup.

Option	Description	Default
Prompt for workspace on startup	If this option is turned on then the workbench will prompt you each time it is started for what workspace to use.	On
Number of recent workspaces to remember	The maximum number of recently-used workspaces that will be remembered and presented in the "Workspace Launcher" dialog.	5
Recent workspaces	The list of recently used workspaces that are presented in the "Workspace Launcher" dialog.	<none>

Team

The Team preferences page contains options which affect the version management Team support.

Option	Description	Default
Show all synchronization information in a resource's text label	Use this option to have the synchronization state of a resource displayed as text in the resource's label. By default, only an icon decorator is used to identify a resource's synchronization state.	Disabled
Show the file author in compare editors	Use this option to display the file author in all compare editors opened by an action on a repository provider (if supported by the repository provider)	Disabled
When editing file in non-shared projects, automatically make files writable if prompting is not possible	Use this option to make the file writable in a non-shared project if prompting is not possible.	Disabled
Reuse open compare editors when opening comparisons	Use this option to have a new comparison open in an existing compare editor. Enabling this option will decrease number of compare editors open.	Enabled
Run Project Set import in the background	Use this option to run Project Set import in the background.	Disabled
Choose the presentation to be used when displaying Workspace projects	Use this option to configure the default layout used when a synchronization is first added to the Synchronize view. The layout can be subsequently changed in the view drop down menu.	Compressed Folders
Open the associated perspective when a synchronize operation completes	Use this option to configure what happens when a synchronization operation is run. The options are: Always: always switch perspectives Never: never switch perspectives Prompt: prompt to switch perspectives	Prompt
Perspectives	Use this option to configure which perspective is to be shown when a synchronize operation is run.	Team Synchronizing

Team File Content

On the Team > File Content preference page, you can associate file names or extensions with the type of data the file contains. The two choices for file content type are ASCII and Binary. Repository providers such as CVS can then use this information to provide content type appropriate behavior. For example, for ASCII files, CVS ensures that line terminators conform to those of the OS platform.

Entries are added to the File Content page in two ways. The first is through contributions from workbench plug-ins. Tools integrated into the workbench provide the workbench with the file content types for file extensions specific to the tool. The workbench itself also defines the file content type for file extensions that are common and appear frequently in the workbench (e.g. html,.png, etc.).

The second method is for users to add file content types explicitly on the File Content preference page. To do so, you can simply click on the **Add Extension** button and enter an extension or click on the **Add Name** button and add a name. Following this, you can toggle the type associated with the name or extension by selecting the entry for the extension in the table and clicking **Change**. Entries can be removed from the list by selecting them and clicking **Remove**. Only those entries that were manually added can be removed. Entries contributed through the plug-in manifest can only be changed.

Team Ignored Resources

On the Team > Ignored Resources preference page, you can specify file name patterns to exclude from the version control management system. There is a list of file patterns against which resources will be matched before they are considered as version control candidates. These patterns may contain the wildcard characters "*" and "?". This pattern "*" represents any sequence of zero or more characters. This pattern "?" represents any one character. For example, you can specify a pattern of "*~", which would match any temporary files that end with "~". Any file or directory that matches any one of the patterns will be ignored during update or commit operations.

To add a file type to the ignore list, simply click **Add Pattern**. In the dialog, enter a file type (e.g. *.class). To remove a file type from the ignore list, simply select the file type in the ignore list and click **Remove**.

You can temporarily disable ignoring the file pattern by de-selecting it from the list. You do not have to remove the specified file pattern from the list to temporarily disable it.

Team Models

The Team > Models preference page allows you to select which model providers are to be included in Team operations.

Text Editors

The following preferences can be changed on the General > Editors > Text Editors page.

Appearance options

Option	Description	Default
Undo history size	This option allows you to set the size of the undo history for text editors.	200
Displayed tab width	This option allows you to set the displayed tab width for text editors.	4
Insert spaces for tabs	This option allows you to insert space characters in place of tab characters	Off
Highlight current line	This option controls whether or the current line is highlighted or not.	On
Show print margin	This option controls whether the print margin is visible or not.	Off
Print margin column	This option allows you to set the print margin column position.	80
Show line numbers	This option controls whether or not line numbers are shown on the left side of the text editor.	Off
Show range indicator	This option controls whether or not range indicators are shown in the text editor.	On
Show whitespace characters	This option controls whether to display whitespace characters in text editors.	Off
Enable drag and drop of text	This option controls whether text drag and drop is enabled.	On
Warn before editing a derived file	This option controls whether to warn if a derived file is going to be edited.	On

Option	Description	Default
Smart caret positioning at line start and end	This option controls whether the editor automatically positions the caret and the start or end of a line.	On
Show affordance in hover on how to make it sticky	This option controls whether to show an affordance in the hover on how to make it sticky	On
Appearance color options	This option controls various appearance colors.	

Web Browser

The following preferences can be changed on the General> Web Browser preference page.

Option	Description	Default
Use internal Web browser	This option enables you to use an internal Web browser.	On
Use external Web browser	This option enables you to use an external Web browser. Select the required browser from the list of available external web browsers.	Off

Workspace

On the General > Workspace preference page, you can manage various IDE-specific workspace preferences settings in the Workbench.

Option	Description	Default
Workspace save interval (in minutes)	This number indicates how often the state of the workspace is automatically saved to disk	5
Refresh automatically	If this option is turned on then the workspace resources will be synchronized with their corresponding resources in the file system automatically. **Note**: This can potentially be a lengthy operation depending on the number of resources you have in your workspace.	Off
Open referenced projects when a project is opened	If this option is enabled opening a project will also open and closed projects it references. Select prompt if you wish to be asked first.	Prompt
Text file encoding	Use this option to specify the encoding to use when saving text files in editors.	Default (CP1252)
Text File line delimiter	Use this option to specify the line delimiter to use for new text files. **Note:** This will generally not effect the file line delimiter for existing files.	Default

Team Support with CVS

Contents

CVS

The Workbench is shipped with a built in client for the Concurrent Versions System (CVS). With this client you can access CVS repositories.

You can find more information on CVS at http://ximbiot.com/cvs/.

CVS Checkout Wizard

This wizard helps you check out one or more projects from a CVS repository. It is available from the Import menu, the New > Project menu and the toolbar of the CVS Repository Exploring perspective. It is also opened when performing a Checkout As from the CVS Repositories view.

The first page of the checkout wizard allows you to choose an existing repository location or create a new one. If you choose to create a new location, the page from the New Repository Location wizard is shown.

After a repository location is selected, you can now choose one or more modules to check out. You can either enter a module path or select one or more from the list of existing modules. A dot (.) can be entered to checkout the entire contents of the repository for those rare cases where the repository represents a single project.

Once one or more modules are selected, you can choose to check out one or more selected modules as a new projects or into an existing project or folder. If one module is selected and it does not contain a .project file (the Workbench project configuration file), you will also have the option to configure the project using the new project wizard. The option to check out subfolders is selected by default. Also, you are given the option to add the checked out modules to a working set.

For a single project, you can configure the location of the project to be either the default location or a custom location outside the workspace. For multiple projects, you can configure the parent folder where all the projects should be located.

CVS Console

The CVS Console is shown in the Workbench Console view. The CVS entry in this view shows the output from CVS commands. This output is similar to that of the CVS command line client. If an error occurs during a CVS operation opening the console can help troubleshoot the cause of the error.

CVS History View

This view provides a list of all the revisions of a resource in the repository as well as all the revisions of a resource in the local history. From this view you can compare revisions, load a revision, revert the corresponding workspace file to a revision, tag with an existing tag, show annotations and open an editor on a revision.

Columns

The History View is able to display both local and remote revisions. Local History revisions have minimal information associated with them - only a time stamp. As a result, the majority of the column items will appear blank for Local History revisions. Remote revisions, on the other hand, will have a much more complete set of information. Here are the columns:

Revision

Since the CVS History view can display both remote and local revisions. For remote revisions:

- the column displays the remote revision icon:
- the column displays the revision number in the history.

For local history revisions:

- the column displays the local history revision icon:

Tags

The tags that are associated with the revision. Selecting a revision line will list the tags in the lower left pane of the view. *(Remote revisions only)*

Revision Time

This column displays the creation date and time of the revision in the history.

Author

This column displays the name of the person who created and authored the version. *(Remote revisions only)*

Comment

This column displays the comment (if any) supplied for this revision at the time it was committed. Selecting a revision line will show the complete comment in the lower right pane of the view. Also, mousing over a dotted comment will result in showing a tooltip with complete comment. *(Remote revisions only)*

Toolbar

Refresh

This command refreshes the contents of the view, fetching the latest history information for the resource from the server.

Link with Editor and Selection

When enabled, the view will display the history for the resource of the active editor or of the active selection.

Pin

When enabled will pin the view and its contents. Any new requests for history will open a new instance of the History View.

Navigation History

When enabled will allow you to select from a drop down menu a resource for which a history has been viewed in the past. The action is disabled when the History view is opened for the first time.

Group Revisions by Date

When enabled, all history items will be sorted into one of the following date categories:

- Today
- Yesterday
- This Month
- Previous

Local and Remote Revisions

When enabled will display both local and remote revisions of the resource in the History view.

Local Revisions

When enabled will display only the local revisions of the resource in the History view.

Remote Revisions

When enabled will display only the remote revisions of the resource in the History view.

Collapse All

Will the view is in the Group by Date mode, Collapse All will collapse all of the date categories.

Compare Mode

When enabled, a double click (or a single click depending on your Open Mode strategy set in Preferences > General) will open a compare editor. When it is disabled, clicking on a revision will open that revision.

Drop Down Menu

The drop down menu contains the following items:

- **Wrap Comments**: Will wrap comments in the Comment Viewer.
- **Show Comment Viewer**: shows the Comment Viewer.
- **Show Tag Viewer**: shows the Tag Viewer.
- **Show Search Field**: shows the Search Field.
- **Filter....**: Opens the Filter Dialog.
- **Remove Filters**: Removes any filtering from the History View.

Context menu

From the context menu of the CVS Resource History view you can perform a number of interesting operations.

Open

This command will open the contents of the selected revision in a read only editor. (The editor used is the one that is registered as the default editor for the file type). *Note*: If the revision being opened is the current version of the file, then it will be opened in a regular editor.

Open With

This menu allows choosing an editor to open the contents of the selected revision. *Note*: If the revision being opened is the current version of the file, then it will be opened in a regular editor, otherwise a read only editor is used.

Compare Current With Revision/Compare With Each Other

The compare command differs based on the number of revisions selected in the history view. For a single selection, **Compare Current With Revision** will the compare the current version of the resource with the selected revision. For two selections, **Compare with Each Other** will compare the selected revisions.

Get Contents

This command will load the contents of the selected revision into the local copy of the file whose history is displayed in the view. The local file can then be committed to make the contents of latest revision in HEAD (or a branch) match the contents of the selected revision.

CVS Repositories View

The CVS Repositories view, which is part of the CVS Repository Exploring perspective, shows the CVS repository locations that you have added to your Workbench. Expanding a location reveals the main trunk (HEAD), project versions and branches in that repository. You can further expand the project versions and branches to reveal the folders and files contained within them.

The pop-up menu for this view also allows you to specify new repository locations. Use the CVS Repositories view to check out resources from the repository to the Workbench, configure the branches and versions shown by the view, view resource history, and compare resource versions.

CVS Workspace Synchronization

A CVS workspace synchronization launched using the Team > Synchronize menu command or the Synchronize toolbar command in the Team Perspective will appear in the Synchronize view. This view allows you to inspect the differences between the local Workbench resources and their remote counterparts as well as update resources in the Workbench and commit resources from the Workbench to a repository.

CVS Merge Synchronization

CVS Merge synchronizations launched using the **Team > Merge** menu command will appear in the Synchronize view. This view allows you to inspect the differences between the local Workbench resources and their remote counterparts that are being merged and allows you to update the local resources. Committing is not supported when merging.

Features

The features of the CVS Merge Synchronization are similar to those of the CVS Workspace Synchronization with the following differences.

- **Modes:** When merging, only incoming and conflicting changes are possible. As such, only the Incoming Mode and Conflicts mode are supported.
- **Operations:** Only updating is supported when merging.
- **Ongoing merges:** In order to compensate for the poor ongoing merge support of CVS, a CVS Merge Synchronization can be pinned and kept indefinitely. The timestamps of merged changes are recorded and, on refresh, only new changes since the last merge are shown in the view.

Merge Wizard

This wizard helps you merge changes between two states of a project into your workspace. Often the Merge Wizard is used to move changes from one branch into another, for example after splitting a branch to work on a bug fix. The merge operation takes changes between two points in a branch, the **start point** and the **end point**, and merges them into your workspace. Typically the start point will be the root of a branch (version tag) and the end point can either be the tip (latest and best) of the branch or another version tag

It is very important to understand that the destination of the merge is always the project in your workspace. After the merge has been completed you can test the changes locally and then commit them to the new branch (usually HEAD).

To start a merge, select a project (or one or more resources), and select **Team > Merge...** from the pop-up menu.

First, you should choose the end tag of the merge. This is the branch or version being merged into the workspace. Choose a version when you want to merge the differences between two versions of a project into your workspace. Choose a branch if you want to merge the changes made in the branch into your local workspace.

The wizard will try to pick an appropriate tag for the start tag or base tag. If one cannot be determined, you should enter it manually. If there is no start tag, you can choose to merge without a preview, in which case a start tag is not required but the merge will happen directly into the workspace. The drawback of this is that CVS uses a text base markup to identify conflicts and this is cumbersome to work with.

If the merge preview option is selected, after the finish button is pressed, the changes between the start point and end point are calculated and displayed as a CVS Merge Synchronization in the Synchronize view. Depending on the size of the project you are merging, this may take some time but can be run in the background. In the Synchronize view you can update or merge changes into your Workbench. Also, you have the option to automatically merge non-conflicting changes which means you will only need to deal with the conflicting changes in the Synchronize view.

Add CVS Repository Wizard

This wizard helps you create a repository location.

Fields

The following are the editable fields for a new CVS repository location:

Option	Description	Default
Host	The address of the host (e.g., "hostmachine.com").	<blank>
Repository Path	The path to the repository on the host (e.g., "/x/y/z", "d:/myrepo").	<blank>
User Name	The user name under which you want to connect to the server.	<blank>
Password	The password required for the above user name to access the above host.	<blank>
Connection Type	The type of CVS connection for the repository, either "pserver", "extssh", "pserverssh2" or "ext".	pserver
Port	Option to use a custom port for the connection.	Use Default Port
Validate connection on finish	Check this option if you want to attempt a connection to the host upon completion of the wizard to validate that the entered information is correct. If this option is not enabled, you will not know whether the information entered is correct until you try to access the contents of a repository for the first time.	On

Option	Description	Default
Save Password	Check this option if you want the password to be saved between sessions. To persist login credentials in a safe, encrypted form Secure Storage is used.	Off

CVS Sharing Wizard

This wizard helps you share a project with others using a CVS repository. It is available from the **Team > Share Project** menu command of views that display resources such as one of the navigation views.

The first page of the Sharing wizard allows you to choose the type of repository you would like to share you project with. If the repository type you wish to share with is not listed, you can click the Show All Wizards checkbox to see repository types that are installed but not enabled.

Selecting CVS will bring you to the CVS Sharing wizard. On the first page of the CVS wizard, you can select the repository you wish to share with. If the repository is not in the list, you can select to create a new repository. If you choose to create a new location, the page from the New Repository Location wizard is shown. After you have selected the repository, you must indicate the name of the module that will be used to share the project. You can choose to use the same name as the local project or you can type in a path containing one or more folder segments indicating the remote module. There is also an option to reconnect the local project to an existing module.

The final page of the CVS Sharing wizard allows you to decide which resources to commit and which to ignore. There are toolbar items for changing the layout of the view and for committing and ignoring resources. If there are resources left in the view when Finish is pressed, you will be prompted to commit the remaining resources in the background (if the *Launch the Commit Wizard* option is enabled) which will allow you to return to work while the commit takes place.

CVS Label Decorations

Label Decorations are used by CVS to show important synchronization information about workspace resources. Decorations can effect the text or the icon of the label associated with a resource. The text decorators are configurable on the CVS Label Decorations Preference page which can also be used to indicate which icons are to be enabled. Here is a description of the icons used by CVS.

Decoration	Resource type	Description
🟧	Any	Indicates that the resource is under version control.
✱	Any	Indicates that the resource or one of its children contains outgoing changes.
✎	File	Indicates that the project containing the file has been configured to use watch/edit and the file is being edited locally.
⊞	File	Indicates that the file contains a merge conflict which has not yet been resolved. The conflict is considered resolved once the file is modified and saved.
🟩	Project or Folder	Indicates that the folder is under version control but does not correspond to an existing remote directory but is instead most likely a module defined in the CVSROOT/modules file.
?	File or Folder	Indicates that the resource is not under version control.

NOTE: Resources that are ignored by CVS will not be decorated by any of the above CVS decorations.

Ignoring Resources from Version Control

When synchronizing resources, it is possible that there are some resources that you do not want to commit to the repository. There are two ignore facilities provided, allowing you to specify what resources should be excluded from update and commit operations.

The first is a global ignore facility, provided by the Workbench. The second is the CVS ignore facility, which reads the contents of a special file *.cvsignore* to determine what to ignore.

Why ignore files when synchronizing?

There are many files that a user may not want to commit to the repository. For example, external editors may create temporary files in your project. In addition, they may be re-generated whenever a build is performed, resulting in many outgoing changes. Typically these are not files that one wants to share with other members of a team or persist in the repository.

Global ignore facility

A global ignore facility is provided by the Workbench via the Team preference page. There is a list of file patterns against which resources will be matched before they are considered as version control candidates. These patterns may contain the wildcard characters "*" and "?". "*" represents any sequence of zero or more characters. "?" represents any one character. For example, you can specify a pattern of "*~", which would match any temporary files that end with "~". Any file or directory that matches any one of the patterns will be ignored during update or commit operations. When you specify a file pattern to ignore, you can temporarily disable ignoring the file pattern by de-selecting it from the list; you do not have to remove the specified file pattern from the list.

The patterns in the global ignore facility are matched against resource names during a synchronize operation. It is important to note that the path leading up to the resource name is not included in the matching. For example, for the file "/path/to/file.txt", only the string "file.txt" is matched against the patterns. This facility is not intended for specifying fully-qualified path names but for specifying globally-applicable patterns.

CVS ignore facility

The Eclipse CVS client recognizes a file named ".cvsignore" in each directory of a project. This is a standard CVS facility and many existing CVS projects may contain this file.

This text file consists of a list of files, directories, or patterns. In a similar way to the global ignore facility, the wildcards "*" and "?" may be present in any entry in the .cvsignore file. Any file or sub-directory **in the current directory** that matches any one of the patterns will be ignored. It is important to note that the semantics of this file differs from that of the global ignore facility in that it applies only to files and directories in the same directory as the .cvsignore file itself. A project may contain one .cvsignore file in each directory. For more information, please visit http://ximbiot.com/cvs/.

Resources that have not been added to CVS control can be ignored by selecting **Team > Add to .cvsignore** from the pop-up menu of the resource in one of the navigation views. This menu option is also available in the Synchronize view.

Secure Storage

The secure storage saves data in an encrypted form. On some operating systems, it uses your operating system account information to provide a single sign-on experience.

Frequently asked questions

How do I change a password?

You can change password via the **Change Password...** button on the **Secure Storage** preference page.

The **Secure Storage** preference page can be found under General > Security > Secure Storage.

Resolving problems

- **Lost secure storage password or invalid secure storage password**

  If you have specified password recovery questions/answers, you can use the **Secure Storage** preference page to recover the master password. After the master password has been recovered, it is strongly recommended that you change the master password using the **Change Password...** button on the **Secure Storage** preference page.

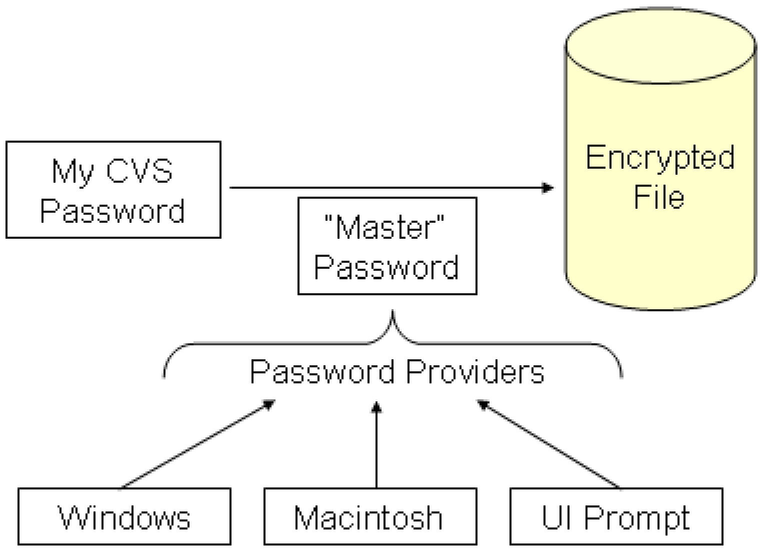
  The recovered password is not displayed, but stored in the memory cache until the application is closed.

- If all else fails

  If you have a problem with secure storage that you can not find a workaround for, the secure storage can be deleted using the **Delete** button on the **Secure Storage** preference page. This, of course, will delete all the contents of the secure storage. It is highly recommended that you restart the application after secure storage has been deleted.

How secure storage works

Let's consider a concrete example of CVS integration. When you specify a password for a CVS connection, the application offers you an option to save your user name and password using secure storage.



Your CVS password is passed as data to secure storage. Secure storage uses a "master" password to encrypt it and store the encrypted CVS password in a file on disk.

The master password is obtained from a password provider module. The master passwords are obtained in a "lazy" fashion, only when they are about to be used. Password providers can use different techniques:

- on *Windows*, the master password is generated as a random value that is encrypted based on your Windows login information and stored in secure storage;
- on *Macintosh*, the master password is initially created as a random value that is stored in the OS keyring;
- the default password provider simply prompts you for a password;
- other password providers might be supplied in your application.

When data is saved with secure storage, the password provider is selected based on the priorities from the list of enabled password providers. Only that provider can be used in future to decrypt the data.

Password recovery

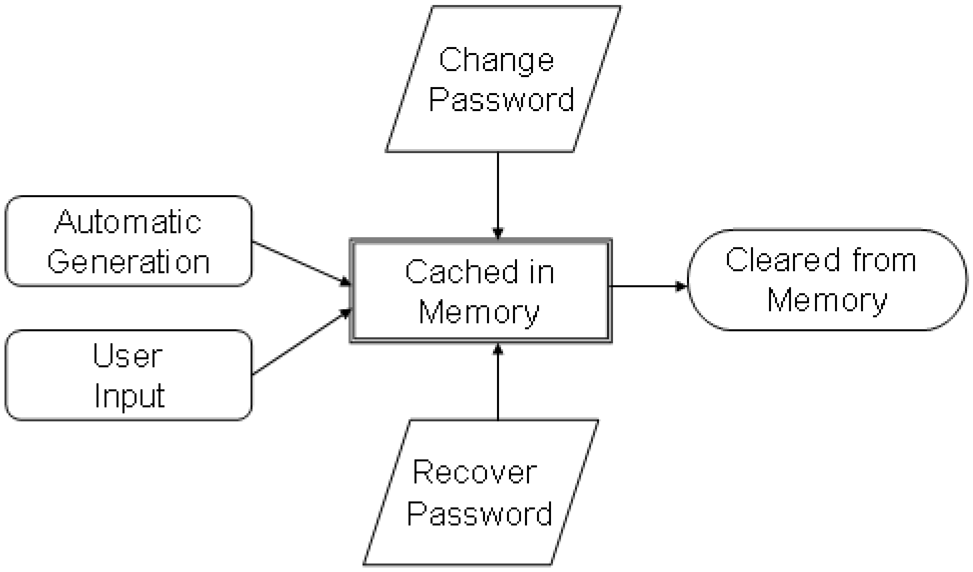
When the secure storage is used for the first time, it will generate a master password used to encrypt data. This same master password will be required to retrieve data from secure storage in future. In case the master password becomes unavailable, secure storage provides optional support for password recovery. If the master password is lost or forgotten, providing exactly the same answers to the questions will recover the master password and will allow data to be retrieved from secure storage. This can help prevent data loss in case you forget or lose your master password.

Pick password recovery questions that will be easy for you to answer, but hard for other people to guess. For example, consider asking about a memorable date, a place you liked when you were a child, or your favorite quote. The strength of secure storage is determined by its weakest link; don't use answers that are very short or easy to guess.

Life of a Master Password

The "master" password is used to encrypt and decrypt data stored by the secure storage. The master passwords are specific to providers: each provider has a separate master password.

The life of a master password begins when a password provider is asked for it for a first time. Depending on the provider, it will either generate a master password behind the scenes, or will ask you for some input. The same master password is then used for all subsequent use of this password provider.



Once the master password is obtained from the password provider, it is cached in memory until the application is closed or the password cache is cleared using the **Secure Storage** preference page.

The master password can be changed using the **Secure Storage** preference page. Depending on the provider, the password change operation might require some input from you or might happen completely behind the scenes.

In case the master password is lost, it can be recovered if password recovery questions and answers were specified. The password recovery allows working around both human and machine problems. For instance, if a UI prompt was used to enter a password and the user forgot the password. If an operating system integration module was used, the operating system might have been re-installed or an entry deleted in the system keyring that was used for the master password.

Secure storage runtime options

Changing location of secure storage

By default, secure storage is located in your home directory. On Windows that typically resolves to *"C:\Documents and Settings\<user_name>\.eclipse\org.eclipse.equinox.security"*. This location is selected to allow multiple Eclipse-based applications to share the same secure storage.

If you would like to modify the location of the default secure storage, you can use the "-eclipse.keyring <file_path>" runtime option. The <file_path> is a path to the file which is used to persist the secure storage data.

The current location of the default secure storage can be found on the General > Security > Secure Storage preferences page.

Password file

The password provider mechanism can be bypassed if you specify the "-eclipse.password <file path>" runtime option. In this case the contents of the file you specified as the argument will be used as a master password. While this option is valuable in some circumstances (such as headless applications), the protection of the password file becomes a consideration. The password file can be protected by the operating system access rights and/or by placing it on a removable storage, such as a USB key.

User interface information

Contents

Development Environment

Contents

Workbench Toolbar

The Workbench toolbar is displayed at the top of the Workbench window, directly underneath the menu bar. The contents of the toolbar change based on the active editor. Actions in the toolbar may apply to particular views, so these actions may be enabled or disabled based on the state of the currently active view or editor.

New Wizard

This command brings up a dialog where you can choose the type of resource to create.

Save The Open Editor Contents

This command saves the file currently displayed in the editor area.

Print

This command opens a dialog which allows you to specify where you would like to print the contents of the file currently being displayed in the editor.

External Tools

This command presents a drop-down menu which allows you to run or configure external tools.

Search

This command opens the search dialog, which allows you to search the workspace for specified text.

Navigation

This tool group contains a variety of editor navigation commands.

Perspective Bar

The perspective bar allows quick access to perspectives that are currently open, as well as providing an easy way to open a new perspective. The perspective bar may be docked in three different positions. It may be docked in the upper right corner (the default position), the upper left corner (under the main toolbar) and to the far left.

Open Perspective

This command opens a new perspective that is selected from a drop-down menu. All of the perspectives that are open within a single Workbench window are shown on the shortcut bar.

Perspective Buttons

These buttons provide a quick way to switch to one of the open perspectives in the current Workbench window.

Available Perspectives

There are several available perspectives, while one is set as a default, others can be manually added to the perspective bar. To add new perspectives to your workspace, click **Open Perspective**, select **Other** and choose from the following available perspectives:

- CVS Repository Exploring;
- Debug;
- Plug-in Development;
- Resource;
- Team Synchronizing.

Fast View Bar

The fast view bar is the place where fast views are docked. It may be docked on any side of the workbench

View Buttons

These buttons provide a quick way to display the fast views in the current perspective. Fast views are essentially minimized views that have been dragged onto the shortcut bar. Fast views pop up when selected, and revert back to their minimized state when you click outside of the view. They may be oriented either horizontally or vertically according to their configuration. To convert a fast view back into a normal view, click **Fast View** in the view's menu or drag it back to the workbench. To add a view to the fast view bar you can:

- Click the button on the left of the fast view bar. This will open a menu containing the views appropriate to the current perspective.
- Right click the view in the workbench and select **Fast View**.
- Drag the view from the workbench to the Fast View bar.

View Toolbars

View toolbars contain actions that apply only to the particular view in which they appear. The view toolbar also contains a context menu that contains other actions for that view. This menu is opened by clicking on the downwards pointing triangle. If there is enough space, view toolbars are in the view tab area. Otherwise they appear in the view.

Title Bar

View title bars contain the view name, its icon, and the view toolbar.

Perspective

A perspective defines the initial set and layout of views in the Workbench window. One or more perspectives can exist in a single Workbench window.

Perspectives can be opened either in the same (existing) Workbench window, hiding the current perspective, or in a new Workbench window

Perspectives define visible action sets, which you can change to customize a perspective. You can save a perspective that you build in this manner, making your own custom perspective that you can open again later.

The Workbench defines the *Resource* perspective by default. This perspective shows views relevant to resource management.

Local History

Local history of a file is maintained when you create or modify a file. Each time you edit and save a file, a copy of it is saved. This allows you to compare your current file state to a previous state, or replace the file with a previous state. Each state in the local history is identified by the date and time the file was saved.

Neither projects nor folders have local history. Local History is displayed in the History View.

To view the local history of a file, choose **Team > Show Local History** from the pop-up menu. This will bring up the History view and populate it with the revisions of the selected file. You can open different revisions from the table (by using **Open** from the context menu or by double clicking on a revision), compare them against the latest revision or against a previous revisions (by using **Compare With Revision** from the context menu), and replace the current revision with the contents of a previous revisions (by using **Get Contents** from the context menu).

NOTE: Projects that are managed by CVS now have the local history revisions displayed alongside remote revisions. In order to see the local history for a CVS managed file, select the **Team > Show History** menu item and make sure that the History View is one of the modes capable of showing local history.

Toolbar - Refresh

This command refreshes the contents of the view, fetching the latest history information for the resource from the server.

Link with Editor and Selection

When enabled, the view will display the history for the resource of the active editor or of the active selection.

Pin

When enabled will pin the view and its contents. Any new requests for history will open a new instance of the History View.

Group Revisions by Date

When enabled, all history items will be sorted into one of the following date categories:

- Today
- Yesterday
- This Month
- Previous

Collapse All

Will the view is in the Group by Date mode, Collapse All will collapse all of the date categories.

Compare Mode

When enabled, a double click (or a single click depending on your Open Mode strategy set in **Preferences > General**) will open a compare editor. When it is disabled, clicking on a revision will open that revision.

Context menu

From the context menu of the Local History view you can perform the following operations:

Open

This command will open the contents of the selected revision in a read only editor. (The editor used is the one that is registered as the default editor for the file type). *Note*: If the revision being opened is the current version of the file, then it will be opened in a regular editor.

Compare Current With Revision/Compare With Each Other

The compare command differs based on the number of revisions selected in the history view. For a single selection, **Compare Current With Revision** will the compare the current version of the resource with the selected revision. For two selections, **Compare with Each Other** will compare the selected revisions.

Get Contents

This command will load the contents of the selected revision into the local copy of the file whose history is displayed in the view.

List of Key Binding

The list of available key bindings in Eclipse depends on many factors, including what view or editor is selected, whether a dialog is open, what plug-ins are installed, and what operating and windowing system is being used. At any time, you can obtain a list of available key bindings using Key Assist (Help > Key Assist... or Ctrl+Shift+L). The following tables list some popular key bindings available in the Eclipse SDK.

New	Create a element or a new resource. Configure which elements are shown in the submenu in Window > Customize Perspective.	Ctrl + N
Close	Close the current editor. If the editor contains unsaved data, a save request dialog will be shown	Ctrl + F4
Close All	Close all editors. If editor contains unsaved data, a save request dialog will be shown.	Ctrl + Shift + F4
Save	Save the content of the current editor. Disabled if the editor does not contain unsaved changes.	Ctrl + S
Save As	Save the content of the current editor under a new name.	
Save All	Save the content of the current editor under a new name.	Ctrl + Shift + S
Print	Prints the content of the current editor. Enabled when an editor has the focus.	Ctrl + P
Properties	Opens the property pages of the select elements.	Alt + Enter

Switching Workspaces

The current workspace for Eclipse can be switched by using the **File->Switch Workspace** command. If you have already switched your workspace previously the previous workspaces will be available for selection in the Switch Workspace menu.

The Switch Workspace --> Other menu item will open the switch workspace dialog. The dialog will allow you to browse for or manually enter a new workspace location. The combo will also allow you to select your previously selected workspaces.

Settings Transfers

When you switch your workspace you can select settings than will be transferred to the new workspace. These settings are supplied by the org.eclipse.ui.preferenceTransfer extension.

The SDK supplies transfers for:

- Workspace Layout: Opened views, their size, and selected perspectives.
- Working Sets: The user defined working sets.

Views and Editors

Contents

Workbench Window Layout

You can rearrange the layout of Workbench windows as follows:

- Drag views to different positions within the Workbench window.
- Drag views to the shortcut bar to create a fast view.
- Drag editors such that they are simultaneously visible beside, above, or below another editor.
- Resize views and editors by dragging the sashes which separate them.
- Drop cursors

Drop cursors indicate where a view will dock when you release your mouse button. This indication is relative to the view or editor area underneath the cursor.

⬆	Dock above: If the mouse button is released when a dock above cursor is displayed, the view will appear above the view underneath the cursor.
⬇	Dock below: If the mouse button is released when a dock below cursor is displayed, the view will appear below the view underneath the cursor.
➡	Dock to the right: If the mouse button is released when a dock to the right cursor is displayed, the view will appear to the right of the view underneath the cursor.
⬅	Dock to the left: If the mouse button is released when a dock to the left cursor is displayed, the view will appear to the left of the view underneath the cursor.
🗗	Stack: If the mouse button is released when a stack cursor is displayed, the view will appear as a tab in the same pane as the view underneath the cursor.
🚫	Restricted: If the mouse button is released when a restricted cursor is displayed, the view will not dock there. For example, a view cannot be docked in the editor area.

Fast views

From a view's title bar context menu, you can select Fast View to minimize the view as a button on the shortcut bar.

If a view is minimized in this way, you can click its button on the shortcut bar to bring it up in a fast view. To revert the fast view back to a docked view again, select the *Fast View* button on its title bar.

Double-Click

Double-clicking a view or editor's title bar maximizes the part in the Workbench window.

Title bar context menu and fast view toolbars

From the context menu of a view or editor's title bar, you can select how you want the view to appear within the Workbench window.

Editor Area

The editor area is where you modify the contents of files in the Workbench.

Marker bar

The marker bar is the vertical bar located at the left of the editor area.

Markers

Markers are displayed in the marker bar, to the left of the text editor.

Depending on the type of file displayed in the editor area, three kinds of markers may be displayed:

- Bookmarks
- Task markers (for associated tasks)
- Debugging breakpoints

You can create and associate a marker with a specific line in a file by accessing the context menu from the marker bar, which is directly to the left of that line.

Types of editors

The Workbench uses three types of editors:

- Internal: These editors are launched inside the editor area in the Workbench window.
- External: You can go outside the Workbench in the file system, edit a Workbench file outside the Workbench, and save the edited file. For example, imagine that you add an SGML file to the Workbench. Later, you go into the file system and open the file in an SGML editor, then save the file. The edited SGML file is still represented in the Workbench, even though you did not edit the file in the Workbench. If you associate a file type with an external editor in the Workbench (General > Editors > File Associations preference page), then the Workbench will launch this external editor.
- ActiveX: On Microsoft Windows platforms, the Workbench makes use of ActiveX controls for applications that allow for them. For example, Microsoft Word supports being embedded as an OLE document. Thus if you have a *.doc* file in the Workbench, and Word is registered as the editor for *.doc* files in your operating system, then opening the file will launch Word as an OLE document within the Workbench editor area. Notice how OLE documents also add such features as menus and toolbar buttons.

Compare Editor

You can view the differences between two files by comparing them. You can compare different files, you can compare versions in the Workbench with versions in the repository, or with the local edit history. In some cases you can compare three files (when a common ancestor exists).

After a comparison is carried out, the compare editor opens in the editor area. In the compare editor, you can browse through all the differences and copy highlighted differences between the compared resources. You can save changes to resources that are made in the comparison editor.

Compare editor allows for two kind of navigation: using differences or changes. A change is a portion of text that has been modified within the line, and the difference is a section of file consisting of one or more lines, and can contain many changes.

Differences are marked with blue color, changes with red

Toolbar

The toolbar of the Compare editor includes the following buttons:

Switch Compare Viewer

> Basing on the content type the system determines which compare viewer should be used during a comparison. The button opens a drop down menu which allows to choose between other viewers registered for the same content type. If there is no alternative content viewer available the button is hidden.

Control Visibility of Ancestor Pane

> There are two conditions under which a three way compare will occur, both when using the Team version management support: when comparing a file that is in conflict, and when comparing a file being merged from a branch. In both cases, the system will determine a common ancestor in the repository to compare the conflict or merge against. This button determines the visibility of the third editor. By default, the ancestor pane is not visible.

Perform Three way/Two way Compare

> The compare editor can be toggled between performing a three way compare or a two way compare which ignores the common ancestor.

Copy All from Left to Right

> Copies the entire contents of the file in the left pane into the file in the right pane, making the contents of the two files identical.

Copy All Non-Conflicting Changes from Right to Left

> Copies all the non-conflicting changes from the right pane into the left pane. Conflicting changes must be copied individually.

Copy Current Change from Left to Right

> Merges changes in two files by copying the highlighted change in the left pane into the highlighted fragment on the right. This will overwrite the highlighted fragment in the right pane.

Copy Current Change from Right to Left

> Does the opposite of the one just described.

Select Next Difference

> Highlights the next difference that is found between the compared resources.

Select Previous Difference

> Highlights the previous difference that is found between the compared resources.

Select Next Change

Highlights the next change that is found between the compared resources.

Select Previous Change

Highlights the previous change that is found between the compared resources.

Search View

This view displays the results of a search.

Text searches will only search for expressions in files with extensions (file types) specified in the search dialog.

Show Next Match

This command highlights the next match of the search expression in the editor area, opening the file if required.

Show Previous Match

This command highlights the previous match of the search expression in the editor area, opening the file if required.

Remove Selected Matches

Removes all highlighted matches from the search results.

Remove All Matches

Remove all matches from the search results.

Expand All

Expand all matches in the hierarchical view.

Collapse All

Collapse all matches in the hierarchical view.

Run the current search again

Re-runs the most recent search.

Cancel Current Search

Cancel a search that is running.

Previous Search Results

This command allows you to browse previously conducted searches and repeat a previous search. You can select a previous search from the drop-down menu or clear the search history.

Pin the Search view

Pinning the search view means that subsequent searches will shown their results in another search view and that the pinned view remains unchanged.

Bookmarks View

The Bookmarks view displays user defined bookmarks

To add the Bookmarks view to the current perspective, see **Window > Show View > Other... > General > Bookmarks**.

The Description column contains a description of the bookmark. You can edit the description by selecting **Properties** from the context menu.

The Resource and Path columns provide the name and location of the resource associated with each bookmark.

The Location column indicates the line number of the bookmark within its associated resource.

Toolbar

The toolbar of the Bookmarks view includes the following buttons.

Delete: Delete the selected bookmark.

Go to: Open the bookmark's resource and navigate to the bookmarked region.

Menus

Click the icon at the left end of the view's title bar to open a menu of items generic to all views. Click the black upside-down triangle icon to open a menu of items specific to the Bookmarks view. Right-click inside the view to open a context menu.

Properties View

This view displays property names and basic properties of a selected resource.

Toolbar buttons allow you to toggle whether to display properties by category and whether to filter advanced properties. Another toolbar button allows you to restore the selected property to its default value.

To see more detailed information about a resource than the Properties view gives you, right-click the resource name in one of the navigation views and select Properties from the pop-up menu.

To add the Properties view to the current perspective, see **Window > Show View > Other... > General > Properties.**

Outline View

This view displays an outline of a structured file that is currently open in the editor area, and lists structural elements. The contents of the outline view are editor-specific.

CVS Views

There are several CVS specific views, the CVS Repositories view, the CVS Resource History view, the CVS Console view and the Synchronize view.

See Team Support in the Reference section for details on these views.

Tasks View

The Tasks view displays tasks that you add manually. You can associate a task with a resource in the Workbench, but this is not required.

By default, the Tasks view is included in the Resources perspective. To add it to the current perspective, see **Window > Show View > Other... > General > Tasks.**

Problems View

The Problems view displays system-generated errors, warnings, or information associated with a resource. These are typically produced by builders.

Wizards

Contents

New Project Wizard

This wizard helps you create a new project in the Workbench.

When you first bring up the New Project wizard, you need to select the type of project you want to create. Select the General type if you want to create a generic project. To assist in locating a particular wizard, the text field can be used to show only the wizards that match the entered text.

New Folder Wizard

This wizard helps you create a new folder in the Workbench.

Field	Description	Default
Enter or select the parent folder	The resource in which the new folder will be created. Type or navigate the list to select the resource	The resource that was selected when you chose to create the new folder
Folder name	The name for the new folder.	<blank>

Advanced

The **Advanced** button reveals or hides a section of the wizard used to create a linked folder. Check the **Link to folder in the file system** checkbox if you want the new folder to reference a folder in the file system. Use the field below the checkbox to enter a folder path or the name of a path variable. Use the **Browse...** button to browse for a folder in the file system. Use the **Variables...** button if you want to use a path variable to reference a file system folder.

New File Wizard

This wizard helps you create a new file in the Workbench.

Field	Description	Default
Enter or select the parent folder	The resource in which the new file will be created. Type or browse the list to select the resource.	The resource that was selected when you invoked the New File wizard.
File name	The name for the new file, including the file extension.	<blank>

Advanced

The **Advanced** button reveals or hides a section of the wizard used to create a linked file. Check the **Link to file in the file system** checkbox if you want the new file to reference a file in the file system. Use the field below the checkbox to enter a file path or the name of a path variable. Use the **Browse...** button to browse for a file in the file system. Use the **Variables...** button if you want to use a path variable to reference a file system file.

CVS Wizards

There are several CVS specific wizards. These include the Import CVS Project wizard (which is also available from the New Project wizard) and the New CVS Repository Location wizard. There are also wizards for several CVS operations, such as committing and merging.

See Team Support in the Reference section for details on these wizards.

Import Wizard

This wizard helps you import resources into the Workbench.

When the Import wizard first comes up, you must choose what type of import to do. To assist in locating a particular wizard, the text field can be used to show only the wizards that match the entered text**.**

Archive File

If you choose this option, you will import files from an archive file.

Field	Description	Default
Archive File	The file from which to import. Type in the full path or Browse to select the path on the file system.	<blank>
Filter Types...	Dialog to select which file types to import. Use this to restrict the import to only certain file types.	*N/A*
Select All	Check off all resources for import	*N/A*
Deselect All	Uncheck all resources.	*N/A*
Folder	The folder into which the resources will be imported. Type the path or Browse to select a path in the Workbench.	The folder holding the selected resource
Overwrite existing resources without warning	Determines whether importing a resource should silently overwrite a resource which already exists in the Workbench. If this option is off, you will be prompted before a given resource is overwritten, in which case you can either overwrite the resource, skip it, or cancel the import.	Off

Export Wizard

This wizard help you export resources from the Workbench.

When the Export wizard first comes up, you must choose what type of export to do. To assist in locating a particular wizard, the text field can be used to show only the wizards that match the entered text.

Archive File

Choose this option to export files to an archive file.

File System

If you choose this option, you will export files to the file system.

Help Contents

Contents

- Workbench User Guide
- Working with Cheat Sheets

Workbench User Guide

The Help view displays help related to using the Workbench. If you select Workbench User Guide in the list of books displayed in the Help window, you will see help topics related to using the Workbench in the Contents tab. The Workbench User Guide is broken down into four main sections, described below.

Getting started

This section contains tutorials that will help you when you start using the Workbench.

Concepts

Concepts are high level descriptions of the schema and functions of the Workbench. This section helps provide a general understanding of how the Workbench functions. For example, the Concepts section includes a discussion of what perspectives and views are and how they relate to one another.

Tasks

Task descriptions are step by step instructions for performing specific actions and tasks in the Workbench. For example, the Tasks section contains step by step instructions for creating a repository location, and for importing a file from the file system into the Workbench.

Reference

Reference materials are helpful resources that will assist you while you are using the Workbench. This includes descriptions of various wizards, dialogs, and fields as well as Workbench resources such as specific views and perspectives.

The Reference section also includes a glossary of terms that you might find useful while using the Workbench.

Working with Cheat Sheets

Eclipse provides cheat sheets to guide you through some of its application development processes. Each cheat sheet is designed to help you complete some task, and it lists the sequence of steps required to help you achieve that goal. As you progress from one step to the next, the cheat sheet will automatically launch the required tools for you. If there is a manual step in the process, the step will tell you to perform the task and click a button in the cheat sheet to move on to the next step. Also, relevant help information to guide you through a task is retrieved in a single click so that lengthy documentation searches will no longer be required.

A composite cheat sheet is another kind of cheat sheet that provides guidance for larger tasks or a group of related tasks.

Launching a cheat sheet

To launch a cheat sheet from the Workbench

1. Select **Help > Cheat Sheets** from the menu bar. If this command is not in the menu, it can be added from **Window > Customize Perspective > Commands**, and check **Cheat Sheets**.
2. The available cheat sheets are listed, select one and click **OK**. Alternatively if you know the location of a cheat sheet content file on your file system or on the web you can enter its path.

The cheat sheet opens as a view. At any time, only one cheat sheet is open and active. When you launch a cheat sheet, any opened cheat sheet is closed before the new one is opened. The completion status of closed cheat sheet is saved.

The cheat sheet has a toolbar at the top right edge. These icons appear in the toolbar:

- Collapses all the expanded steps except the current step or expands steps to the last expanded state. Click to toggle between these two states.
- Allows you to select and open another cheat sheet. The completion status of the active cheat sheet is saved. Then, the active cheat sheet is closed and the selected cheat sheet is opened.
- Hides the cheat sheet.
- Saves the completion status of the active cheat sheet and closes it.

Note that some cheat sheets can also be launched from the welcome page by first selecting **Tutorials** and then selecting one of the tutorials.

Starting the cheat sheet

Each cheat sheet has a list of steps and it always begins with an Introduction step. When you launch a fresh cheat sheet, the Introduction step is expanded so that you can read a brief description of the cheat sheet. To start working with the cheat sheet, click **Click to Begin** in that step. The next step is expanded and highlighted. You should also see one or more actions buttons, such as **Click to Perform** in the highlighted step. You can now begin working through the tasks using the cheat sheet. At any time, the only highlighted step in the cheat sheet is the current step.

Restarting the cheat sheet

Any time after starting a cheat sheet, you can restart from the first step by clicking **Click to Restart** in the Introduction step. If you have already created some artifacts, you will have to manually clean up the workspace before restarting the cheat sheet.

Progressing through the steps

In the current step, when you click **Click to Perform**, a tool (which can be a wizard), will be launched and you will be required to work with that tool. When you finish working with that tool, the next step is automatically highlighted and it becomes the current step. When the current step is a manual task, you will need to perform the work and click **Click to Complete** to move to the next step. A check mark appears in the left margin of each completed step.

Getting help information for tasks

To get step-by-step instructions for that step, click the help link in the step before you click **Click to Perform**, and the step-by-step instructions on how to work with that tool will be displayed in the Help window.

Additional help for entry fields in the tool or wizard may be available by focusing on the field (use the **Tab** key to position to that entry) and pressing **F1**.

Skipping a step

If a current step has a **Click to Skip** option, then it is an optional step. You must click **Click to Skip** to skip the current step, when you do, the step will have the skip mark in the left margin. If the task does not present **Click to Skip**, you must perform that step and you cannot skip it.

Redoing a step

You can redo any step that you may have completed or skipped in the current cheat sheet. To redo the step, expand the step by clicking its expand icon and then clicking **Click to Redo**. After redoing a step, the cheat sheet will continue from the redo step.

Closing the cheat sheet

When you finish the last step in a cheat sheet, it automatically restarts. You can also close the active cheat sheet by clicking the close icon in the cheat sheet's toolbar. The active cheat sheet saves its completion status when it is closed so that you can continue where you left off at a later time.

Search

Contents

File Search

In the Search Dialog, the File Search tab allows you to search for files or text in the Workbench. You can bring up the Search Dialog by clicking on the Search toolbar button.

Containing text

Type the expression for which you wish to do the text search. Leave this field empty to search for files.

From the drop-down menu, you can choose to repeat or modify a recent search.

Wildcards

The available wildcards for search expressions are displayed in the search dialog:

- "*" matches any set of characters, including the empty string
- "?" matches for any character
- "\" is the escape for a literal; if you want to search for an asterisk, question mark, or backslash character, type a backslash before it to indicate that you are not using these characters as wildcards (e.g., "*", "\?", or "\\")

File name patterns

In this field, enter all the file name patterns for the files to find or search through for the specified expression.

Wildcards

The available wildcards for file name patterns are displayed in the search dialog:

- "*" matches any set of characters, including the empty string
- "?" matches for any character

Case sensitive

Turn this option on if you want the text search to be case sensitive.

Scope

Choose the scope of your search. You can either search the whole workspace, pre-defined working sets, previously selected resources or projects enclosing the selected resources.

Workbench Menus

Contents

File Menu

The **File** menu enables you to create, save, close, print, import, and export Workbench resources and to exit the Workbench.

New (Shift+Alt+N)

Enables you to create new resources. Before you can create a new file, you must create a project in which to store the file.

Open File

Enables you to open a file for editing - including files that do not reside in the Workspace.

Close (Ctrl+W)

Closes the active editor. You are prompted to save changes before the file closes.

Close All (Shift+Ctrl+W)

Closes all open editors. You are prompted to save changes before the files close.

Save (Ctrl+S)

Saves the contents of the active editor.

Save As

Enables you to save the contents of the active editor under another file name or location.

Save All (Shift+Ctrl+S)

Saves the contents of all open editors.

Revert

Replaces the contents of the active editor with the previously saved contents.

Move

Enables you to move the currently selected resources to a different project.

Rename (F2)

Enables you to change the name of the currently selected resource.

Refresh (F5)

Refreshes the resource with the contents in the file system.

Convert Line Delimiters To

Alters the line delimiters for the selected files. Changes are immediate and persist until you change the delimiter again - you do not need to save the file.

Print (Ctrl+P)

Prints the contents of the active editor.

Switch Workspace

Opens the **Workspace Launcher**, from which you can switch to a different workspace. This restarts the Workbench.

Import

Launches the **Import** wizard, which enables you to add resources to the Workbench.

Export

Launches the **Export** wizard, which enables you to export resources from the Workbench.

Properties (Alt+Enter)

Opens the **Properties** dialog for the currently selected resource. You can learn:

- The path to the resource in your file system
- The date of the last modification
- Whether a file is writable or executable, and what its encoding is
- Whether a project's resources inherit their encoding and line delimiters or whether they are set to a particular value.

Recent file list

Contains a list of the most recently accessed files in the Workbench. You can open any of these files from the **File** menu by simply clicking the file name. You can control the number of files in this list from the Editors preference page.

Exit

Closes and exits the Workbench.

Edit Menu

This menu helps you manipulate resources in the editor area.

Undo

This command reverses your most recent editing action.

Redo

This command re-applies the editing action that has most recently been reversed by the Undo action.

Cut

This command removes the selection and places it on the clipboard.

Copy

This command places a copy of the selection on the clipboard.

Paste

This command places the text or object on the clipboard at the current cursor location in the currently active view or editor.

Delete

This command removes the current selection.

Select All

This command selects all text or objects in the currently active view or editor.

Find/Replace

This command allows you to search for an expression in the active editor, and optionally replace the expression with a new expression.

Find Next

This command allows you to search for the next occurrence of the current selection, or for the next occurrence of the most recent expression found using the Find/Replace action.

Find Previous

This command allows you to search for the previous occurrence of the current selection, or for the previous occurrence of the most recent expression found using the Find/Replace action.

Incremental Find Next

This command allows you to search for expressions in the active editor. As you type the search expression, it will incrementally jump to the next exact match in the active editor. While in this mode, the up and down cursor keys can be used to navigate between matches, and the search can be cancelled by pressing left or right cursor keys, the enter key, or the escape key.

Incremental Find Previous

This command allows you to search for expressions in the active editor. As you type the search expression, it will incrementally jump to the previous exact match in the active editor. While in this mode, the up and down cursor keys can be used to navigate between matches, and the search can be cancelled by pressing left or right cursor keys, the enter key, or the escape key.

Add Bookmark

This command adds a bookmark in the active file on the line where the cursor is currently displayed.

Add Task

This command adds a task in the active file on the line where the cursor is currently displayed.

Word Completion

This action will attempt to complete the word currently being entered in the active editor.

Set Encoding

This action launches a dialog that allows you to change the file encoding used to read and write the file in the active editor.

Navigate Menu

This menu allows you to locate and navigate through resources and other artifacts displayed in the Workbench.

Go Into

This command refocuses the active view so that the current selection is at the root. This allows web browser style navigation within hierarchies of artifacts.

Go To

- **Back**: This command displays the hierarchy that was displayed immediately prior to the current display. For example, if you Go Into a resource, then the Back command in the resulting display returns the view to the same hierarchy from which you activated the Go Into command. This command is similar to the Back button in an HTML browser.
- **Forward**: This command displays the hierarchy that was displayed immediately after the current display. For example, if you've just selected the Back command, then selecting the Forward command in the resulting display returns the view to the same hierarchy from which you activated the Back command. This command is similar to the Forward button in an HTML browser.
- **Up one level**: This command displays the hierarchy of the parent of the current highest-level resource.
- **Resource**: This command allows you to navigate quickly to a resource. For more information see the links to related tasks below.

Open Resource

This command displays a dialog that lets you select any resource in the workspace to open it in an editor. For more information see the links to related tasks below.

Show In

This sub-menu is used to find and select the currently selected resource in another view. If an editor is active, these commands are used to select the resource currently being edited in another view.

Next

This command navigates to the next item in a list or table in the active view. For example, when the search results view is active, this navigates to the next search result.

Previous

This command navigates to the previous item in a list or table in the active view. For example, when the search results view is active, this navigates to the previous search result.

Last Edit Position

This command allows you to jump the last edit position.

Go to Line

This command allows you to jump to a specific line in the active editor.

Back

This command navigates to the previous resource that was viewed in an editor. Analogous to the **Back** button on a web browser.

Forward

This command navigates to undo the effect of the previous **Back** command. Analogous to the **Forward** button on a web browser.

Project Menu

The Project menu allows you to perform actions (builds or compilations) on projects in the Workbench.

Open Project

This command opens the currently selected project or projects. The selected projects must currently be closed for this command to be available.

Close Project

This command closes the currently selected project or projects. The selected projects must be currently open for this command to be available. Closing a project will remove all of that project's state from memory, but the contents on disk are left untouched.

Build All

This command performs an incremental build on all projects in the Workbench. That is, it builds (compiles) all resources in the Workbench that are affected by any resource changes since the last incremental build. This command is only available if auto-build is turned off. Auto-build is turned off via the **Build Automatically** menu option or from the **General > Workspace** preference page.

Build Project

This command performs an incremental build on the currently selected project. That is, it builds (compiles) all resources in the project that are affected by any resource changes since the last build. This command is only available if auto-build is turned off. Auto-build is turned off via the **Build Automatically** menu option or from the **General > Workspace** preference page.

Build Working Set

This menu allows you to performs an incremental build on a working set. That is, it builds (compiles) all resources in the working set that are affected by any resource changes since the last build. This command is only available if auto-build is turned off. Auto-build is turned off via the **Build Automatically** menu option or from the **General > Workspace** preference page.

Properties

This command opens a dialog showing the properties of the selected project or of the project that contains the selected resource.

Window Menu

This menu allows you to display, hide, and otherwise manipulate the various views, perspectives, and actions in the Workbench.

New Window

This command opens a new Workbench window with the same perspective as the current perspective.

New Editor

This command opens an editor based on the currently active editor. It will have the same editor type and input as the original.

Open Perspective

This command opens a new perspective in this Workbench window. This preference can be changed on the **General > Perspectives** preference page. All of the perspectives that are open within the Workbench window are shown on the shortcut bar.

The perspectives you will likely want to open are listed first. This list is dependent on the current perspective. From the **Other...** submenu you can open any perspective.

Show View

This command displays the selected view in the current perspective. You can configure how views are opened on the **General > Perspectives** preference page. Views you are likely to want to open are listed first. This list is dependent on the current perspective. From the **Other...** submenu you can open any view. The views are sorted into categories in the Show View dialog.

Customize Perspective

Each perspective includes a predefined set of actions that are accessible from the menu bar and Workbench toolbar.

Help Menu

This menu provides help on using the Workbench.

Welcome

This command will open the welcome content.

Help Contents

This command displays the help contents in a help window or external browser. The help contents contains help books, topics, and information related to the Workbench and installed features.

Search

This command displays the help view opened on the Search page.

Dynamic Help

This command displays the help view opened to Related Topics page.

Key Assist

This command will display a list of key bindings

Tips and Tricks.

This command will open a list of interesting productivity features that you may not have discovered.

Cheat Sheets

This command will open the cheat sheet selection dialog.

Check for Updates

This command will check for updates to the installed software.

Install New Software

This command allows you to download and install new software.

Icons and Buttons

Contents

Editor Area Marker Bar

The following markers can appear in the marker bar (to the left of the editor area):

Icon	Description
🔖	Bookmark
●	Breakpoint
☑	Task marker
⇨	Search result
⊗	Error Marker
⚠	Warning marker
ℹ	Information marker

Tasks View

The following markers can appear in the Tasks view:

Icon	Description
!	High Priority Task
⇩	Breakpoint
☑	Completed Task

Toolbar Buttons

The following buttons may appear in the Workbench toolbar, toolbars for views, and the shortcut bar:

Button	Description	Button	Description
⊞	Open a new perspective	💾	Save the active editor contents

Button	Description	Button	Description
💾	Save the contents of all editors	💾	Save editor contents under a new name or location
🔦	Opens the search dialog	🖨	Print editor contents
🗔	Open a resource creation wizard	📄	Open a file creation wizard
📁	Open a folder creation wizard	📁	Open a project creation wizard
📥	Open the import wizard	📤	Open the export wizard
✂	Cut selection to clipboard	📄	Copy selection to clipboard
📋	Paste selection from clipboard	↶	Undo most recent edit
↷	Redo most recent undone edit	⇩	Navigate to next item in a list

ER/Studio Software Architect

Welcome to ER/Studio Software Architect 1.0 for Eclipse, a design-driven environment for modeling applications.

The following offers additional assistance, information, and resources

- For information on how to use this Help system, see Help on Help.
- For points to additional information not included in this document see Additional Product Information.
- Getting Started with ER/Studio Software Architect
- Embarcadero’s Home Page
- Embarcadero’s Product Support

Overview

ER/Studio Software Architect is a design-driven environment for modeling software applications. It features support for UML 2.0, OCL, patterns, quality assurance audits and metrics, import/export support, and automated documentation generation. The goal is to model applications using UML.

ER/Studio Software Architect is an Eclipse-based RCP application, and the primary components are a Model Navigator, Diagram Editor, Palette, and Properties View. ER/Studio Software Architect will have only one perspective, based on the Modeling perspective.

Features

This section provides an overview of the features available in the ER/Studio Software Architect application.

Projects

Work in ER/Studio Software Architect will be done in the context of a project. A project is a logical structure that holds all resources required for the user's work. All projects located in the selected Workspace will be listed in the Model Navigator. Project properties can be set up when the project is being created, and modified further, using the Properties dialog.

The following is the projects that can be created in ER/Studio Software Architect:

- **Pattern Definition project** is a profiled UML 2.0 modeling project that allows creation of new patterns.
- **Profile Definition project** is a profiled modeling project that allows you to create new profiles.
- **UML 2.0 project** is a design project with no source code support.

Packages

The notion of a package has two facets: logical and physical.

Logically, a model consists of one or more packages. A package is a model element used to group elements, and provides a namespace for the grouped elements. A package can contain packageable elements (the elements that can be directly owned by a package) and the other packages. A model itself is a package.

Physically, a package is a folder containing the files that store diagrams and model elements. Contents of a package can be displayed on a special type of the Class Diagram that is synchronized with the package contents, i.e. all the classifiers directly owned by this package automatically appear on the package diagram. Each package contains the single package diagram that is created automatically and cannot be added explicitly.

The root package of a project (Model) is usually referenced as the default package. The package diagram of this package is called the default diagram. This diagram is created and opened just after the modeling project creation.

Diagrams

Each modeling project contains a set of diagrams that are graphical representations of parts of the model. Diagrams contain graphical elements (nodes connected by paths) that represent model elements.

Each diagram will belong to a certain diagram type (for example, UML 2.0 Class Diagram). The diagram type will define the typical contents of the diagram (the kind of elements that are usually placed on this diagram) and the notation used to represent the model elements. For example, a Class in a UML 2.0 project can be added to the Class Diagram and to the Composite Structure Diagram and will have different representations there. Each diagram will have the specific Palette and context menu that allows creation of the model elements specific to this diagram type. These tools can be customized.

Diagrams will exist within the context of a project. You will have to create or open a project before creating a new diagram. The set of available diagram types will depend on the type of project. In UML 2.0 project you will have a set of standard UML diagrams defined in UML2.0 specification. Along with the design diagrams that are explicitly created by the user, ER/Studio Software Architect models will have so-called "Package" diagrams. These diagrams will have the Class Diagram type, but they will be generated automatically for each package and show its contents.

Source code generation and synchronization for diagrams will not be supported.

Diagram Format

The diagrams created with ER/Studio Software Architect are stored in XML-based files with the extension *.txv<diagram_type>. For example, the file <name>.txvcls would correspond to a class diagram. Design elements are stored either in the package files (default.txaPackage) or in separate XML-based format files with the extension .txa<element_type>, depending on your choice when creating a project in the New Project wizard.

The XML-based diagram format is compatible across the product line, and it you can copy and reuse diagrams created in our different products.

Containment Metamodel

ER/Studio Software Architect will handle the logical and physical containment of design elements as follows:

- Design elements are created as children of packages.
- All elements shown on diagrams are shortcuts (or references) to actual model elements, therefore when a new element is created on a diagram, ER/Studio Software Architect creates this element in the package and adds its shortcut to the diagram.
- Clipboard actions will operate with references, if the source and target containers of the action are diagrams.
- You can also create design elements in separate files (standalone design elements) or in one file (filemates).

Model Elements

Each model in a modeling project is a set of entities that are instances of metaclasses of the metamodel chosen for the project. These instances are the Model Elements.

Each model element will have a set of properties and notations defined for its metaclass. For example, when you create a UML 2.0 project, every element created in this project instantiates a metaclass from the UML 2.0 metamodel. That is, each actor on a use case diagram in a UML 2.0 project is an instance of usecases/Actor, and each component is an instance of components/Component.

The model elements will have graphical notation and can be explicitly placed on diagrams, are nodes and links.

Model Shortcuts

A shortcut is a representation of a model element placed on a diagram. You can create multiple shortcuts to the same element on different model diagrams. Modifications to an element itself can be made from any diagram containing its shortcut and are propagated to all of its shortcuts. Modifications to shortcut view properties made from any diagram wouldn't affect the representations of this element on other diagrams. Shortcuts can be removed from a diagram without removing the element from the model.

You can create shortcuts to elements within the same project. To create a shortcut to an element from another workspace project, this project would have to be added to the Model Path of the current project.

Model Hyperlinking

You can create hyperlinks from diagrams or model elements to other system artifacts and browse directly to them. Hyperlinks can be used for the following purposes:

- Link diagrams that are generalities or overviews to specifics and details.
- Link diagrams or elements to external documentation.

You can create hyperlinks to:

- An existing diagram or diagram element from any project in the workspace.
- A resource in the workspace.
- An external document (file or URL)

Use case diagrams typically represent the context of a system and system requirements. Usually, one begins at a high level and specifies the main use cases of the system. Next, you would determine the main system use cases at a more granular level. As an example, a "Conduct Business" use case can have another level of detail that includes use cases such as "Enter Customers" and "Enter Sales". Once you have achieved the desired level of granularity, it is useful to have a convenient method of expanding or contracting the use cases to grasp the scope and relationships of the system's use case views.

The hyperlinking feature of ER/Studio Software Architect will allow you to create browse-through sequences comprised of any number of use cases or any other diagrams. By browsing the hyperlink sequence, you can follow the relationships between the use case diagrams. You can use hyperlinking to link diagrams and elements based on requirements. For example, you could create a hierarchical browse-through sequence of use case diagrams, creating hyperlinks within the diagrams that follow a specific actor through all use cases that reference the actor.

Model Annotations

The Tools Palette for UML diagram elements display note and note link icons for all UML diagrams. These elements are used to place annotation nodes and their links on the diagram.

Notes can be free floating or a note link can be drawn to some other element to show that a note pertains specifically to it. A note link can be attached to another link.

Interoperability

Interoperability is supported in the following ways:

- For UML models created in other tools, it will use the various types of import and export functions, such as XMI.
- Transformations will enable users to exchange model information.

Modeling

The topics in this section provide an overview of modeling, information on UML diagrams, and supported technologies.

UML Modeling

Effective modeling with ER/Studio Software Architect will simplify the design stage of a project. The primary objective of modeling is to organize and visualize the structure and components of software-intensive systems. Models visually represent requirements, subsystems, logical and physical elements, and structural and behavioral patterns.

Supported UML Specifications

The Object Management Group's Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems. ER/Studio Software Architect supports UML to help you specify, visualize, and document models of your software systems, including structure and design.

"UML In Color" is an optional profile that supports the 'modeling in color' methodology. Color modeling makes it possible to analyze a problem domain and easily spot certain classes during analysis. ER/Studio Software Architect supports the use of four main groups of the color-modeling stereotypes:

- Role
- Moment-interval, Mi-detail
- Party, Place, Thing
- Description

When applying a stereotype to one of the diagram elements listed above, the view of the associated diagram element changes on the diagram. The stereotype field displays directly above the name field for the element, and the color of the element changes depend on the stereotype chosen. For each of these stereotypes you can choose a specific color to make the model more understandable at a glance.

UML 2.0 Diagrams

ER/Studio Software Architect provides support for the most frequently needed diagrams and notations defined by the UML 2.0 specification:

- Activity Diagrams
- Class Diagrams

- Use Case Diagrams
- Component Diagrams
- Composite Structure Diagrams
- Deployment Diagrams
- State Machine Diagrams
- Interaction (Sequence and Communication) Diagrams

UML Profiles

ER/Studio Software Architect includes several pre-installed profiles and allow you to create your own profile definitions using a Profile Definition project.

UML Profiles Basics

UML is a standard modeling language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. While the general modeling concepts of UML are quite suitable for the majority of developers, in some situations, a further extension of these concepts is useful to allow a more refined rendering of domain specific concepts and techniques. UML extension mechanisms address the definition of additional semantics of model elements that cannot be expressed directly using UML constructs. This technique is known as UML Profiling.

Profiles provide mechanisms that allow metaclasses from existing metamodels to be extended to adapt them for different purposes. All kinds of model elements can get stereotypes and tagged values defined in profile applied to the model.

The UML standard provides refinement mechanisms for profile creation, such as stereotypes, tagged values, constraints, and notation icons that collectively specialize and tailor the UML for a specific domain or process. These elements can be used to adapt the UML semantics without changing the UML metamodel. This means that you can interpret the semantics of a profile in the context of the UML specification.

Profile Definition Projects

A Profile Definition project is a profiled modeling project that allows creation of new profile definitions. One Profile Definition project would correspond to a single profile. Inside the Profile Definition project you can use some packages to locate different elements. For example, you can put all enumerations to one package, all palette contributions to another package, and all stereotypes to the third one. All the elements would be deployed to the same profile.

Profile Definition adds the following elements to the class diagram Tools Palette:

- Stereotype
- Palette Contribution
- Extension
- Contribution

Supported Metamodels

In ER/Studio Software Architect, you can create profiles on the basis of the UML 2.0 metamodel only. Stereotypes, tagged values, and OCL constraints declared in the profile, can only refer to the UML 2.0 metamodel.

Stereotypes

Stereotypes contain properties that extend a linked metaclass, and enable the use of platform- or domain-specific terminology or notation in addition to the ones used for the extended metaclass. When a model element has a stereotype, it indicates that this is a special kind of element that conforms to a rather rigid specification, defined for this stereotype in the profile definition.

Palette Contributions

Palette Contributions enable users to add creation tools for Stereotypes and pure metaclasses to the selected Diagram Tools Palette.

A contributed Stereotype will be associated with a Palette Contribution by means of a Contribution link. It will also be possible to associate a Palette Contribution with a shortcut to metaclass from the metamodel. This allows customizing the palette by adding some elements to the tool bars of different diagrams. For example, if you wanted to have the possibility to create classes from the component diagram toolbar, you could associate the shortcut to uml20::classes::Class with a Palette Contribution in the profile definition.

A Palette Contribution can extend another Palette Contribution; so doing, the child Palette Contribution will inherit all the parent diagrams, contributed stereotypes, and pure metaclasses.

Extension Links

Extension links indicate that the properties of a parent metaclass are extended with the new properties through a stereotype. An Extension link is drawn from a Stereotype to a shortcut to a metaclass whose properties are extended.

Contribution Links

A contribution link will connect a Palette Contribution element with a Stereotype.

Model Compare and Merge

ER/Studio Software Architect provides a comprehensive solution for comparing and merging models in a project.

EMF and UML Model Compare

ER/Studio Software Architect supports two-way and three-way comparison of EMF and UML models, or model elements of a similar type, in a tree view.

In a two-way compare, the compared models are called Left and Right. Model Compare/Merge traverses the compared models, going level by level down the containment tree. On each level, objects are matched using ID features which will be set in the ID Features page of the Preferences dialog (Window > Preferences > Modeling > EMF Model Compare > ID Features). After that, Model Compare/Merge compares values of attributes and non-containment references.

It should be possible to export compare results to an EMF XMI file.

Shared Models Compare

ER/Studio Software Architect provides integration with version control systems and allows two-way and three-way comparison and merge of shared (version controlled) models.

When comparing shared models, the Left model represents the local version while the Right model represents the remote version. In three-way compare, the third model is called the Ancestor. It represents a common ancestor version of the two versions taken from VCS.

ER/Studio Software Architect utilizes standard Eclipse synchronization APIs and is able to compare models stored in any version control system which supports the 'Synchronize' view from Eclipse.

Comparing and merging of shared models requires one (for two-way comparison) or two (for three-way comparison) remote versions of the compared model.

ER/Studio Software Architect copies the local model to a temporary project, applies changes reported by the repository provider, and then displays these changes in the 'Synchronize' view. Temporary models are read-only, and ER/Studio Software Architect uses a modal Model Compare dialog, instead of the standard Compare editor.

Merging Models

The merging capability enables you to transfer elements from one model to another.

Template Elements

ER/Studio Software Architect supports templates, as defined in the UML 2.0 superstructure specification. This support provides the ability to show templates, template signatures, parameters, and template bindings in a UML 2.0 diagram.

An element that can be templated may contain a template signature which specifies the formal template parameters. An element that can be templated contains a template signature is a template. A template signature displays in diagram as a rectangle in the top-right corner of the owning element. In the Properties View of a template element, the property 'isTemplate' is set to true.

A template binding represents a relationship between a templateable element and a template. A template binding specifies the substitutions of actual parameters for the formal parameters of the template.

Model Import and Export

ER/Studio Software Architect allows sharing of model information with other systems by importing and exporting model information, or by sharing project files as outlined in the table below.

Feature Description	Description
Exporting diagrams to images	It will be possible to save diagrams in several formats, including Bitmap images (BMP), Enhanced windows metafiles (EMF), Graphics interchange (GIF), JPEG file interchange (JPG), and Scalable Vector Graphics (SVG).
Importing/Exporting from XMI	ER/Studio Software Architect will support the following UML versions/platforms for XMI import/export: XMI for UML 2.0.
Export a Quality Assurance metric chart to image	Create a chart and then export it to image.

OCL Support

The Object Constraint Language (OCL) is a formal language that describes expressions on UML models. OCL expressions specify operations or actions that, when executed, alter the state of the system. UML modelers can use OCL to specify application-specific constraints in their models. UML modelers also can use OCL to specify queries on the UML model, which are completely programming language independent.

ER/Studio Software Architect allows you to use the following capabilities of OCL 2.0 to work with your model:

- Add the OCL constraints to the types defined in a model, providing them as constraint notes linked to the context elements on the diagram. The constraint text opens in a powerful editor that provides syntax highlighting, errors validation, and code completion functionality.
- Use OCL as a query language operating with types defined in the metamodel. You can perform a Search In Model by OCL query, write and run Model Audits and Metrics, and use OCL expressions in the documentation templates for the documentation generator.

OCL support is provided for the diagrams listed in the following table.

Diagram Type	Support Provided
All diagram types	Object constraints. The default language of constraints depends on context element type and project type.
Interaction (Sequence and Communication)	State invariant constraints for lifelines and constraints for the operands of the combined fragments as OCL expressions. Pre- and post-condition for the Interactions. These elements are realized as inner constraint elements available via the element's Properties.
State Machine	Guard conditions of transitions as OCL expressions. Pre- and post conditions of a State Machine, and a StateInvariant for a State. These elements are realized as inner constraints available via the element's Properties.
Activity	Pre- and post-conditions for activity; local pre- and post-conditions for an action.

OCL Constraints and Expressions

OCL is a formal language used to describe expressions on UML models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model.

The icons on the diagram Palette allows creation of OCL constraints as design elements on diagrams, and linkage of these constraints with the desired context. The OCL Expressions view provides an OCL editor where one can develop and validate OCL expressions. Any OCL constraint contains an OCL expression.

OCL support for constraints provides syntax and error highlighting in the OCL Editor view. The text of the constraint are validated when the constraint is linked to its context.

OCL on Non-Class Diagrams

Constraints on non-class diagrams fall into two categories:

- Inner constraints
- External constraints

The OCL editor provides syntax highlighting, error validation, and code completion functionality.

The OCL editor is available for the constraints inside the elements. You can expand the element node in the Model Navigator and open a constraint in the editor.

Constraint	Context
Element with defined constraint	Correct context for the constraint.
StateMachine	Operation selected as a Specification association of this StateMachine (a class will be selected for the context property and specification will be a method of the selected class).
Activity	Specification of the activity.
Action	Specification of the activity that contains this action.
State and Transition	Class selected in the context property of the StateMachine that contains this State or Transition.
Operation	Operation itself. Note that the operation will have a valid OCL context only if it is owned by Class or Interface. Operations owned by other classifiers will get no OCL context, and their constraints should have text as a constraint language.
Interaction	Context of this Interaction or Specification if it is defined.
State invariant and Interaction constraint	Class selected as type of the Lifeline that contains this element. It will be either a class directly selected as the type of the Lifeline or part selected as the Lifeline representation.
Extends	Context cannot be specified and constraint can only be defined as text (language=text).

In addition to the inner constraints described above, all the elements on the non-class diagrams that are Types (i.e. various classifiers) that can be furnished with external OCL constraints.

An external constraint will be created as a Constraint element linked to the constrained element by the context link. Unlike inner constraints, the external constraints always use the linked type as a context. Thus, the StateMachine constraints could have the context of the assigned specification if it is an inner precondition, or the StateMachine itself if it is the external linked constraint.

Patterns

Patterns provide software developers with powerful reuse facilities. Rather than trying to tackle each design problem from the very outset, you can use predefined patterns supplied with ER/Studio Software Architect. The hierarchy of patterns are defined in a Pattern Registry. You can manage and logically arrange their patterns using a Pattern Organizer.

Patterns are pluggable extensions for ER/Studio Software Architect enabling you to:

- Create new and frequently used elements
- Modify existing elements

Each pattern describes a set of model elements, relations between them, and constraints applied to those elements. Patterns are represented by special modeling projects covering all the aspects of patterns. Patterns are independent of any programming or markup language. You can use them to create or modify any type of element. Concrete patterns are designed to work with elements of a specific type. The Pattern Registry is used to manage patterns.

Pattern instances appear as a result of recognition of the existing model or creating new instances (along with model elements playing pattern roles) in the model. Pattern instances contain information about the pattern name and the role of each participant. They will be shown in the Pattern Explorer view and under the Patterns node in the Model Navigator.

When applied to a diagram, such patterns will create their entities and will be presented on the diagram itself, with the links to the created entities. Such patterns will enable further modification by means of adding new participants (new pattern parts). All patterns that appear in the Pattern Explorer will be represented in the project model in the form of entities with metaclass "pattern". Visually, pattern instances will be displayed as ovals (like collaboration occurrences). Pattern entities will have children links to pattern participants and this will allow map links on diagrams from pattern instances to pattern participants. Actions on pattern instances in the model will be the same as in the pattern explorer.

During the lifetime of a pattern instance the model could change (some elements from the instance may be deleted, some may be changed so that they don't satisfy the pattern definition anymore) and the pattern instance could become invalid. You would then need to perform pattern instance validation regularly.

Pattern Definition Projects

Using the pattern support subsystem in ER/Studio Software Architect, you can easily work with patterns via pattern definitions. You can use well known predefined patterns, define new ones, delete, rename, edit them, etc. You can also manage pattern instances: recognize patterns in an existing model, create elements by pattern, create new participants for particular roles for existing pattern instance, etc.

When new pattern instances are created from existing project elements, the creation process uses participants from the current selection and enables you to modify the pattern properties. You can easily view and/or modify properties of pattern instances using the standard Properties view. Any change that is made to a pattern property is applied immediately to the pattern participants (via refactoring).

All pattern definitions are stored in com.embarcadero.em.patterns\patterns subfolder of the ER/Studio Software Architect plug-ins folder. Each pattern definition is an archive file, packed by zip utilities provided by JDK, containing compilation results suitable for recognition and completion engines and the whole definition project in order to allow definition editing. Folder and shortcuts structure will be stored in the pattern.registry file in the same location.

Pattern Recognition

Pattern recognition identifies pattern instances from existing elements in the project. The identification process determines pattern participants and parameters. You can start the pattern recognition process from the project's context menu to perform pattern recognition, validation, and problem reporting.

Quality Assurance

Quality Assurance in ER/Studio Software Architect provides teams and managers with measures of the quality of their projects. As with any Quality Control, the team should understand what is measured, and why. Although audits and metrics are similar in that they both analyze your project, they serve different purposes. Audits and metrics are run as separate processes. Because the results of these two processes are different in nature, ER/Studio Software Architect provides different features for interpreting and organizing the results.

Model Audits

ER/Studio Software Architect supports a wide range of model audits. The list of available model audits can be viewed in the Preferences dialog. You can define, save, and reuse sets of model audits. Model audits are OCL queries that produce a Boolean result and operate in the context of existing metamodels. You can also employ additional OCL operations provided for the metamodel, specified in the OCL operations and OCL library operations tabs in the Preferences dialog.

After model audits are run, the results are displayed in the Model Audits View. The view provides detailed descriptions for all found errors and allows you to navigate to the corresponding problem element from this view by double-clicking the error message.

Model Metrics

ER/Studio Software Architect supports a wide range of model metrics. The list of available model metrics can be viewed in the Modeling Quality Assurance Model node of the Preferences dialog. You can define, save, and reuse sets of model metrics. Model metrics will be OCL queries that produce an Integer result and operate in the context of existing metamodels. You can also employ additional OCL operations provided for the metamodel, specified in the OCL operations and OCL library operations tabs in the Modeling OCL page.

After model metrics are run, the results are displayed in the Model Metrics view. You can navigate to the corresponding elements listed in the Model Metrics view by double-clicking the element name.

Metrics Graphical Representation

Metrics results can also be viewed graphically. Two graphic views allow users to summarize metrics results: bar charts and Kiviat charts. Both charts are invoked from the context menu of the table. The Kiviat chart is used for rows and the bar chart is used for columns.

The bar chart displays the results of a selected metric for all packages, classes, and/or operations. The bar color will reflect conformance to the limiting values of the referenced metric:

The Kiviat chart demonstrates the analysis results of the currently selected class or package for all the metrics that have predefined limiting values. The metrics results are arranged along the axes that originate from the center of the graph. Each axis has a logarithmic scale with the logarithmic base being the axis metric upper limit so that all upper limit values are equidistant from the center. In this way, limits and values are displayed:

Exporting and Importing Audits and Metrics

You are able to import and export model (but not code) metrics and audits all at once, including sets of named OCL queries on metamodels, and other settings. Model audits and metrics are saved to files with extensions .ModelMetrics and .ModelAudits. When importing such a file, you will completely replace their currently defined model audits or metrics.

Refactoring

ER/Studio Software Architect leverages refactoring operations provided by the Eclipse platform. Refactoring is available for the model elements in ER/Studio Software Architect projects by means of context menus.

Version Control

ER/Studio Software Architect supports several version control systems that can be integrated in Eclipse. They would include but not be limited to CVS Version control in ER/Studio Software Architect enables several users to work with one modeling project.

ER/Studio Software Architect provides context menus to work with CVS. The version control system is set up so that only one user can work with a shared model at a time. In case several users edit the model at a time, users can use the model compare and merge functionality of ER/Studio Software Architect. They can compare the structure of their models and merge inconsistencies if necessary. Alternatively they can revert to the saved version of the model.

Task-aware features

Task-aware features of ER/Studio Software Architect make it easier for you to work with modeling resources under version control by automatically finding all the resources that need to be checked out for the modeling resource to be modified. To enable task-aware features, you will use the following Team options for locking files and managing modeling resources:

Path to Option	Options
Preferences > Team > Modeling Resources	Auto Checkout modeling resource on edit and Checkout before model modification

While editing some model elements, a lock dialog appears, which will list all resources that need to be locked. By choosing Yes, you will apply a locking mechanism to the selected resources and they will become writable. After modification, when checking in the files, they will become unlocked and read-only. Every time you try to edit a read-only file that is under version control, a dialog appears suggesting that the file be locked for editing. To automatically lock files on checkout, you can check auto lock check box in the dialog.

You can also use the 'Ignore default package diagrams' option to not store and synchronize default package diagrams. This option will add default.txvpck and default.txvClassDiagram2 patterns to the ignored resources.

Project Documentation

ER/Studio Software Architect enables you to create external documentation for the open projects, or from the command line. You can use the generated reports to illustrate their projects with the documentation in one of the available formats. Documentation generation is available for all types of ER/Studio Software Architect projects, including Pattern Definition and Profile Definition projects. The generated documentation can also include the results of Audits.

ER/Studio Software Architect allows you to generate documentation in one of the following output formats:

- RTF
- HTML
- TXT
- PDF
- XSL-FO

By default, documentation will be generated in HTML format.

All the documentation that ER/Studio Software Architect generates is written to a single directory that can be specified in the documentation generation dialogs. By default, this will be the output folder of the Eclipse workspace. The generated documentation opens in the appropriate viewer, associated with the output format.

Project reports will be created by applying documentation templates to ER/Studio Software Architect projects. The templates will contain commands to the documentation generator; the projects will provide the source of project-specific data. Documentation templates are *.tpl text files with formatting instructions and tags for the commands.

Additional Product Information

The Embarcadero Web site is an excellent source for additional product information, including white papers, articles, FAQs, discussion groups, and the Embarcadero Knowledge Base.

Go to www.embarcadero.com/support, or click any of the links below, to find:

- Documentation
- Online Demos
- Technical Papers
- Discussion Forums
- Knowledge Base

Getting Started with ER/Studio Software Architect

This section contains an introduction to modeling with ER/Studio Software Architect. The sample projects and Cheat Sheets are designed to help you explore the features while working with projects. Some of the special features include: patterns, generating project documentation, reverse engineering and so forth.

Overview	Provides a brief introduction to the feature set of ER/Studio Software Architect. Use this application to build a UML model of your application.
Documentation Set	Describes the documentation set for ER/Studio Software Architect.
Sample Projects and Cheat Sheets	Provides a list of sample projects and cheat sheets.
Help on Help	Explains how to use the ER/Studio Software Architect online Help and where to find additional resources.
Tour of ER/Studio Software Architect	Tour of ER/Studio Software Architect.

Overview

Welcome to ER/Studio Software Architect, the design-driven environment for modeling applications. ER/Studio Software Architect includes features such as support for UML 2.0, OCL, patterns, Quality Assurance audits and metrics, XMI format import and export, and automated documentation generation.

ER/Studio Software Achitect is shipped with two versions: one version using the regular installer and one using the Instant On build.

- The Installation Guide for a regular installation is available for download from Embarcadero's web site: http://www.embarcadero.com.
- The Instant On build (ION), is a Xenocode virtualized application. Click here for more information on using the Instant On Application Launcher.

The ER/Studio Software Architect features are tightly integrated with the Eclipse environment. When ER/Studio Software Architect support is activated, the following items are added or modified:

- Diagram Editor
- Model Navigator
- Properties View
- Palette

NOTE: If your Internet access is limited by network security, or if your computer is protected by a personal firewall, the Web-based links in this Help system might not function properly.

Related Concepts

Help on Help

Documentation Set

Documentation Set

The documentation set consists of the following items:

Item	Description	Location
Release Notes (ReadMe)	Late-breaking information including: Last minute notes System requirements Installing and starting ER/Studio Software Architect Known issues and limitations	readme.html
Setting Up Licensing for ER/Studio Software Architect.	ER/Studio Software Architect licensing setup.	ER/Studio Software Architect main menu: Help/ER/Studio Software Architect Licensing
Online help	General, context, and dynamic help for ER/Studio Software Architect including the following comprehensive information most relevant to the user: — Conceptual topics — Getting Started and Concepts — Procedural topics — Working with Projects, Creating and using profiles, Working with diagrams, Working with different types of modeling, Refactoring procedures, Using OCL, Working with patterns, Quality assurance, and Documentation generation procedures. — Reference topics — dialogs, wizards and GUI elements	ER/Studio Software Architect main menu: Help/Help Contents
Cheat Sheets	Interactive tutorial that help you start using basic product features. Each cheat sheet helps you complete a single task. A list of cheat sheets is available in the Sample projects and cheat sheets topic.	ER/Studio Software Architect main menu: Help/ Cheat Sheets

Related Concepts

Help on Help

Sample Projects and Cheat Sheets

ER/Studio Software Architect ships with sample projects and cheat sheets that help you get acquainted with the application and its features.

The sample projects are available under **File > New > Other > Examples.**

The following is a list of sample projects:

- Profile Definition
- UML 2.0 Notation

Cheat sheets provided with ER/Studio Software Architect are basically interactive tutorials that help you to start using some of the new features. Each cheat sheet helps you complete some task. The cheat sheets are available under: **Help > Cheat Sheets**.

The following is a list of available cheat sheets

Open this cheat sheet...	to learn how to perform the following
OCL Features	
Define and Run Model Audits	Create custom model audits using OCL
OCL Constraint Creation	Create OCL constraint in your project
Search in Model Using OCL Expressions	Perform model search using OCL.
Patterns	
Create a pattern from Existing Elements	
Create a Pattern from Scratch	
Export Pattern Definitions	
UML Profiles	
Apply UML Profile	Apply created UML profile for Web Services modeling
Define UML Profile	Create custom UML profile for Web Services modeling
Deploy UML Profile	Deploy created UML Profile

Help on Help

ER/Studio Software Architect allows you to view various help topics that will assist you while you are completing your tasks.

Online Help

The Online Help includes conceptual overviews, procedural how-to's, and reference information, which allow you to navigate from general to more specific information as needed.

NOTE: When you use a link to navigate from one topic to another topic, the context of the Help topic you are viewing might not be obvious. To find the context of a topic within the **Contents** pane, click **Show in Table of Contents** on the toolbar of the Eclipse Help viewer.

Concepts

Concepts introduce the main features and methods that will help you learn and understand ER/Studio Software Architect techniques. At the end of most conceptual topics, you will find links to related, more detailed information.

How-To Procedures

The how-to procedures provide step-by-step instructions.

All procedures are listed under **Procedures** in the **Contents** pane of the Help window.

Reference Topics

The reference topics provide detailed information on subjects such as configuration options, GUI elements, dialoges, and wizards references.

All of the reference topics are listed under the **Reference** section in the **Contents** pane of the Help window.

Typographic Conventions Used in the Help

The following typographic conventions are used throughout ER/Studio Software Architect online Help.

Typographic conventions

Convention	Used to indicate
Monospace type	Source code, file and folder names, and text that you must type.
Boldface	GUI elements and dialogs.
Italics	Book titles and to emphasize new terms.
KEYCAPS	Keyboard keys, for example, the CTRL or ENTER key.

Related Concepts

Documentation Set

Tour of ER/Studio Software Architect

Modeling User Interface

The Modeling perspective is the default perspective. The Modeling perspective provides the following views:

View	Description
Add linked results	Shows results of applying the Add Linked command.
Model Audits	Displays the results of the model audits that are run.
Model Bookmarks	Lists bookmarked model elements.
Model Metrics	Displays the results of the model metrics that are run.
Model Navigator	Provides the logical representation of the model of a project: namespaces (packages) and diagram nodes.
Diagram Editor	Displays created and opened diagrams. When multiple diagrams are used, the diagram editor provides a tab for each diagram.
Properties	Displays the properties for a selected element. The properties for each element are usually divided into different categories.

View	Description
Profile Constraints	Lists available profile constraints.
Profile Validation	Displays results of the profile validation process.
OCL Expression	Enables quick evaluation of OCL expressions in the explicitly specified context (an EM/Studio or EMF model element), or in the context of the current selection.
Last Validation Results	Displays results of the latest validation of a pattern definition.
Pattern Explorer	Enables logical organization of patterns (using virtual trees, folders and shortcuts), and management of recognized instances of patterns.
Pattern Registry	Defines the virtual hierarchy of patterns.

Model Menu

View	Description
Generate Class Diagram	
Run Model Metrics	Displays the results of the model metrics you run.
Run Model Audits	Displays the results of the model audits you run
Compare with	Each Other (as model elements): Compares two or three selected model elements against each other and shows differences in a separate view Local Version: Compares a shared resource with a version stored on your disk.
Profile	Uninstall Profiles: Uninstalls the selected profile Open Profile Definition: Opens the profile definition project. Deploy Profile: Starts the creating profile plug-in process. Run Profile Constraints: Runs profile-specific audits. Convert Properties: Converts profile-specific properties of the projects created in the previous version of ER/Studio Software Architect for Eclipse to the new format. For more information see Converting Profile-Specific Properties topic in the Procedures section. Preferences: Opens the Profile preferences in the Modeling node.
Documentation	Generate HTML: Opens the Generate HTML Documentation dialog. Generate Documentation Using Template: Opens the Generate Documentation Using Template dialog.

Diagram Menu

The Diagram menu includes commands relevant to working with the diagram currently opened in the **Diagram** editor. The commands include, but are not limited to, layout and align, different levels of zoom, switching grid and rulers, hiding and showing elements, and so forth. For more information, refer to: Diagram View

Related Procedures

Choosing a Perspective

Concepts

This section provides an overview of the features provided by ER/Studio Software Architect.

Basics	Basic information about features.
Interoperability	This section describes interoperability with the other editions and versions from the legacy versions.
Modeling Overview	Describes UML modeling in general.
Model Import and Export Overview	Describes the features for importing and exporting entire models or parts of the models.
OCL Support	Overview of OCL support.
Patterns	Overview of patterns.
Quality Assurance	Describes quality assurance facilities.
Version Control	This topic provides an overview of version control features.
Project Documentation	This part describes the documentation generation facility and documentation template basics.

Basics

This section provides information about the basic features.

Project Overview	Describes the projects.
Package Overview	Describes namespaces and packages.
Diagram Overview	Describes the UML diagram.
Diagram Format	This section describes the XML-based diagram format that is common for all modeling tools of Embarcadero's applications.
Containment Metamodel	Brief description of containment metamodel.
Model Element Overview	Describes the model elements.
Model Shortcut Overview	Describes the shortcuts on UML diagrams.
Model Hyperlinking Overview	Describes the feature of model element hyperlinking.
Model Annotation Overview	Describes the feature for annotating UML diagrams.

Project Overview

Work in ER/Studio Software Architect is done in the context of a **project**. A project is a logical structure that holds all resources required for your work. All projects located in the selected Workspace are listed in the Model Navigator.

You can set up project properties when the project is being created, and modify them further, using the **Properties** dialog.

The following is a list of projects that can be created.

- **Projects from CVS** version control system
- **EMF** (Eclipse Modeling Framework) Project
- **Pattern Definition project** is a profiled UML 2.0 modeling project that allows you to create new patterns.
- **Profile Definition project** is a profiled modeling project that allows you to create new profiles.
- **UML 2.0 project** is a design project with no source code support.

NOTE: The project settings are initially specified on project creation. Further, you can update properties for the existing project.

Related Procedures

Projects

Package Overview

The notion of a package has two facets: logical and physical.

- Logically, a model consists of one or more packages. A package is a model element used to group elements, and provides a namespace for the grouped elements. A package can contain packageable elements (the elements that can be directly owned by a package) and the other packages. A model itself is a package.
- Physically, a package is a folder containing the files that store diagrams and model elements.

Contents of a package can be displayed on a special type of the Class Diagram that is synchronized with the package contents, i.e. all the classifiers directly owned by this package automatically appear on the package diagram. Each package contains the single package diagram that is created automatically and cannot be added explicitly.

The root package of a project (Model) is usually referenced as the **default** package. The package diagram of this package is called the default diagram. This diagram is created and opened just after the modeling project creation.

Related Concepts

Containment Metamodel

Related Procedures

Working with a Package

Diagram Overview

Each modeling project contains a set of diagrams that are graphical representations of parts of the model. Diagrams contain graphical elements (nodes connected by paths) that represent model elements.

Each diagram belongs to a certain diagram type (for example, UML 2.0 Class Diagram). The diagram type defines the typical contents of the diagram (the kind of elements that are usually placed on this diagram) and the notation used to represent the model elements. For example, a Class in a UML 2.0 project can be added to the Class Diagram and to the Composite Structure Diagram and has different representations in each. Each diagram has the specific Palette and context menu that allow to create the model elements specific to this diagram type. These tools can be customized.

Diagrams exist within the context of a project. You have to create or open a project before creating a new diagram.

The set of available diagram types depends on the type of project. In UML 2.0 project you have a set of standard UML diagrams defined in UML2.0 specification. Along with the design diagrams that are explicitly created by the user, ER/Studio Software Architect models have the Package diagrams. These diagrams have the ClassDiagram type, but they are generated automatically for each package and show its contents.

Related Concepts

Diagram Format

Related Procedures

Creating a Diagram

Diagram Format

The diagrams created with ER/Studio Software Architect for Eclipse are stored in XML-based files with the extension *.txv<diagram_type>. For example, the file <name>.txvClassDiagram20 corresponds to a class diagram. Design elements are stored either in the package files (default.txaPackage) or in separate XML-based format files with the extension .txa<element_type>, depending on your choice when creating a project in the **New Project** wizard.

Related Concepts

Interoperability

Containment Metamodel

ER/Studio Software Architect for Eclipse handles the logical and physical containment of design elements as follows:

- Design elements are created as children of packages.
- All elements shown on diagrams are shortcuts (or references) to actual model elements, therefore when you create a new element on a diagram, this element is created in the package and adds its shortcut to the diagram.
- Clipboard actions operate with references, if the source and target containers of the action are diagrams.
- You can optionally create design elements in separate files (standalone design elements) or in one file (filemates).

Related Concepts

Model Shortcut Overview

Model Element Overview

Each model in a modeling project is a set of entities that are instances of metaclasses of the metamodel chosen for the project. These instances are the Model Elements.

Each model element has a set of properties and notation defined for its metaclass. For example, when you create a UML 2.0 project, every element created in this project instantiates a metaclass from the UML 2.0 metamodel that is, each actor on a use case diagram in a UML 2.0 project is an instance of usecases/Actor, each component is an instance of components/Component.

The model elements that have the graphical notation and can be explicitly placed on diagrams, are nodes and links.

Related Concepts

Diagram Overview

Model Shortcut Overview

Containment Metamodel

Model Shortcut Overview

A **shortcut** is a representation of a model element placed on a diagram. One can create multiple shortcuts to the same element on different model diagrams. The modifications of element itself can be made from any diagram containing its shortcut and are propagated to all its shortcuts. The modifications of shortcut view properties made from any diagram don't affect the representations of this element on other diagrams. Shortcut can be removed from diagram without removing the element from the model.

You can create shortcuts to the elements within the same project. To create a shortcut to an element from another workspace project, add this project to the Model Path of the current project.

The small special symbol appears over a node to indicate a shortcut. For the package diagrams it appears only if this node belongs to a different namespace or package.

Select a shortcut on your diagram and choose **Select in Model Navigator** on the context menu to navigate to the source element in the Model Navigator.

Related Procedures

Creating a Shortcut

Establishing Cross-Project References

Model Hyperlinking Overview

You can create hyperlinks from diagrams or model elements to other system artifacts and browse directly to them.

Why Use Hyperlinking?

Use hyperlinks for the following purposes:

- Link diagrams that are generalities or overviews to specifics and details.
- Link diagrams or elements to external documentation.

Create a hyperlink from an existing diagram or one of its elements to any other diagram or model element, or create a new diagram that will be hyperlinked to the current element.

You can also create hyperlinks from your diagrams to external documents such as files or URLs.

Hyperlink Types

You can create hyperlinks to:

- An existing diagram or diagram element from any project in the workspace.
- A resource in the workspace.
- An external document (file or URL)

Browse-through Sequence

Use case diagrams typically represent the context of a system and system requirements. Usually, you begin at a high level and specify the main use cases of the system. Next, you determine the main system use cases at a more granular level. As an example, a "Conduct Business" use case can have another level of detail that includes use cases such as "Enter Customers" and "Enter Sales". Once you have achieved the desired level of granularity, it is useful to have a convenient method of expanding or contracting the use cases to grasp the scope and relationships of the system's use case views.

The hyperlinking feature of ER/Studio Software Architect allows you to create browse-through sequences comprised of any number of use case or any other diagrams. By browsing the hyperlink sequence, you can follow the relationships between the use case diagrams.

ER/Studio Software Architect does not confine hyperlinking to such sequences, however. You can use hyperlinking to link diagrams and elements based on your requirements. For example, you can create a hierarchical browse-through sequence of use case diagrams, creating hyperlinks within the diagrams that follow a specific actor through all use cases that reference the actor.

Related Procedures

Hyperlinking Diagrams

Creating a Browse-Through Sequence of Diagram

Model Annotation Overview

The tools Palette for UML diagram elements displays note and note link icons for all UML diagrams. Use these elements to place annotation nodes and their links on the diagram.

Notes can be free floating or you can draw a note link to some other element to show that a note pertains specifically to it. You can attach a note link to another link.

Related Procedures

Annotating a Diagram

Interoperability

ER/Studio Software Architect supports the possibility to exchange models created in the different products of Embarcadero’s product line and in the other modeling tools.

Interoperability is supported in the following ways:

- ER/Studio Software Architect opens projects created with the other tools of Embarcadero’s product line. In so doing, the project roots and diagram formats are considered and processed.
- For the models created in the other tools, use the various types of import and export, such as XMI.
- Also, transformations enable the users to exchange model information. Refer to the concept section “Model Transformation Support” for details.

Reusing Legacy Projects

Reusing the legacy *.tpr, *.tpx and *.jpx projects is an important interoperability goal. However, this task faces a number of problems related to the differences between the products, which are summarized in the following table:

Legacy Projects	New Projects
Support multiple modeling roots.	All modeling information is stored in a single folder.
It is possible to specify package prefix for a root.	The notion of package prefix does not exist.
Support two diagram formats (DF format and TXV format)	Supports TXV format only.
Old containment metamodel stores diagrams and model elements together.	New containment metamodel separates the diagram information from the model elements.

These problems are resolved by means of a new migration tool implemented as **Import Project Wizard**, which takes a legacy project as input and produces one or more ER/Studio Software Architect Eclipse projects.

The resulting projects meet the following common requirements:

- Folder structure of the resulting project is created considering the package prefixes if any.
- All diagrams are converted from the old containment metamodel to the new containment metamodel. If a model root contains diagrams in DF format, these diagrams are converted to TXV format. If a model root contains diagrams in both DF and TXV formats, then only TXV diagrams are considered.
- Optionally, you can create the resulting project with the design elements stored in different files. In this case, the model elements of the source project are converted to standalone design elements.

Reusing Artifacts

Due to different platforms, ER/Studio Software Architect does not support complete migration of the legacy custom artifacts.

Instead ER/Studio Software Architect provides the possibility to create your own artifacts and extensions using its functionality.

Modules	You can create modules using Eclipse API and EMF API. Use Eclipse API for IDE-related parts, and EMF API for working with models.
Custom diagrams and custom properties	Use profiles to customize diagrams and define custom properties. Refer to Profile Definition Project Overview.

Related Concepts

UML Profiles

Diagram Format

Profile Definition Project

Patterns Overview

Modeling Overview

The topics in this section provide an overview of modeling, information on UML diagrams, and supported technologies.

UML Modeling Overview	Describes what modeling with ER/Studio Software Architect means in general.
UML Profiles	Describes UML profiles.
Model Compare and Merge	Describes model compare and merge functionality.

Related Concepts

UML 2.0 Diagrams

UML Modeling Overview

Effective modeling simplifies the development stage of your **project**.

The primary objective of modeling is to organize and visualize the structure and components of software intensive systems. **Models** visually represent requirements, subsystems, logical and physical elements, and structural and behavioral patterns.

Supported UML Specifications	Describes supported UML specifications.
UML 2.0 Diagrams	Gives a general notion of UML 2.0 diagrams.

Supported UML Specifications

The Object Management Group's Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.

ER/Studio Software Architect supports UML to help you specify, visualize, and document models of your software systems, including their structure and design.

Refer to UML documentation for the detailed information about UML semantics and notation. The *UML (version):Superstructure* document defines the user level constructs required for UML. It is complemented by the *UML (version): Infrastructure* document which defines the foundational language constructs required for UML. The two complementary specifications constitute a complete specification for the UML modeling language.

UML 2.0

The set of available diagrams depends on your project type.

Design projects support UML 2.0 specifications.

The version of UML is selected when a project is created. It cannot be changed later.

UML In Color

"**UML In Color**" is an optional profile to support the **modeling in color** methodology. Color modeling makes it possible to analyze a problem domain and easily spot certain classes during analysis. The use of the four main groups of the color-modeling stereotypes is supported:

- Role
- Moment-interval, Mi-detail
- Party, Place, Thing
- Description

When applying a stereotype to one of the diagram elements listed above, the view of the associated diagram element changes on the diagram. The stereotype field displays directly above the name field for the element, and the color of the element depends on the stereotype chosen. For each of these stereotypes you can choose a specific color to make your model more understandable at a glance. Note that the other stereotypes do not have associated colors.

Related Concepts

UML Modeling Overview

UML 2.0 Diagrams

Support is provided for the most frequently needed diagrams and notations defined by the UML 2.0.

UML 2.0 Activity Diagram Definition	Provides UML 2.0 activity diagram definition.
UML 2.0 Class Diagram Definition	Provides UML 2.0 class diagram definition and example, and notes about using class diagrams.
UML 2.0 Use Case Diagram Definition	Provides UML 2.0 use case diagram definition.
UML 2.0 Component Diagram Definition	Provides UML 2.0 component diagram definition.
UML 2.0 Composite Structure Diagram Definition	Provides UML 2.0 composite structure diagram definition.
UML 2.0 Deployment Diagram Definition	Provides UML 2.0 deployment diagram definition.
UML 2.0 State Machine Diagram Definition	Provides UML 2.0 state machine diagram definition and example.
Interaction (Sequence and Communication) Diagrams	Describes UML 2.0 Interaction diagrams.

UML 2.0 Activity Diagram Definition

The activity diagram enables you to model the system behavior, including the sequence and conditions of execution of the actions. Actions are the basic units of the system behavior.

An Activity diagram enables you to group and ungroup actions. If an action can be broken into a sequence of other actions, you can create an activity to represent them.

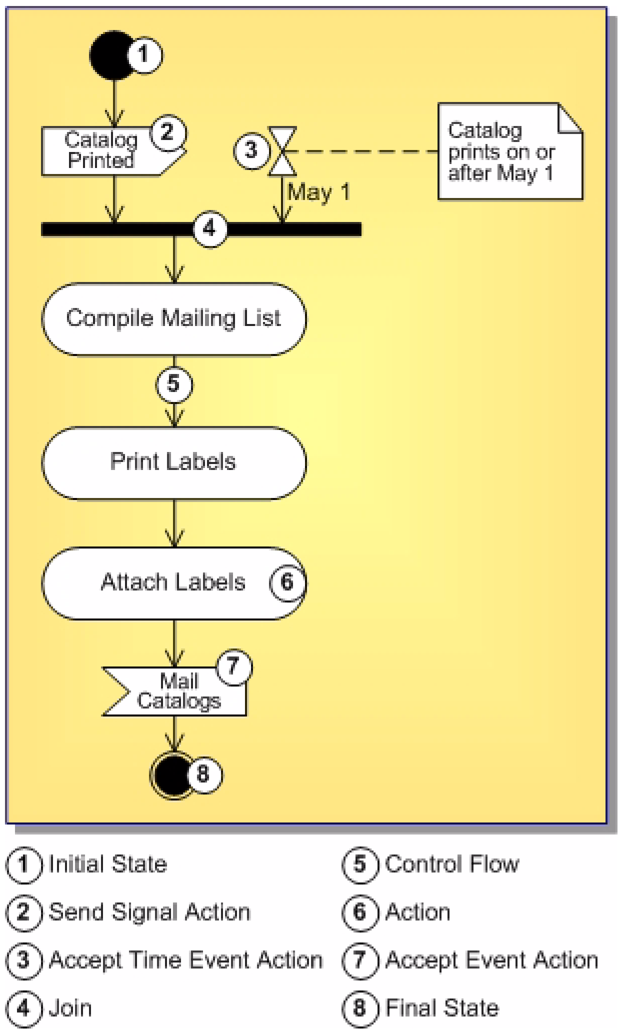
In UML 2.0, activities consist of actions. An action represents a single step within an activity, that is, one that is not further decomposed within the activity. An activity represents a behavior which is composed of individual elements that are actions. An action is an executable activity node that is the fundamental unit of executable functionality in an activity, as opposed to control and data flow among actions. The execution of an action represents some transformation or processing in the modeled system, be it a computer system or otherwise.

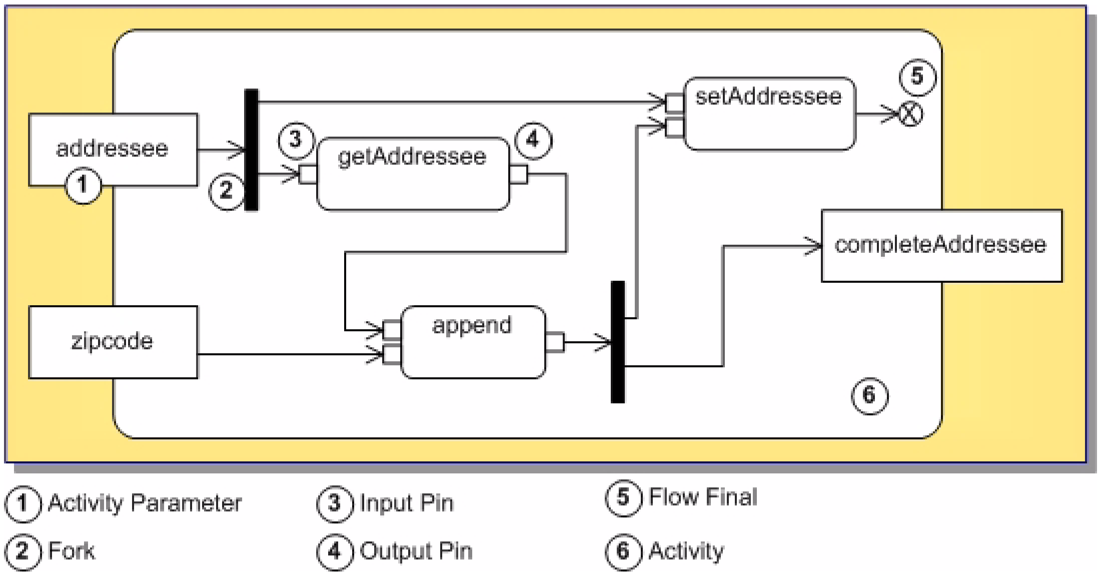
The semantics of activities is based on token flow. By flow, we mean that the execution of one node affects and is affected by the execution of other nodes, and such dependencies are represented by edges in the activity diagram. Data and control flows are different in UML 2.0.

A control flow may have multiple sources (it joins several concurrent actions) or it may have multiple targets (it forks into several concurrent actions).

Each flow within an activity can have its own termination, which is denoted by a flow final node. The flow final node means that a certain flow within an activity is complete. Note that the flow final may not have any outgoing links.

Using decisions and merges, you can manage multiple outgoing and incoming control flows.





Related Procedures

UML 2.0 Activity Diagrams Procedures

UML 2.0 Class Diagram Definition

UML 2.0 Class diagrams feature the same capabilities as the UML 1.4 diagrams. The UML 2.0 class diagrams offer new diagram elements such as ports, provided and required interfaces.

According to the UML 2.0 specification, an instance specification can instantiate one or more classifiers. You can use classes, interfaces, or components as a classifier.

Interfaces

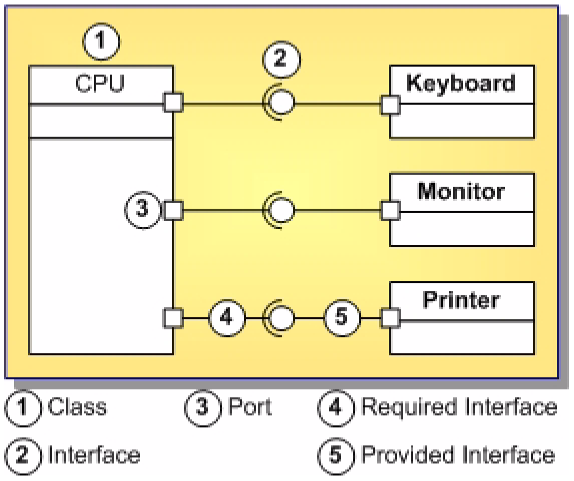
A class implements an interface via the same generalization/implementation link. In addition to the implementation interfaces, there are provided and required interfaces. Interfaces can be represented in class diagrams as rectangles or as circles. For the sake of clarity of your diagrams, you can show or conceal interfaces.

UML 2.0 class diagram supports the ball-and socket notation for the provided and required interfaces.

NOTE: Applying a provided interface link between a class and an interface creates a regular generalization/ implementation link. To create provided interface, apply the provided interface link to a port on the client class.

Sample Diagram

The figure below shows a class diagram with some of the new elements.



Related Procedures

UML 2.0 Class Diagrams Procedures

UML 2.0 Use Case Diagram Definition

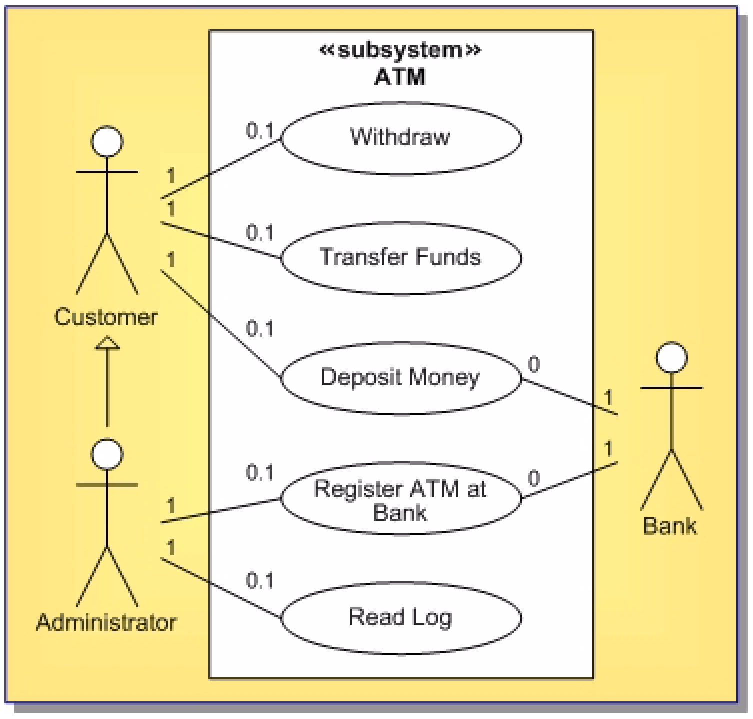
Diagram courtesy of the Unified Modeling Language: *Superstructure version 2.0. August 2003. p. 536*.

Definition

Use case diagram describes required usages of a system, or what a system is supposed to do. The key concepts that take part in a use case diagram are actors, use cases, and subjects. A subject represents a system under consideration with which the actors and other subjects interact. The required behavior of the subject is described by the use cases.

Sample Diagram

The following diagram shows an example of actors and use cases for an ATM system.



Related Procedures

UML 2.0 Use Case Diagrams Procedures

UML 2.0 Component Diagram Definition

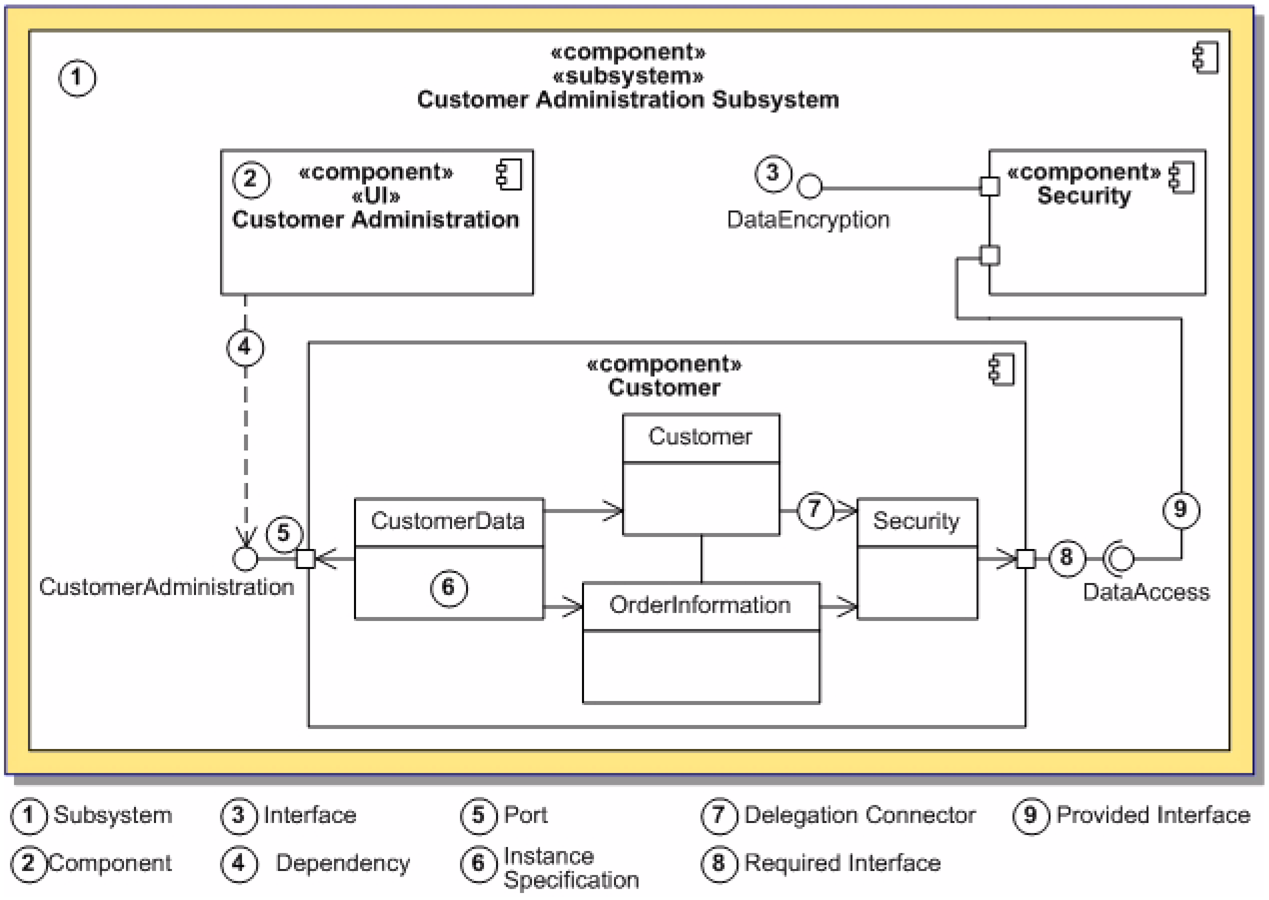
This topic describes the UML 2.0 Component Diagram.

Definition

According to the UML 2.0 specification, a component diagram can contain instance specifications. An instance specification can be defined by one or more classifiers. You can use classes, interfaces, or components as a classifiers. You can instantiate a classifier using the **Properties Window**, or the in-place editor.

Sample Diagram

The following component diagram specifies a set of constructs that can be used to define software systems of arbitrary size and complexity.



Related Procedures

UML 2.0 Component Diagrams Procedures

UML 2.0 Composite Structure Diagram Definition

Diagram courtesy of the Unified Modeling Language: *Superstructure version 2.0. August 2003. p. 178*.

Definition

Composite structure diagrams depict the internal structure of a classifier, including its interaction points to the other parts of the system. It shows the configuration of parts that jointly perform the behavior of the containing classifier.

A collaboration describes a structure of collaborating parts (roles). A collaboration is attached to an operation or a classifier through a Collaboration Use.

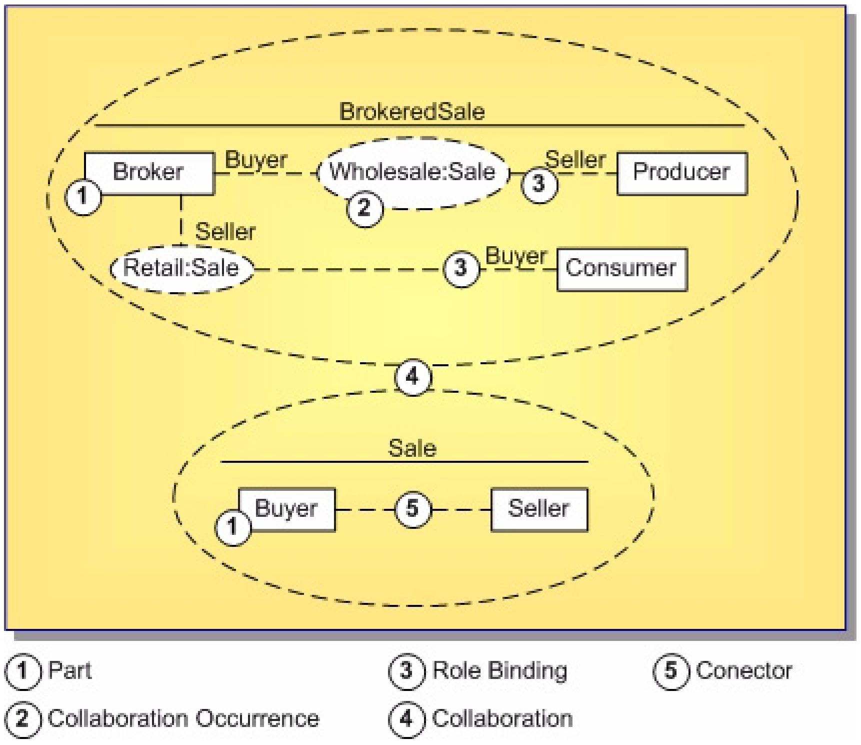
Classes and collaborations in the Composite Structure diagram can have internal structure and ports. Internal structure is represented by a set of interconnected parts (roles) within the containing class or collaboration. Participants of a collaboration or a class are linked by the connectors.

A port can appear either on a contained part, or on the boundary of the class.

The contained parts can be included by reference. Referenced parts are represented by the dotted rectangles.

Composite Structure diagram supports the ball-and-socket notation for the provided and required interfaces. Interfaces can be shown or hidden in the diagram as needed.

Sample Diagram



Related Procedures

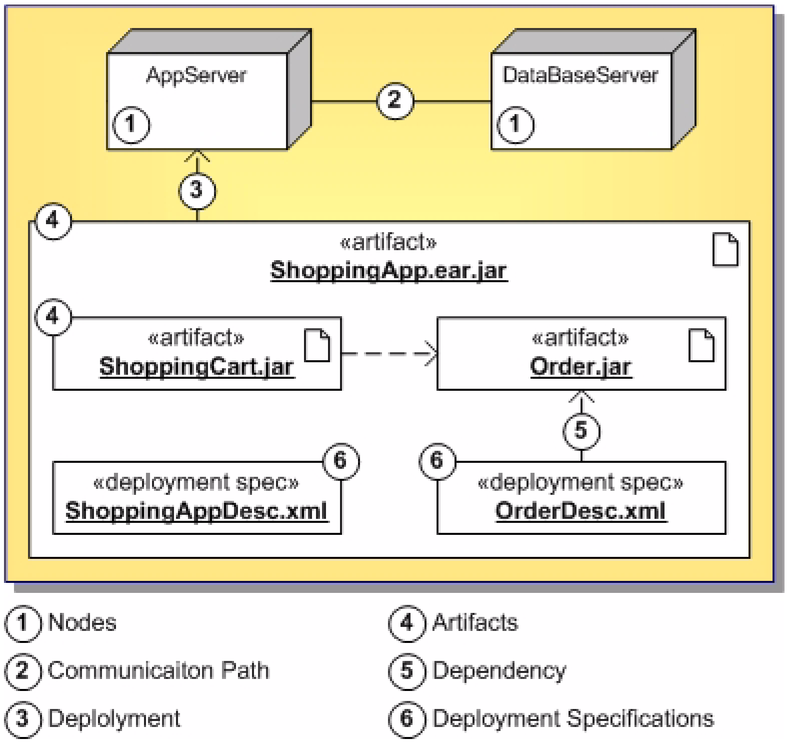
UML 2.0 Composite Structure Diagrams Procedures

UML 2.0 Deployment Diagram Definition

The deployment diagram specifies a set of constructs that can be used to define the execution architecture of systems that represent the assignment of software artifacts to nodes. Nodes are connected through communication paths to create network systems of arbitrary complexity. Nodes are typically defined in a nested manner, and represent either hardware devices or software execution environments. Artifacts represent concrete elements in the physical world that are the result of a development process.

Diagram courtesy of the Unified Modeling Language: *Superstructure version 2.0. August 2003. pp. 207, 212*.

Sample Diagram



Related Procedures

UML 2.0 Deployment Diagrams Procedures

UML 2.0 State Machine Diagram Definition

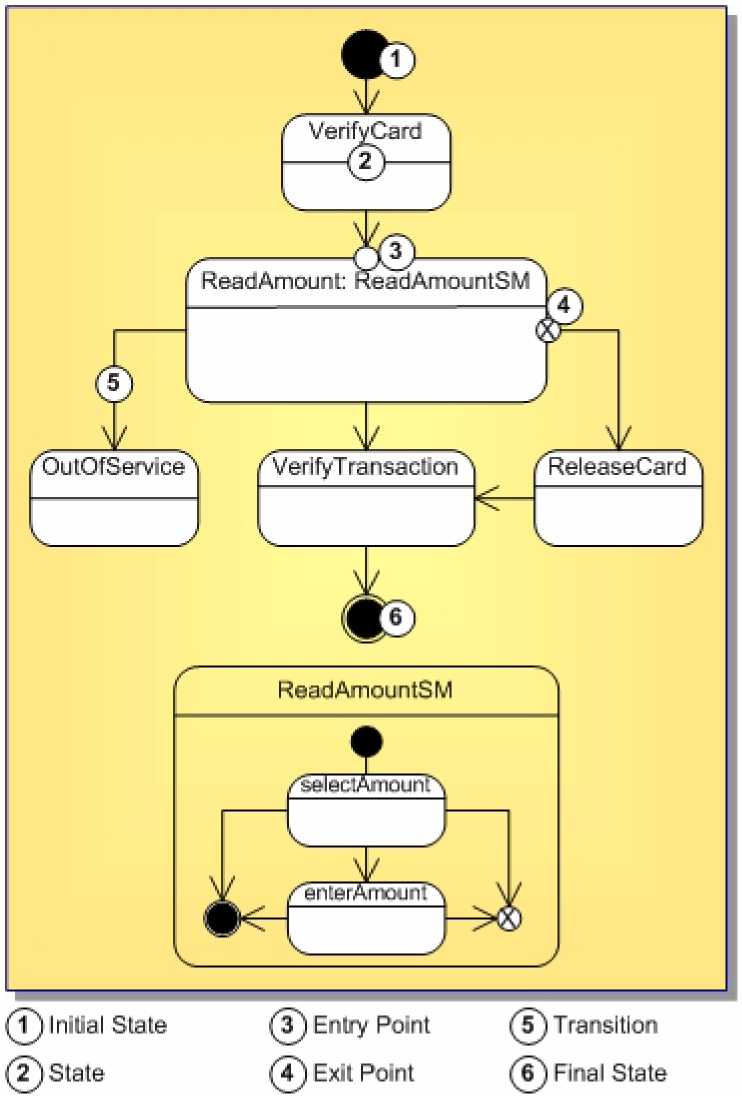
States are the basic units of the state machines. In UML 2.0 states can have substates.

Execution of the diagram begins with the Initial node and finishes with Final or Terminate node or nodes. Please refer to UML 2.0 Specification for more information about these elements.

Definition

State Machine diagrams describe the logic behavior of the system, a part of the system, or the usage protocol of it.On these diagrams you show the possible states of the objects and the transitions that cause a change in state.

Sample Diagram



Related Procedures

UML 2.0 State Machine Diagrams Procedures

Interaction (Sequence and Communication) Diagrams

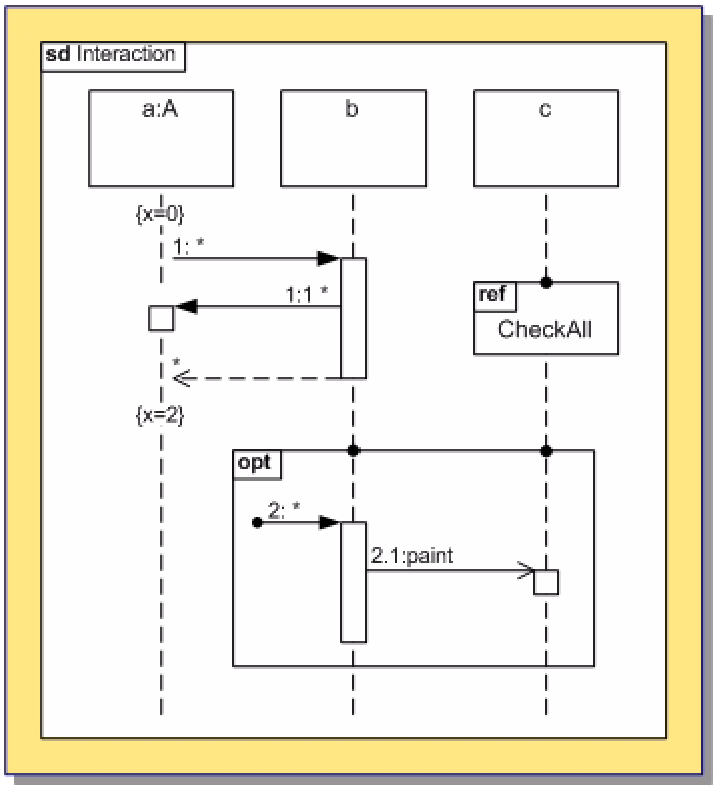
Using ER/Studio Software Architect you can create interactions for the detailed description and analysis of inter-process communications. Interactions can be visually represented in your projects by means of the two most common interaction diagrams: Sequence and Communication. On the other hand, interactions can exist in projects without visual representation.

Whenever an interaction diagram is created, the corresponding interaction entity is added to the project. Interactions are represented as nodes in the Model Navigator and can be placed inside classes and use cases.

You can view an interaction in two ways: as a sequence diagram, or as a communication diagram. An interaction diagram contains a reference to the underlying interaction.

It is not possible to switch a diagram that already exists from sequence to communication and vice versa. However, it is possible to create a sequence diagram and a communication diagram based on the same interaction.

Sequence diagram can contain shortcuts to other diagram elements. However, you cannot create shortcuts to the elements nested in Interactions.



Related Procedures

UML 2.0 Interaction Diagrams Procedures

UML Profiles

Several pre-installed profiles are provided and allows you to create your own profile definitions using Profile Definition project.

UML Profiles Basics	Provides overview of UML profiles
Profile Definition Project	Provides overview of profile definition project.
Supported Metamodels	Provides a list of metamodels supported.
Stereotype	Describes the stereotype element in the Profile definition project.
Palette Contribution	Describes the stereotype element in the Profile definition project.
Extension Link	Describes the Extension link element in the Profile definition project.

Contribution Link	Describes the Contribution link element in the Profile definition project.

UML Profiles Basics

UML is a standard modeling language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. While the general modeling concepts of UML are quite suitable for the majority of developers, in some situations, a further extension of these concepts is useful to allow a more refined rendering of domain-specific concepts and techniques. UML extension mechanisms address the definition of additional semantics of model elements that cannot be expressed directly using UML constructs. This technique is known as UML Profiling.

Profiles provide mechanisms that allow metaclasses from existing metamodels to be extended to adapt them for different purposes. All kinds of model elements can get stereotypes and tagged values defined in profile applied to the model.

The UML standard provides refinement mechanisms for profile creation, such as stereotypes, tagged values, constraints, and notation icons that collectively specialize and tailor the UML for a specific domain or process. These elements can be used to adapt the UML semantics without changing the UML metamodel. This means that you can interpret the semantics of a profile in the context of the UML specification.

Related Concepts

Stereotype

Profile Definition Project

Profile definition project is a profiled modeling project that allows you to create new profile definitions.

One Profile Definition project corresponds to a single profile. Inside the Profile Definition project you can use some packages to locate different elements, for example, you can put all enumerations to one package, all palette contributions - to another package, all stereotypes - to the third one. All the elements will be deployed to the same profile.

The Profile Definition adds the following elements to the class diagram Tools Palette:

- Stereotype
- Palette Contribution
- Extension
- Contribution

NOTE: Stereotype and Palette Contribution elements are also added to the diagram context menu: **New Profile > Definition**.

NOTE: Metaclasses referenced in a profile must be taken from the corresponding target UML metamodel selected when creating a project.

Related Concepts

Interoperability

Related Procedures

Profiles

Supported Metamodels

You can create profiles on the base of the following supported metamodels:

- ER Physical
- UML 2.0

Stereotypes, tagged values and OCL constraints declared in the profile, can only refer to the selected metamodel.

Related Concepts

Interoperability

Related Procedures

Profiles

Stereotype

Stereotype contains properties that extend a linked metaclass, and enables the use of platform or domain specific terminology or notation in addition to the ones used for the extended metaclass.

When a model element has a stereotype, it indicates that this is a special kind of element that conforms to a rather rigid specification, defined for this stereotype in the profile definition.

The following important issues should be taken into consideration:

- Stereotype is created as a Class20 element in your project with the <stereotype>.
- Stereotype extends a metaclass through Extension link.
- When an instance of a stereotype is created in the target diagram, it gets the attributes (tagged values) of the stereotype in question.
- The following types of attributes (tagged values) of a stereotype are valid: Primitive, Enumeration, Metaclass. Attributes of all the other types are ignored during deployment.
- Tagged values of a stereotype may be defined in two ways: as attributes of the valid types and as association links drawn from a stereotype element to the valid attribute types.
- If a tagged value of a stereotype is defined as an outgoing association, its *name* and *multiplicity* properties are used as follows: name property is set to supplier role of the association link. If the supplier role is not defined, association name is used instead; for multiplicity values other than 1 or 0 to 1 the multivalued attributes are created.
- Stereotype may extend another stereotype via Generalization and inherit its attributes and viewmap. Note that the child stereotype inherits extensions from the parent stereotype, and the parent viewmap (unless it defines its own viewmap).
- Stereotype has the extended metaclass property, which is defined in the Profile Definition node of the Properties View.
- An abstract metaclass may be chosen as extended metaclass. In this case the new element will not appear in any toolbar but all the elements of the child metaclasses will get this stereotype in the list of predefined stereotypes.
- If, after profile deployment, an element belonging to the type of the extended metaclass is used in the target diagram, the extending stereotypes are added to the list of predefined stereotypes for this metaclass.

Related Procedures

Creating Stereotypes

Palette Contribution

Palette Contribution enables you to add creation tools for Stereotypes and pure metaclasses to the selected Diagram Tools Palette.

A contributed Stereotype is associated to a Palette Contribution by means of a Contribution link. It is also possible to associate a Palette Contribution with a shortcut to metaclass from the metamodel. This allows to customize the palette by adding some elements to the tool bars of different diagrams. For example, if you want to have a possibility to create classes from the component20 diagram toolbar, you can associate the shortcut to uml20:: classes::Class with a Palette Contribution in your profile definition.

A Palette Contribution can extend another Palette Contribution; so doing, the child Palette Contribution inherits all the parent diagrams, contributed stereotypes and pure metaclasses.

Related Procedures

Creating Palette Contributions

Extension Link

Extension link indicates that the properties of a parent metaclass are extended with the new properties through a stereotype. An Extension link is drawn from a Stereotype to a shortcut to a metaclass whose properties are extended.

Contribution Link

Contribution link connects a Palette Contribution element with a Stereotype.

Model Compare and Merge

A comprehensive solution is provided for comparing and merging models in your project.

EMF and UML Models Compare

Two-way and three-way comparison of EMF, or UML models, or model elements of the similar type in a tree view are provided.

In two-way compare, the compared models are called *Left* and *Right*. Model Compare/Merge traverses the compared models, going level by level down the containment tree. On each level, objects are matched using ID features which you set in the **ID Features** page of the **Preferences** dialog (**Window Preferences > Modeling > EMF Model Compare > ID Features**). After that, Model Compare/Merge compares values of attributes and noncontainment references.

You can export compare results to an EMF XMI file.

Shared Models Compare

Integration with version control systems is provided and allows two-way and three-way comparison and merge of shared (version controlled) models.

When comparing shared models, *Left* model represents the local version while *Right* model represents the remote version. In three-way compare, the third model is called *Ancestor*. It represents a common ancestor version of the two versions taken from VCS.

ER/Studio Software Architect utilizes standard Eclipse synchronization APIs and is able to compare models stored in any version control system which supports the Eclipse **Synchronize** view.

Comparing and merging of shared models requires one (for two-way comparison) or two (for three-way comparison) remote versions of the compared model.

ER/Studio Software Architect copies your local model to a temporary project, then applies changes reported by the repository provider, and then displays these changes in the **Synchronize** view. Temporary models are read-only, and ER/Studio Software Architect uses a modal **Model Compare** dialog, instead of the standard **Compare** editor.

Merging Models

Merging capability enables you to transfer elements from one model to another.

Related Procedures

Comparing and Merging Models

Model Import and Export Overview

You can share model information with other systems by importing and exporting model information, or by sharing project files:

Feature	Description
Exporting diagrams to images	You can save diagrams in several formats, including: Bitmap image (BMP) Enhanced windows metafile (EMF) Graphics interchange (GIF) JPEG file interchange (JPG) Scalable Vector Graphics (SVG)

Feature	Description
Importing from XMI Exporting to XMI	XMI (XML Metadata Interchange) enables the exchange of metadata information. Using XMI, you can exchange models across languages and applications. For example, if you have a modeling project created with a tool other than ER/Studio Software Architect, you can import it as an XMI file for extension or as the basis of a new project. Likewise, you can export projects for use in other applications. The result in each case is a single, portable .xml file. XMI for UML 2.0 was introduced in IBM® Rational® Software Architect and allows to exchange models that comply with UML 2.0 specification. The models are exchanged via files with .uml2 extension. For import and export, ER/Studio Software Architect supports the following UML versions/platforms: • XMI for UML 2.0 • XMI for UML 2.0 compliant with OMG standard (XMI created without usage of some non-OMG-standard tags such as Annotations.) • XMI for UML 2.1
Export a Quality Assurance metric chart to image	Create a chart and then export it to image.

Related Concepts

Interoperability

Related Procedures

Exporting a Diagram to an Image

Importing a Project in XMI Format

Exporting a Project to XMI Format

XMI Export and Import of the Models with Cross-Project References

OCL Support

This section provides an overview of OCL.

About OCL Support	This topic describes support for Object Constraint Language.
OCL Constraints and Expressions	Describes OCL constraints and expressions.
OCL on Non-Class Diagrams	Describes OCL usage for non-class diagrams.

About OCL Support

This topic describes support for Object Constraint Language.

About OCL

The Object Constraint Language (OCL) is a formal language that describes expressions on UML models. OCL expressions specify operations or actions that, when executed, alter the state of the system. UML modelers can use OCL to specify application-specific constraints in their models. UML modelers also can use OCL to specify queries on the UML model, which are completely programming language independent. For more information about OCL, refer to the OCL 2.0 specification.

You can use all the capabilities of OCL 2.0 to work with your model:

- Add the OCL constraints to the types defined in your model, providing them as constraint notes linked to the context elements on the diagram. The constraint text opens in a powerful editor that provides syntax highlighting, errors validation, and code completion functionality.
- Use OCL as a query language operating with types defined in the metamodel. You can perform a Search In Model by OCL query, write and run Model Audits and Metrics, and use OCL expressions in the documentation templates for the documentation generator.

NOTE: Portions of this product include the Object Constraint Language Library, courtesy of Kent University, United Kingdom. See http://www.cs.kent.ac.uk/projects/ocl/

Supported Diagram Types

OCL supports the diagrams listed in the following table.

Diagram Type	Support Provided
All diagram types	Object constraints. The default language of constraints depends on context element type and project type.
Interaction (Sequence and Communication)	State invariant constraints for lifelines and constraints for the operands of the combined fragments as OCL expressions. Pre— and post- condition for the Interaction. These elements are realized as inner constraint elements available via element's Properties
State Machine	Guard conditions of transitions as OCL expressions. Pre- and post- conditions of a StateMachine, and a StateInvariant for a State. These elements are realized as inner constraints available via element's Properties
Activity	Pre- and post- conditions for activity; local pre- and post-conditions for an action.

Related Concepts

UML Modeling Overview

Related Procedures

Object Constraint Language (OCL)

OCL Constraints and Expressions

As the OMG's specification describes it, OCL is a formal language used to describe expressions on UML models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model.

The icons on the diagram **Palette** allow to create **OCL constraints** as design elements on diagrams, and link these constraints with the desired **context**. The **OCL Expressions** view provides an OCL editor where you can to develop and validate OCL expressions. Any OCL constraint contains an **OCL expression**.

OCL support for constraints provides syntax and error highlighting in the **OCL Editor** view. The text of the constraint is validated when the constraint is linked to its context.

Related Concepts

UML Modeling Overview

Related Procedures

Object Constraint Language (OCL)

OCL on Non-Class Diagrams

Constraints on non-class diagrams fall into two categories:

- Inner constraints
- External constraints

OCL editor provides syntax highlighting, errors validation and code completion functionality.

Inner Constraints

The following properties are defined by creating nested constraints inside the elements. Generally, each property is a property tab that contains two properties - language (ocl/text) and body (constraint body).

- guard in transition/internal transition
- precondition and postcondition in StateMachine, Activity, Interaction
- local precondition and local postcondition in Action
- state invariant in State
- Condition in Extends on Use Case diagram
- state invariant and Interaction constraint on Sequence diagram
- body, precondition, postcondition, and ownedRule in Operation

The OCL editor is available for the constraints inside the elements. You can expand the element node in the Model Navigator and open a constraint in the editor.

NOTE: For the properties of elements that can have class as their context, the OCL context is set automatically.

Constraint	Context
Element with defined constraint	Correct context for the constraint
StateMachine	Operation selected as a Specification association of this StateMachine (a class is selected for the context property and specification is a method of the selected class)
Activity	Specification of the activity
Action	Specification of the activity that contains this action
State and Transition	Class selected in the context property of the StateMachine that contains this State or Transition
Operation	Operation itself Note that the operation has a valid OCL context only if it is owned by Class or Interface. Operations owned by other classifiers get no OCL context, and their constraints should have text as a constraint language.
Interaction	Context of this Interaction or Specification if it is defined
State invariant and Interaction constraint	Class selected as type of the Lifeline that contains this element. It can be either a class directly selected as the type of the Lifeline or part selected as the Lifeline representation.
Extends	Context cannot be specified and constraint can only be defined as text (language=text).

NOTE: If a context for a constraint with language=ocl is not specified or cannot be specified, such constraints are shown as invalid.

External Constraints

In addition to the inner constraints described above, all the elements on the non-class diagrams that are Types (i.e.various classifiers) can be furnished with external OCL constraints.

External constraint is created as a Constraint element linked to the constrained element by the context link.

Unlike inner constraints, the external constraints always use the linked type as a context. Thus, the StateMachine constraints may have the context of the assigned specification if it is inner precondition or the StateMachine itself context if it is the external linked constraint.

Related Concepts

UML Modeling Overview

Related Procedures

Object Constraint Language (OCL)

Patterns

This section describes patterns.

Patterns Overview	Overview of patterns.
Pattern Definition Project	Describes pattern definition project.
Pattern recognition	Describes how pattern recognition works.

Patterns Overview

Patterns provide software developers with powerful reuse facilities. Rather than trying to tackle each design problem from the very outset, you can use the predefined patterns supplied with ER/Studio Software Architect. The hierarchy of patterns is defined in the **Pattern Registry**. You can manage and logically arrange your patterns using the **Pattern Organizer**.

Patterns are pluggable extensions for ER/Studio Software Architect enabling you to:

- Create new and frequently used elements
- Modify existing elements

Patterns

Each pattern describes of a set of model elements, relations between them, and constraints applied to those elements. Patterns are represented by special modeling projects covering all the aspects of patterns. Patterns, in general, are independent of any programming or markup language. You can use them to create or modify any type of element. However, concrete patterns are designed to work with elements of a specific type. Use **Patter Registry** to manage patterns.

NOTE: ER/Studio Software Architect is shipped with some predefined patterns that cannot be deleted or otherwise edited.

Pattern Instances

Pattern instances appear as a result of recognition of the existing model or creating new instances (along with model elements playing pattern roles) in the model. Pattern instances contain information about the pattern name and the role of each participant. They are shown in the **Pattern Explorer** view and under the Patterns node in the **Model Navigator**.

When applied to a diagram, such patterns create their entities and are presented on the diagram itself, with the links to the created entities. Such patterns enable further modification by means of adding new participants (new pattern part). All patterns that appear in the Pattern Explorer are represented in the project model in the form of entities with metaclass “pattern”. Visually pattern instances are displayed as ovals (like collaboration occurrences). Pattern entities have children links to pattern participants and this allows to viewmap links on diagrams from pattern instances to pattern participants. Actions on pattern instances in the model are the same as in pattern explorer.

During the lifetime of the pattern instance the model can change (some elements from the instance may be deleted, some may be changed so that they don't satisfy the pattern definition anymore) and the pattern instance can become invalid. This is why you need to perform pattern instance validation regularly.

Related Concepts

Pattern Definition Project

Pattern Definition Project

Using pattern support subsystem in ER/Studio Software Architect you can easily work with patterns via pattern definitions. You can use well known predefined patterns, define new ones, delete, rename, edit them etc. You can also manage pattern instances: recognize patterns in existing model, create elements by pattern, create new participants for particular roles for existing pattern instance etc.

Pattern definition project is a profiled UML 2.0 modeling project with the following modifications comparing to a pure UML 2.0 project:

The following elements are allowed in a pattern definition project:

- Instance specifications
- Slots
- Pattern definition links (derived from Kernel Association class, able to connect instance specifications)
- Constraints
- Value specifications
- Pattern constraint links (derived from Binary link class, aimed to define constraint parameters)
- Class diagrams
- All other elements are prohibited

The extensions are added to the allowed metaclasses:

- Instance Specification: able to aggregate pattern definition links.
- Slot: the following new properties are added: Use for recognition (boolean) — Controls whether to use this property on recognition; Use for generation (boolean) — Controls whether to set this property on creating elements by pattern; Is configurable (boolean) — Set to true means that in “create by pattern” wizard user will be able to modify value of this property to be set on element creation. This property should be false if use for generation property is set to false.
- Constraint: able to aggregate pattern constraint link
- Class diagram: patternPartWizardDefinition (boolean) — Set to true means that new “create pattern part” wizard will be generated for the instances of this pattern

When you create new pattern instances from existing project elements, the creation process uses participants from your current selection and enables you to modify the pattern properties. You can easily view and/or modify properties of pattern instances using the standard Properties view. Any change that you make to a pattern property applies immediately to the pattern participants (via refactoring).

All pattern definitions are stored in com.borlang.tg.patterns\patterns subfolder of the ER/Studio Software Architect plugins folder. Each pattern definition is an archive file, packed by zip utilities provided by JDK, containing compilation results suitable for recognition and completion engines and the whole definition project in order to allow definition editing. Folder and shortcuts structure are stored in the pattern.registry file in the same location.

Related Procedures

Creating Pattern Definition

Pattern Recognition

Pattern recognition identifies pattern instances from existing elements in the project. The identification process determines pattern participants and parameters. You can start the pattern recognition process from the project's context menu to perform pattern recognition, validation, and problem reporting.

Related Procedures

Recognizing Patterns

Quality Assurance

Quality Assurance provides teams and managers with measures of the quality of their project. As with any Quality Control, the team should understand what is measured, and why. Although audits and metrics are similar in that they both analyze your project, they serve different purposes. Audits and metrics are run as separate processes. Because the results of these two processes are different in nature, ER/Studio Software Architect provides different features for interpreting and organizing the results.

Model Metrics	Describes model metrics.
Metrics Graphical Representation	Describes Kiviat chart representation of metrics.
Exporting and Importing Audits and Metrics	Introduces import and export of audits and metrics.

Model Audits

A wide range of model audits are supported. The list of available model audits can be viewed in the **Preferences** dialog. You can define, save, and reuse sets of model audits. Model audits are OCL queries that produce Boolean result and operate in the context of existing metamodels. You can also employ additional OCL operations provided for a metamodel, specified in the **OCL operations** and **OCL library operations** tabs in the **Preferences** dialog.

After you run model audits, the results are displayed in the Model Audits View. The view provides detailed descriptions for all found errors and you can navigate to the corresponding problem element from this view by double clicking the error message.

Related Concepts

Quality Assurance

Model Metrics

ER/Studio Software Architect supports a wide range of model metrics. The list of available model metrics can be viewed in the **Modeling > Quality Assurance > Model** node of the **Preferences** dialog. You can define, save, and reuse sets of model metrics. Model metrics are OCL queries that produce Integer result and operate in the context of existing metamodels. You can also employ additional OCL operations provided for a metamodel, specified in the **OCL operations** and **OCL library** operations tabs in the **Modeling > OCL** page.

After you run model metrics, the results are displayed in the **Model Metrics** view. You can navigate to the corresponding elements listed in the **Model Metrics** view by double clicking the element name.

Related Concepts

Quality Assurance

Metrics Graphical Representation

Metrics results can also be viewed graphically. Two graphic views allow you to summarize metrics results: bar charts and Kiviat charts. Both charts are invoked from the context menu of the table. Use the Kiviat chart for rows and the bar chart for columns.

Bar Chart

The bar chart displays the results of a selected metric for all packages, classes, and/or operations. The bar color reflects conformance to the limiting values of the metric in reference:

- Green represents values that fall within the permissible range.
- Red represents values that exceed the upper limit.
- Blue represents values that are lower than the minimal permissible value.
- A thin vertical red line represents the upper limit and a thin vertical blue line represents the lower limit.

Kiviat Chart

The Kiviat chart demonstrates the analysis results of the currently selected class or package for all the metrics that have predefined limiting values. The metrics results are arranged along the axes that originate from the center of the graph.

Each axis has a logarithmic scale with the logarithmic base being the axis metric upper limit so that all upper limit values are equidistant from the center. In this way, limits and values are displayed using the following notation:

- Upper limits are represented by a red circle. Any points outside the red circle violate the upper limit.
- Lower limits are represented by blue shading, showing that any points inside the blue area violate the lower limit. Note that blue shading does not show up in areas of the graph with lower limits of 1 or 0.

  TIP: To see the value of an individual data point on the Kiviat graph, hover your mouse pointer over it to display a popup.

- The actual metrics show up in the form of a star with metric values drawn as points.
- Green points represent acceptable values.
- Blue points represent values below the lower limit.
- Red points represent values exceeding the upper limit.
- Scale marks are displayed as clockwise directional ticks perpendicular to the Kiviat ray.
- Lower limit labels are displayed as counterclockwise directional blue ticks perpendicular to the Kiviat ray.

Related Concepts

Quality Assurance

Exporting and Importing Audits and Metrics

Introduces import and export functionality for audits and metrics.

Version Control

This topic provides an overview of version control features.

Overview

Several version control systems are provided that can be integrated in Eclipse. They include but not limited to CVS. Version control in ER/Studio Software Architect enables several users to work with one modeling project.

ER/Studio Software Architect provides context menus to work with CVS version control systems.

Your version control system should be set up so that only one user can work with a shared model at a time. In case several users edit the model at a time, use model compare and merge functionality of ER/Studio Software Architect. You can compare the structure of your models and merge inconsistencies if necessary. Alternatively you can revert to the saved version of the model.

NOTE: For more information about each version control system, refer to the appropriate program documentation.

Task-aware features

Task-aware features of ER/Studio Software Architect make easier your work with modeling resources under version control by automatically finding all the resources that need to be checked out for the modeling resource to be modified. To enable task-aware features, use the following Team options for locking files and managing modeling resources:

Path to option	Options
Windows > Preferences > Team > Modeling resources	Auto Checkout modeling resource on edit and Ignore default package diagrams.

You can use the Ignore default package diagrams option to not store and synchronize default package diagrams. This option adds default.txvpck and default.txvClassDiagram20 patterns to the ignored resources

Project Documentation

This part describes the documentation generation facility and documentation template basics.

Related Procedures

Generating HTML Documentation

Documentation Generation

You can create external documentation for the open projects, or from the command line. Use the generated reports to illustrate your projects with the documentation in one of the available formats.

Documentation generation is available for all types of projects, including Pattern Definition or Profile Definition projects.

Documentation output formats

You can generate documentation in one of the following output formats:

- RTF
- HTML
- TXT
- PDF
- XSL-FO

By default, documentation is generated in HTML format.

Documentation Files

All the documentation that ER/Studio Software Architect generates is written to a single directory that you specify in the documentation generation dialogs. By default, this is out folder of your Eclipse workspace.

The generated documentation opens in the appropriate viewer, associated with the output format.

Documentation Templates

Project reports are created by applying documentation templates to projects. The templates contain commands to the documentation generator; the projects provide the source of project-specific data. Documentation templates are *.tpl text files with formatting instructions and tags for the commands.

Related Concepts

Organization of a Documentation Template

Related Procedures

Generating HTML Documentation

Documentation Template

The documentation generator uses projects and templates to produce project reports. You can use predefined templates that are delivered with the product, or create your own custom templates, using the Template designer. This part discusses the structure of templates, its zones, sections and controls.

Documentation Generator Metamodel	Documentation Generator makes use of its own metamodel that defines the hierarchy of metatypes.
Organization of a Documentation Template	This topic describes organization of a documentation template and the correspondence between the elements of a template and the generated output.

Documentation Template Sections	This topic describes sections of a documentation template.
Documentation Template Controls	Controls are the items in documentation templates that determine the contents of reports
Multi-frame Documentation Templates	This topic describes multi-frame documentation templates structure.
Hyperlinks in Documentation	A hypertext link connects a link reference (starting point or source) to a link destination (target).
Javadoc Link References	Javadoc References (or JDRefs) are supported. These are expressions associated with Javadoc tags.
Enable Conditions	Enable conditions are boolean expressions for turning section processing on or off.

Documentation Generator Metamodel

Documentation generator has its own metamodel described in the metamodel definition file plugins \com.embarcadero.gendoc.core_8.1.0\templates\MetaModel.mm.

It is a textual file that defines the metatypes hierarchy, how metatypes correspond to the model elements, the types of elements another element can contain, and the properties of each metatype. The beginning of each model definition file lists the properties that DocGen knows. These include DocGen-specific properties and others. Properties are defined as follows:

```
property_name = "[name of property localization key]"
```

The remainder of each model definition file contains the metatype definitions. The major fields in the definitions are as follows:

- name: metatype name
- extends: parent metatype
- full_name: the name displayed in the Documentation Template Designer
- metatype_filter: defines the correspondence between metatype and model element
- rwi_entity: the type of the related element in API
- properties: a list of properties available for this type. Descendant metatypes inherit their properties from parent metatype.
- excluded_properties: items listed among the properties that aren't documented when using 'all properties' scope in Property iterator
- contained_metatypes: metatypes that can be contained by this metatype

The name field for each type is always present. The existence of the other fields varies with the type. Below is the examples of metatype definition

```
<metatype>

name=NODE

extends=ELEMENT rwi_entity=node

full_name="[gendoc/gen_doc_by_template1/full_name_NODE]”

properties = { %package }

contained_metatypes = { NODE; MEMBER; LINK }

</metatype>
```

An element iterator or folder can contain nested element iterators whose type is listed among its contained metatypes, the contained metatypes of its parent, or indirectly through the contained metatypes of one of its contained metatypes. For example, an element iterator with DIAGRAM scope can contain nested element iterators with the following scopes:

- hyperlink (inherited from ELEMENT)
- diagram reference,
- diagram,
- node,
- link
- member (indirectly through the contained metatype, NODE)

Element properties are inherited. An element iterator can contain nested property iterators whose type is inherited from its ancestor or listed directly among its properties. For example, an element iterator with DIAGRAM scope can contain nested property iterators for the following type scopes: shape type, name, documentation, annotation, hyperlink, url (inherited from ELEMENT), or package, stereotype, alias.

Related Concepts

Organization of a Documentation Template

Organization of a Documentation Template

Documentation template is a *.tpl text file that contains instructions to the Documentation Generator. Project reports are created by applying documentation templates to projects.

In this section you will learn about:

- Zones of a template
- Body of a template, and its representation in a generated report
- Root object metatype
- Current model elements

Zones of a Template

Documentation templates consist of headers, footers, and body sections. The Documentation Template Designer divides templates into five major zones:

- Page header
- Report header
- Body
- Report footer
- Page footer

The zones are horizontal bands that go across two panes. The *scope pane*, which is on the left, reveals the template structure. The *details pane* on the right shows the contents of the zones, which include commands to the DocGen engine. Context menus for each zone are different in the scope pane and in the details pane.

Headers and footers are at the top and the bottom of the Designer window. The report header and the report footer apply only once per document. Page headers and footers apply once per page for RTF documentation; they are ignored for HTML and text documentation.

The body zone of a template contains the commands that produce the body of the generated report. DocGen builds a report into horizontal regions. Each region in the report corresponds to a section in the template that determines the data for that region and how that data should appear.

Body of a Template

The body of a documentation template is organized into a hierarchy of sections. Some sections in the body are nested inside others. Some sections have siblings. Sections that are not nested within any others are children of the root. The scope pane reveals the tree structure, indenting each section according to its level in the tree.

Root Object Metatype

Every section in the body of a template has a section scope. Scopes are based on metatypes that correspond to the different types of model elements. The section scope of the body zone corresponds to the root object metatype. The model itself is considered to be a special metatype, which is the default root metatype for a new template.

Current Model Element

Documentation Generator uses a dynamic current model element to go through a template and access specific project information. The type of the current element is the metatype for the section that the engine is currently processing. The value of the current element changes according to when the processing for the section takes place.

The body of a report is created starting from the root element, going in a “depth-first” fashion. In other words, processing starts with the first root section, visiting it along with any of its nested subsections before continuing to the next root section. This pattern is recursive: visit the sub-tree rooted at a section before going to the next sibling section. For each sibling of a section, DocGen begins its processing with the same current element.

Documentation Template Sections

The body of a newly created template consists of a generic element iterator and a static section nested within. It provides a minimal base for constructing the tree of sections. Every new section must be a sibling or a child of an existing section.

There are six different types of body zone sections:

- Static sections
- Element iterators
- Element property iterators
- Folder sections
- Calls to stock sections
- Calls to template sections

Static Section

Static sections contain the commands to the Documentation Generator for getting project data.

Of all kinds of body sections, only static sections contain controls for producing actual output. Headers and footers can also contain controls. Folders and iterators, which cannot directly contain controls, must have at least one static section nested somewhere within.

You can edit properties of a static section. Refer to the *Static section* reference for details.

Element Iterators

Element iterators provide a way of looping through elements of a model. Each element iterator has its own metatype, which must be consistent with the metatype of the iterator's parent's.

If you want an iterator to be able to access an entire model, choose Package as the metatype.

In an element iteration section, a new current element is calculated according to the current element of the parent section and the metatype of the iterator. Documentation Generator loops through an element iteration section using each possible new element as the current element for that iteration. The properties of an element iterator affect the way a new current element is calculated and how it changes during iterations. If no elements are encountered corresponding to the iterator's metatype, no documentation is produced.

Element iterators can have headers and footers. If the section execution does not result in output, then the iterator's headers and footers are ignored.

Scope options determine which elements of the model this iterator will document. Each iterator works over the subtree of the model that is rooted at the current element (the element that starts the iteration).

You can edit properties of an element iterator. Refer to the *Element Iterator* reference for details.

Element Property Iterators

Element property iterators are for looping through the properties of model elements.

Element iterators traverse model elements. Element property iterators traverse element properties instead of elements.

A property iterator can reside inside an element iterator, folder, or property iterator. A property iterator must contain at least one static section, folder section, or call to a stock section or template. A property iterator may also contain an element iterator, or another property iterator.

A property iterator is described by its properties: iteration scope, sorting etc. Refer to the *Property Iterator* reference for details.

Folder Sections

Folder sections group other sections together. A folder has at least one nested section, and it may have a header or footer. In that sense, folders are similar to element iterators, except that DocGen executes folders only once.

Folders inherit their metatypes from their parents. The sections nested within a folder must be consistent with its metatype. Folders provide a way to put section-level properties on their contents. This includes enabling conditions for toggling its processing on and off.

Folders can have headers and footers. If the sections in a folder do not result in output, then the folder's headers and footers are ignored.

A folder section is described by its properties: output style, enable condition etc. Refer to the *Folder section* reference for details.

Calls to Stock Sections

Stock sections are reusable folders or iterators that reside in the template's collection of stock sections. They are not shared among different templates. When a call to a stock section is processed, it is the same as if the called stock section were simply embedded at the position of the call.

Stock sections are especially convenient for frequently used constructs. You can insert a call to a stock section from any section whose metatype is consistent with the metatype of the stock section. Stock sections may contain calls to other stock sections as well recursive calls to themselves.

You can edit properties of a call to stock section. Refer to the *Call to stock section* reference for details.

Calls to template sections

With a call to a template, DocGen can produce documentation using a different template without terminating the current one.

When a template is called, the current element of the calling template becomes the root element of the called template. A calling template can pass additional information to the called template through template parameters.

Calls to templates make it possible to construct a library for generating documentation for particular model elements (class, actor, use case, and so on).

You can edit properties of a call to template section. Refer to the *Call to template section* reference for details.

Documentation Template Controls

Of the six kinds of body sections, only static sections contain controls for producing actual output. Headers and footers can also contain controls. Folders and iterators, which cannot directly contain controls, must have at least one static section nested somewhere within.

When you insert a new control, the Documentation Template Designer displays a dialog box for setting the control's properties. The template shows each control as a shaded rectangle in the details pane. You can change the properties of a control after it is created.

The contents of this section includes:

- Label
- Image
- Panel
- Formula
- Data
- Include Text

Label, Image, and Panel Controls

The simplest kinds of controls are labels, panels, and images.

Label

A **label** generates static text that is independent of its containing section. The text does not depend on the metatype of the section or where the section belongs in the template. Placing identical labels in a header and a static section results in the same output as long as the header and static section are not skipped. Label properties include the label's text, style (font, color, and border), and if and how to hyperlink the output.

Image

Depending on its type, an **image** can be external to the project or it can be a project diagram. You can put an image control in a static section to include an image of a diagram in the generated document. Documentation Generator, while processing the section, will create an image only if the current model element represents a model diagram.

Panel

A **panel** is simply a container for other controls. Panels are convenient for grouping controls together to provide a uniform style and precise alignment. You can set the panel's background color, border, and style, and the parameters that will be passed to the controls within the panel.

Data Controls

Data controls provide the major mechanism of placing data from a project into a report. When a data control is processed, the actual data are obtained from the current model element.

The source of information for a data control can be one of the following:

Element Property	A property of the current element. The Data Control dialog box displays a list of every property belonging to the metatype of the current model element.
Generator's Variable	A variable used by DocGen. You can use this in report headers or footers to insert the project name or the date and time the report is created.

Document Field	A field of the report such as page number or bookmark. You can select Document Field to insert page numbers and number of pages into page headers or footers. The Document Field list is empty for report headers and footers.

Formula and Text Controls

Formulae provide a way to place data into a report that DocGen calculates when it processes the control. You must enter the formula that DocGen evaluates to calculate that output. Both formula controls and text controls rely on such formulas.

Formula controls

A formula is an expression that Documentation Generator can evaluate to a string. The expression can be a combination of string literals, DG variables, and OCL or legacy RWI functions.

DG variables are special variables that are available to DocGen at runtime when it is producing a report. DG variables include items such as current element, the date and time, and template parameters. Find the complete list of DG variables, OCL functions and legacy RWI functions in the section *"Documentation Generator and Template Designer Reference"*

Supported formula types are Legacy and OCL. Syntax depends on the selected formula type, as shown in the following table.

Supported formulae type	Syntax
Legacy	single quotes for string literals; + for string concatenation -> for calls to functions via pointers
OCL	OCL

The following examples demonstrate the usage of formulae expressions for the different formulae types:

Example 1:

From a section with **class** metatype put Package followed by the name of the containing package into the report:

Syntax	Formulae expression
Legacy	"Package " + getContainingPackage() -> getProperty("$name")
OCL	context uml14::kernel::classes::Class 'Package '.concat(self.getContainingPackage().name)

Example 2

From a section with a **generic class** metatype, put Interface in the report if the current element is an interface and Class if it is not.

Syntax	Formulae expression
Legacy	if (hasProperty("$interface"), "Interface", "Class")
OCL	context uml14::kernel::classes::Class if self.interface then 'Interface' else 'Class' endif

Include Text controls

Include Text controls are used for copying text from other files into a template. When you insert an **Include Text** control, you must enter an expression for the location of the text file. The expression can be hard coded as a string literal, or it can use a formula as described above. **Include Text** controls have formatting properties identical to those for formula and label controls.

Multi-frame Documentation Templates

Multi-frame HTML documentation divides project reports into frames to give multiple views within the same browser window. Multi-frame HTML documentation consists of two kinds of HTML files:

- A collection of HTML files to define the content for each frame
- A frameset file to specify the layout of frames

A frameset template consists of two major parts. One part describes the frameset file that can be defined through the template properties. The other part, which is the body of the frameset template, contains calls to the templates that provide the contents of the frames.

The body of a frameset template is similar to the body of an ordinary document template. A frameset template body can contain any number of iteration sections (element iterators and property iterators), folder sections, and stock section calls. However, static sections and headers and footers for folder sections and iterators are prohibited. Calls to template sections replace static sections to produce the actual output.

The section properties of a call to template determine how the output for a template call can be used. With multiframe HTML documentation, calls to template sections typically generate separate files that can be loaded into a frame of the resulting HTML project documentation.

When Documentation Generator processes a frameset template, it produces the frameset HTML file and the separate HTML files for the frame content. Documentation Generator begins processing a frameset template at its body. When it encounters a call to a template section, the engine suspends the current template execution, loads the called template, and processes it to produce a separate HTML document. The root element for the called template is the current model element of the calling template. After processing the called template is completed, Documentation Generator resumes executing the calling template. After processing the body of the frameset template is completed, Documentation Generator produces the special HTML frameset file. This file corresponds to the frameset structure specified in the template properties. The name of the frameset file matches the name of the frameset template. It is the starting point of the generated documentation.

Hyperlinks in Documentation

A hypertext link connects a link reference (starting point or source) to a link destination (target). The link reference is a text or image in the HTML document. The link destination is a file (usually an HTML document or an anchor in an HTML document). Document templates support both references and targets. Link references are properties of controls. Link targets are properties of static sections, headers, and footers.

Any generated output that contains an anchor or bookmark can be a link target. Documentation templates have facilities for inserting anchors at the “main documentation” of model elements.

It is occasionally necessary to provide link references to several different documents (or locations in HTML files) created with the same model element. For example, along with the main documentation file created for a package, there could be a different HTML document that simply lists all classes in the package. If this listing document were in a separate "navigation" pane, it would serve as an index for the package. Clicking the package on a diagram (or in some more general text) could load that listing document in the navigation frame. The Documentation Template Designer enables you to target different documentation locations generated by the same model element.

Link references in multi-frame documentation may have multiple targets. Clicking on such a reference could simultaneously load two different documents in two different frames. For example, suppose a diagram element represents a package. Clicking on this element could load the image of the package diagram in one frame and the main (textual) documentation for the package in another. Such link references are named "compound".

Javadoc Link References

Javadoc References (or JDRefs) are the expressions associated with Javadoc tags such as {@link} and @see. You can use them to create link references inside documentation text ({@link}) as well as with some other documenting tags. Documentation Generator can convert JDRefs into real hypertext links. Each JDRef should conform to the rules described in the standard Javadoc documentation. There are three types of Javadoc references.

- Element reference refers to an element of the model (method, class, package, etc.). The general form of an element reference is: package.class#member label, where package.class#member is the referenced model element and label is optional text to be displayed with the link. (If label is omitted, the name of the referenced element is displayed.) Documentation Generator can convert each element reference into a hyperlink to the main documentation of the element.
- URL reference represents a link to a relative or absolute URL. The general form of a url reference is:<a href="URL#value">label</a>
- Text reference has the form "string" (a text string in double-quotes). A text reference is simply information that does not represent a hyperlink.

A JDRef appears in one of two forms:

- inside {@link} tags embedded in documentation text. The JDRef is the value of the $doc property and other Javadoc element's properties.
- as the value of some Javadoc element's properties such as see.

The Documentation Template Designer provides conversions for both cases. You need to specify the conversion in the properties of the control.

Enable Conditions

Enable conditions are boolean expressions for turning section processing on or off. They are created using the OCL or legacy notation.

Enable condition is evaluated before stepping into this section, so the properties of the metatype of a section are not available to Documentation Generator at the moment of expression evaluating. Enable conditions typically have sub-expressions that are calls to special DG functions returning DG options and template parameters. (See the list of DG functions and variables in the Documentation Template Designer Reference.) They can also use the properties of the upper level section metatype. The results may be joined together with logical operators under the usual precedence rules. Here are two examples of the enable conditions in the Legacy and OCL notation:

Legacy	OCL
getDGVariable('reportScope') != 'current_diagram'	context OclAny getDGVariable('reportScope') <> 'current_diagram'
getContainingNode() -> hasProperty ("$interface")	context uml::kernel::Element self.getContainingNode().oclAsType (uml14::kernel::classes:: Class).interface

Procedures

This section provides how-to information for the various areas of software development for the ER/Studio Software.Architecht application.

Getting Started Procedures	Provides how-to information that will help you start using the product.
Diagrams	This section describes how to create diagrams, customize their appearance, and populate diagrams with elements and shortcuts.
Projects	This section provides how-to information on using projects.
Profiles	This section provides how-to information about Profiles.
Comparing and Merging Models	Describes how to compare models and model elements with each other, and perform history comparison with the earlier versions of the model stored in Version Control Systems (VCS).
Object Constraint Language (OCL)	This section provides how-to information on using the OCL facilities.
Patterns	This section provides how-to information on using patterns.
Quality Assurance	This section provides how-to information on using Audits and Metrics.
Using Version Control and Teams	This section describes the use of Version Control Systems (VCS).
Generating Project Documentation	How-to information on using Documentation Generation facilities.
Documentation Templates Procedures	This section provides how-to information on creating and editing custom documentation templates using the Documentation Template Designer.
Interoperability and Migration	This section provides how-to information on exchanging model information between the various products of the ER/Studio Software Architect product line.

Getting Started Procedures

This section provides how-to information on configuring ER/Studio Software Architect, working with projects, and more.

Instant On Application Launcher	How to launch and use the Instant On Application Launcher
Adding a Single Model Element to a Diagram	How to create a single model element.
Bookmarking Model Elements	How to bookmark model elements for easy access.
Choosing a Perspective	How to choose a perspective.

Configuring Preferences on the Workspace and Diagram Levels	How to define preferences on the workspace and diagram levels.
Creating a Browse-Through Sequence of Diagrams	How to create a browse-through sequence of diagrams.
Creating a Diagram	How to create a diagram in a project.
Creating a Project	How to create a project in ER/Studio Software Architect for Eclipse.
Creating a Shortcut	How to create a shortcut.
Creating a Simple Link	How to create a simple link.
Deleting a Diagram	How to delete a diagram
Deleting Elements	How to delete an element from diagram.
Hiding and Showing Model Elements	How to hide or show model elements.
Opening a Diagram	How to open an existing diagram in the Diagram Editor.
Printing Diagrams	How to print your diagrams.
Selecting Model Elements	How to select model elements.
Using Drag-and-Drop	How to use drag-and-drop.
Using Example Projects	How to use sample projects in ER/Studio Software Architect for Eclipse.

Instant On Application Launcher

ER\Studio Software Architect is shipped with an Instant On (ION) build, which is a Xenocode virtualized application.

You can use tags to launch virtualized applications, for example:

```
<ERStudio Executable> tag[parameters]
```

The following tags are valid:

Tag	Description
cmdLine	start cmdLine.cmd file
gendoc	start gendoc.cmd
genhtml	start genhtml.cmd
model-audit	start model-audit.cmd
model-metric	start model-metric.cmd
XMIExport	start XMIExport

Adding a Single Model Element to a Diagram

You can create a single node element using the diagram Palette, or the **New** command of the diagram context menu.

To create a single model element

1 Open a target diagram in the Diagram Editor.

2 On the Palette, click the icon for the element you want to place on the diagram. The icon stays highlighted.

TIP: Icons are identified with tooltips.

3 Click the diagram background in the place where you want to create the new element. This creates the new element and activates the in-place editor for its name.

TIP: Alternately, you can right-click the diagram background in the Diagram Editor, or the diagram node in the Model Navigator, and choose **New** on the context menu. The submenu displays all of the basic elements that can be added to the current diagram, and the **Shortcuts** command.

Related Procedures

Adding Multiple Elements to a Diagram

Creating a Simple Link

Bookmarking Model Elements

You can bookmark model elements. Bookmarked elements are listed in the **Model Bookmarks** view.

To add or remove a bookmark

1 To add a bookmark, right click an element on the diagram editor and choose **Model Bookmarks > Add Bookmark**.

2 To remove a bookmark, right click an element on the diagram editor and choose **Model Bookmarks > Remove** Bookmark. Alternatively, you can use the context menu of the **Model Bookmarks** view to remove a bookmark.

To navigate to a bookmarked element

1 Open the **Model Bookmarks** view.

2 Right click a bookmark and choose either **Show in Model Navigator** or **Select on Diagram**.

NOTE: Double click a bookmark in the **Model Bookmarks** view to select the element on the diagram. To open the Model Bookmarks view click **Windows > Show Views > Model Bookmarks**.

Choosing a Target Perspective

ER/Studio Software Architect changes the user interface according to how you want to work by providing several perspectives. By default ER/Studio Software Architect starts with Modeling perspective.

To choose a Perspective

1 On the main menu, choose **Window > Open Perspective > Other**. The **Open Perspective** dialog opens.

2 Select one of the Perspectives from the list, and click **OK**.

After you select a perspective, ER/Studio Software Architect automatically customizes the interface to provide ready access to only the relevant elements of the interface, and to show only the information in the model that best supports the chosen perspective. Interface elements and/or model information that are not generally relevant to the perspective are hidden. You can still access hidden information by changing the relevant configuration options and restoring hidden panes manually, but you may find it easier to just switch perspectives.

Configuring Preferences on the Workspace and Diagram Levels

You can flexibly change configuration. Use the **Preferences** dialog to tune modeling features to best fit your requirements.

The **Preferences** dialog window provides a number of diagram customization settings. You can configure the appearance and layout of the diagrams, specify font properties, member format, and level of detail on the diagram and workspace levels.

To configure settings on the workspace level

1. On the main menu, choose **Window > Preferences**.
2. In the **Preferences** dialog window, expand the **Modeling** category.
3. Click the desired subcategory.
4. Edit configuration options as required.
5. Click **OK** to apply changes and close the dialog window.

You can configure certain diagram-specific options (Diagram, Layout, View management and Print) on the diagram level

To enable configuration changes on the diagram level

1. On the main menu, choose **Diagram > Preferences**.
2. Set the check box **Enable diagram-specific settings**.
3. Click the desired subcategory (Diagram, Layout, Print, and View management).
4. Edit configuration options as required.
5. Click **OK** to apply changes and close the dialog window.

To disable configuration changes on the diagram level

1. On the main menu, choose **Diagram > Preferences**.
2. Clear the check box **Enable diagram-specific settings**.
3. Click **OK** to apply changes and close the dialog window.

Creating a Browse-Through Sequence of Diagram

You can link entire diagrams at one level of detail to the next diagram up or down in a sequence of increasing granularity, or you can link from key use cases or actors to the next diagram.

To create a browse-through sequence

1 Open the main diagram of the sequence you are going to create.

2 Select the source model element, or right-click the diagram background to link the entire diagram.

   TIP: It is recommended to use some common approach for all links in your sequence.

3 Create a hyperlink to the next diagram or model element you would like to participate in the sequence. The titles of source and destination model elements turn blue.

4 Open the destination diagram.

5 Repeat steps 3–4 for all parts of your sequence.

6 Optionally, create hyperlinks in the reverse motion.

Related Concepts

Model Hyperlinking Overview

Creating a Diagram

Diagrams exist within the context of a project. Create or open a project before creating any new diagrams.

To create a new diagram from the Model Navigator

1 In the Model Navigator, right click on a package or the project root.

2 From the context menu, select **New Diagram** and choose the diagram type from the submenu.

To create a new diagram using the Diagram Editor toolbar

1 Click the arrow to the right of the **New Diagram** icon ( ) on the diagram editor toolbar.

2 Choose the diagram type from the submenu.

   OR

3 Click directly on the **New Diagram** icon. The Create Diagram wizard dialog opens.

4 In the resulting dialog: Select a diagram type from the drop down list. Select the package where the new diagram will be created. Click **Browse** to choose a package. Enter a name for the new diagram.

5 Click **Finish**.

To create a new diagram using the New Diagram Wizard

1 Select **File >New >Diagram** and the Create Diagram wizard opens.

2 Specify the properties for the new diagram as follows:

   - **Type:** Use the drop down list to select a diagram type. By default, the Class Diagram is selected.
   - **Location:** By default, the new diagram is created in the package selected before the wizard displays.
   - **Name:** Use the text field to type a name for the new diagram.

3 Click **Finish**.

To create a class diagram from a package diagram

1. On the package diagram, select classes that you would like to display in a separate class diagram.
2. On the main menu, select **Model > Generate Class Diagram**.

   NOTE: This action is only available when several classes are selected.

Creating a Project

ER/Studio Software Architect provides several projects that you can work with. The projects are created in the same manner. While creating a project you will specify different options depending on the type of project.

To create a Project

1. Select **File > New > UML 2.0 Project** on the main menu. The **New Project** wizard displays.
2. Enter the name of your new project. You can use the default location of browse to a new one.
3. You can also choose to add the new project to working sets. Working sets allow you to customize your view and determine which resources are to be displayed.
4. Click **Next**.
5. Follow the wizard to specify necessary options for a new project and click **Finish** to complete the wizard.

Related Concepts

Project Overview

Creating a Shortcut

You can create a shortcut to a model element from the current project or from the projects connected by cross projects references, by using two methods:

- By choosing **New > Shortcut** on the Diagram Editor context menu.
- By dragging and dropping a shortcut from the Model Navigator

To create a shortcut by using the Shortcuts dialog window

1. Right-click the diagram background.
2. Choose **New > Shortcuts** on the context menu.

   TIP: Use CTRL+SHIFT+N keyboard shortcut

3. In the **Shortcuts** dialog window, choose the required element from the tree view of available contents.

   NOTE: If the project has cross-project references to the other projects in the workspace, the contents of these projects is available for being added as shortcut.

4. Click **Add** to place the selected element to the list of the existing or ready to add elements.
5. When the list of ready to add elements is complete, click **OK**.

To create a shortcut by using drag-and-drop

1. Select the element in the Model Navigator.
2. Drag-and-drop the element onto the diagram.

Related Concepts

Model Shortcut Overview

Related Procedures

Establishing Cross-Project References

Adding a Single Model Element to a Diagram

Creating a Simple Link

You can create a link to another node, or a shortcut of an element of the same or another project (these projects must be of the same UML version).

To create a simple link between two nodes

1. On the diagram Palette, click the icon for the type of link you want to draw in the diagram. The icon stays highlighted.
2. Click the source element.
3. Drag to the destination element and drop when the target element is highlighted.

Related Procedures

Creating a Link with Bending Points

Creating Model Element by Pattern

Deleting a Diagram

CAUTION: You cannot delete the default diagram created automatically for a package.

1. In the **Navigator**, select the diagram to be deleted.
2. On the context menu, choose **Delete**. A confirmation dialog opens asking to confirm your delete request.

   NOTE: You can also preview the diagram to be deleted or cancel the request.

3. Click **OK** and the diagram is deleted.

Related Procedures

Creating a Diagram

Closing a Diagram

Deleting Elements

All elements shown on diagrams are shortcuts to actual model elements. When deleting an element on a diagram you have the option to delete either the shortcut from view or delete the element from the model (except classes on synchronized package diagrams). This behavior is configured in the Modeling Preferences.

To delete an element

1. Select the element on diagram.
2. Choose **Delete** on the context menu of the element.

   TIP: Alternately, click the DELETE key.

3. Confirm deletion, if this behavior is selected in the Modeling Preferences.

To delete an element from View

1. Select the element on the diagram.
2. Choose **Delete from View** on the context menu of the element.
3. Confirm deletion, if this behavior is selected in the Modeling Preferences.

Related Procedures

Adding a Single Model Element to a Diagram

Hiding and Showing Model Elements

You can control the visibility of elements on a diagram by using the **Hide** command (available on the context menu for individual diagram elements), and the **Show/Hide** command (available on the diagram context menu).

To hide by using one of the following methods

1. Open the Diagram Editor.
2. Do one of the following:
   - Select the element on the diagram, right-click and choose **Hide** on the context menu.
   - Select multiple elements on the diagram using CTRL+CLICK or by lassoing, and select Hide from the context menu.
   - Right-click the diagram background and choose **Hide/Show** on the context menu. The **Show Hidden** dialog opens, as discussed below.

To show or hide diagram elements using the Show Hidden dialog

1. Right-click the diagram and choose **Show/Hide** on the context menu. The **Show Hidden** dialog opens.
2. Select the element(s) that you wish to hide from the Diagram Elements list.

3 To add elements in the Diagram Elements list to the Hidden Elements list, do one of the following:
   - Double-click the element.
   - Click the element once and click Add.
   - Select multiple elements using CTRL+CLICK and click Add.
4 To remove items from the Hidden Elements list do one of the following:
   - Double-click the element.
   - Click the element once and click **Remove**.
   - Select multiple elements using CTRL+CLICK and click **Remove**.
   - To remove all items from the Hidden Elements list, click **Remove All**.
5 Click **OK** to close the dialog.

Related Procedures

Adding a Single Model Element to a Diagram

Opening a Diagram

You can open diagrams from the Model Navigator, Navigator or by using the Diagram Editor toolbar. In this section you will learn how to open an existing diagram

To open a diagram

1 In the Model Navigator view navigate to the diagram you want to open.
2 Select the diagram node in the tree-view and do one of the following:
   - Double click the diagram
   - Select **Open** from the context menu
   - Select **Open in Active Editor** from the context menu (this replaces the contents of any currently opened diagram)

You can use the Diagram Editor toolbar to open the parent diagram.

Related Procedures

Opening a Parent Diagram

Printing Diagrams

To print a diagram or multiple diagrams

1 Open the diagram or select the tab in the Diagram Editor that displays the diagram.
2 Select **File Print** on the main menu. The **Print Diagram** dialog displays.
3 Select the scope for printing.
4 If you check the option **Print whole diagram as an image**, the option **Print diagram as black and white image** becomes enabled. Make your desired selections.

5 Click **Preview** to see how the diagram or diagrams look with the current print settings.

6 Click the drop down arrow to set the **Preview Scale** factor

7 Click **Print Options** to define Print Preferences. You can use the scroll bars to scroll around the diagram or to view other diagrams that were included in the scope.

8 Click **Print** to proceed to the standard print dialog where you can select your printer.

Selecting Model Elements

Most manipulations with diagram elements and links involve dragging the mouse or executing context menu commands on the selected elements.

In this section you will learn how to

- Select one or more elements
- Cancel selection

To select an element

1 Open a diagram in the Diagram Editor.

2 On the diagram Palette, click **Select**.

3 In the Diagram Editor, click any element or a member to select it.

TIP: To select multiple elements, do one of the following:

- Hold down the CTRL key and click each element individually
- Click the background and drag a lasso around an area to select all the elements it contains.
- Press CTRL+A to select all elements on a diagram.
- Right-click the diagram background and choose **Select All** on the context menu.

To cancel selection

- Press **ESC**.

Related Procedures

Aligning Model Elements

Using Drag-and-Drop

Drag-and-drop applies to the members as well as to the node elements. You can move or copy members (methods, fields, properties, and so on) by using drag-and-drop in the Diagram Editor or in the Model Navigator. You can also change the origin and destination for links on your diagrams using drag-and-drop. Drag-and-drop functionality from the Model Navigator to the Diagram Editor and within the Model Navigator works as follows:

- Selecting an element in the Model Navigator. and using drag-and-drop to place the element onto the diagram creates a shortcut.
- Using drag-and-drop while pressing the SHIFT key moves the element to the selected container.
- Using drag-and-drop while pressing the CTRL key copies the element to the selected container.

To move a link to a new destination

1. Select a link in the Diagram Editor.
2. Hover the cursor over the destination arrow.
3. Drag the arrow and drop it on the new destination. If the destination element is not in view, drag the link in the appropriate direction, and the diagram will scroll with you.

    TIP: Follow the same instructions to move the link source to an allowable location.

Related Procedures

Selecting Model Elements

Using Example Project

A set of predefined sample projects are provided. To use them, you have to create in your workspace a new project on the base of an example.

To use a Example Project

1. Select **File > New > Other** on the main menu. The **New Project** wizard displays.
2. Expand the **Examples** node in the tree view list, and select the desired project. Click **Next**.
3. Follow the wizard to specify the necessary options for a new project and click **Finish** to complete the wizard.

Related Concepts

Project Overview

Diagrams

This section describes how to create diagrams, customize their appearance, and populate diagrams with elements and shortcuts.

Common Diagrams Procedures	This section describes procedures that apply to all types of diagrams.
Populating Diagrams	This topic provides How-To information about creating node elements, links and members in all types of diagrams.
Editing Diagrams	Lists the Editing Diagrams Procedures.

Common Diagram Procedures

This section describes procedures that apply to all types of diagrams.

Annotating a Diagram	How to annotate a diagram

Browsing a Diagram with Overview Pane	How to browse.
Changing the Default Diagrams Directory	
Closing a Diagram	How to close a diagram.
Creating a Diagram	How to create a diagram in a project.
Deleting a Diagram	How to delete a diagram.
Exporting a Diagram to an Image	How to export a diagram to an image.
Hiding and Showing Model Elements	How to hide or show model elements.
Hyperlinking Diagrams	How to hyperlink diagrams.
Opening a Diagram	How to open an existing diagram in the Diagram Editor.
Opening a Parent Diagram	How to open parent diagram of the current diagram.
Printing Diagram Elements	How to print one or more diagram elements.
Printing Diagrams	
Searching Model Elements	How to search model elements on diagrams.
Searching Model with OCL queries	How to search for model elements using OCL queries.
Using a Class Diagram as a View	How to use a class diagrams as a view.
Zooming a Diagram	How to zoom a diagram.

Annotating a Diagram

Use the following actions to annotate a diagram:

1. Draw a note
2. Draw a note link
3. Type comments

To draw a note

1. In the Diagram Editor, you can:
   - Hyperlink the note to another diagram or element.
   - Edit the text when its in-place editor is active.
   - Edit the properties of a note using Properties View.
2. In the Properties View for the note, you can:
   - Edit the text.
   - Change the foreground and background colors.
   - Change the text-only property.

To draw a note link

1. Click **Note Link** on the Palette.
2. In the Diagram Editor, click the source element.

3 Drag the link to the destination element.

4 Drop when the second element is highlighted.

> TIP: You can use the Properties View to view both the client and supplier sides of the link.

Related Concepts

Model Annotation Overview

Related Procedures

Adding a Single Model Element to a Diagram

Creating a Shortcut

Browsing a Diagram with Overview Pane

To open the Overview pane

1 Open a diagram and click **Overview.** The pane expands to show a thumbnail image of the current diagram.

2 Click the shaded area and drag it. This is a convenient way to scroll around the diagram.

3 Resize the **Overview** pane by clicking the upper-left corner of the pane and dragging it.

4 Close the **Overview** pane by clicking the diagram.

Related Procedures

Zooming a Diagram

Changing the Default Diagrams Directory

By default, diagram files are contained within the default design root folder, which is called the Model Folder.

To change the default diagrams directory

1 Right click the project root in the Model Navigator view, and select **Properties**. The **Properties** dialog displays.

2 Choose **Design root path** from the **properties** list on the left.

3 Specify the path in the **Design root path** field, and press OK.

CAUTION: The path name can contain only the folder name the existing design root will be renamed to, not the path to the folder.

Closing a Diagram

To close a diagram

1 Switch to the Diagram Editor.

2 Click the cross icon to close the current view.

TIP: Alternately, choose **File Close** on the main menu, or CTRL+W.

NOTE: Closing a diagram does not remove it from your project.

Related Concepts

Diagram Overview

Creating a Diagram

Diagrams exist within the context of a project. Create or open a project before creating any new diagrams.

To create a new diagram from the Model Navigator

1 In the Model Navigator, right click on a package or the project root.

2 From the context menu, select **New Diagram** and choose the diagram type from the submenu.

To create a new diagram using the Diagram Editor toolbar

1 Click the arrow to the right of the **New Diagram** icon on the diagram editor toolbar.

2 Choose the diagram type from the submenu.

OR

3 Click directly on the **New Diagram** icon. The UML Diagram dialog opens.

4 In the resulting dialog: Select a diagram type from the drop down list. Select the package where the new diagram will be created. Click **Browse** to choose a package. Enter a name for the new diagram.

5 Click **Finish**.

To create a new diagram using the New Diagram Wizard

1 Select **File New Diagram**.

2 Specify the properties for the new diagram as follows:

- **Location:** By default, the new diagram is created in the package selected before the wizard displays.
- **Type:** Use the drop down list to select a diagram type. By default, the Class Diagram is selected.
- **Name:** Use the text field to type a name for the new diagram.

3 Click **Finish**.

To create a class diagram from a package diagram

1 On the package diagram, select classes that you would like to display in a separate class diagram.

2 On the main menu, select **Model Generate Class Diagram**.

NOTE: This action is only available when several classes are selected.

Deleting a Diagram

CAUTION: You cannot delete the default diagram created automatically for a package.

1 In the **Package Explorer**, select the diagram to be deleted.

2 On the context menu, choose **Delete**.

3 Confirm deletion, if required.

Related Procedures

Creating a Diagram

Closing a Diagram

Exporting a Diagram to an Image

To export a diagram to an image

1 Place the focus on the diagram you want export in the Diagram Editor.

2 Choose **File Export** on the main menu. The **Export** wizard opens.

3 In the **Select** page of the wizard, choose **Modeling Image (GIF, JPEG, Bitmap, EMF, SVG**, and click **Next**.

4 In the **Export to Image** page, specify the following settings:

- **Destination file**: enter the fully qualified name of the resulting file, or click Browse and navigate to
- the desired location.
- **Diagrams scope**: click a radio button to select the diagrams or diagram elements to be exported.
- **Format**: select the desired format from the drop-down list.
- **Scale**: enter magnification factor.
- **Export heading**: check the option to save the image together with the diagram title.
- **Open in viewer**: check the option to launch the default image viewer.

5 Click **Next** to preview, or **Finish** to complete export.

Related Concepts

Model Import and Export Overview

Hiding and Showing Model Elements

You can control the visibility of elements on a diagram by using the **Hide** command (available on the context menu for individual diagram elements), and the **Show/Hide** command (available on the diagram context menu).

To hide by using one of the following methods

1 Open the Diagram Editor.

2 Do one of the following:

- Select the element on the diagram, right-click and choose **Hide** on the context menu.
- Select multiple elements on the diagram using CTRL+CLICK or by lassoing, and select Hide from the context menu.
- Right-click the diagram background and choose **Hide/Show** on the context menu. The **Show Hidden** dialog opens, as discussed below.

To show or hide diagram elements using the Show Hidden dialog

1 Right-click the diagram and choose **Show/Hide** on the context menu. The **Show Hidden** dialog opens.

2 Select the element(s) that you wish to hide from the Diagram Elements list.

3 To add elements in the Diagram Elements list to the Hidden Elements list, do one of the following:

- Double-click the element.
- Click the element once and click Add.
- Select multiple elements using CTRL+CLICK and click Add.

4 To remove items from the Hidden Elements list do one of the following:

- Double-click the element.
- Click the element once and click **Remove**.
- Select multiple elements using CTRL+CLICK and click **Remove**.
- To remove all items from the Hidden Elements list, click **Remove All**.

5 Click **OK** to close the dialog.

Related Procedures

Adding a Single Model Element to a Diagram

Hyperlinking Diagrams

Select Hyperlinks from the diagram context menu to create, view, remove, and browse hyperlinks.

Use the following techniques to create a hyperlink

1 Create a hyperlink to an existing diagram or element

2 Create a hyperlink to a new diagram

3 Create a hyperlink to an external URL or file

4 Browse hyperlinks

5 Remove a hyperlink

To create a hyperlink to an existing diagram or element

1 Open an existing diagram or create a new diagram from which to create the hyperlink.

2 Select the element that you want to link to another diagram or element.

3 To link the entire diagram, click the diagram background to deselect all elements.

NOTE: Do not select the actual package in the Model Navigator to create a hyperlink. Rather, expand the package node, and select the desired diagram.

4 Right-click and choose **Hyperlinks Edit**. The **Edit Hyperlinks** dialog window (Selection Manager) opens.

5 Select the Model Elements tab to view the pane containing a tree view of the available project contents.

6 Select the desired diagram or element from the list, and click **Add**.

7 For element selection, expand diagram nodes in the **Model Elements** tab.

8 To remove an element from the selected list, select the element and click **Remove**.

9 Click **OK** to close the dialog and create the link.

To create a hyperlink to a new diagram

1 Open a diagram in the Diagram Editor, or select it in the Model Navigator.

2 On the context menu, choose **Hyperlinks New Diagram**.

3 In the **New Diagram** dialog, select the diagram type, enter the diagram name and click OK.

To create a hyperlink to an external URL or file

1 Open an existing diagram or create a new diagram from which to create the hyperlink.

2 Select the element that you wish to link to the external document.

To link the entire diagram, click the diagram background to deselect all elements.

3 Right-click and choose **Hyperlinks Edit**. The **Edit Hyperlinks** dialog opens.

4 Select the **External Documents** tab to view the Recently Used Documents list which contains a list of previously selected files or URLs.

To add a file to the Recently Used Documents list:

1 Click **Browse**. The **Open file** dialog opens.

2 Navigate to the desired file and click **Open**.

To add a URL to the Recently Used Documents list:

1 Click URL.

2 In the dialog that opens, enter the appropriate URL and click **OK**.

NOTE: You can create a hyperlink to an external document by entering a relative URL path.

To remove an element from the selected list, select the element and click **Remove**.

1. To clear the Recently used Documents list, click **Clear**.
2. Click **OK** to close the dialog and create the link.

To browse hyperlinks

1. To view hyperlinks to a diagram, element or external document, right-click on the diagram background or element, and choose Hyperlinks from the context menu. All hyperlinks created appear under the Hyperlinks submenu. On a diagram, all names of diagram elements that are hyperlinked are displayed in blue font. When you select a link from the submenu, the respective element appears selected in the Diagram Editor.
2. Once you have defined hyperlinks for a selected diagram or element, use the context menus to browse to the linked resources.

   NOTE: Browsing to a linked diagram opens it in the Diagram Editor. or makes it the current diagram if already open.

   Browsing to a linked element causes its parent diagram to open or become current, and the diagram scrolls to the linked element and selects it.

To remove a hyperlink

1. Open the diagram that displays the link you want to remove.
2. Choose **Hyperlinks Edit** from the diagram or element context menu. The **Edit Hyperlinks** dialog opens.
3. In the selected list on the right of the dialog, click the hyperlink that you wish to remove.
4. Click **Remove**.
5. Click **OK** to close the dialog.

   NOTE: To remove a hyperlink from a specific element, select the element first. Then choose **Hyperlinks Edit** on the context menu.

Related Concepts

Model Hyperlinking Overview

Opening a Diagram

You can open diagrams from the Model Navigator, or by using the Diagram Editor toolbar. In this section you will learn how to

- open an existing diagram
- cancel diagram opening process

To open a diagram

1. In the Model Navigator view navigate to the diagram you want to open.
2. Select the diagram node in the tree-view and do one of the following:
   - Double click the diagram
   - Select **Open** from the context menu
   - Select **Open in Active Editor** from the context menu (this replaces the contents of any currently opened diagram)

You can use the Diagram Editor toolbar to open the parent diagram.

Related Procedures

Opening a Parent Diagram

Opening a Parent Diagram

You can open parent diagram from the Diagram Editor toolbar.

To open the parent diagram

1. Click **Open Parent Diagram** to open the parent of the active diagram. If a diagram has no parent, the button is disabled.

Related Procedures

Opening a Diagram

Printing Diagram Elements

To print one or more diagram elements

1. Open the diagram or select the tab in the Diagram Editor that displays the diagram containing the diagram elements you wish to print.
2. Right click the diagram element or multiple elements, and select **Print** from the context menu. The **Print Diagram** dialog displays.
3. Selecting option **Print whole diagram as an image** enables the option **Print diagram as black and white image**. Make your desired selections.
4. At this point you can click **Preview** to see how the selected diagram element or elements look with the current print settings. You can click **Print Options** to set up Print Preferences.
5. Click the drop down arrow to choose the **Preview Scale** factor from the list of available scales to best fit the printed image on the page.
6. Click **Print** to proceed to the standard print dialog where you can select your printer.

Printing Diagrams

To print a diagram or multiple diagrams

1. Open the diagram or select the tab in the Diagram Editor that displays the diagram.
2. Select **File Print** on the main menu. The **Print Diagram** dialog displays.
3. Select the scope for printing.
4. If you check the option **Print whole diagram as an image**, the option **Print diagram as black and white image** becomes enabled. Make your desired selections.
5. Click **Preview** to see how the diagram or diagrams look with the current print settings.
6. Click the drop down arrow to set the **Preview Scale** factor
7. Click **Print Options** to define Print Preferences. You can use the scroll bars to scroll around the diagram or to view other diagrams that were included in the scope.
8. Click **Print** to proceed to the standard print dialog where you can select your printer.

Searching Model Elements

ER/Studio Software Architect enables you to use the search facilities to locate model elements on model diagrams. This function enables you to search the current diagram or all opened diagrams for the specified string in a certain scope. You can create search strings using wildcards and regular expressions. The function is case sensitive.

To find model elements that fall under specified criteria, perform the following steps:

1. On the main menu, choose **Search Model**. Search dialog opens, with the **Model Search** tab selected.
2. Specify the search string in the **Search String** field. Check the following options if necessary:
   - **Case sensitive**: Searches for text that matches uppercase and lowercase characters
   - **Regular expression**: Enables using regular expressions.
3. In the **Search for** section click the desired radio button to select the name or any other property to search for.
4. In the **Scope** section click the desired radio button to select the search area. The possible options are workspace, selected resources, the current project or a predefined working set. To select a working set, click **Choose**. In the **Select Working Set** dialog, choose the desired working set and click **OK**. If there are no available working sets, use **New** to create one.
5. Click **Search.**

Related Procedures

Searching Model with OCL Queries

Searching Model with OCL Queries

You can search for models using OCL queries.

To find model elements that match the specified OCL query

1. On the main menu, choose **Search > Model**. The **Search** dialog displays.
2. Click the **OCL Model Search** tab.
3. Specify the context for your expression in the **Context** field.

   TIP: Use the drop-down list, or Content Assistant. To open the Content Assistant, click on the **Context** field and press CTRL +SPACE. Choose the desired element from the list.

4. In the **Invariant** field type the query expression.
5. In the **Scope** section click the desired radio button to select the search area. The possible options are workspace, selected resources, the current project or a predefined working set. To select a working set, click **Choose**. In the **Select Working Set** dialog, choose the desired working set and click **OK**. If there are no available working sets, use **New** to create one.
6. Click **Search**.

A tree with the list of matching elements will be opened. You can navigate to the corresponding diagram from this view by double-clicking the selected element.

Related Concepts

OCL Support

Using a Class Diagram as a View

Class diagrams can also be used to create subviews of the project.

To use a class diagrams as a view:

1. Create a new class diagram.
2. Create shortcuts to the original diagram to easily and quickly build subset views for easier management.

   TIP: Using this feature, you can create views of distributed classes into one diagram, automatically displaying any relationships that the gathered classes may have with each other.

Zooming a Diagram

Use the zooming commands of the main menu, or toolbar icons, to obtain the required magnification in the Diagram Editor.

To specify the magnification in the Diagram Editor

1. On the main menu, choose **Diagram**.
2. Choose one of the available zooming commands on the menu: **Zoom In, Zoom Out, Fit to Window, Actual Size**.

   TIP: Alternately, use the diagram Palette or keyboard shortcuts.

Populating Diagrams

This topic provides How-To information about creating node elements, links and members in all types of diagrams.

Adding a Member to a Container	How to add a member to a container.
Adding a Single Model Element to a Diagram	How to create a single model element.
Adding Multiple Elements to a Diagram	How to create multiple elements.
Creating a Link with Bending Points	How to create a link with bending points.
Creating a Shortcut	How to create a shortcut.
Creating a Simple Link	How to create a simple link.
Creating an Inner Classifier	How to create an inner classifier.

Adding a Member to a Container

You can add members to class diagram elements (containers) by using the respective context menu for the diagram element in the Diagram Editor or model Navigator or available shortcut keys to add members to a container element.

To add a member to a container

1 Right-click the desired container element.

2 On the context menu, choose **New <Member type>**, where the Member type corresponds to the target container.

TIP: You can also use keyboard shortcuts to add fields and methods to a container allowing such members. Click CTRL+W (for fields) and CTRL+M (for methods, functions).

3 You can edit the member using the in-place editor, or the Properties View.

Related Procedures

Adding a Single Model Element to a Diagram

Adding a Single Model Element to a Diagram

You can create a single node element using the diagram Palette, or the **New** command of the diagram context menu.

To create a single model element

1 Open a target diagram in the Diagram Editor.

2 On the Palette, click the icon for the element you want to place on the diagram. The icon stays down.

TIP: Icons are identified with tooltips.

3 Click the diagram background in the place where you want to create the new element. This creates the new element and activates the in-place editor for its name.

> **TIP:** Alternately, you can right-click the diagram background in the Diagram Editor, or the diagram node in the Model Navigator, and choose **New** on the context menu. The submenu displays all of the basic elements that can be added to the current diagram, and the **Shortcuts** command.

Related Procedures

Adding Multiple Elements to a Diagram

Creating a Simple Link

Adding Multiple Elements to a Diagram

You can place several elements of the same type on a diagram without returning to the Palette or by using the diagram context menu. Each element will have a default name that can be edited with the in-place editor or in the Properties View.

To create multiple elements

1 Holding down the CTRL key, click the Palette icon for the element you want to create (the icon stays down). Continue to hold down the CTRL key. The cursor changes colors and has a plus symbol located below the cursor (✚).

2 Click the desired location on the diagram background. The new element is placed on the diagram at the point where you click.

3 Click the next location on the diagram background. The next new element is placed on the diagram.

4 Repeat the previous step until you have the desired number of elements of that type.

5 To stop multiple element creation, click the right mouse button and the cursor changes.

Related Procedures

Adding a Single Model Element to a Diagram

Creating a Simple Link

Creating a Link with Bending Points

If your diagram is densely populated, you can draw bent links between the source and target elements to avoid other elements that are in the way.

To create a link with bending points

1 Click the link icon on the Palette.

2 Click the source element.

3 Drag the link line, clicking the diagram background each time you want to create a section of the link. Sections on a link lie between two blue bullets. The bullets display whenever you select the link on the diagram.

> **TIP:** You can cancel each section of a link pressing the BACKSPACE key.

4 Click the destination element to terminate the link.

TIP: Once you have created a link, you can add bending points to it. Click on the desired point of the link, and drag ше to the desired position.

Related Procedures

Creating a Simple Link

Creating a Shortcut

You can create a shortcut to a model element from the current project or from the projects connected by cross projects references, by using three methods:

- By choosing **New > Shortcut** on the Diagram Editor context menu.
- By dragging and dropping a shortcut from the Model Navigator
- By choosing **Add as Shortcut** on the Model Navigator context menu

To create a shortcut by using the Shortcuts dialog window

1 Right-click the diagram background.

2 Choose **New > Insert > Shortcuts** on the context menu.

TIP: Use CTRL+SHIFT+N keyboard shortcut

3 In the **Shortcuts** dialog window, choose the required element from the tree view of available contents.

NOTE: If the project has cross-project references to the other projects in the workspace, the contents of these projects is available for being added as shortcut.

4 Click **Add** to place the selected element to the list of the existing or ready to add elements.

5 When the list of ready to add elements is complete, click **OK**.

To create a shortcut by using drag-and-drop

1 Select the element in the Model Navigator.

2 Drag-and-drop the element onto the diagram.

To create a shortcut by using the Model Navigator context menu

1 Open the diagram where the shortcut will be added.

2 In the Model Navigator, select the element to be added to the current diagram as a shortcut.

3 Right-click the element in the Model Navigator, and choose **Add as Shortcut** on the context menu.

Related Concepts

Model Shortcut Overview

Related Procedures

Establishing Cross-Project References

Adding a Single Model Element to a Diagram

Creating a Simple Link

You can create a link to another node, or a shortcut of an element of the same or another project (these projects must be of the same UML version).

To create a simple link between two nodes

1. On the diagram Palette, click the icon for the type of link you want to draw in the diagram. The icon stays highlighted.
2. Click the source element.
3. Drag to the destination element and drop when the target element is highlighted.

Related Procedures

Creating a Link with Bending Points

Creating Model Element by Pattern

Creating an Inner Classifier

You can add inner classifiers to class diagram elements (containers) using the respective context menu for the diagram element in the Diagram Editor or Model Navigator. You can also select a classifier in the Palette and click the container element in the Diagram Editor to add the inner classifier to the container element.

TIP: You can use drag-and-drop or clipboard operations to remove an inner classifier from the container element.

To create an inner classifier using the context menu

1. Right-click the container element.
2. Choose **Add <Inner classifier type>**

To create an inner classifier using the clipboard operations

1. Use the clipboard operations to either cut or copy an existing classifier.
2. Select the container element.
3. Use the clipboard operations to paste the selected classifier into the container element.

To create an inner classifier using drag-and-drop:

1. Select an existing classifier in the Diagram Editor.
2. Drag-and-drop it onto an existing container in the Diagram Editor. A border highlights the location that is recognized as a valid destination for the inner classifier.

Related Procedures

Adding a Single Model Element to a Diagram

Editing Diagrams

Aligning Model Elements	How to align model elements.
Assigning a Stereotype to an Element	How to specify and define an element stereotype.
Changing Type of an Association Link	How to change type of an Association link
Copying and Pasting Model Elements	How to copy and paste model elements.
Deleting Elements	How to delete an element from diagram.
Laying Out a Diagram Automatically	How to lay out a diagram automatically.
Laying out a Diagram for Printing	How to use optimized layout for printing.
Moving Model Elements	How to move model elements.
Renaming a Diagram	How to rename a diagram.
Rerouting a Link	How to reroute a link.
Resizing Model Elements	How to change a size of a model element.
Selecting Model Elements	How to select model elements.
Working with Rulers Guides and Grid	How to use ruler guides and grids in your diagram.

Aligning Model Elements

You can automatically rearrange all or selected model elements on a diagram according the desired order. The following options are available:

- Top
- Bottom
- Right
- Left
- Center Horizontally
- Center Vertically

To align model elements on a diagram

1 Select several nodes or inner classifiers on a diagram.

2 On the main menu choose **Diagram > Align > <option>**.

TIP: Alternately, use the diagram Palette icons ( ).

Related Procedures

Laying Out a Diagram Automatically

Assigning a Stereotype to an Element

You can assign a stereotype in the diagram by using the in-place editor, or the Properties View.

To assign a stereotype by using the in-place editor

1. Double-click the stereotype name to activate the in-place editor.
2. Enter the new name.
3. Press ENTER.

To assign a stereotype by using the Properties View

1. Select an element on your diagram.
2. In the Properties View, select the **Stereotype** field.
3. Click browse.
4. In the Edit Property Values dialog click **Add** and enter required stereotype.

   NOTE: You can also select a stereotype from the drop-down menu. When you click Add, a drop-down arrow appears in the blank field.

Changing Type of an Association Link

Use the following techniques to change the type of an Association link

- Set the link type by using the Properties View

To set the Association link type by using the Properties View

1. Select a link on the diagram.
2. Open the Properties View.
3. In the Properties View, select the **Associates Type** field.
4. Click the drop-down arrow and select the appropriate property from the list. Your available choices are association, aggregation, or composition.

Related Procedures

Creating a Simple Link

Copying and Pasting Model Elements

The move and copy operations are performed by drag-and-drop, context menu commands, or keyboard shortcut keys.

NOTE: You can move or copy an entire diagram. In this case, all elements addressed on this diagram are not copied, and a new diagram contains shortcuts to these elements.

To copy and paste one or more elements

1. Select the desired element or elements.
2. To copy the selection, do any of the following:
   - Right-click and choose **Copy** on the context menu
   - Press CTRL+C on the keyboard

3 To paste the selection, do any of the following:

- Right-click the target location and choose **Paste Element** on the context menu
- Select the target location and press CTRL+V

Related Procedures

Adding a Single Model Element to a Diagram

Keyboard Shortcuts

Deleting Elements

All elements shown on diagrams are shortcuts to actual model elements. When deleting an element on a diagram you have the option to delete either the shortcut from view or delete the element from the model (except classes on synchronized package diagrams). This behavior is configured in the Modeling Preferences.

To delete an element

1 Select the element on diagram.

2 Choose **Delete** on the context menu of the element.

TIP: Alternately, click the DELETE key.

3 Confirm deletion, if this behavior is selected in the Modeling Preferences.

To delete an element from View

1 Select the element on diagram.

2 Choose **Delete from View** on the context menu of the element.

3 Confirm deletion, if this behavior is selected in the Modeling Preferences.

Related Procedures

Adding a Single Model Element to a Diagram

Laying Out a Diagram Automatically

To lay out a diagram by using one of the algorithms

1 Right-click the diagram background.

2 On the context menu, select **Layout**, and choose a command from the submenu.

There are several Layout commands on the **Layout** submenu:

- **Layout All**: Sets the layout of all elements according to the layout algorithm defined for the current diagram.
- **Layout for Printing**: Sets the layout of all elements using the algorithm, regardless of the option selected on any level.
- **Layout All and Optimize Sizes**: Set the layout of all elements and enlarges or shrinks all elements on the diagram to the optimal size.

To set up the diagram layout

1. On the main menu choose **Window > Preferences > Modeling > Layout**.
2. Select the desired layout for links in the **Links layout** section (direct or rectilinear)
3. Choose the desired algorithm from the **Algorithm** drop-down list, and specify the algorithm-specific options (if any.)
4. To enable layout of the inner substructure in diagrams, check the **Recursive** option As a result, you can observe results of layout tuning when apply one of the *Layout* commands to the diagram.

The context menu available in the Diagram Editor provides access to the automated layout optimization features.

Related Procedures

Aligning Model Elements

Laying out a Diagram for Printing

There is an automated layout optimization for printing diagrams. Using automated layout for printing ensures that all diagram elements fall within page borders. Invoke automated layout immediately before printing a diagram.

To lay out you diagram elements for printing

1. Right-click the diagram background
2. On the context menu, choose **Layout > Layout All for Printing**.

NOTE: You can revert to your manual layout after a Layout and optimize operation by using Undo. For example, you might invoke Layout and optimize, print the diagram, then call Undo to restore your manual layout.

Moving Model Elements

Create your own layout by selecting and moving single or multiple diagram elements.

You can:

- Select a single element and drag it to a new position.
- Select multiple elements and change their location.
- Manually reroute links.
- Use Cut and Paste operations.

NOTE: If you drag an element outside the borders of the Diagram Editor, the diagram automatically scrolls to follow the dragging.

TIP: Manual layouts are saved when you close a diagram or project and restored the next time you open it. Manual layouts are not preserved when you run one of the auto-layout commands (**Layout All** or **Layout All and Optimize Size**).

To move one or more elements

1 Select the element or elements to be moved.

2 Drag-and-drop the selection to the target location.

TIP: If you have selected several model elements in certain diagrams (State Machine, Use Case, Activity or Business Process), it is recommended to use the heading area of one of the selected elements to drag the entire group. The attempt to drag by an internal area of an element results in switching the Diagram Editor to the Select mode, and loosing the current selection. However, if you still hold the mouse button down and press ESC, the new selection is cancelled and the current selection is preserved.

Related Procedures

Selecting Model Elements

Renaming a Diagram

CAUTION: The automatically created package diagram cannot be renamed.

To rename a diagram

1 In the Properties View, double-click the diagram name to initiate the inline editor.

2 Enter a new name.

3 Press **Enter**.

Alternately

1 Select the diagram in the Model Navigator.

2 Press F2 or right-click and choose **Rename** on the context menu.

3 Enter a new name.

4 Press **Enter.**

Related Procedures

Creating a Diagram

Rerouting a Link

To reroute a link

1 Select a link

2 Click on the bullet at the end of the link you want to reroute.

3 While holding down the cursor, drag and drop the client or supplier end of the link to the desired destination object.

Related Concepts

Model Element Overview

Related Procedures

Laying Out a Diagram Automatically

Resizing Model Elements

You can resize diagram elements automatically or manually. When new items are added to an element that has never been manually resized, the element automatically grows to enclose the new items.

To resize an element manually

1 Click an element. The selected element is highlighted with bullets.

2 Drag one of the bullets in the desired direction.

When the element contents change, for example, when members are added or deleted, and the element size is too small to display all members, scroll bars are displayed to the right of compartments.

To optimize a node element size

1 Right-click an element.

2 Choose **Optimize Size**.

To optimize the elements on an entire diagram

1 Right-click the diagram background.

2 Choose **Layout All and Optimize Size**.

Related Procedures

Laying Out a Diagram Automatically

Selecting Model Elements

Most manipulations with diagram elements and links involve dragging the mouse or executing context menu commands on the selected elements.

In this section you will learn how to

- Select one or more elements
- Cancel selection

To select an element

1 Open a diagram in the Diagram Editor.

2 On the diagram Palette, click **Select** ( ).

3 In the Diagram Editor, click any element or a member to select it.

TIP: To select multiple elements, do one of the following:

- Hold down the CTRL key and click each element individually
- Click the background and drag a lasso around an area to select all the elements it contains.
- Press CTRL+A to select all elements on a diagram.

To cancel selection

1 Press ESC.

Related Procedures

Aligning Model Elements

Working with Rulers, Guide, and Grid

ER/Studio Software Architect provides means to use ruler guides in the **Diagram Editor** for aligning purposes.

To add or remove a ruler guide

1 To add a ruler guide, click either the vertical or horizontal ruler. The guide appears at the click point.

NOTE: Alternately, right click a ruler and choose **Create Guide**. The guide is created at the zero point of the ruler.

2 To remove a ruler guide, click a guide on the ruler, drag it out of the ruler space until your pointer becomes a normal select shape, and release your mouse button.

Once you created several guides, you can connect your elements to ruler guides. If you connect several elements to a guide, all elements move when you move the guide.

Aligning elements with ruler guides

1 Move or resize an element on the diagram to place one side of the element close to a rule guide.

2 Drop the element when the guide highlights as a red line.

3 Repeat the previous steps to connect other elements to the guide.

4 Move the guide. Notice how all connected elements move with connected ruler guide.

NOTE: To disconnect an element from the guide, simply move the element from the guide. You can optionally display or hide a design grid on the diagram background and have elements “snap” to the nearest grid coordinate when you place or move them. Grid options are configured in the **Diagram** page of the **Preferences** dialog.

To show grid

1 Open **Windows > Preferences** dialog.

2 Choose **Modeling > Diagram**.

3 In the **Rulers, Grid, and Snapping** group adjust the options.

NOTE: Grid display and snap are enabled by default.

Projects

This section provides how-to information on using projects.

Changing the Default Diagrams Directory	
Choosing a Perspective	How to choose a perspective.
Creating a Project	How to create a project for Eclipse.
Enabling UML Profiles	Describes how to enable profile support for a project.
Establishing cross-project references	Describes how to establish cross-project references between the projects located in the same workspace.
Exporting a Project to XMI Format	How to export a project to XMI format.
Exporting a Project to XMI Format Using Command Line	How to export a project to XMI format using command line.
Importing a Project in XMI Format	How to import XMI data.
Navigating between the Tree View, abd Diagram	How to synchronize the Tree View, and Diagram
Troubleshooting a Model	How to troubleshoot a model.
Using Example Projects	How to use sample projects for Eclipse.
Working with a Package	How to work with a package.
XMI Export and Import of the Models with Cross-Project References	You can import and export multi-root projects using XMI. Note that XMI import and export is implemented differently for UML 2.0 projects.

Changing the Default Diagrams Directory

By default, diagram files are contained within the default design root folder, which is called the Model Folder.

To change the default diagrams directory

1. Right click the project root in the Model Navigator, or Navigator view, and select **Properties**. The **Properties** dialog displays.
2. Choose **Design root path** from the **properties** list on the left.
3. Specify the path in the **Design root path** field, and press OK.

CAUTION: The path name can contain only the folder name the existing design root will be renamed to, not the path to the folder.

Choosing a Perspective

ER/Studio Software Architect changes the user interface according to how you want to work by providing several perspectives. By default ER/Studio Software Architect starts with Modeling perspective.

To choose a Perspective

1. On the main menu, choose **Window > Open Perspective >Other**. The **Open Perspective** dialog appears. The options are:
   - CVS Repository Exploring
   - Modeling (default)
   - Resource
   - Team Synchronizing
2. Select one of the Perspectives from the list, and click **OK**.

   After you select a perspective, The interface is automatically customized to provide ready access to only the relevant elements of the interface, and to show only the information in the model that best supports the chosen perspective. Interface elements and/or model information that are not generally relevant to the perspective are hidden. You can still access hidden information by changing the relevant configuration options and restoring hidden panes manually, but you may find it easier to just switch perspectives.

Creating a Project

Several projects are provided that you can work with. The projects are created in the same manner. While creating a project you will specify different options depending on the type of project.

To create a Project

1. Select **File > New > Other** on the main menu. The **New Project** wizard displays.
2. Expand the **Modeling** node in the tree view list, and select the type of project you want to create and click **Next**.
3. Follow the wizard to specify necessary options for a new project and click **Finish** to complete the wizard.

Related Concepts

Project Overview

Enabling UML Profiles

There are several ways to enable UML profiles for projects.

To enable UML profiles support while creating a project

1. On the main menu choose **File > New > UML 2.0 Project**. The **New Project** wizard displays.
2. Enter the project and click **Next**. You also have the option to use the default location for the project and add the project to an existing working set.
3. On the Modeling Settings dialog you can use the default setting for which diagram to open or select a new diagram from the drop-down list.

   NOTE: If you choose to change the default setting, you can name the diagram that will be opened.
4. You also have the option to create the design elements in separate files. When you place elements on the diagram, each element appears as its own folder.
5. Click Next and the Profiles dialog appears. All available profiles are listed. If you do not want to use any of these profiles, click **Finish**.

To enable UML profiles support for existing projects

1. In the Model Navigator, right click the root project folder, and select **Properties** on the context menu. The **Properties for <project>** dialog displays.
2. From the list on the left, select **UML Profiles**.
3. Select any of the UML profiles that you want to enable. More than one can be activated.
4. Click **OK**.

NOTE: You can also access the **Properties for <project>** dialog through the Navigator view.

To specify the default set of UML profiles enabled for all new workspace projects

1. Choose **Window > Preferences** on the main menu.
2. In the left pane of the **Preferences** dialog, expand the **Modeling** node.
3. Select the **UML Profiles** node.
4. Select the profiles you want to enable for UML 2.0 projects.

NOTE: The selected UML profiles are automatically enabled for projects created after you changed profile preferences. Profiles support of existing projects is not changed.

Establishing Cross-Project References

You can establish references between the projects of similar type within your workspace. This capability is enabled in the **Model Path** page of the **Project Properties** dialog. When cross-project referencing is enabled, the imported project is included in the project in question as read-only root and becomes visible in the selection dialogs. WHen you do this action, any changes in the referenced projects are propagated across the target project as well. For example, renaming elements in the referenced project is immediately reflected in the target project.

To enable cross-project references

1. Select the desired project in the Model Navigator.
2. Right-click on the project node and choose **Properties** on the context menu and the dialog **Properties for <Project Name>** opens
3. Select **Model Path**.
4. In the **Model Path** page, click **Add Project** and the **Select Projects to Import** dialog appears, displaying the list of available projects in the workspace. Only the projects of similar types are included in the list.
5. Check the desired projects in the field **Available Projects in the Workspace**, and click **OK**.
6. Click **OK** to confirm your settings and close the **Properties for** dialog.

CAUTION: It is strongly recommended to avoid establishing recurrent references.

Related Concepts

Project Overview

Related Procedures

Creating a Project

Exporting a Project to XMI Format

You can export projects or sections of projects created in ER/Studio Software Architect for use by other applications/languages using XMI. ER/Studio Software Architect supports several XMI formats. The availability of formats depends on the types of projects currently opened.

To export a project to XMI format

1. Select **File > Export** on the main menu and the **Export** dialog opens.
2. Expand **Modeling**, choose **XMI File** and click **Next**.
3. In the **Export Project to XMI FIle** dialog specify the following:
   - Select the project to export.
   - Select the XML and UML version you want the file to support under **Select XMI Type**. UML 2.0 projects can be exported to XMI for UML 2.0 only.
   - Select an appropriate XMI Encoding requirement in the **XMI Encoding** list.
   - Specify the destination in the **Select the export destination**. You can include the path as well as the name of the file that will be created, or you can accept the default. The name consists of <project_folder>\out\xmi\<project_name>.uml2.
4. Click **Next** and the **Run Audits on Exported Project** dialog appears. You can choose to run a selected audit or run them all.
5. Click **Finish** to generate the XMI file.

A dialog displays indicating that the XMI export is completed. If there are any warnings produced during XMI export, the **XMI Export** dialog notifies you to refer to the **Task** view. To open the **Task** view, from the main menu, select **Window > Show View Other > General > Tasks**.

NOTE: For UML 2.0 projects with applied profiles or projects that contain any stereotypes or primitive types, during the export process the following files are created in addition to the model .uml2 file:<model name>. profile.uml2 – for stereotypes and primitive types<profile name>.profile.uml2 – for applied profiles

Related Concepts

Model Import and Export Overview

Related Procedures

XMI Export and Import of the Models with Cross-Project References

Exporting a Project to XMI Format Using Command Line

A way is provided for command line XMI export of UML 2.0. Use XMIExport.cmd on the Windows platform.

To export a project to XMI format under Windows:

1. Locate the XMIExport.cmd file in the installation folder.
2. Run the XMIExport.cmd file with necessary parameters.

NOTE: For usage instructions and command-line parameters, run XMIExport.cmd -help.

Related Concepts

Model Import and Export Overview

Importing a Project in XMI Format

You can import projects or sections of projects created in other modeling tools and saved in XMI format.

To import a project from XMI file

1. Select **File > Import** on the main menu and the **Import** dialog opens.
2. Expand the Modeling node, select **XMI File** and click **Next**.
3. In the **Import Project from XMI File** dialog, specify the following:
   - **Select source file**: Full path to .xml, .xmi, or .uml2 file you want to import.
   - **Select destination project**: Select the ER/Studio Software Architect project to which your XMI data will be imported.
4. Click **Finish**.

   NOTE: .uml2 file can be imported to UML 2.0 projects.

After you are notified that the import process is complete, you can view the results in the **Model Navigator**.

NOTE: When importing UML 2.0 models with profile files related to the model, for the models originally exported from ER/Studio Software Architect for Eclipse, select model .uml2 file as a source and make sure that all the profile files are located in the same folder with the model file.

If there are any warnings produced during XMI import, the XMI Import dialog notifies you to refer to the **Task** view. To open the Task view, from the main menu, select **Window > Show View Other > General > Tasks**.

Related Concepts

Model Import and Export Overview

Related Procedures

XMI Export and Import of the Models with Cross-Project References

Navigating Between the Tree View and Diagram

Constant synchronization is provided between different aspects of your project:

- Model hierarchy, presented in the tree view (**Model Navigator View**)
- Model graphical representation in the **Diagram Editor**

TIP: You can also use the Refresh function of the **Model Tree View** to update the entire model, and the Refresh function of the **Diagram Editor**.

You can navigate between the Model Tree and Diagram Editor in the following directions:

1. Navigate to the **Diagram Editor** from the **Model Tree View**.
2. Navigate to a model element from the **Model Tree View** to the **Diagram Editor**
3. Navigate from the **Diagram Editor** to the **Model Tree View**
4. Navigate from a lifeline to its classifier in the **Model Navigator View** or a Class diagram

To navigate to the Diagram Editor from the Model Navigator View:

1 In the **Model Navigator View**, right-click the diagram node.

2 Choose **Select on Diagram.**

NOTE: If the element exists in more than one diagram, a dialog opens displaying the diagram. Select the desire one and click **OK**.

Alternatively, double-click the diagram node in the **Model Navigator View**.

To navigate to a model element from the Model Navigator View to the Diagram Editor:

1 Right-click a model element in the **Model Navigator View**.

2 Choose **Select on Diagram** on the context menu.

NOTE: Click Link with Editor ( ) on the Model Navigator toolbar and elements selected in Model Navigator will be automatically selected on diagrams.

To navigate from the Diagram Editor to the Model Navigator View:

1 Right-click the selected element or diagram background in the **Diagram Editor**.

2 Choose **Select in Model Tree** on the context menu.

To navigate from a lifeline to its classifier in the Model Navigator View or a Class diagram:

1 Right-click the selected lifeline on a UML 2.0 Sequence diagram in the **Diagram Editor**.

2 Choose **Select Type in Model Navigator View** to navigate to the classifier in the **Model Navigator View**,

OR

Choose **Select Type On Diagram** to navigate to the classifier on a Class diagram in the **Diagram Editor**.

Using Example Projects

A set of predefined sample projects are provided. To use them, you have to create in your workspace a new project on the base of an example.

To use a Example Project

1 Select **File > New > Other** on the main menu. The **New Project** wizard displays.

2 Expand the **Examples** node in the tree view list, and select the desired project. Click **Next**.

3 Follow the wizard to specify the necessary options for a new project and click **Finish** to complete the wizard.

Related Concepts

Project Overview

Working with a Package

By default, a package element on diagram displays the package contents.

Use the following techniques for a package:

- Open a package
- Modify package contents
- Delete a package
- Rename a package
- Move a package

To open a package

1. Select package in the Diagram Editor or in the Model Navigator.
2. Choose the **Open** or **Open in Active Editor** command on the package context menu.

   TIP: Alternately, double-click the package element on diagram.

To modify package contents

1. To add an element, choose New <element> on the package context menu.

   TIP: You can use the context menu of a nested element in a package to add its fields and subelements directly, without opening it in diagram.

2. To delete an element from a package, press DELETE.

To delete a package

1. Select package in the Diagram Editor or in the Model Navigator.
2. Choose **Delete** on its context menu.

CAUTION: Deleting a package also deletes all of its contents.

To rename a package

1. Select package in the Diagram Editor or in the Model Navigator.
2. To rename a package, including changing its name in all of its source files, do one of the following:
   - Choose **Rename** on the context menu of a package in the Diagram Editor or in the Model Navigator.
   - Press F2 to invoke the in-place editor for the package element in the Diagram Editor or in the Model Navigator.
   - Edit the Name field in the Properties View

To move a package

1. Select package in the Diagram Editor or in the Model Navigator.
2. Drag the package and drop it to the target location.

CAUTION: It is not recommended to undo move operations for packages.

Related Concepts

Package Overview

XMI Export and Import of the Models with Cross-Project References

You can import and export multi-root projects using XMI.

- **UML 2.0 projects:** When a project that contains cross-project references is exported to XMI file, special files *.imports.uml2 are created for each referenced root. The exported XMI file contains references to these files. When an XMI file is imported, the resulting project contains the main model only. If the referenced roots still exist in the workspace, then the resulting UML 2.0 model recognizes them. References to the elements from these roots may be resolved only if their UINs have not been changed since export. Note that UIN may change when an element is moved.

Related Concepts

Interoperability and Migration

Model Import and Export Overview

Related Procedures

Importing a Project in XMI Format

Exporting a Project to XMI Format

Profiles

You can model diagrams using several preinstalled profiles as well as profiles created with Profile Definition projects.

A Typical User Scenario of Working With Profiles	General information how to create a profile definition.
Adding Attributes to Stereotypes	How to add attributes (tagged values) to stereotypes, and define inspector grouping.
Adding Shortcuts to Metaclasses	How to create shortcuts to metaclasses.
Applying Profiles	How to apply profile.
Creating Palette Contributions	How to create a palette contribution in the profile.
Creating Profile-Specific Constraints	How to create profile-specific constraints.
Creating Stereotypes	How to create stereotypes in your profile.
Defining Profile Properties	How to specify profile definition properties
Deploying Profiles	How to deploy a profile.
Enabling UML Profiles	Describes how to enable profile support for a project.
Exporting and Importing Profiles	How to import and export profile plugins.
Opening Profile Definition	How to view and modify definition of the custom profiles and profiles that come bundled with ER/Studio Software Architect
Setting Viewmap Properties for Stereotypes	How to specify visual representation of the elements in the created profile.
Uninstalling Profiles	How to uninstall a profile and correctly remove it from the platform

Verifying a Model Against Profile Constraints	How to verify a model against the specified constraints, provided that a profiles is applied to this model.
Working with Required Stereotypes	How to define required stereotypes and filter their manifestation in diagrams.

A Typical User Scenario of Working With Profiles

To create, deploy and apply a new profile definition, perform the following general steps

1. Create a Profile Definition project. While creating a Profile Definition project, specify a UML version that the profile is targeted at (UML 2.0 by default). Metaclasses referenced in a profile must be taken from the corresponding target UML metamodel.

   Creating a Project

2. Open the default class diagram of your Profile Definition project and edit the profile properties:

   Defining Profile Properties

3. Create Stereotypes:

   Creating Stereotypes

4. Edit Stereotypes (edit properties, create shortcuts to metaclasses):

   Adding Shortcuts to Metaclasses

5. Add attributes (tagged values) to the stereotypes:

   Adding Attributes to Stereotypes

6. Define view properties for the stereotypes:

   Setting Viewmap Properties for Stereotypes

7. Create Palette Contributions; fill them with the contributed stereotypes or pure metaclasses.

   Creating Palette contributions

8. Deploy profile:

   Deploying Profiles

9. Apply profile:

   Applying Profiles

Related Concepts

UML Profiles Basics

Creating a Project

Several projects are provided that you can work with. The projects are created in the same manner. While creating a project you will specify different options depending on the type of project.

To create a Project

1 Select **File >New > Project** on the main menu. The **New Project** wizard displays.

2 Expand the **Modeling** node in the tree view list, and select the type of project you want to create. Click **Next**.

3 Follow the wizard to specify necessary options for a new project and click **Finish** to complete the wizard.

Related Concepts

Project Overview

Defining Profile Properties

When a Profile Definition project is created, you can specify or edit profile properties that are accessible via the default package diagram of the Profile Definition project. These properties are:

- Textual profile description
- Namespace, which identifies the profile.

To specify profile definition properties

1 Open the default package diagram of the Profile Definition project.

2 On the context menu of the diagram, choose **Properties**. The Properties View opens.

3 In the Properties View, select the **Profile Definition** tab.

4 Click the **description** field and enter the description text. Optionally, click **Edit**.

5 Click the **namespace** field and enter a valid string.

Creating Stereotypes

To create a Stereotype

1 Using the **Profile Definition** palette, add a **Stereotype** node to the diagram background.

2 In the Properties View, choose **Profile Definition** node.

3 In the **Extended metaclass** field, click **Edit.** In the dialog that opens, select the desired metaclasses from the **Model Elements** pane. Use **Add** and **Remove** to make up a list of extended metaclasses. Click **OK** when ready

4 If necessary, specify the required stereotype property

Working with Required Stereotypes

5 Define view properties:

TIP: View properties are not available for the stereotypes that extend multiple metaclasses.

Setting Viewmap Properties for Stereotypes

6 Add attributes (tagged values) to the stereotype

Adding Attributes to Stereotypes

7 Using **Contribution link**, connect the Stereotype to the desired Palette Contribution.

Related Procedures

Working with Required Stereotypes

Setting Viewmap Properties for Stereotypes

Adding Attributes to Stereotypes

Adding Shortcuts to Metaclasses

When creating your profile definition project you can use shortcuts to metatypes from metamodel root. Shortcuts to metaclasses can be created in several ways some of which are similar to adding any other shortcut to your diagram.

To use the Shortcuts dialog

1. Right-click the diagram background and select **Shortcuts New**. The **Shortcuts** dialog displays.
2. Expand the metamodels node, choose the desired metaclass, and click **Add**.
3. Use **Add** and **Remove** to make up a list of extended metaclasses.
4. Click OK when ready.

To use cut, copy, and paste operations

1. Cut, copy, and drag a metaclass in the Model Navigator
2. Paste and drop it to any of your diagram except packages.

To create a shortcut on package diagram using drag-and-drop operation: press and hold CTRL and SHIFT while dragging an element from the Model Navigator to your package.

Adding Attributes to Stereotypes

In this section you will learn how to add attributes to stereotypes, define attribute properties, groupings and descriptions. After applying a profile, stereotype attributes become visible in the Properties View of the elements with this stereotype.

You can add attributes to stereotypes in one of the following ways:

- Using the Properties View
- Using outgoing association links

To add attributes (tagged values) to a stereotype using the Properties View

1. Right-click on the selected stereotype in profile definition diagram, and choose **New > Attribute** on the context menu. Add as many attributes as required.
2. Select an attribute in the stereotype node. Its properties are displayed in the Properties View
3. In the **Properties** tab:
   - Choose the **name** field and enter attribute name.
   - Click the **type** field and specify the valid type using the combo box for primitive types, or selection manager dialog for the enumerations and metaclasses.

   NOTE: invalid types are ignored during profile deployment.

4 In the **Profile Definition** tab, click the **inspector group** field, and select the inspector group from the list of existing groups, or create new one. After applying the profile, the attribute in question will display in a separate tab (group) of the Properties View.

5 In the **Description** tab, choose the Edit tab and enter textual description. After applying the profile, this description will show up as a tooltip in the tab (group) of the Properties View. **Result:** The tagged value name is equal to the attribute name. Multiplicity of the tagged value is equal to attribute's multiplicity.

To add attributes (tagged values) to a stereotype using outgoing association links

1 On the profile definition diagram, create shortcuts to certain types (Primitive types, Enumerations or Metaclasses). Note that invalid types are ignored during profile deployment.

2 In the Class Diagram group of the Tool Palette, select Association link and draw it from the stereotype in question to a shortcut to the desired type.

3 Select the created association link. Its properties are displayed in the Properties View

4 In the **Supplier** tab, specify the following properties: supplier multiplicity, supplier role

5 Specify the inspector group and description, as described in the steps 4 and 5 of the previous procedure, if necessary. **Result:** If supplier role is specified, it is used as the tagged value name. If the supplier role is not specified, then the link name or default name is used as an tagged value name. Multiplicity of the tagged value is equal to the supplier multiplicity. If the upper of the supplier multiplicity is greater than 1, the attribute is treated as multivalued.

NOTE: When the attribute of a metaclass type or the association to metaclass shortcut is added to some stereotype, this attribute or association get the viewmap and icon properties in the Profile Definition section of the Properties View. These properties allow to select the viewmap and icon for the link that appears when this profile-specific property is set in the target project for the element with this stereotype.

Setting Viewmap Properties for Stereotypes

You can specify visual representation of the elements in the profile you create. Values in the icon and viewmap properties affect the way the elements are displayed on your diagram.

NOTE: Viewmap property values are different for stereotypes that extend links.

To set viewmap and icon properties

1 Select a stereotype rectangle. Properties view displays the properties of the selected stereotype.

2 In the **Profile Definition** node of the **Properties** view, choose extended metaclass field and click **Edit**.

3 In the selection manager dialog, navigate to the desired metaclass and click **Add.** You can select several extended metaclasses. Click **OK** when ready.

TIP: Viewmap property is available for the stereotypes that extend one metaclass only.

4 Once a value for the extended metaclass property is provided, viewmap and icon properties are displayed.

5 Select the viewmap field and click **Edit**. This opens the **Viewmap Editor** dialog. Note that viewmap depends on the type of the extended metaclass: there are different sets of viewmap properties for the links and nodes.

6 Specify the viewmap properties: For a stereotype that extends a node: Select **color** to specify color for the element, or select **svg** to browse for an svg file. If you choose **svg** you can also specify the figure from those described in the selected svg file and specify whether the graphical node with selected .svg figure will be resizable. For a stereotype that extends a link: specify values for Source decoration, Target decoration, Foreground, Background, and Line style options.

7 Click **OK** to save the changes and close the dialog.

Creating Palette contributions

In this section you will learn how to create a new tool, assign target diagrams and define the tool icon.

A Palette Contribution is defined by the two basic notions:

- its target diagram
- contributed stereotypes or pure metaclasses

To create a Palette Contribution

1 Using the **Profile Definition** tool, add a Palette Contribution node to the diagram background.

2 In the Properties View, choose **Profile Definition** node.

3 In the **diagrams** field, define the diagram types, where you want the new creation tools to appear: click **Edit** and in the **Select Diagrams** dialog that opens, check the desired diagram types.

4 In the **icon field**, click **Edit** and in the **Select Icon** dialog that opens, navigate to the desired *. gif file, using **Copy From File System**.

> **TIP:** This dialog enables you to arrange your icons in an orderly way. Use **Create New Directory** to create a special folder for storing icons and populate it with the required images. Such approach is useful for the large shared projects.

5 Using Contribution link, define the contributed stereotype set:

- Linking to a stereotype defines the contributed stereotype
- Linking to a shortcut to a metaclass defines the contributed pure metaclass

Related Concepts

UML Profiles Basics

Deploying Profiles

After you create one or more profiles, you can create profile plug-ins to share them with your team members.

To deploy created profile

1 Select the Profile Definition project or any project element in the Diagram Editor or Navigator view.

2 Choose **Model Profile Deploy profile**. **Deploy Profile** wizard opens.

3 In the **Profile Content Project Settings** page, update project and plug-in settings as required and click **Next**.

4 In the **Behavior** page, define the way the new profile will be deployed. Follow the notes of the wizard. Click **Next**.

5 In the **Target Directory** page, select the directory where the plugin will be deployed. You can choose from the default location, linked folders or external location outside of the Eclipse platform. Click **Finish**.

6 Any errors that occur during the profile validation are reported in the **Profile Validation Results** view. You can navigate from an error message to the respective profile definition element and correct the error. When ready, click **Deploy** in the view.

7 After profile plug-in is deployed, you will be prompted to restart the workbench and make the new profile available in the list of supported profiles. Click **Yes** to restart.

NOTE: The profiles are internationalized on creation, therefore you can edit the properties file inside your new profile plugin to provide any strings.

Related Procedures

Uninstalling Profiles

Applying Profiles

Profile plugins that you create can be distributed among your team members.

If you have just created a profile plugin, you need to restart the application for the changes to take affect and for the plugin to become available in the program.

To enable a created profile

1 Select the **Navigator** view tab. If this view is not open, select **Window Show View Navigator** on the main menu.

2 In the **Navigator** view, right click the root project folder, and select **Properties** from the context menu. The **Properties** dialog displays.

3 From the list on the left, select **UML Profiles**.

4 Select the profile you created. More than one can be activated.

5 Click **OK**.

NOTE: There is no binary compatibility of compiled profiles across the various operating systems and versions of ER/Studio Software Architect. You can copy a deployed profile to the plugins folder on another computer, if the operating system and the ER/Studio Software Architect version are the same.

Related Procedures

UML Profiles Basics

Profile Definition Project

Adding Attributes to Stereotypes

In this section you will learn how to add attributes to stereotypes, define attribute properties, groupings and descriptions. After applying a profile, stereotype attributes become visible in the Properties View of the elements with this stereotype.

You can add attributes to stereotypes in one of the following ways:

- Using the Properties View

- Using outgoing association links

To add attributes (tagged values) to a stereotype using the Properties View

1. Right-click on the selected stereotype in profile definition diagram, and choose **New > Attribute** on the context menu. Add as many attributes as required.
2. Select an attribute in the stereotype node. Its properties are displayed in the Properties View
3. In the **Properties** tab:
   - Choose the **name** field and enter attribute name.
   - Click the **type** field and specify the valid type using the combo box for primitive types, or selection manager dialog for the enumerations and metaclasses.

   NOTE: invalid types are ignored during profile deployment.
4. In the **Profile Definition** tab, click the **inspector group** field, and select the inspector group from the list of existing groups, or create new one. After applying the profile, the attribute in question will display in a separate tab (group) of the Properties View.
5. In the **Description** tab, choose the Edit tab and enter textual description. After applying the profile, this description will show up as a tooltip in the tab (group) of the Properties View. **Result:** The tagged value name is equal to the attribute name. Multiplicity of the tagged value is equal to attribute's multiplicity.

To add attributes (tagged values) to a stereotype using outgoing association links

1. On the profile definition diagram, create shortcuts to certain types (Primitive types, Enumerations or Metaclasses). Note that invalid types are ignored during profile deployment.
2. In the Class Diagram group of the Tool Palette, select Association link and draw it from the stereotype in question to a shortcut to the desired type.
3. Select the created association link. Its properties are displayed in the Properties View
4. In the **Supplier** tab, specify the following properties: supplier multiplicity, supplier role
5. Specify the inspector group and description, as described in the steps 4 and 5 of the previous procedure, if necessary. **Result:** If supplier role is specified, it is used as the tagged value name. If the supplier role is not specified, then the link name or default name is used as an tagged value name. Multiplicity of the tagged value is equal to the supplier multiplicity. If the upper of the supplier multiplicity is greater than 1, the attribute is treated as multivalued.

   NOTE: When the attribute of a metaclass type or the association to metaclass shortcut is added to some stereotype, this attribute or association get the viewmap and icon properties in the Profile Definition section of the Properties View. These properties allow to select the viewmap and icon for the link that appears when this profile-specific property is set in the target project for the element with this stereotype.

Adding Shortcuts to Metaclasses

When creating your profile definition project you can use shortcuts to metatypes from metamodel root. Shortcuts to metaclasses can be created in several ways some of which are similar to adding any other shortcut to your diagram.

To use the Shortcuts dialog

1 Right-click the diagram background and select **Shortcuts > New**. The **Shortcuts** dialog displays.

2 Expand the metamodels node, choose the desired metaclass, and click **Add**.

3 Use **Add** and **Remove** to make up a list of extended metaclasses.

4 Click **OK** when ready.

To use cut, copy, and paste operations

1 Cut, copy, and drag a metaclass in the Model Navigator

2 Paste and drop it to any of your diagram except packages.

To create a shortcut on package diagram using drag-and-drop operation: press and hold CTRL and SHIFT while dragging an element from the Model Navigator to your package.

Applying Profiles

Profile plugins that you create can be distributed among your team members.

If you have just created a profile plugin, you need to restart ER/Studio Software Architect for the changes to take affect and for the plugin to become available in the program.

To enable a created profile

1 Select the **Navigator** view tab. If this view is not open, select **Window > Show View > Navigator** on the main menu.

2 In the **Navigator** view, right click the root project folder, and select **Properties** from the context menu. The **Properties** dialog displays.

3 From the list on the left, select **UML Profiles**.

4 Select the profile you created. More than one can be activated.

5 Click **OK**.

NOTE: There is no binary compatibility of compiled profiles across the various operating systems and versions of ER/Studio Software Architect. You can copy a deployed profile to the plugins folder on another computer, if the operating system and ER/Studio Software Architect version are the same.

Related Procedures

UML Profiles Basics

Profile Definition Project

Creating Palette Contributions

In this section you will learn how to create a new tool, assign target diagrams and define the tool icon.

A Palette Contribution is defined by the two basic notions:

- its target diagram

- contributed stereotypes or pure metaclasses

To create a Palette Contribution

1. Using the **Profile Definition** tool, add a Palette Contribution node to the diagram background.
2. In the Properties View, choose **Profile Definition** node.
3. In the **diagrams** field, define the diagram types, where you want the new creation tools to appear: click **Edit** and in the **Select Diagrams** dialog that opens, check the desired diagram types.
4. In the **icon field**, click **Edit** and in the **Select Icon** dialog that opens, navigate to the desired *. gif file, using **Copy From File System**.

   TIP: This dialog enables you to arrange your icons in an orderly way. Use **Create New Directory** to create a special folder for storing icons and populate it with the required images. Such approach is useful for the large shared projects.

5. Using Contribution link, define the contributed stereotype set:
   - Linking to a stereotype defines the contributed stereotype
   - Linking to a shortcut to a metaclass defines the contributed pure metaclass

Related Concepts

UML Profiles Basics

Creating Profile-Specific constraints

While defining the profile you can create a set of specific audits available only for projects with the applied profile. Such audits can be created as constraints linked to metaclasses in the profile definition project. It is important to note that constraint context can be only represented by a metaclass from the target metamodel.

For example, let's assume we have defined the stereotype MyStereotype for uml20::classes::Class and want to check that the class with this stereotype only extends class with the same stereotype.

To provide the audit, do the following in your profile definition project:

1. Create a shortcut to uml20::classes::Class metaclass.
2. Create a constraint element.
3. Link the created constraint with the metaclass shortcut (it gets the context uml20::classes::Class)
4. Type the following in the body of the constraint: inv:stereotypes->includes('MyStereotype') implies generalizations->forAll(general.stereotypes->includes('MyStereotype'))
5. Deploy profile.

After the profile is applied to some project, it is possible to run profile-specific audits via **Model > Profile > Run > Profile Constrains** command.

NOTE: **description** and **name** properties of the constraint element, specified in the Properties View, are used in the new audit. Value of the **description** property is used as the audit description, and the constraint name and invariant name are used as the audit name.

Related Concepts

UML Profiles Basics

Creating Stereotypes

To create a Stereotype

1 Using the **Profile Definition** palette, add a **Stereotype** node to the diagram background.

2 In the Properties View, choose **Profile Definition** node.

3 In the **Extended metaclass** field, click **Edit.** In the dialog that opens, select the desired metaclasses from the **Model Elements** pane. Use **Add** and **Remove** to make up a list of extended metaclasses. Click **OK** when ready

4 If necessary, specify the required stereotype property

Working with Required Stereotypes

5 Define view properties:

TIP: View properties are not available for the stereotypes that extend multiple metaclasses.

Setting Viewmap Properties for Stereotypes

6 Add attributes (tagged values) to the stereotype

Adding Attributes to Stereotypes

7 Using **Contribution link**, connect the Stereotype to the desired Palette Contribution.

Related Procedures

Working with Required Stereotypes

Setting Viewmap Properties for Stereotypes

Adding Attributes to Stereotypes

Working with Required Stereotypes

In this section you will learn how to create a required stereotype and how to manage stereotypes in diagrams after applying or removing the parent profile of a stereotype.

To create a required stereotype

1 In a Profile Definition, select a stereotype that extends a metaclass.

2 Select extension link: In the Model Navigator, expand the stereotype node and click the extension link.

TIP: Alternately, add a shortcut to the parent metaclass to the Profile Definition. The extension link to the extending stereotype is drawn automatically.

3 In the Properties View of the extension link, select the **Profile Definition** tab.

4 Set **is required** property to true. The extension link in diagram gets the label {isRequired}, and the field **bind with profile** appears in the Properties View.

5 Set **bind with profile** field as required:

- If the field is set to true: after applying the profile to a project, the appropriate elements get the required stereotype, and after turning off the parent profile this stereotype is removed from the elements.
- If the field is set to false: after applying the profile to a project, the appropriate elements get the require stereotype, and after turning off the parent profile this stereotype is preserved.

  TIP: This feature is useful for the large team projects and helps avoid confusion that might be caused by applying custom profiles.

To filter out required stereotypes in diagrams

1 On the main menu choose **Window > Preferences > Modeling Profiles > View Management**

2 Click the tab that corresponds to the desired metamodel.

3 In the list of available profiles, check the stereotypes you would like to hide in diagrams.

4 Apply changes and close the dialog.

Related Concepts

UML Profiles Basics

Related Procedures

Creating Stereotypes

Setting Viewmap Properties for Stereotypes

You can specify visual representation of the elements in the profile you create. Values in the icon and viewmap properties affect the way the elements are displayed on your diagram.

NOTE: Viewmap property values are different for stereotypes that extend links.

To set viewmap and icon properties

1 Select a stereotype rectangle. Properties view displays the properties of the selected stereotype.

2 In the **Profile Definition** node of the **Properties** view, choose extended metaclass field and click **Edit**.

3 In the selection manager dialog, navigate to the desired metaclass and click **Add.** You can select several extended metaclasses. Click **OK** when ready.

  TIP: Viewmap property is available for the stereotypes that extend one metaclass only.

4 Once a value for the extended metaclass property is provided, viewmap and icon properties are displayed.

5 Select the viewmap field and click **Edit**. This opens the **Viewmap Editor** dialog. Note that viewmap depends on the type of the extended metaclass: there are different sets of viewmap properties for the links and nodes.

6 Specify the viewmap properties: For a stereotype that extends a node: Select **color** to specify color for the element, or select **svg** to browse for an svg file. If you choose **svg** you can also specify the figure from those described in the selected svg file and specify whether the graphical node with selected .svg figure will be resizable. For a stereotype that extends a link: specify values for Source decoration, Target decoration, Foreground, Background, and Line style options.

7 Click **OK** to save the changes and close the dialog.

Adding Attributes to Stereotypes

In this section you will learn how to add attributes to stereotypes, define attribute properties, groupings and descriptions. After applying a profile, stereotype attributes become visible in the Properties View of the elements with this stereotype.

You can add attributes to stereotypes in one of the following ways:

- Using the Properties View
- Using outgoing association links

To add attributes (tagged values) to a stereotype using the Properties View

1. Right-click on the selected stereotype in profile definition diagram, and choose **New > Attribute** on the context menu. Add as many attributes as required.
2. Select an attribute in the stereotype node. Its properties are displayed in the Properties View
3. In the **Properties** tab:
   - Choose the **name** field and enter attribute name.
   - Click the **type** field and specify the valid type using the combo box for primitive types, or selection manager dialog for the enumerations and metaclasses.

   NOTE: invalid types are ignored during profile deployment.
4. In the **Profile Definition** tab, click the **inspector group** field, and select the inspector group from the list of existing groups, or create new one. After applying the profile, the attribute in question will display in a separate tab (group) of the Properties View.
5. In the **Description** tab, choose the Edit tab and enter textual description. After applying the profile, this description will show up as a tooltip in the tab (group) of the Properties View. **Result:** The tagged value name is equal to the attribute name. Multiplicity of the tagged value is equal to attribute's multiplicity.

To add attributes (tagged values) to a stereotype using outgoing association links

1. On the profile definition diagram, create shortcuts to certain types (Primitive types, Enumerations or Metaclasses). Note that invalid types are ignored during profile deployment.
2. In the Class Diagram group of the Tool Palette, select Association link and draw it from the stereotype in question to a shortcut to the desired type.
3. Select the created association link. Its properties are displayed in the Properties View
4. In the **Supplier** tab, specify the following properties: supplier multiplicity, supplier role
5. Specify the inspector group and description, as described in the steps 4 and 5 of the previous procedure, if necessary. **Result:** If supplier role is specified, it is used as the tagged value name. If the supplier role is not specified, then the link name or default name is used as an tagged value name. Multiplicity of the tagged value is equal to the supplier multiplicity. If the upper of the supplier multiplicity is greater than 1, the attribute is treated as multivalued.

   NOTE: When the attribute of a metaclass type or the association to metaclass shortcut is added to some stereotype, this attribute or association get the viewmap and icon properties in the Profile Definition section of the Properties View. These properties allow to select the viewmap and icon for the link that appears when this profile-specific property is set in the target project for the element with this stereotype.

Defining Profile Properties

When a Profile Definition project is created, you can specify or edit profile properties that are accessible via the default package diagram of the Profile Definition project. These properties are:

- Textual profile description
- Namespace, which identifies the profile.

To specify profile definition properties

1. Open the default package diagram of the Profile Definition project.
2. On the context menu of the diagram, choose **Properties**. The Properties View opens.
3. In the Properties View, select the **Profile Definition** tab.
4. Click the **description** field and enter the description text. Optionally, click **Edit**.
5. Click the **namespace** field and enter a valid string.

Deploying Profiles

After you create one or more profiles, you can create profile plug-ins to share them with your team members.

To deploy created profile

1. Select the Profile Definition project or any project element in the Diagram Editor or Navigator view.
2. Choose **Model Profile Deploy profile**. **Deploy Profile** wizard opens.
3. In the **Profile Content Project Settings** page, update project and plug-in settings as required and click **Next**.
4. In the **Behavior** page, define the way the new profile will be deployed. Follow the notes of the wizard. Click **Next**.
5. In the **Target Directory** page, select the directory where the plugin will be deployed. You can choose from the default location, linked folders or external location outside of the Eclipse platform. Click **Finish**.
6. Any errors that occur during the profile validation are reported in the **Profile Validation Results** view. You can navigate from an error message to the respective profile definition element and correct the error. When ready, click **Deploy** in the view.
7. After profile plug-in is deployed, you will be prompted to restart the workbench and make the new profile available in the list of supported profiles. Click **Yes** to restart.

   NOTE: The profiles are internationalized on creation, therefore you can edit the .properties file inside your new profile plugin to provide any strings.

Related Procedures

Uninstalling Profiles

Enabling UML Profiles

There are several ways to enable UML profiles for projects.

To enable UML profiles support while creating a project

1. On the main menu choose **File New Project**. The **New Project** wizard displays.
2. Expand the **Modeling** node in the tree view list, and select the UML project you want to create (UML 2.0). Click **Next**.
3. Follow the wizard to the **Profiles** screen. The **Profiles** screen of the wizard lists available profiles.
4. Select one or more profiles you want to enable and click **Next** to continue creating a new project with the **New Project** wizard.

To enable UML profiles support for existing projects

1. In the Model Navigator, right click the root project folder, and select **Properties** on the context menu. The **Properties for <project>** dialog displays.
2. From the list on the left, select **UML Profiles**.
3. Select any of the UML profiles that you want to enable. More than one can be activated.
4. Click **OK**.

   NOTE: You can also access the **Properties for <project>** dialog through the Navigator view.

To specify the default set of UML profiles enabled for all new workspace projects

1. Choose **Window > Preferences** on the main menu.
2. In the left pane of the **Preferences** dialog, expand the **Modeling** node.
3. Select the **UML Profiles** node.
4. Select the profiles you want to enable for a UML 2.0 project.

   NOTE: The selected UML profiles are automatically enabled for projects created after you changed profile preferences. Profiles support of existing projects is not changed.

Exporting and Importing Profiles

In this section you will learn how to export and import profile plugins.

To export profiles

1. On the main menu choose **File > Export**.
2. In the **Export** dialog, under the **Modeling** node, choose **Profile Plug-ins** and click **Next.**
3. In the list of available profiles, check the ones to be exported.
4. Specify the target directory, entering its fully qualified name in the text field, or clicking **Browse**.
5. Click **Finish**.

To import profiles

1. On the main menu choose **File > Import**.
2. In the **Import** dialog, under the **Modeling** node, choose **Profile Plug-ins** and click **Next**.

3 In the **Search profile plug-ins in directory** field, specify the source directory, where the desired plug-ins are stored. The list of available profiles displays.

4 In the list of profiles encountered in the specified folder, check the ones to be imported, and click **Next.**

5 Specify the target directory.

- If you click **Target plug-ins directory**, the selected profile plug-ins will be imported to the default
- directory.
- If you click **Linked directory**, the selected profile plug-ins will be imported to the linked directory
- of your choice. Optionally, you can link new directories, using **Link New Directory**.

6 Click **Finish**.

Related Procedures

UML Profiles Basics

Opening Profile Definition

In this section you will learn how to view and modify definition of the custom profiles and profiles that come bundled with ER/Studio Software Architect. Profile definition opens as a UML 2.0 project.

To open profile definition

1 On the main menu choose **Model Profile Open Profile Definition**. The **Open Profile Definition** dialog displays the list of profiles that declare their definitions.

2 Check the desired profile and click **Finish**. The selected profile definition opens as a UML 2.0 project.

Related Concepts

Profile Definition Project

Related Procedures

UML Profiles Basics

Setting Viewmap Properties for Stereotypes

You can specify visual representation of the elements in the profile you create. Values in the icon and viewmap properties affect the way the elements are displayed on your diagram.

NOTE: Viewmap property values are different for stereotypes that extend links.

To set viewmap and icon properties

1 Select a stereotype rectangle. Properties view displays the properties of the selected stereotype.

2 In the **Profile Definition** node of the **Properties** view, choose extended metaclass field and click **Edit**.

3 In the selection manager dialog, navigate to the desired metaclass and click **Add.** You can select several extended metaclasses. Click **OK** when ready.

TIP: Viewmap property is available for the stereotypes that extend one metaclass only.

4 Once a value for the extended metaclass property is provided, viewmap and icon properties are displayed.

5 Select the viewmap field and click **Edit.** This opens the **Viewmap Editor** dialog. Note that viewmap depends on the type of the extended metaclass: there are different sets of viewmap properties for the links and nodes.

6 Specify the viewmap properties: For a stereotype that extends a node: Select **color** to specify color for the element, or select **svg** to browse for an svg file. If you choose **svg** you can also specify the figure from those described in the selected svg file and specify whether the graphical node with selected .svg figure will be resizable. For a stereotype that extends a link: specify values for Source decoration, Target decoration, Foreground, Background, and Line style options.

7 Click **OK** to save the changes and close the dialog.

Uninstalling Profiles

Uninstalling profile feature enables you to correctly remove the unused profile plugins, which involves detaching them from all projects in your workspace, and deleting from the file system.

To uninstall a profile

1 On the main menu choose **Model Profile Uninstall Profile**.

2 Select profiles to be uninstalled.

Related Procedures

UML Profiles Basics

Verifying a Model Against Profile Constraints

Verification of a profile involves defining the necessary constraints within the target metamodel, and actual verification of a model against the selected constraints.

To verify a model against an applied profile

1 On the main menu choose **Window Preferences Profile Constraints**, and choose the desired constraints in the appropriate metamodel.

2 On the main menu choose **Model Profile Run Constraints**.

The results of verification display in the **Profile Constraints** view. You can navigate from any entry in the table to the respective element in diagram.

Related Procedures

UML Profiles Basics

Working with Required Stereotypes

In this section you will learn how to create a required stereotype and how to manage stereotypes in diagrams after applying or removing the parent profile of a stereotype.

To create a required stereotype

1 In a Profile Definition, select a stereotype that extends a metaclass.

2 Select extension link: In the Model Navigator, expand the stereotype node and click the extension link.

TIP: Alternately, add a shortcut to the parent metaclass to the Profile Definition. The extension link to the extending stereotype is drawn automatically.

3 In the Properties View of the extension link, select the **Profile Definition** tab.

4 Set **is required** property to true. The extension link in diagram gets the label {isRequired}, and the field **bind with profile** appears in the Properties View.

5 Set **bind with profile** field as required:

- If the field is set to true: after applying the profile to a project, the appropriate elements get the required
- stereotype, and after turning off the parent profile this stereotype is removed from the elements.
- If the field is set to false: after applying the profile to a project, the appropriate elements get the required
- stereotype, and after turning off the parent profile this stereotype is preserved.

TIP: This feature is useful for the large team projects and helps avoid confusion that might be caused by applying custom profiles.

To filter out required stereotypes in diagrams

1 On the main menu choose **Window > Preferences > Modeling Profiles > View Management**

2 Click the tab that corresponds to the desired metamodel.

3 In the list of available profiles, check the stereotypes you would like to hide in diagrams.

4 Apply changes and close the dialog.

Related Concepts

UML Profiles Basics

Related Procedures

Creating Stereotypes

UML 2.0 Diagrams

This section provides how-to information on using UML diagrams.

UML 2.0 Class Diagrams Procedures	Lists the UML 2.0 Class Diagrams Procedures.
UML 2.0 Use Case Diagrams Procedures	Lists the UML 2.0 Use Case Diagrams Procedures.
UML 2.0 Interaction Diagrams Procedures	Lists the UML 2.0 Interaction Diagrams Procedures.
UML 2.0 State Machine Diagrams Procedures	Lists the UML 2.0 State Machine Diagrams Procedures.
UML 2.0 Activity Diagrams Procedures	Lists the UML 2.0 Activity Diagrams Procedures.

UML 2.0 Component Diagrams Procedures	Lists the UML 2.0 Component Diagrams Procedures.
UML 2.0 Deployment Diagrams Procedures	Lists the UML 2.0 Deployment Diagrams Procedures.
UML 2.0 Composite Structure Diagrams Procedures	Lists the UML 2.0 Composite Structure Diagrams Procedures.
Template Elements	This section describes how to create template elements in diagrams and define formal parameters.

UML 2.0 Class Diagrams Procedures

Adding Owned Behavior to a Class	How to add behavior to a class.
Changing Appearance of Compartments	About changing appearance of the class compartments in diagrams.
Creating Data Types	How to create and extend a data type
Creating Enumerations and Enumeration Literals	How to create enumerations which are created as regular diagram elements.
Working with a Constructor	How to create a constructor and define constructor parameters.
Working with a Field	How to rename a field, define its visibility and stereotype.
Working with a Provided or Required Interface	How to work with the provided and required interfaces. These procedures are common to UML 2.0 Class, Component and Composite Structure diagrams.
Working with a Relationship	How to work with a relationship link (common for and 2.0).
Working with Association classes and n-ary associations	How to create and delete association classes and n-ary associations
Working with Inner Classes	How to create inner classes and inner interfaces in diagrams which display within their own compartment field within the class.
Working with Instance Specifications	How to create an instance specification which can instantiate one or more classifiers

Adding Owned Behavior to a Class

You can add behavior to a class. Behavior is defined by an activity, state machine or interaction.

To add a classifier behavior to a class

1 Select a class in diagram.

2 In the Properties View select the field **classifier behavior** and click browse.

3 In the dialog **Select Behavior for ‘classifier behavior’ Property**, select the desired element in the Model Elements pane and click **OK**.

The owned behaviors can be added to classifier by pasting some behavior into classifier (for example, cutting some activity and pasting it to a Class) or via the context menu. This way the interaction can be added to class, activity and interaction can be added to use case.

Related Concepts

UML 2.0 Class Diagram Definition

Changing Appearance of Compartments

You can collapse or expand compartments for the different members of class, interface, and package elements. Use the **Preferences** dialog to set viewing preferences for compartment controls. Adding compartment controls is particularly useful when you have large container elements with content that does not need to be visible at all times.

To show compartment controls

1. On the main menu, choose **Window > Preferences**
2. Open **Modeling > View Management**.
3. On the Details tab, check the option **Always show "Attributes" and "Operations" compartments**.

   NOTE: This option is not selected by default.

4. Click **OK.**

To collapse or expand compartments

1. Select the class (or interface) on the diagram.
2. Click the "**+**" or "**-**" located to the left of the compartment name.

Creating Data types

Data types are created as regular diagram elements, using the Tool Palette or **New > Data Type** command on the diagram context menu. You can add attributes and operations to data types, using the context menu.

To extend a data type

1. Select a data type in diagram.
2. In the Properties View, select the field **extends** and click Browse.
3. In the **Select Data Type for 'extends' Property** dialog, select the desired element in the Model Elements pane.
4. Use **Add** and **Remove** to make up a list of Selected Elements.
5. Click **OK** when your list is complete.

Related Concepts

UML 2.0 Class Diagram Definition

Related Procedures

Adding Owned Behavior to a Class

Creating Enumerations and Enumeration Literals

Enumerations are created as regular diagram elements, using the Tool Palette or **New > Enumeration** command on the diagram context menu.

To add an enumeration literal

1. Select an enumeration in diagram.
2. On the context menu, choose **New > Enumeration literal** and the new literal is added to the enumeration element.
3. In the Properties View, select the field **name** and enter the enumeration name. The name of the literal is displayed in diagram.
4. In the **specification** field enter a value.

   NOTE: The specification value is displayed only in the Property View for the enumeration literal.

To extend an enumeration

1. Select an enumeration in diagram.
2. In the Properties View, select the field **extends** and click browse.
3. In the dialog **Select Enumeration for 'extends' Property,** select the desired enumeration element in the Model Elements pane.
4. Use **Add** and **Remove** to make up a list of Selected Elements.
5. Click **OK** and the selected enumeration elements are connected by a link.

Related Concepts

UML 2.0 Class Diagram Definition

Working with a Constructor

You can create as many constructors in a class as needed, using the **New Constructor** command of the context menu of a class.

To define the constructor parameters

1. Select the desired constructor in a class.
2. In the Properties View, in the **parameters** field click **Browse**.
3. In the **Select Parameters for Operation** dialog, click **Add**. A parameter is added with the default values. Edit the values as required, or use the defaults. Use **Add** and **Remove** to make up the list of parameters, and click **OK** when ready.

   TIP: Alternately, you can type the list of parameters in the text area. Use comma as a delimiter.

Working with a Provider or Required Interface

To create a provided interface

1 Create class and interface node elements using the Palette icons.

2 On the diagram Palette, click **Provided Interface** ( ).

3 Click the client class and drag the mouse to the interface node.

To create a required interface

1 Create class and interface node elements using the Palette icons.

2 On the diagram Palette, click **Required Interface** ( ).

3 Click the client class and drag the mouse to the interface node.

Working with a Relationship

This section describes how to change the link type and properties.

To change the type of an association link

1 Select an Association Link on diagram.

2 In the Properties View, select the drop-down arrow in the **associates type** field.

3 Select the link type (association, aggregation, or composition) from the drop-down list.

Related Procedures

Creating a Simple Link

Changing Type of an Association Link

Working with Association Classes and n-ary Associations

Refer to the Getting Started Procedures to learn how to draw a link. This section describes how to change the link type and properties.

To change the type of an association link

1 Select an Association Link on the diagram.

2 In the Properties View, select the **associates type** field.

3 Choose the link type (association, aggregation, or composition) from the drop-down list.

Related Procedures

Creating a Simple Link

Changing Type of an Association Link

Working with Inner Classes

Both inner classes and inner interfaces in diagrams display within their own compartment field within the class. To create an inner class, do one of the following:

When the class already exists

1 Select the class you want to create as an inner class.

2 Drag it over the target class and drop it

Using the context menu:

1 Right click the parent class.

2 Select **New > Class** from the context menu.

Using Cut, Copy, and Paste:

Use the clipboard operations to either cut or copy an existing inner class.

1 Select the class you want to use as the inner class and click **Cut** from the context menu.

2 Select the parent class and select **Paste** from the context menu.

TIP: Classes do not keep the same visibility after removing them from the parent class.

Working with Instance Specifications

According to the UML 2.0 specification, an instance specification can instantiate one or more classifiers. You can instantiate a classifier using the **instantiates** property in the Properties View or the in-place editor.

To instantiate a classifier using the Properties View

1 Select an instance specification in your diagram.

2 In the **Properties** node of the Properties View, select the **instantiates** field.

3 Click Chooser.

4 In the **Choose Classifier for 'instantiates' property**, select the classifiers from the available contents, using **Add/Remove**.

5 Click OK to save your changes.

To instantiate a classifier using the in-place editor

1 Select an instance specification in your diagram.

2 Press F2 to open the in-place editor. Alternatively, click twice on the instance specification name.

3 Type the name of an existing classifier, delimited by a colon, next to the instance specification name. For example, InstanceSpecifcation1:Class1.

4 Press **Enter.** To define the features of an instance specification, you can insert slots into an instance specification element, associate the slots with the attributes of the instantiated classifiers, set value, and define the slot stereotype.

To add a slot to an instance specification element

1. Add an instance specification element to your diagram.
2. Right-click the instance specification element and choose **New Slot** on the context menu.

To associate a slot with a structural feature

1. Select a slot in an instance specification element.
2. Click the **Properties** tab of the Properties View.
3. In the defining **feature field**, select the desired attribute from the list of attributes owned by the classifiers, which is instantiated by the instance specification (or their parents).

To set the slot value, do one of the following

1. In the Properties View of the slot, select the value field, click **Editor**, and type the desired string in the **Edit property values** editor.
2. Invoke the in-place editor for the slot and type the value next to the slot name, delimited by an equal sign.

Related Procedures

UML 2.0 Component Diagrams Procedures

UML 2.0 Composite Structure Diagrams Procedures

UML 2.0 Class Diagrams Procedures

UML 2.0 Use Case Diagrams Procedures

This section outlines the procedures related to UML 2.0 Use Case diagrams.

Creating an Extension Point	How to create an extension point.
Defining Includes and Extends Links	How to create Includes and Extends links which are created between Use Cases.
Setting Subject for a Use Case	How to create a classifier for a Use Case.

Creating an Extension Point

To create an extension point:

1. Right-click the use case element.
2. Choose **Add Extension Point** on the context menu.
3. Type in a name.

Defining Includes and Extends Links

Includes and Extends links are created between Use Cases. The created links are marked with their stereotype.

To define condition properties of an Extends link

1. Select Extends link in diagram.
2. In the Properties View expand **condition** node.
3. In the **language** field, choose the condition type (OCL or plain text).
4. In the **body** field, specify the condition text.

   TIP: Alternatively, you can double click on the condition element in diagram and enter the condition text in the editor window.

Related Procedures

Setting Subject for a Use Case

Object Constraint Language (OCL)

Setting Subject for a Use Case

To set a subject for a Use Case:

1. Select a Use Case in diagram.
2. In the Properties View, select **subject.**
3. Click Browse.
4. In the **Select Classifier for Subject Property** dialog, select the desired classifiers from the metamodel. Use **Add** and **Remove** to create the list of selected elements.
5. Click **OK**.

UML 2.0 Interaction Diagrams Procedures

This section outlines the procedures related to UML 2.0 Sequence and Communication diagrams.

A Typical Scenario of Designing a UML 2.0 Interaction Diagram	How to design a UML 2.0 sequence or communication diagram.
Associating a Lifeline with a Classifier	How to associate a lifeline with a classifier.
Associating a Lifeline with a Referenced Element	How to associate a lifeline with a referenced element.
Copying and Pasting an Execution or Invocation Specification	How to copy and paste an execution or invocation specification.
Creating a Full-Screen Sequence or Communication Diagram from an Interaction	How to create a full-screen sequence or a communication diagram from an interaction.
Creating a State Invariant	How to create a state invariant.
Creating an Interaction Use	How to create an interaction use.
Defining Decomposition of a Lifeline	How to define decomposition of a lifeline.
Working with a Combined Fragment	How to work with a combined fragment.

Working with a UML 2.0 Message	About working with UML 2.0 messages.
Working with Interactions	How to create an interaction, open it in a sequence or collaboration diagram and associate it with a class.

A Typical Scenario of Designing a UML 2.0 Interaction Diagram

Use the following tips and techniques when you design a UML 2.0 Sequence or Communication Diagrams. Usually you create Interaction Diagrams after Class Diagrams.

Whenever an interaction diagram is created, the corresponding interaction is added to the project. Interactions are represented as nodes in the Model Navigator.

You can view an interaction in two ways: as a Sequence Diagram, or as a Communication Diagram. So doing, any actions performed with either view are automatically reflected in the other views. Thus, adding or deleting an element in an interaction results in the modification of the corresponding interaction diagram, and vice versa. An interaction diagram contains a reference to the underlying interaction.

To design a UML 2.0 Sequence Diagram, perform the following general actions

1. Create an interaction with one or more lifelines, and open it in a sequence diagram. Associate the interaction with a class and operation.

   Working with Interactions

2. Create interaction use

   Creating an Interaction Use

3. Associate a lifeline with a classifier

   Associating a Lifeline with a Classifier

4. Define decomposition of a lifeline

   Defining Decomposition of a Lifeline

5. Repeat the steps to create all required lifelines.

6. Link the created lifelines by using messages

   Working with a UML 2.0 Message

7. Add combined fragments to the lifeline

   Working with a Combined Fragment

8. Add state invariants

   Creating a State Invariant

Working with Interactions

You can start designing your sequence or communication diagram with creating an interaction. An interaction can be opened in a sequence or communication diagram.

In this section you will learn how to:

- Create an interaction
- Open an interaction in a sequence or communication diagram

- Define context and specification for an interaction

To create an interaction

1. In the Model Navigator, right-click a project or a package node.
2. On the context menu, choose **New > Interaction diagram elements >Interaction**

To open an interaction in diagram

1. Select the desired interaction in the Model Navigator.
2. On the context menu, choose **Open full screen communication diagram** or **Open full screen sequence diagram**.

To define context and specification for an interaction

1. Select an interaction in the Model Navigator.

   TIP: Alternately, click on the interaction diagram background.

2. In the Properties View select the **Properties** tab.
3. In the **context** field, click Chooser and in the **Choose referenced classifier** dialog select the desired context.
4. In the **specification** field, click Chooser and in the **Choose operation** dialog select the operation for the specified context.

Creating an Interaction Use

1. In the diagram Palette, choose **Interaction Use**.
2. Click on the target lifeline.
3. In the Properties View for the newly created interaction use, choose the **Properties** tab.
4. In the **interaction name** field, click Chooser.

   TIP: Alternately, just type the interaction name.

5. In the **Choose Referenced Interaction** dialog, select the desired interaction and click **OK.** An interaction use is initially created attached to a lifeline. Further you can expand it over several lifelines, detach from and reattach to lifelines.

Associating a Lifeline with a Classifier

In this section you will learn how to:

- Associate a lifeline with an existing classifier using the lifeline context menu
- Associate a lifeline with a new classifier using the context menu
- Associate a lifeline with a classifier using the Properties View

To associate a lifeline with a classifier using the lifeline context menu

1. Select a lifeline on an Interaction diagram.
2. Right-click the lifeline and select **Choose Type > <connectable element's type>** on the context menu.

3 If the desired type is not in the list, choose **More**. The **Choose represented connectable element's type** dialog opens.

4 Select a classifier to be associated with the lifeline from the tree of available model elements, and click **OK**.

To associate a lifeline with a new type using the lifeline context menu

1 Select a lifeline on an Interaction diagram.

2 Right-click the lifeline and select **New > Type** on the context menu.

3 Select **Class** or **Interface** on the submenu. The new connectable element adds to the model.

To associate a lifeline with a classifier using the Properties View

1 Select a lifeline on an Interaction diagram.

2 In the Properties View of the lifeline, select **type** field.

3 Enter the classifier name in the text area, or click **Browse**. The **Choose represented connectable element's type** dialog opens.

4 Select a classifier to be associated with the lifeline from the tree of available model elements, and click **OK**.

Defining Decomposition of a Lifeline

To define decomposition for a lifeline

1 Select the desired lifeline in the Model Navigator or the Diagram Editor.

2 In the Properties View, select the **decomposition** field.

3 Click **Browse**.

4 In the **Choose Referenced Interaction** dialog, select the desired interaction.

5 Click **OK**.

TIP: Decomposition, type, stereotype, and referenced element properties are also reflected in the corresponding Communication diagram.

Working with a UML 2.0 Message

This section describes techniques for working with messages in sequence and communication diagrams. Although the two diagram types are equivalent, the techniques for dealing with messages differ.

In this section you will learn how to:

- Show or hide reply message
- Create nested messages (Sequence diagram)
- Create a message from a lifeline back to itself
- Create a message link that corresponds to an operation call
- Create an asynchronous call, which enables you to extend or reduce the time of invocation specification and execution specification independently.
- Create a found execution on a lifeline, that is a message that comes from an object that is not shown on the diagram. (Sequence diagram)

To show or hide reply message

1. Select a call message on a diagram.
2. In the Properties View, set the **show reply message** value to true to show reply message or to false to hide reply message.

To create a nested message

1. Choose **Message** icon on the diagram Palette.
2. Click the desired execution specification to originate the message and drag the link to the target lifeline.

   NOTE: The nested message inherits the numbering of the parent message. For example, if the parent message has the number 1, its first nested message is 1.1.

To create a message from a lifeline back to itself

1. Choose **Message** icon on the diagram Palette.
2. Double-click the target lifeline.

To create a message link that corresponds to an operation call

1. Create a message link between two lifelines in an interaction.
2. Make sure that the target lifeline has its type defined, and the associated classifier contains at least one operation.
3. In the Properties tab of the Properties View, select the **signature** field and click **Browse**.
4. In the **Choose Operation** dialog, select the desired operation.
5. Click **OK.** The message link is named according to the name of the operation.

To de-synchronize invocation specification and execution specification

1. Select an invocation specification on a lifeline.
2. Click the **sort** property in the Properties View and select asynchCall in the list.

To create a found execution

1. In the Palette, click **Found execution**.
2. Click the desired place of a lifeline. An execution specification bar is created in the target lifeline.

Related Procedures

Working with Instance Specifications

Working with a Combined Fragment

In this section you will learn how to:

- Create a combined fragment
- Create nested combined fragments
- Create nested operators

- Sever nested operators
- Create operands
- Expand combined fragments across several lifelines
- Detach a combined fragment from a lifeline

To create a combined fragment

1 Choose **Combined Fragment** in the diagram Palette, and click on the target lifeline.

2 In the **New Combined Fragment** dialog that opens, choose the desired operator from the list of available operators and set the combined fragment options (operator name, arguments, or number of operands).

3 Click **OK.** Result: the combined fragment is added to the target lifeline or execution specification. Each new combined fragment has different color, to tell it from the other combined fragments within the same cluster of nested frames.

To create a nested combined fragment

1 Choose **Combined Fragment** in the diagram Palette, and click on the target combined fragment that already exists in a lifeline.

NOTE: Each new node has different color that is selected at random. You can work with the inner frames same way as with the outer frames: move along a lifeline, spread them over several lifelines, detach and tie frames. Note that drawing a message link from a frame automatically expands it, together with its outer frames, if any.

To create nested operators

1 Select the desired combined fragment.

2 In the **other operators** field of the Properties View, click **Browse**. The **Interaction Operators** dialog opens, displaying the list of already defined operators in the current combined fragment.

3 Click **Add**. A new line displays below the existing entry in the list of operators.

4 If a certain operator enables arguments, enter them in the adjacent field in the Arguments column. Use comma as a delimiter.

5 Use **Add** and **Remove** to make up the desired list of the nested operators. Use **Up** and **Down** to specify the proper order of nested operators.

6 Click **OK** to apply changes. Result: the nested operators are listed in the descriptor of the combined fragment in the specified order.

To sever operators

1 Right-click a combined fragment that contains nested operators.

2 On the context menu choose **Sever operators between**.

3 On the submenu select the pair of operators, between which the combined fragment will be divided. Result: A nested combined fragment is created.

To combine with an outer fragment

1 Right-click an inner fragment.

2 On the context menu, choose **Combine with an outer fragment**.

To create an operand

1 Select the desired combined fragment or an operand in the Model Navigator or in the Diagram Editor.

2 On the context menu of the selection, choose **New Interaction Operand**.

3 In the **Interaction constraint** tab of the Properties View, select the language to be used for describing constraint. To do this, click the Language drop-down list and choose OCL or plain text.

4 Type the constraint expression.

5 Add as many operands as required.

6 Apply changes. Result: a new operand is created. If the operand was created from the context menu of a combined fragment, it will be added to the end of the combined fragment. If the operand was created from the context menu of an operand, it will be added just before this operand. Constraint text displays in the operand section of the combined fragment.

To expand a combined fragment across several lifelines

1 Select the desired combined fragment.

TIP: You can expand both outer and inner combined fragments.

2 Click the anchor icon and drag it to the target lifeline. Result: the fragment is spans across lifelines, with the mounting links on each lifeline.

To detach a combined fragment from a lifeline

1 Select the mounting link of a combined fragment.

2 Choose **Delete** on the context menu.

TIP: You cannot delete the only mounting link of a combined fragment. A combined fragment must be attached to at least one lifeline.

Creating a State Invariant

A state invariant is a constraint placed on a lifeline. This constraint is evaluated at runtime prior to execution of the next execution specification. State invariants are represented in the interaction diagrams in two ways: as OCL expressions or as references to the state diagrams. You can use the state invariants to provide comments to your interaction diagrams and to connect interactions with states.

Validation is provided of the state invariants represented as OCL expressions. An OCL editor with highlighting and validation is provided for typing you OCL expression.

The typed OCL expression is correct only if the context is specified for it. The context of the State Invariant connected to the LifeLine is the type of the part this LifeLine represents. If no type is set, the OCL expression is reported as invalid with OCL as the language of the expression. If no correct context can be specified, select text as the expression language.

To create a state invariant as an OCL expression:

1 On the diagram Tools Palette, click **state invariant** .

2 Click the target lifeline or execution specification.

3 In the **Properties** view of the state invariant, expand the **Common properties** node.

4 In the invariant kind field, choose OCL expression from the list. The shape of the state invariant diagram element changes to braces.

5. In the OCL invariant view that opens, select the language of the comment from the Language list. The possible options are OCL and plain text.
6. Type your expression text and apply changes.

To connect a state invariant to a state:

1. On the diagram Tools Palette, click the **state invariant**.
2. Click the target lifeline or execution specification.
3. In the **Properties** view of the state invariant, expand the **Common properties** node.
4. In the invariant kind field, choose States/Regions from the list.
5. In the States/Regions field, click **Edit**.
6. In the **Choose States and/or Regions** dialog, select the desired states and/or regions from the model. Use **Add** to add them to the **Selected** list.
7. Click **OK** to save your changes.
8. Alternatively, you can type the state or region name using the in-place editor. If the state or region belongs to a different package, specify its fully-qualified name.

Related Concepts

OCL Support

Associating a Lifeline with a Classifier

In this section you will learn how to:

- Associate a lifeline with an existing classifier using the lifeline context menu
- Associate a lifeline with a new classifier using the context menu
- Associate a lifeline with a classifier using the Properties View

To associate a lifeline with a classifier using the lifeline context menu

1. Select a lifeline on an Interaction diagram.
2. Right-click the lifeline and select **Choose Type <connectable element's type>** on the context menu.
3. If the desired type is not in the list, choose **More**. The **Choose represented connectable element's type** dialog opens.
4. Select a classifier to be associated with the lifeline from the tree of available model elements, and click **OK**.

To associate a lifeline with a new type using the lifeline context menu

1. Select a lifeline on an Interaction diagram.
2. Right-click the lifeline and select **New Type** on the context menu.
3. Select **Class** or **Interface** on the submenu. The new connectable element adds to the model.

To associate a lifeline with a classifier using the Properties View

1. Select a lifeline on an Interaction diagram.
2. In the Properties View of the lifeline, select **type** field.
3. Enter the classifier name in the text area, or click **Browse**. The **Choose represented connectable element's type** dialog opens.
4. Select a classifier to be associated with the lifeline from the tree of available model elements, and click **OK**.

Associating a Lifeline with a Referenced Element

To associate a lifeline with a referenced element

1. Make sure that your Interaction context or Interaction specification contains the referenced elements that should be represented by the lifelines.
2. Select the desired lifeline in the Model Navigator or the Diagram Editor.
3. In the Properties View, select the **represents** field.
4. Click **Browse**.
5. In the **Choose Represented Connectable Element** dialog, select the desired part from the project or Favorites.
6. Click **OK**.

To navigate to a referenced interaction

1. Right-click on an interaction use that refers to another interaction.
2. On the context menu, choose **Select**.
3. Choose the desired destination on the submenu.

Copying and Pasting an Execution or Invocation Specification

Clipboard operations are supported for the execution and invocation specifications.

To copy and paste an execution or invocation specification:

1. **Cut**, **Copy**, and **Paste** commands are available on the context menu of an execution specification and invocation specification. It is possible to copy or move these elements within the same diagram or to another diagram.
2. When an execution or invocation specification is copied, it means that the entire branch of messages is copied also. Pasting the clipboard contents to a target lifeline results in changing the message numbers according to the numbering of messages in the target lifeline.
3. If you paste an invocation or execution specification to another diagram, the entire outgoing bunch of messages will be pasted also, with all the respective lifelines. If the target diagram does not contain lifelines for this execution specification, they will be created automatically.

   TIP: It is also possible to move and copy message branches using the drag-and-drop technique. To move an execution or invocation specification, drag-and-drop it to the target location. To create a copy, drag-and-drop while holding the CTRL key down.

Related Procedures

Working with a UML 2.0 Message

Creating a Full-Screen Sequence or Communication Diagram from an Interaction

To create a full-screen sequence or a communication diagram from an interaction

1. In the Diagram Editor or in the Model Navigator, choose an Interaction element.
2. Right-click the Interaction node and choose **Open Full-Screen Sequence diagram** or **Open Full-Screen Communication diagram**. Results: If such diagram does not exist, it will be created. Then this diagram opens in the Diagram Editor.

Creating a State Invariant

A state invariant is a constraint placed on a lifeline. This constraint is evaluated at runtime prior to execution of the next execution specification. State invariants are represented in the interaction diagrams in two ways: as OCL expressions or as references to the state diagrams. You can use the state invariants to provide comments to your interaction diagrams and to connect interactions with states.

Validation of the state invariants represented as OCL expressions is provided. An OCL editor with highlighting and validation is provided for typing you OCL expression.

The typed OCL expression is correct only if the context is specified for it. The context of the State Invariant connected to the LifeLine is the type of the part this LifeLine represents. If no type is set, the OCL expression is reported as invalid with OCL as the language of the expression. If no correct context can be specified, select text as the expression language.

To create a state invariant as an OCL expression:

1. On the diagram Tools Palette, click the **state invariant**.
2. Click the target lifeline or execution specification.
3. In the **Properties** view of the state invariant, expand the **Common properties** node.
4. In the invariant kind field, choose OCL expression from the list. The shape of the state invariant diagram element changes to braces.
5. In the OCL invariant view that opens, select the language of the comment from the Language list. The possible options are OCL and plain text.
6. Type your expression text and apply changes.

To connect a state invariant to a state:

1. On the diagram Tools Palette, click the **state invariant**.
2. Click the target lifeline or execution specification.
3. In the **Properties** view of the state invariant, expand the **Common properties** node.
4. In the invariant kind field, choose States/Regions from the list.
5. In the States/Regions field, click **Edit**.
6. In the **Choose States and/or Regions** dialog, select the desired states and/or regions from the model. Use **Add** to add them to the **Selected** list.

7 Click OK to save your changes.

8 Alternatively, you can type the state or region name using the in-place editor. If the state or region belongs to a different package, specify its fully-qualified name.

Related Concepts

OCL Support

Creating an Interaction Use

To create an interaction use

1 In the diagram Palette, choose **Interaction Use**.

2 Click on the target lifeline.

3 In the Properties View for the newly created interaction use, choose the **Properties** tab.

4 In the **interaction name** field, click **Browse**.

TIP: Alternately, just type the interaction name.

5 In the **Choose Referenced Interaction** dialog, select the desired interaction and click **OK.** An interaction use is initially created attached to a lifeline. Further you can expand it over several lifelines, detach from and reattach to lifelines.

Defining Decomposition of a Lifeline

To define decomposition for a lifeline

1 Select the desired lifeline in the Model Navigator or the Diagram Editor.

2 In the Properties View, select the **decomposition** field.

3 Click **Browse**.

4 In the **Choose Referenced Interaction** dialog, select the desired interaction.

5 Click **OK**.

TIP: Decomposition, type, stereotype, and referenced element properties are also reflected in the corresponding Communication diagram.

Working with a UML 2.0 Message

This section describes techniques for working with messages in sequence and communication diagrams. Although the two diagram types are equivalent, the techniques for dealing with messages differ.

In this section you will learn how to:

- Show or hide reply message
- Create nested messages (Sequence diagram)
- Create a message from a lifeline back to itself

- Create a message link that corresponds to an operation call
- Create an asynchronous call, which enables you to extend or reduce the time of invocation specification and execution specification independently.
- Create a found execution on a lifeline, that is a message that comes from an object that is not shown on the diagram. (Sequence diagram)

To show or hide reply message

1 Select a call message on a diagram.

2 In the Properties View, set the **show reply message** value to true to show reply message or to false to hide reply message.

To create a nested message

1 Choose **Message** icon on the diagram Palette.

2 Click the desired execution specification to originate the message and drag the link to the target lifeline.

NOTE: The nested message inherits the numbering of the parent message. For example, if the parent message has the number 1, its first nested message is 1.1.

To create a message from a lifeline back to itself

1 Choose **Message** icon on the diagram Palette.

2 Double-click the target lifeline.

To create a message link that corresponds to an operation call

1 Create a message link between two lifelines in an interaction.

2 Make sure that the target lifeline has its type defined, and the associated classifier contains at least one operation.

3 In the Properties tab of the Properties View, select the **signature** field and click **Browse**

4 In the **Choose Operation** dialog, select the desired operation.

5 Click **OK.** The message link is named according to the name of the operation.

To de-synchronize invocation specification and execution specification

1 Select an invocation specification on a lifeline.

2 Click the **sort** property in the Properties View and select asynchCall in the list.

To create a found execution

1 In the Palette, click **Found execution**.

1 Click the desired place of a lifeline. An execution specification bar is created in the target lifeline.

Related Procedures

Working with Instance Specifications

Working with Interactions

You can start designing your sequence or communication diagram with creating an interaction. An interaction can be opened in a sequence or communication diagram.

In this section you will learn how to:

- Create an interaction
- Open an interaction in a sequence or communication diagram
- Define context and specification for an interaction

To create an interaction

1. In the Model Navigator, right-click a project or a package node.
2. On the context menu, choose **New > Interaction diagram elements >Interaction**

To open an interaction in diagram

1. Select the desired interaction in the Model Navigator.
2. On the context menu, choose **Open full screen communication diagram** or **Open full screen sequence diagram**.

To define context and specification for an interaction

1. Select an interaction in the Model Navigator.

   TIP: Alternately, click on the interaction diagram background.

2. In the Properties View select the **Properties** tab.
3. In the **context** field, click **Browse** and in the **Choose referenced classifier** dialog select the desired context.
4. In the **specification** field, click **Browse** and in the **Choose operation** dialog select the operation for the specified context.

UML 2.0 State Machine Diagrams Procedures

Associating a Transition or a State with a Behavior	How to associate a transition with an activity (UML 2.0 State Machine Diagram).
Changing Regions Order in a State	If you have several regions in your State or State Machine, you can rotate them to place either in horizontal or vertical order.
Creating an OCL Guard Condition for a Transition	How to create a guard condition for a transition
Creating and Editing States	How to create and edit states in a state diagram.
Creating History Elements	How to create a history.
Creating Members for State Machines, States, and Regions	How to create a member for a state.
Designing a UML 2.0 State Machine Diagram	How to design a UML 2.0 state machine diagram

Working with a Complex State	How to create a composite (nested) state (UML 2.0 State Machine Diagram).
Working with Activities and States Machines Full Screen Diagrams	How to open an activity or a state machine in a full-screen view.

Associating a Transition or a State with a Behavior

You can associate an activity (created on some UML 2.0 Activity Diagram) with a state (on entering the state, while doing the state activity, and on exiting the state), or with a transition between states.

To associate a transition with an activity

1 Select a transition or a state on a UML 2.0 State Machine diagram.

2 In the Common properties tab of the **Properties** view, click the **Effect** (for a transition) or **Do Behavior**, **Entry Behavior** or **Exit Behavior** (for a state) field.

3 Click **Browse** to open the **Select Behavior for property** dialog.

4 In the model tree view, locate the desired activity and click **Add**.

5 Click **OK** to save the changes.

Changing Regions Order in a State

If you have several regions in your State or State Machine, you can rotate them to place either in horizontal or vertical order.

To change regions order, perform the following:

1 Right click a State or a State Machine on your diagram

2 Select either **Order Regions Vertically** or **Order Regions Horizontally**.

Related Concepts

UML 2.0 State Machine Diagram Definition

Creating an OCL Guard Condition for a Transition

To create a guard condition for a transition:

1 Select a transition on a diagram.

2 Select the **guard** tab in the **Properties** view.

3 Specify the language for your guard condition (OCL by default).

4 Select the **body** field and click **Edit**.

5 Type the condition expression and click OK to apply changes.

Related Concepts

OCL Support

Creating and Editing States

NOTE: All State Diagram elements are created inside State Machines, therefore, create at least one State Machine before attempting to create other elements.

To create a state

1 On the Palette click **State**.

2 Click the diagram background.

Alternately:

Right-click a region of the StateMachine element and select **New State** from the context menu. When a new state is placed on a diagram, you can use the Properties View to edit its properties.

- Configure standard properties of the element.
- In the **State Invariant** tab, select the language of the expression from the Language list box. The possible options are OCL and plain text.
- In the **Properties** page, configure the behavior of the state by setting these additional properties:

Field	Description
Composite	Set to True if there is one or more regions in this state (not editable)
Orthogonal	Set to True if there are two or more regions in this state (not editable)
Simple	Set to True if there are no regions in this state (not editable)
Do Behavior	Specify the activity to be performed during execution of the current state by using the Properties View. This activity may be selected from any Activity diagram of the project
Entry Behavior	Specify the activity to be performed when the current state starts executing by using the Properties View. This activity may be selected from any Activity diagram of the project
Exit Behavior	Specify the activity to be performed when the current state finishes executing by using the Properties View. This activity may be selected from any Activity diagram of the project

NOTE: You may place a state inside of the existing state. It is possible to hide individual states. For example, you can hide the content of composite states for better understanding of the whole diagram.

To make a state a submachine state

1 Select a state you want to make a submachine state.

2 In the Properties View of the state click the **submachine** property and choose any model state machine. For your convenience a StateMachine element can be opened in a separate Diagram Editor view.

To open a StateMachine element in a separate Diagram Editor view

1 Right-click a **StateMachine** element in the Diagram Editor.

2 Select **Open Full Screen in New Diagram**.

To define the do activity, entry or exit activities of a state

1. Select a state and click the appropriate field in the Properties view.
2. Click **Edit**. This opens the **Select Activity20 for property** dialog.
3. Locate the desired activity and use **Add** and **Remove** to add and remove activities.
4. Click **OK** to save your changes.

Related Concepts

OCL Support

UML 2.0 State Machine Diagram Definition

Related Procedures

Changing Regions Order in a State

Creating History Elements

The Shallow History and Deep History elements are placed on regions of the states. Please refer to UML 2.0 Specification for more information about these elements.

You can create none or one Deep History, and none or one Shallow History elements in each region.

To add a history to a state, do one of the following:

1. Right-click a region in a state, point to **New**, and select one of the **History** elements on the shortcut menu.
2. Click one of the **History** elements on the Palette and then click the target state region.

Related Concepts

UML 2.0 State Machine Diagram Definition

Related Procedures

Changing Regions Order in a State

Creating Members for State Machines, States, and Regions

To create a member for a state:

1. Open the **Diagram View**.
2. Right-click an existing state and choose **New (member)** on the context menu.

   The following members are available:

   - Internal transition (also available on the context menu) – A shorthand for handling events without leaving a state and dispatching its exit/entry activities.
   - Region (also available on the context menu) – Use regions inside the states to group the substates. The regions may have different visibility settings and history elements. Each state has one region immediately after creation though it can be deleted.
   - Reference to Entry point and Reference to Exit point – Use references to entry/exit points as sources/targets of transitions respectively.
   - In the Regions, you can create all the elements that are available for the States except Internal transition. In addition to regions you can create the following members for State Machines:
   - Entry point – Execution of the state starts at this point. It is possible to create several entry points for one state, that makes sense if there are substates.
   - Exit point – Execution of the state finishes at this point. It is possible to create several exit points for one state, that makes sense if there are substates.

Designing a UML 2.0 State Machine Diagram

Following are tips and techniques that you can use when working with UML 2.0 State Machine Diagram.

To design a UML 2.0 State Machine Diagram, follow this general procedure:

1. Create initial and final nodes.
2. Create main states and substates.
3. Create regions.
4. Create entry and exit points.
5. Create pins.
6. Create transitions.
7. Create history nodes.
8. You can optionally create shortcuts to related elements of other diagrams.

Related Procedures

Creating a Shortcut

Working with a Complex State

The techniques in this section pertain to models of particularly complex composite states. These procedures are common for the State and Activity diagrams.

Create a composite state by nesting one or more levels of states within one state and draw transitions among the nested elements. You can place the following elements in a state:

- activity
- signal sending
- signal receipt
- start/end states
- history

TIP: You can nest multiple levels of states inside one state. For especially complex state modeling, however, you can find it more convenient to create different diagrams, model each of the state levels individually, and then hyperlink the diagrams sequentially.

Use the following techniques to create a composite (nested) state

1. Create a nested state using drag-and-drop
2. Create a nested state using the context menu of the state element

To create a nested state using drag-and-drop

1. Place a state element on the diagram background.
2. Drag a new state on top of an existing state.
3. Drop a new state.

To create a nested state using the context menu of the state element

1. Right-click the state (region) that will be the container.
2. Select **New State** on the context menu.

TIP: Using the **Shortcuts** command on the context menu of the diagram, you can reuse existing elements from the other state diagrams. Right click the diagram and choose **New Shortcuts**, navigate within the pane containing the tree view of the available project contents to the existing diagram, and select its elements, states, histories, forks, and/or joins.

Related Concepts

Model Hyperlinking Overview

Related Procedures

Creating a Shortcut

Working with Activities and States Machines Full Screen Diagrams

It is possible to work with the individual activities and state machines in specific diagrams. You can open an activity or a state machine in a full-screen view, the element in question being expanded to the whole diagram. The new diagram has the same name as the selected element. Such activity or state machine is transparent, which makes it possible to view the grid. You cannot move the container activity or state machine, but the nested elements are still selectable and movable.

To create a full-screen diagram for an activity or a state machine

1 Select the desired element in diagram or in the Model Navigator.

2 On the context menu of the selection, choose **Open Full Screen in New Diagram**.

UML 2.0 Activity Diagrams Procedures

Creating Activity Parameters	How to add an activity parameter to an activity.
Creating Pins	How to create a pin.
Designing a UML 2.0 Activity Diagram	How to design a UML 2.0 Activity Diagram.
Rotating Activity Partitions	How to change orientation of activity partitions.
Using Control Flow Link	
Working with Activities and States Machines Full Screen Diagrams	How to open an activity or a state machine in a full-screen view.
Working with Activity Element	How to create a transition link to represent a control flow. It can be drawn between the following elements on the state or activity diagrams.
Working with an Object Flow or a Control Flow	How to work with an object flow or a control flow.

Creating Activity Parameters

To add an activity parameter to an activity:

1 On the Palette, click **Activity Parameter**.

2 Click the target activity.

OR

1 Right click an activity

2 Select **New Activity Parameter** on the context menu

An Activity Parameter node is added to the activity as a rectangle. Note that the activity parameter node is attached to its activity. You can only move the node along the activity borders.

Creating Pins

Actions may require some input and produce some output. The input and output are defined by the input and output pins.

To add an input pin, output pin, or value pin, do one of the following:

1 Right-click an action

2 Select **New**, and choose either **Input Pin** or **Output Pin** or **Value Pin** on the context menu.

OR

1 In the Tools Palette choose one of the pins

2 Click the target action.

The created pin is added to the target action as a square. Note that the pins are attached to their actions, and can be only dragged along the action borders.

Designing a UML 2.0 Activity Diagram

Use the following tips and techniques when you design a UML 2.0 Activity Diagram. Usually you create Activity Diagrams after State Machine Diagrams.

To design a UML 2.0 Activity Diagram, follow this general procedure:

1 Create one or more activities. You can place several activities on a single diagram, or create a separate diagram for each.

CAUTION: You cannot create nested activities.

2 Usually activities are linked to states or transitions on State Machine Diagrams. Switch to your State Machine Diagrams and associate the activities you just created with states and transitions.

TIP: After that you can find that some more activities must be created, or the same activity can be used in several places.

3 Switch back to the Activity Diagram. Think about flows in your activities. You can have an object flow (for transferring data), a control flow, both or even several flows in each activity.

4 Create starting and finishing points for every flow. Each flow can have the following starting points:

- Initial node
- Activity parameter (for object flow)
- Accept event action
- Accept time event action

Each flow finishes with a **Activity Final** or **Flow Final** node.

If your activity has several starting points, they can be used simultaneously.

5 Create object nodes. You do not link object nodes to classes on your Class Diagrams. However, you can use hyperlinks for better understanding of your diagrams.

6 Create action nodes for your flows. Flows can share actions.

CAUTION: You cannot create nested actions.

7 For object flows, add pins to actions. Connect actions and pins by flow links.

8 Add pre- and post- conditions. You can create plain text or OCL conditions.

9 You can optionally create shortcuts to related elements of other diagrams.

Related Procedures

Creating a Shortcut

Rotating Activity Partitions

By default, activity partitions are created horizontally aligned. You can rotate those elements to fit your diagram.

To rotate activity partitions

1. Create one or more activity partitions within activities.
2. Right click the Activity with activity partitions and select **Rotate Activity Partitions**.

Related Procedures

Creating a Shortcut

Using Control Flow Link

A transition link represents a control flow. It can be drawn between the following elements on the state or activity diagrams. It can also be drawn from an initial node, or to an activity final or final flow element from those elements listed below.

- state
- activity invocation
- decision
- fork/join
- history

To create a control link between two elements

1. Click the Control Flow link icon on the tools palette.
2. On the Diagram, click the source element.
3. Drag the link to the destination element.
4. Drop when the second element is highlighted.

Once the link has been drawn on the diagram, use the Properties view to update the link. Properties set for the link are shown on the diagram. For more information on available properties, see Control Flow Link Properties.

Working with Activities and States Machines Full Screen Diagrams

It is possible to work with the individual activities and state machines in specific diagrams. You can open an activity or a state machine in a full-screen view, the element in question being expanded to the whole diagram. The new diagram has the same name as the selected element. Such activity or state machine is transparent, which makes it possible to view the grid. You cannot move the container activity or state machine, but the nested elements are still selectable and movable.

To create a full-screen diagram for an activity or a state machine

1. Select the desired element in diagram or in the Model Navigator.
2. On the context menu of the selection choose **Open Full Screen in New Diagram**.

Working with Activity Element

Because all activity diagram elements are enclosed into the Activity element, you should create at least one Activity to start modeling in the Activity diagram.

To create an activity element:

1. On the Palette click **Activity**.
2. Click the diagram background.

For your convenience an Activity element can be opened in a separate Diagram Editor view.

To open an activity element in a separate Diagram Editor view:

1. Right click an activity element in the Diagram Editor
2. Select **Open Full Screen in New Diagram**

Working with an Object Flow or a Control Flow

You can create control flow or object flow as an ordinary link between the two node elements. The valid nodes are highlighted when the link is established.

You can scroll to the target element if it is out of direct reach, or you can use the context menu command to avoid scrolling.

There are certain limitations stipulated by UML 2.0 specifications:

- Object flow link must have an object at least on one of its ends.
- It is impossible to connect two actions with an object flow except through an output pin on the source action.
- Control flow link may not connect objects and/or activity parameters.

Use the following techniques with an object flow or a control flow:

1. Create a flow. Flows are created as the regular links.
2. Create a fork or a join
3. Create a decision or a merge

To create a fork or a join

1. Identify the actions involved. If necessary, place all of the actions on the diagram first. Lay them out as desired.
2. Place either a fork or a join on the diagram. Resize as needed.
3. If depicting multiple sources, draw control flow from each of the source actions to the join, and from the join to the target action. If depicting multiple targets, draw control flow from the source action to the fork, and from the fork to each of the target actions.

To create a decision or a merge

1. Identify the actions involved. If necessary, place all of the actions on the diagram first. Lay them out as desired.
2. Place either a decision or a merge on the diagram. Resize as needed.
3. If merging multiple actions, draw control flow from each of the source actions to the merge, and from the merge to the target action. If making a decision, draw control flow from the source action to the decision, and from the decision to each of the target actions.

Related Procedures

Creating a Simple Link

UML 2.0 Component Diagrams Procedures

Designing a UML 2.0 Component Diagram	How to design a UML 2.0 Component Diagram.
Working with a Provided or Required Interface	How to work with the provided and required interfaces. These procedures are common to UML 2.0 Class, Component and Composite Structure diagrams.
Working with Instance Specifications	How to create an instance specification which can instantiate one or more classifiers.

Designing a UML 2.0 Component Diagram

Following are tips and techniques that you can use when working with UML 2.0 Component Diagrams. It can be convenient to start creation of a model with Component Diagrams if you are modeling a large system. For example, a distributed, client-server software system, with numerous interconnected modules. You use Component Diagrams for modeling a logical structure of your system, while you use Deployment Diagrams for modeling a physical structure.

To design a UML 2.0 Component Diagram, follow this general procedure:

1. Create a hierarchy of components. The largest component can be the whole system or its major part (for example, *server application*, *IDE*, *service*).

   TIP: You can create nested component nodes. There are two methods for creating a nested component node: You can select an existing component and add a child component inside. Alternatively, you can create two separate components and connect them with an Association-Composition link.

2. In the hierarchy of components, you can end up by adding concrete classes and instance specifications. You can create them on a Component Diagram directly, or create them on a Class Diagram and put shortcuts on a Component Diagram.
3. Create interfaces. Each component can have a provided interface and a required interface.
4. Optionally, create artifacts. Usually, you describe physical artifacts of your system on Deployment Diagrams. But if some component is closely connected with its physical store, add and link an artifact to a Component Diagram.

   TIP: You can create nested artifacts.

5 Optionally, create ports for your components. You can attach a port to a component and link it with several classes or components inside. In this case, when a message arrives, this port decides which class must handle it.

6 Draw links between elements.

7 You can optionally create shortcuts to related elements of other diagrams.

Related Procedures

Creating a Shortcut

Working with a Provided or Required Interface

To create a provided interface

1 Create class and interface node elements using the and Palette icons.

2 On the diagram Palette, click **Provided Interface**.

3 Click the client class and drag the mouse to the interface node.

To create a required interface

1 Create class and interface node elements using the and Palette icons.

2 On the diagram Palette, click **Required Interface**.

3 Click the client class and drag the mouse to the interface node.

Working with Instance Specifications

According to the UML 2.0 specification, an instance specification can instantiate one or more classifiers. You can instantiate a classifier using the **instantiates** property in the Properties View or the in-place editor.

To instantiate a classifier using the Properties View

1 Select an instance specification in your diagram.

2 In the **Properties** node of the Properties View, select the **instantiates** field.

3 Click **Browse**.

4 In the **Choose Classifier for 'instantiates' property**, select the classifiers from the available contents, using Add/Remove.

5 Click OK to save your changes.

To instantiate a classifier using the in-place editor

1 Select an instance specification in your diagram.

2 Press F2 to open the in-place editor. Alternatively, click twice on the instance specification name.

3 Type the name of an existing classifier, delimited by a colon, next to the instance specification name. For example, InstanceSpecifcation1:Class1.

4 Press **Enter**.

To define the features of an instance specification, you can insert slots into an instance specification element, associate the slots with the attributes of the instantiated classifiers, set value, and define the slot stereotype.

To add a slot to an instance specification element

1. Add an instance specification element to your diagram.
2. Right-click the instance specification element and choose **New Slot** on the context menu.

To associate a slot with a structural feature

1. Select a slot in an instance specification element.
2. Click the **Properties** tab of the Properties View.
3. In the defining **feature field**, select the desired attribute from the list of attributes owned by the classifiers, which is instantiated by the instance specification (or their parents).

To set the slot value, do one of the following

1. In the Properties View of the slot, select the **value** field, click **Editor**, and type the desired string in the **Edit property values** editor.
2. Invoke the in-place editor for the slot and type the value next to the slot name, delimited by an equal sign.

Related Procedures

UML 2.0 Component Diagrams Procedures

UML 2.0 Composite Structure Diagrams Procedures

UML 2.0 Class Diagrams Procedures

UML 2.0 Deployment Diagrams Procedures

This section provides how-to information about designing UML 2.0 deployment diagrams.

Designing a UML 2.0 Deployment Diagram	How to design a UML 2.0 Deployment Diagram.
Working with Artifacts	How to create an artifact which represents a physical entity and is depicted in diagram as a rectangle with the <<artifact>> stereotype.

Designing a UML 2.0 Deployment Diagram

Use the following tips and techniques when you design a UML 2.0 Deployment Diagram. It can be convenient to start creation of a model with Deployment Diagrams if you are modeling a large system that is comprised of multiple modules, especially if these modules reside on different computers. You use Deployment Diagrams for modeling a physical structure of your system, while you use Component Diagrams for modeling a logical structure.

To design a UML 2.0 Deployment Diagram, follow this general procedure

1 Create a hierarchy of execution environments, devices, and nodes. Execution environments usually represent software environment used to execute your system, such as an operating system. Devices usually represent hardware equipment, such as a printer, a hard disk, or a computer. Nodes represent the rest of physical entities, such as a file.

TIP: You can create nested execution environments, devices, and nodes. For example, you can add a node inside of an execution environment, or a node inside of a device.

2 Create artifacts.

3 Create deployment and instance specifications. By doing this, you arrange physical locations of objects and other entities of your system.

4 Add operations to artifacts.

5 Once an operation is added, you can define its properties in the Properties View, which includes parameters, stereotype, multiplicity and more.

6 You can optionally create shortcuts to related elements of other diagrams.

To deploy an artifact to a target node

1 In the diagram Palette, choose **deployment**.

2 Drag-and-drop the deployment link from a node to an artifact.

To define parameters of an operation

1 Select the desired operation in an artifact.

2 In the Properties View, expand the **General** node and choose **Parameters** field.

3 Click **Browse** to open **Add/Remove Parameters** dialog.

4 Click **Add**. This creates an entry in the parameters list.

5 Enter the parameter's name, type multiplicity, default value, and direction. Note that parameter type can be selected from the list of predefined types, or from the model.

6 Using **Add** and **Remove**, create the list of parameters.

7 Click **OK** when ready.

Related Procedures

Creating a Shortcut

Working with Artifacts

An artifact represents a physical entity and is depicted in diagram as a rectangle with the <<artifact>> stereotype. An artifact may have properties that define its features, and operations that can be performed on its instances.

Physically the artifacts can be model files, source files, scripts, binary executable files, a table in a database system, a development deliverable, a word-processing document, or a mail message.

A deployed artifact is the one that has been deployed to a node used as a deployment target. Deployed artifacts are connected with the target node by deployment links.

TIP: You can create complex artifacts, by nesting artifact icons.

To add operation to an artifact

1 Right-click an artifact icon in the diagram.

2 Choose **New Operation** on the context menu.

Once an operation is added, you can define its properties in the Properties View.

To define parameters of an operation, follow these steps

1 Select the desired operation in an artifact.

2 In the Properties view, expand the Common properties node and click the parameters field.

3 Click **Edit** to open the **Select Parameters for Operation** dialog.

4 Click **Add**. This creates an entry in the parameters list.

5 Enter the parameter's name, type, direction kind, multiplicity, default value. Note that the parameter type and direction kind can be selected from the list of pre-defined values.

6 Repeat steps 4-5 to create the list of parameters and click OK when ready.

To deploy an artifact to a target node:

1 Click the deployment icon in the Palette.

2 Click the element to be deployed. The valid source is highlighted.

3 Drag-and-drop the deployment link to a target node. The valid target is highlighted.

UML 2.0 Composite Structure Diagrams Procedures

Creating a Port	How to create a port.
Creating a Referenced Part	How to create a referenced part.
Creating an Internal Structure for a Node	How to create an internal structure for a node.
Working with a Collaboration Use	How to work with a collaboration use
Working with a Provided or Required Interface	How to work with the provided and required interfaces. These procedures are common to UML 2.0 Class, Component and Composite Structure diagrams.
Working with Instance Specifications	How to create an instance specification to instantiate one or more classifiers.

Creating a Port

To create a port:

1 Choose the port icon on the Palette.

2 Click the target class or part.

3 Create as many ports as required.

Creating a Referenced Port

To create a referenced part:

1. Open the **Diagram View**.
2. Do one of the following:
    - Use the referenced part icon on the diagram Palette.
    - Right-click a target container and choose **New Referenced part** on the context menu.
    - Select a part, open the Model Navigator, and check the option aggregated by reference.

Creating an Internal Structure for a Node

To create an internal structure for a node

1. Choose the part icon on the diagram Palette.
2. Click the valid container (class or collaboration).
3. Repeat these steps to create as many participants as needed.

    TIP: Choose the part icon on the diagram Palette while holding down the CTRL key. Each click on a valid container produces a new part.

4. Link the collaborating parts by connectors.
5. Use the Properties View to set up the properties of the part.

Working with a Collaboration Use

To create a collaboration use

1. On the Tools Palette, choose **Collaboration Use**.
2. Click the target container.
3. Specify the name of the Collaboration Use.

To link to a collaboration type

1. Select a Collaboration Use element.
2. Specify the type of Collaboration Use using one of the following methods:
    - In the type field of the Collaboration Use in the Tools Palette, click choose, and select the collaboration, which the Collaboration Use instantiates, from the model.
    - Next to the name of the Collaboration Use, insert a colon and the name of the collaboration, which the Collaboration Use instantiates. Result: The type of collaboration use is indicated next to its name.

To bind the roles (parts) of the different classifiers via the collaboration use

1. Create a collaboration use and define its type.
2. Create one or more parts in the collaboration that represents the type.

3 In the Composite Structure Diagram tool palette click the Role Binding icon ( ).

4 Click the collaboration use occurrence and drag the cursor and click the target role. Result: A role link is created from the collaboration use to the role in the target classifier. The role link is now marked with the name of the role selected in the collaboration.

NOTE: Each role can be used for binding only once.

To define an owner

1 Right-click a collaboration use and choose Properties on its context menu.

2 In the owning classifier field of the Properties View, click **Browse**.

3 In the **Select Owning Classifier** dialog, navigate to the owner class or collaboration and click OK. Result: A link is created between the owner as supplier, and the collaboration use as the client. The link is marked with the label <<occurrence>>.

Working with a Provided or Required Interface

To create a provided interface

1 Create class and interface node elements using the and Palette icons.

2 On the diagram Palette, click **Provided Interface**.

3 Click the client class and drag the mouse to the interface node.

To create a required interface

1 Create class and interface node elements using the and Palette icons.

2 On the diagram Palette, click **Required Interface**.

3 Click the client class and drag the mouse to the interface node.

Working with Instance Specifications

According to the UML 2.0 specification, an instance specification can instantiate one or more classifiers. You can instantiate a classifier using the **instantiates** property in the Properties View or the in-place editor.

To instantiate a classifier using the Properties View

1 Select an instance specification in your diagram.

2 In the **Properties** node of the Properties View, select the **instantiates** field.

3 Click **Browse**.

4 In the **Choose Classifier for 'instantiates' property**, select the classifiers from the available contents, using Add/Remove.

5 Click OK to save your changes.

To instantiate a classifier using the in-place editor

1. Select an instance specification in your diagram.
2. Press F2 to open the in-place editor. Alternatively, click twice on the instance specification name.
3. Type the name of an existing classifier, delimited by a colon, next to the instance specification name. For example, InstanceSpecifcation1:Class1.
4. Press Enter.

To define the features of an instance specification, you can insert slots into an instance specification element, associate the slots with the attributes of the instantiated classifiers, set value, and define the slot stereotype.

To add a slot to an instance specification element

1. Add an instance specification element to your diagram.
2. Right-click the instance specification element and choose **New Slot** on the context menu.

To associate a slot with a structural feature

1. Select a slot in an instance specification element.
2. Click the **Properties** tab of the Properties View.
3. In the defining **feature field**, select the desired attribute from the list of attributes owned by the classifiers, which is instantiated by the instance specification (or their parents).

To set the slot value, do one of the following

1. In the Properties View of the slot, select the **value** field, click **Editor**, and type the desired string in the **Edit property values** editor.
2. Invoke the in-place editor for the slot and type the value next to the slot name, delimited by an equal sign.

Related Procedures

UML 2.0 Component Diagrams Procedures

UML 2.0 Composite Structure Diagrams Procedures

UML 2.0 Class Diagrams Procedures

Template Elements

This section describes how to create template elements in diagrams and define formal parameters

Creating Constraints	This topic describes how to create an OCL constraint.
Creating Template Elements	This topic provides how-to information about creating template elements. This procedure is common for UML 2.0 diagrams.

Defining Formal Parameters	This topic provides how-to information about adding formal parameters to templates. This procedure is common for UML 2.0 diagrams.
Editing Constraint Expressions	How to edit a constraint expression.

Creating Constraints

You can create constraints for all elements of the UML 2.0 diagrams. To describe a constraint, you can use plain text, or OCL.

To create a constraint in a UML 2.0 diagram

1. Click **Constraint Link** on the diagram Palette and point to the model element which defines the context of your constraint (i.e. Class, Attribute or Operation), then hold down the left mouse button and draw the link to the place where you want to create the **Constraint** element.
2. Release the mouse button to insert the element. The element displays with the in-place editor open.
3. Type the constraint expression, save your changes, and close the **Constraint** editor.

TIP: Alternately, use one of the following methods:

- On the context menu of an element, choose **New Linked Constraint**, and enter the constraint expression.
- Use the **Constraint** and **Constraint link** on the Tools Palette to place a constraint node on the diagram and link it to the context element.

Related Concepts

OCL Support

Comparing and Merging Models

Describes how to compare models and model elements with each other, and perform history comparison with the earlier versions of the model stored in VCS.

Comparing and Merging Shared Models	How to compare and merge models shared with VCS.
Comparing Models	How to compare two or three models against each other and review differences.
Merging Models	How to merge models using the **Compare** editor.

Comparing and Merging Shared Models

Use the **Synchronize** view to compare shared models.

To compare and merge shared models

1 In the Team **Synchronize** view, select a model or model element which version stored in the repository you want to compare with the local version.

2 Choose **Model Compare With Local Version** from the main menu

TIP: Alternately, right-click a model or model element and choose **Open In Model Compare Dialog** from the context menu.

The comparison results display in the **Model Compare** dialog.

3 Review the differences, apply your changes, and then commit the model to the repository using menu commands specific to your VCS.

NOTE: When merging a shared model, you can change your local version only. models consist of a large number of files, so you need to have all of these files locally.

CAUTION: A merge is performed on the model level, and existing file conflicts may still remain after the model merge. To commit these changes, use the "forced commit" mechanism provided by your version control system, e.g. the "Override and Commit" option in CVS.

Related Concepts

Model Compare and Merge

Related Procedures

Merging Models

Comparing Models

You can compare two or three models against each other and review differences.

To compare models

1 In the **Model Navigator** or **Navigator** view, select two or three models or model elements.

2 Choose **Compare With Each Other (as Models)** from the context menu. The comparison results display in the **Compare** editor.

Related Concepts

Model Compare and Merge

Related Procedures

Merging Models

Merging Models

How to merge models using the **Compare** editor.

To merge models

1. Compare models or model elements.

   Comparing Models

2. Double-click the first difference displayed in the **Structure Compare** section of the **Compare** editor. The difference details display in the **Substructure/Properties Merge** section of the **Compare** editor.

   TIP: Use **Show Containment References** on the editor toolbar to toggle plain/treelike view of the comparison results.

3. In the **Substructure Merge** tab, select the desired element of the input model and click **Copy to the Left** or **Copy to the Right** to copy it to the target model.

4. In the **Properties Merge** tab, select the desired property of the input model and click **Copy to the Left** or **Copy to the Right** to copy it to the target model.

   TIP: You can undo and redo operations, using **Undo** and **Redo** toolbar icons, or keyboard shortcuts CTRL +Z and CTRL+SHIFT+Z.

Related Concepts

Model Compare and Merge

Comparing Models

You can compare two or three models against each other and review differences.

To compare models

1. In the **Model Navigator** or **Navigator** view, select two or three models or model elements.
2. Choose **Compare With Each Other (as Models)** from the context menu. The comparison results display in the **Compare** editor.

Related Concepts

Model Compare and Merge

Related Procedures

Merging Models

Object Constraint Language (OCL)

This section provides how-to information on using OCL facilities.

Creating an OCL Guard Condition for a Transition	How to create a guard condition for a transition.
Creating Constraints	This topic describes how to create an OCL constraint.
Editing Constraint Expressions	How to edit a constraint expression.

OCL In Documentation Templates	How to use OCL expressions in the templates for generating project documentation.
Searching Model with OCL queries	How to search for model elements using OCL queries.
Using OCL in Model Audits and Metrics	How to use OCL expressions in Audits and Metrics.
Working with a Combined Fragment	How to work with a combined fragment.
Working with Custom OCL Operations	How to create, edit, import and export OCL operations.

Creating an OCL Guard Condition for a Transition

To create a guard condition for a transition:

1. Select a transition on a diagram.
2. Select the **guard** tab in the **Properties** view.
3. Specify the language for your guard condition (OCL by default).
4. Select the **body** field and click **Edit**.
5. Type the condition expression and click OK to apply changes.

Related Concepts

OCL Support

Creating Constraints

You can create constraints for all elements of the UML 2.0 diagrams. To describe a constraint, you can use plain text, or OCL.

To create a constraint in a UML 2.0 diagram

1. Click **Constraint Link** on the diagram Palette and point to the model element which defines the context of your constraint (i.e. Class, Attribute or Operation), then hold down the left mouse button and draw the link to the place where you want to create the **Constraint** element.
2. Release the mouse button to insert the element. The element displays with the in-place editor open.
3. Type the constraint expression, save your changes, and close the **Constraint** editor.

TIP: Alternately, use one of the following methods:

- On the context menu of an element, choose **New Linked Constraint**, and enter the constraint expression.
- Use **Constraint** and **Constraint link** on the Tools Palette to place a constraint node on the diagram
- and link it to the context element.

Related Concepts

OCL Support

Editing Constraint Expressions

Constraint expressions are represented in plain text or in OCL language. You can use the Editor view, or OCL tab of the Properties View to create or modify the constraint body.

To edit a constraint expression in the Editor view

1 Double-click a constraint element. The constraint test opens in its own tab of the Editor view.

2 In the **Language** drop-down list in upper-right corner of the view, select the desired language of the expression.

NOTE: If OCL is selected, the OCL editor provides syntax control and error highlight. Red or green mark to the right indicate the validity of the OCL expression.

3 Apply changes.

To edit a constraint expression in the Properties View

1 Select a constraint element in diagram.

2 In the Properties View select the **OCL** tab.

3 In the **language** field, select the desired language of the expression.

4 In the **body** field, enter the expression in the text area, or click **Edit** and enter text in the **Enter constraint** dialog.

NOTE: Alternately, select a constraint element and press `F2`. Edit the constraint in the in—place editor.

Related Concepts

OCL Support

Related Procedures

Creating Constraints

OCL in Documentation Templates

You can compose model queries and define enable conditions using OCL syntax, and use them in a template for generating documentation. OCL or Legacy type expressions can be entered in the template element's properties dialog using the provided Expression Editor. Where applicable the editor is either opened in the tab or you can use Edit Expression to open the editor.

The standard OCL operations, with the special native OCL extensions are provided for the functions that are specific for Documentation Generation. Native OCL extensions mostly have the same signature and meaning as the legacy Documentation Generation functions have. Code sense suggests these operations along with the standard OCL ones.

To add an expression to your template

1 Open a template where you want to add OCL expression.

2 In the **Properties** dialog, open the tab where you want to type the expression. If the Expression Editor is not opened in the tab, click **Edit Expression**.

3 Specify the context for your expression in the **Context** field.

4. In the **Body** area you can type the expression text. The code sense, syntax highlighting and validating are available.
5. Click **OK** to save the expression in the template.

Related Concepts

OCL Support

Searching Model with OCL Queries

You can search for models using OCL queries.

To find model elements that match the specified OCL query

1. On the main menu, choose **Search Model**. The **Search** dialog displays.
2. Click the **OCL Model Search** tab.
3. Specify the context for your expression in the **Context** field.

   TIP: Use the drop-down list, or Content Assistant. To open the Content Assistant, click on the **Context** field and press CTRL +SPACE. Choose the desired element from the list.

4. In the **Invariant** field type the query expression.
5. In the **Scope** section click the desired radio button to select the search area. The possible options are workspace, selected resources, the current project or a predefined working set. To select a working set, click on **Choose**. In the **Select Working Set** dialog, choose the desired working set and click **OK**. If there are no available working sets, use **New** to create one.
6. Click **Search.** A tree with the list of matching elements will be opened. You can navigate to the corresponding diagram from this view by double-clicking the selected element.

Related Concepts

OCL Support

Using OCL in Model Audits and Metrics

You can run audits and metrics in your design model using OCL expressions.

You can create custom OCL audits and metrics that operate with metamodel types and run them against the model that is an instance of the same metamodel. ER/Studio Software Architect also contains a set of sample audits (the ideas of most of them are taken from Ambler and Fowler books). These audits can be used as examples for custom rules creation.

To define audits and metrics

1. Choose **Window > Preferences** from the menu and the **Preferences** dialog displays.
2. Expand the **Modeling** node and select **QA Source**.
3. Select either **Audits** or **Metrics** tab.
4. Click **New** to add an audit or metric. The **Edit Audit** or **Edit Metric** dialog displays respectively.

5 Specify your audit or metric name, description, severity, and select the context of the OCL expression. The code for your new audit or metric is displayed in the standard OCL editor in the **Body** text area. The audit expression should be a valid invariant that returns Boolean. Each metric expression should return Integer value.

To run defined audits and metrics

1 In the diagram, select model elements against which you want to run audits or metrics.

2 Select **Model Run Model Audits** or **Model Run Model Metrics** from the main menu. The results display in **Model Audit** or **Model Metrics** view respectively.

NOTE: The scope depends on the current selection made on diagram: if the project default diagram is selected, the entire project will be checked. If a single element is selected, this element will be checked only.

Related Concepts

OCL Support

Working with a Combined Fragment

In this section you will learn how to:

- Create a combined fragment
- Create nested combined fragments
- Create nested operators
- Sever nested operators
- Create operands
- Expand combined fragments across several lifelines
- Detach a combined fragment from a lifeline

To create a combined fragment

1 Choose **Combined Fragment** in the diagram Palette, and click on the target lifeline.

2 In the **New Combined Fragment** dialog that opens, choose the desired operator from the list of available operators and set the combined fragment options (operator name, arguments, or number of operands).

3 Click **OK**. Result: the combined fragment is added to the target lifeline or execution specification. Each new combined fragment has different color, to tell it from the other combined fragments within the same cluster of nested frames.

To create a nested combined fragment

1 Choose **Combined Fragment** in the diagram Palette, and click on the target combined fragment that already exists in a lifeline.

NOTE: Each new node has different color that is selected at random. You can work with the inner frames same way as with the outer frames: move along a lifeline, spread them over several lifelines, detach and tie frames. Note that drawing a message link from a frame automatically expands it, together with its outer frames, if any.

To create nested operators

1. Select the desired combined fragment.
2. In the **other operators** field of the Properties View, click **Browse**. The **Interaction Operators** dialog opens, displaying the list of already defined operators in the current combined fragment.
3. Click **Add**. A new line displays below the existing entry in the list of operators.
4. If a certain operator enables arguments, enter them in the adjacent field in the Arguments column. Use comma as a delimiter.
5. Use **Add** and **Remove** to make up the desired list of the nested operators. Use **Up** and **Down** to specify the proper order of nested operators.
6. Click **OK** to apply changes. Result: the nested operators are listed in the descriptor of the combined fragment in the specified order.

To sever operators

1. Right-click a combined fragment that contains nested operators.
2. On the context menu choose **Sever operators between**.
3. On the submenu select the pair of operators, between which the combined fragment will be divided. Result: A nested combined fragment is created.

To combine with an outer fragment

1. Right-click an inner fragment.
2. On the context menu, choose **Combine with an outer fragment**.

To create an operand

1. Select the desired combined fragment or an operand in the Model Navigator or in the Diagram Editor
2. On the context menu of the selection, choose **New Interaction Operand**.
3. In the **Interaction constraint** tab of the Properties View, select the language to be used for describing constraint. To do this, click the Language drop-down list and choose OCL or plain text.
4. Type the constraint expression.
5. Add as many operands as required.
6. Apply changes. Result: a new operand is created. If the operand was created from the context menu of a combined fragment, it will be added to the end of the combined fragment. If the operand was created from the context menu of an operand, it will be added just before this operand. Constraint text displays in the operand section of the combined fragment.

To expand a combined fragment across several lifelines

1. Select the desired combined fragment.

   TIP: You can expand both outer and inner combined fragments.

2. Click the anchor icon and drag it to the target lifeline. Result: the fragment is spans across lifelines, with the mounting links on each lifeline.

To detach a combined fragment from a lifeline

1 Select the mounting link of a combined fragment.

2 Choose **Delete** on the context menu.

TIP: You cannot delete the only mounting link of a combined fragment. A combined fragment must be attached to at least one lifeline.

Working with Custom OCL Operations

Custom OCL operations can be used in OCL queries throughout ER/Studio Software Architect - in audits, metrics, search expressions, gendoc templates, etc. In this section you will learn how to perform the following actions with the custom OCL operations:

- Create
- Delete
- Edit
- Export
- Import
- Clone

To create a custom OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling OCL** and open **OCL Operations** tab.

3 On the toolbar of the tab, click **New.**

4 In the Edit Operation dialog that opens, choose context and enter the valid OCL expression as the body of the operation.

5 Click **OK**.

To delete an OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling OCL** and open **OCL Operations** tab.

3 Select the operation to be deleted.

4 On the toolbar of the tab, click **Remove**.

To edit an OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling > OCL** and open **OCL Operations** tab.

3 Select operation to be modified.

4 On the toolbar of the tab, click **Edit.**

5 In the Edit Operation dialog that opens, update context and the body of the operation.

6 Click **OK**.

TIP: Alternately, you can update the body of the operation in the OCL text area.

To export an OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling OCL** and open **OCL Operations** tab.

3 Select the operation to be exported.

4 On the toolbar of the tab, click **Export**.

5 In the export dialog that opens, navigate to the desired target location and save the operation as *.oclOperations type.

To import an OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling OCL** and open **OCL Operations** tab.

3 On the toolbar of the tab, click **Import**.

4 In the import dialog that opens, find the desired file of the *.oclOperations type, and click **OK**.

To clone an OCL operation

1 On the main menu choose **Window > Preferences.**

2 In the **Preferences** dialog, choose **Modeling OCL** and open **OCL Operations** tab.

3 Select operation to be copied.

4 On the toolbar of the tab, click **Clone**.

Related Concepts

OCL Support

Related Procedures

Object Constraint Language (OCL)

Patterns

This section provides how-to information on using patterns.

Adding a Pattern Part	How to add a part to a pattern
Building Pattern	How to build a pattern from a pattern definition project.
Creating Model Element by Pattern	How to create elements by pattern.
Creating Pattern Definition	How to create a pattern definition project on the base of the selected model elements.

Deleting Patterns Instances	How to delete pattern instances from the model.
Managing Pattern Definitions in the Pattern Registry	How to use the Pattern Registry to create, edit and export pattern definitions.
Recognizing Patterns	How to recognize patterns in a project.
Validating Pattern Definition Projects	How to validate a pattern definition project.
Verifying Pattern Instances	How to verify a pattern instance and delete invalid instances.
Working with the Pattern Instances	How to use pattern instances (create elements by pattern, verify pattern instances, add pattern parts)

Adding a Pattern Part

To add a part to a pattern

1. Right-click an oval that represents pattern instance in diagram.
2. On the context menu of the pattern instance, choose **Create pattern part**.
3. On the submenu of this menu node, select the desired part ti be created. **Create Pattern Part** wizard opens.
4. On the first page of the wizard, specify the target package where the part will be created. Click **Next**.
5. On the second page of the wizard, specify properties for the new pattern part, using the **Set values** section.
6. Click **Finish**.

Related Concepts

Patterns

Building Pattern

In this section you will learn how to build a pattern from a pattern definition project.

To build a pattern from a pattern definition project

1. Right-click your pattern definition project in the Model Navigator, and choose **Patterns Build pattern** on the context menu.
2. Specify the name of pattern and select a folder in pattern registry (in local workspace-specific part) where the shortcut to the new pattern will be created.
3. Click **Finish** to start the compilation process.

NOTE: During compilation process the project validation is performed. If an error preventing the project from compiling is found, the compilation is aborted and errors are displayed in the Pattern definition validation results view.

Creating Model Element by Pattern

To create model elements by pattern

1 On the main menu choose **File > New > Other**.

2 Expand the **Modeling** node and select **Model element by pattern**.

3 Select patterns to apply and click **Next.** You can optionally define the values for user-editable properties of the pattern and select whether to add pattern instance into the model. If you choose to add pattern instance, the new entity appears in the Patterns root or your project and in the Pattern Explorer, and the oval appear in diagram.

4 Click **Finish** to create elements based on the selected pattern definition.

TIP: If your diagram looks entangled after adding pattern instances, use **Layout > Layout All** command on the diagram context menu.

Related Concepts

Patterns

Creating Pattern Definition

New pattern definition projects are created the same way as any other project in ER/Studio Software Architect. While creating a new project, select **Pattern definition** under node. The project is created with empty model for you to design your pattern from scratch.

ER/Studio Software Architect suggests a simplified way to create a pattern definition project by meant of exporting a selection of model elements to a pattern definition.

To create a pattern definition project from selected model elements

1 Select the elements you want to transform to pattern definition.

2 Right click the selection and choose **Export Pattern definition** from the context menu.

3 In the **Create pattern from elements** dialog, specify the name of the pattern definition and the target category. You can opt to display the existing patterns, and select the transformation profile is necessary. Click **Next**.

4 In the **Set role names** page, edit the role names of the elements involved in the pattern definition, or accept defaults. Click **Next**.

5 In the **Set default values** page, specify the default values for each role. Click **Finish.** Newly created model will be filled with images (represented as **InstanceSpecifications**) of selected elements of the source model. Properties of the source model elements are represented by **Slots**. New pattern definition displays in the Pattern Registry view.

Related Procedures

Creating a Project

Deleting Patterns Instances

You can delete elements of the patterns, using Diagram Editor, the Model Navigator, or the **Pattern Explorer**.

To delete a pattern instance

1 In the Diagram Editor or Model Navigator, select the pattern instance to be deleted.

2 On the context menu choose **Delete** to delete the selected pattern instance from the diagram and model.

TIP: Alternately, select the desired pattern instance in the **Pattern Explorer**, and choose **Delete instances** on the context menu. The selected instance is deleted from the model.

To delete all instances of a pattern

1 In the **Pattern Explorer**, select the desired pattern node.

2 On the context menu of the selection, choose **Delete instances**.

Related Concepts

Patterns

Managing Pattern Definitions in the Pattern Registry

In this section you will learn how to:

- Open the Pattern Registry
- Create a new pattern from registry
- Edit pattern definitions from registry
- Export pattern definitions as a plugin

To open Pattern Registry

1 Select **Window > Show view > Other** from the main menu.

2 Expand the **Patterns** node and select **Pattern Registry**.

To create a new pattern from registry

1 Right click the Workspace folder in the **Pattern Registry**, and choose **New Pattern** on the context menu.

2 Specify a name for the new pattern and select a pattern definition project that will be compiled to a pattern.

To edit definition from registry

1 Right click a pattern in the Pattern Registry

2 Choose **Edit Definition** on the context menu.

To export definition as a plugin

1 Click **Export** on the toolbar of the **Pattern Registry**.

2 Specify necessary parameters for a new plugin and select patterns you want to include in the plugin.

3 Click **Finish** to create a new Eclipse extension plugin.

Recognizing Patterns

You can examine your project for patterns.

To recognize patterns in a project

1. Select a project in the Model Navigator.
2. On the context menu of the project, choose **Patterns Recognize Patterns**. The **Recognize Patterns** wizard opens.
3. In the **Choose Pattern Definitions** page, choose definitions of the patterns you want to find in the project. To do that, check the desired patterns in the tree of available patterns, and click **Next**.
4. In the **Recognition Scope** page, choose the desired model elements. Use **Add** and **Remove** to make up the list of model elements to be examined. Click **Next**.
5. Observe the list of recognized patterns, or repeat the process with the other patterns and scope.

CAUTION: For the large projects, recognizing patterns increases memory consumption, which can result in the “out of memory” exception. Change the JVM parameters to increase memory limit. To do that, restart the application with the following Eclipse command line argument: -vmagrs -Xmx600M, thus adding 600 Mb to RAM. You can try the different additional memory values, depending on your hardware configuration.

Recognition results are presented in a dialog allowing you to choose instances that should be created and stored in the project model. After some of them are selected new model entities are created in a special model node called "Patterns" and grouped by pattern definition instances in the Pattern Explorer view. References to pattern instances from model can be created on any diagram by using the Add shortcut command.

Related Concepts

Pattern Recognition

Validating Pattern Definition Project

To avoid errors and inconsistencies you need to check if the pattern you designed is valid and can be compiled before creating a pattern definition from your Pattern Definition project.

The following is checked when performing pattern validation:

- Every instance specification in the project has non-null classifier assigned via InstantiatesLink. The classifier is a class from the “metamodels” root of the model.
- Every slot of every instance in the project has defining feature which can be found as an attribute of the class representing the metaclassifier of the slot’s owning instance specification.
- Slots are assigned only those values that are acceptable for corresponding features.
- Every instance specification representing a link must be tied by an aggregation-shaped link (PatternDefinitionAssociationLink with property instanceAggregated set to true) to an instance of non-link metaclass.
- Every link instance must have two participants attached to it by means of PatternDefinitionAssociationLink with property instanceAggregated set to false. Link stereotype must be set to client or supplier.
- Link client and supplier attached to a link instance by association links must comply to metamodel description (classifiers of the link participants must inherit metaclasses mentioned in the descriptions of the corresponding link participant role).

- No cyclical aggregations allowed in the pattern definition model.
- Each constraint in the pattern definition project must have constraintType property set to the name of one of the available constraints.
- Each constraint must be linked by ConstraintParameterLinks to all parameters this constraint is checked against. Each such link must have parameter role set in the parameterRole property. Roles of the parameter links must not duplicate one another, all parameters of the constraint should be defined.
- Union of all participant sets selected for pattern parts should not be equal to the set of all participants in the definition.

To validate a pattern definition

1. In Model Navigator view select your Pattern definition project
2. Right click the selection and choose **Patterns Check pattern definition project validity** from the context menu. The validation results are displayed in the **Pattern definition validation results** view.

Verifying Pattern Instances

If modification of a pattern instance violates its validity, such pattern instance shows red in diagram. Verification of a pattern instance helps update relationships between the pattern and participants, and make the pattern valid, if possible. In case such update is not possible, a warning message is displayed.

To verify a pattern instance

1. Right-click the desired pattern oval in diagram.
2. On the context menu, choose **Patterns Verify pattern**

   TIP: Alternately, choose **Verify pattern** on the context menu of pattern in the **Pattern Explorer**. You might want to get rid of invalid pattern instances. Use the Pattern Explorer context menu command.

To delete invalid pattern instances

1. In the **Pattern Explorer**, right-click on a pattern node.
2. On the context menu of the node, choose **Clear Invalid Instances**.

Related Concepts

Patterns

Working with the Pattern Instances

Managing pattern instances involves the following procedures

1. Create a pattern instance in diagram:

   Creating Model Element by Pattern

2. Create additional parts in a pattern:

   Adding a Pattern Part

3. Check validity of a pattern instance:

   Verifying Pattern Instances
4. Examine a diagram for patterns:

   Recognizing Patterns
5. Delete pattern instances

   Deleting Patterns Instances

Related Concepts

Patterns

Creating Model Element by Pattern

To create model elements by pattern

1. On the main menu choose **File > New > Other**.
2. Expand the **Modeling** node and select **Model element by pattern**.
3. Select patterns to apply and click **Next.** You can optionally define the values for user-editable properties of the pattern and select whether to add pattern instance into the model. If you choose to add pattern instance, the new entity appears in the Patterns root or your project and in the Pattern Explorer, and the oval appear in diagram.
4. Click **Finish** to create elements based on the selected pattern definition.

   TIP: If your diagram looks entangled after adding pattern instances, use **Layout > Layout All** command on the diagram context menu.

Related Concepts

Patterns

Adding a Pattern Part

To add a part to a pattern

1. Right-click an oval that represents pattern instance in diagram.
2. On the context menu of the pattern instance, choose **Create pattern part**.
3. On the submenu of this menu node, select the desired part ti be created. **Create Pattern Part** wizard opens.
4. On the first page of the wizard, specify the target package where the part will be created. Click **Next**.
5. On the second page of the wizard, specify properties for the new pattern part, using the **Set values** section.
6. Click **Finish**.

Related Concepts

Patterns

Verifying Pattern Instances

If modification of a pattern instance violates its validity, such pattern instance shows red in diagram. Verification of a pattern instance helps update relationships between the pattern and participants, and make the pattern valid, if possible. In case such update is not possible, a warning message is displayed.

To verify a pattern instance

1 Right-click the desired pattern oval in diagram.

2 On the context menu, choose **Patterns Verify pattern**

TIP: Alternately, choose **Verify pattern** on the context menu of pattern in the **Pattern Explorer**. You might want to get rid of invalid pattern instances. Use the Pattern Explorer context menu command.

To delete invalid pattern instances

1 In the **Pattern Explorer**, right-click on a pattern node.

2 On the context menu of the node, choose **Clear Invalid Instances**.

Related Concepts

Patterns

Recognizing Patterns

You can examine your project for patterns.

To recognize patterns in a project

1 Select a project in the Model Navigator.

2 On the context menu of the project, choose **Patterns Recognize Patterns. Recognize Patterns** wizard opens.

3 In the **Choose Pattern Definitions** page, choose definitions of the patterns you want to find in the project. To do that, check the desired patterns in the tree of available patterns, and click **Next**.

4 In the **Recognition Scope** page, choose the desired model elements. Use **Add** and **Remove** to make up the list of model elements to be examined. Click **Next**.

5 Observe the list of recognized patterns, or repeat the process with the other patterns and scope.

CAUTION: For the large projects, recognizing patterns increases memory consumption, which can result in the “out of memory” exception. Change the JVM parameters to increase memory limit. To do that, restart the application with the following Eclipse command line argument: -vmagrs -Xmx600M, thus adding 600 Mb to RAM. You can try the different additional memory values, depending on your hardware configuration.

Recognition results are presented in a dialog allowing you to choose instances that should be created and stored in the project model. After some of them are selected new model entities are created in a special model node called "Patterns" and grouped by pattern definition instances in the Pattern Explorer view. References to pattern instances from model can be created on any diagram by using the Add shortcut command.

Related Concepts

Pattern Recognition

Deleting Patterns Instances

You can delete elements of the patterns, using Diagram Editor, the Model Navigator, or the **Pattern Explorer**.

To delete a pattern instance

1. In the Diagram Editor or Model Navigator, select the pattern instance to be deleted.
2. On the context menu choose **Delete** to delete the selected pattern instance from the diagram and model.

   TIP: Alternately, select the desired pattern instance in the **Pattern Explorer**, and choose **Delete instances** on the context menu. The selected instance is deleted from the model.

To delete all instances of a pattern

1. In the **Pattern Explorer**, select the desired pattern node.
2. On the context menu of the selection, choose **Delete instances**.

Related Concepts

Patterns

Quality Assurance

This section provides how-to information on using Audits and Metrics.

Creating a Metrics Chart	How to create a chart for Quality Assurance metric results.
Creating and Using Code QA Sets	How to create and use your own QA sets
Exporting and Importing Model Audits/Metrics	How to export and import model audits and metrics.
Exporting QA Results	How to export audit and metric results to an XML or HTML file(s) to share them with team members or to review them later.
Flagging Audits in Code	How to flag audits in code.
Generating QA Report	How to create a QA report on you audits or metrics data.
Grouping and Ungrouping	How to group and ungroup audit results.
Hiding and Showing Audit Results	How to hide and unhide audit results
Navigating to Problems	How to navigate to problems listed in QA results.

Refreshing QA Results	How to refresh the QA results table.
Saving and Loading Audit Results	How to save and load audit results.
Saving and Loading Metric Results	How to save and load metric results.
Specifying Quality Assurance Preferences	How to perform quality assurance tasks.
Using OCL in Model Audits and Metrics	How to use OCL expressions in Audits and Metrics.
Using QA History	How to use QA results history.
Viewing Audit Results	How to view audit results.
Viewing Metric Results	How to view metric results.
Viewing Metrics as Graphs	How to view metrics as graphs.
Viewing Problem Detection Audits (Detection Metrics)	How to view Problem Detection Audits.

Creating a Metrics Chart

You can create a chart in the **Metric Results Pane**.

Metrics charts are created in temporary files which are deleted when the charts are closed. However, you can save graphical information in text files, export it to the desired graphical format, and include graphics in project.

To create a Kiviat chart

1. Select the row that contains the results for the desired element.
2. Right-click and choose **Kiviat Graph**.

To save a chart as image

1. In the chart view pull-down menu choose **Save Image As BMP Picture...** or **Save Image As SVG Picture...**.
2. In the **Save Chart As Image** dialog, navigate to the target location and click **Save**.

To print a chart

1. Browse to the chart you want to print.
2. In the chart view pull-down menu choose **Print...**.

Related Concepts

Quality Assurance

Grouping and Ungrouping

You can group the results of generated audits using one of five different categories:

- Severity
- Description
- Resource

- In Folder

To group audit results:

1. Right click the audit results table.
2. Select **Group by** and a category.
3. The results are grouped based on your choice.

To ungroup audit results:

1. Right click the audit results table.
2. Select **UnGroup**.

Hiding and Showing Audit Results

You can hide or show specific audit results using the following criteria:

- Row(s)
- Description
- Resource
- Folder

To hide results:

1. Right click the audit results table.
2. Select **Hide Selected**.
3. Select a category from the submenu.
4. That category is hidden based on the audit row selected.

   TIP: Use SHIFT+CLICK or CTRL+CLICK to select more than one audit row.

To show hidden results:

1. Right click the audit results table.
2. Select **Show All Hidden**.

Navigating to Problems

You can jump directly from problems identified in the QA results table to the corresponding section of your code.

To navigate to the corresponding section of your code from the QA results table

1. Right click the result.
2. Select **Go To.** Alternatively double click the result in the table. This will open the source file in the editor and highlight the problem line. If the source file is already open in the editor, it brings that tab to the front.

Refreshing QA Results

To rerun the same QA check on selected resources from the QA results table:

- Click **Refresh** on the **Audit** view toolbar.

Specifying Quality Assurance Preferences

A full range of code and model audits and metrics are provided to run against your project. You can perform the following:

- Select which audits and metrics values to run against your project.
- Save customized "QA sets" to run against future builds.
- Create reports based on QA results.
- Include QA results in project documentation.
- View QA results as Kiviat.
- Sort, search, and copy your QA results.
- Alter the values used for quality assurance.

NOTE: Model audits and metrics results display in the Model Audits View and Model Metrics View.

To open Quality Assurance preferences

1. Choose **Window > Preferences** from the main menu.
2. Choose **Window > Preferences >Modeling> QA Model** for model audits and metrics. The Quality Assurance Preferences contain lists of audits and metrics that you can select as well as tools for saving and loading (exporting and importing for model) QA sets.

Related Procedures

Adding a Single Model Element to a Diagram

Using OCL in Model Audits and Metrics

You can run audits and metrics in your design model using OCL expressions.

You can create custom OCL audits and metrics that operate with metamodel types and run them against the model that is an instance of the same metamodel. ER/Studio Software Architect also contains a set of sample audits (the ideas of most of them are taken from Ambler and Fowler books). These audits can be used as examples for custom rules creation.

To define audits and metrics

1. Choose **Window > Preferences** from the menu. The **Preferences** dialog displays.
2. Expand the **Modeling** node and select **QA Source**.
3. Select either **Audits** or **Metrics** tab.
4. Click **New** to add an audit or metric. The **Edit Audit** or **Edit Metric** dialog displays respectively.

5 Specify your audit or metric name, description, severity, and select the context of the OCL expression. The code for your new audit or metric is displayed in the standard OCL editor in the **Body** text area. The audit expression should be a valid invariant that returns Boolean. Each metric expression should return Integer value.

To run defined audits and metrics

1 In the diagram, select model elements against which you want to run audits or metrics.

2 Select **Model Run Model Audits** or **Model Run Model Metrics** from the main menu. The results display in **Model Audit** or **Model Metrics** view respectively.

NOTE: The scope depends on the current selection made on diagram: if the project default diagram is selected, the entire project will be checked. If a single element is selected, this element will be checked only.

Related Concepts

OCL Support

Using QA History

Each time you run QA, a history of the results are kept.

To use the QA results history

1 On the **Audit** or **Metric** view toolbar, locate the icon that looks like a clock.

2 Click the down arrow next to the icon to see a list of previous results.

NOTE: This list only applies to your current session of ER/Studio Software Architect. Once you close the application, the list is deleted. You can also clear the history by selecting the Clear History command beneath the list.

3 Select an item in the list to display those results in the table.

Using Version Control and Teams

This section describes the use of Version Control Systems (VCS). You can use VCS to keep track of changes and store versions of the work you do.

Comparing and Merging Shared Models	How to compare and merge models shared with VCS.
Setting Up Repositories	How to set up repositories.
Sharing Projects	Describes the sharing project procedures and tips.

Comparing and Merging Shared Models

Use the **Synchronize** view to compare shared models.

To compare and merge shared models

1 In the Team **Synchronize** view, select a model or model element which version stored in the repository you want to compare with the local version.

2 Choose **Model > Compare With > Local Version** from the main menu

TIP: Alternately, right-click a model or model element and choose **Open In Model Compare Dialog** from the context menu.

The comparison results display in the **Model Compare** dialog.

3 Review the differences, apply your changes, and then commit the model to the repository using menu commands specific to your VCS.

NOTE: When merging a shared model, you can change your local version only. The models consist of a large number of files, so you need to have all of these files locally.

CAUTION: A merge is performed on the model level, and existing file conflicts may still remain after the model merge. To commit these changes, use the "forced commit" mechanism provided by your version control system, e.g. the "Override and Commit" option in CVS.

Related Concepts

Model Compare and Merge

Related Procedures

Merging Models

Setting Up Repositories

The following steps describe how to set up CVS repositories for use in ER/Studio Software Architect. This is a prerequisite to sharing projects.

To open the CVS Repository Exploring Perspective

1 From the main menu, select **Window > Open Perspective > Other**. The **Select Perspective** dialog opens.

2 Choose **CVS Repository Exploring** from the list, and click OK. Using this perspective, you can add repositories for various VCS systems. Before you share a project, you need to define the repositories that ER/Studio Software Architect will use. Prior to this, the CVS Repositories view is blank. Use the **Add CVS Repository** wizard to define a repository.

To define a repository

1. Right click in an empty area of the **CVS Repositories** view, and select **New Repository Location**. The **Add CVS Repository** wizard opens.
2. Provide the information required by the wizard as listed below. When finished, click **Next**.

   This list provides option descriptions.

   - **Host** – Type the host name for your CVS server. For example, if your login command begins with: CVS -d :pserver:/jane.doe@CVS-host, enter: CVS-host
   - **Repository path** – Enter the CVS repository as you would in the pserver section of the CVS login command. For example, if your login command begins with: CVS -d :pserver:/jane.doe@CVS-host/ repository_alias, enter: /repository_alias
   - **User** – Enter the user name. For example, if your login command begins with: CVS -d :pserver:/ jane.doe@CVS-host/repository_alias, enter: jane.doe
   - **Password** – Enter your valid password.
   - **Connection type** – Choose from the list: pserver, ext
   - **Use Default Port** and **Use Port** – Select **Use Port** to define a custom port for the connection. **Use Default Port** is enabled by default.
   - **Validate Connection on Finish** – This option is checked by default. Leaving the option checked allows you to attempt to connect with the host server to ensure that all information was entered correctly.
   - **Save Password** – Check to save your password locally on your computer. Note the warning message about the password file at the bottom of the wizard.
3. Click **Finish**. The **CVS Repositories** view is updated with the new CVS repository location.

   For instructions on sharing a project, refer to Sharing Projects topic. Consult the documentation set provided with your IDE for complete details on using version control. From the main menu, choose **Help > Help Contents**. You can find more information on CVS at http://ximbiot.com/cvs/wiki/index.php?title=Main_Page.

Related Procedures

Generating Project Documentation

Generating Project Documentation

This section provides how-to information on using Documentation Generation facilities.

Configuring the Documentation Generation Facility	How to configure the Documentation Generation facility.
Generating HTML Documentation	How to generate project documentation in HTML format, by the predefined template.
Generating Project Documentation from Command Line	How to generate project documentation using batch process.
Generating Project Documentation Using Template	How to generate project documentation in any supported format, using the desired template.

Configuring the Documentation Generation Facility

To configure the documentation generation facility

1. On the main menu, choose **Window > Preferences > Generate Documentation**.
2. Under the Generate HTML category, enter the options for HTML documentation: classes and members to be included in the documentation tags to be included in the documentation title, window title, header, and footer.
3. Under the HTML Output category, choose the option of processing line breaks.
4. Under the RTF Output category, set up text formatting options

Generating HTML Documentation

In this section you will learn how to generate project documentation in HTML format, using the default template, supplied with the product.

To generate HTML project documentation

1. Select **Model > Documentation > Generate HTML** on the main menu.
2. In the **Generate HTML Documentation** dialog that opens, specify the output folder, select your preferred **Scope** and **Options** settings.
3. Click **Finish** to generate documentation.

Related Procedures

Configuring the Documentation Generation Facility

Generating Project Documentation from Command Line

You can update project documentation as part of a periodic automated build process by having the process script call the documentation generator in ER/Studio Software Architect via the command line interface.

The following utilities are provided that enable you to generate project documentation without launching the product:

- genhtml for generating HTML documentation, with the launchers genhtml.cmd
- gendoc for generating documentation by template, in one of the supported output formats, with the launchers gendoc.cmd

To generate project documentation using the documentation generation utility

1. Use the following command: <utility> [project filename] [options] [packagenames]
2. Specify parameters and options as described in the utility references.

   NOTE: If you run the utility without parameters, documentation is generated for the entire workspace and stored in the out subfolder of the workspace.

Generating Project Documentation Using Template

In this section you will learn how to generate project documentation in the desired format, using one of the default or custom templates of your choice.

To generate project documentation using template

1. Select **Model > Documentation > Generate Using Template** on the main menu.
2. In the **Generate Documentation Using Template** dialog that opens, specify the following settings:
   - In the **Output Path** field enter the fully qualified path to the target folder. Alternatively, use **browse** .
   - In the **Format** field, select the desired output format from the drop-down list.
   - In the **Templates** section, select the desired Default or Custom template.
   - In the **Scope** section, click radio button for the desired scope. Note that you can generate documentation for all open projects in the workspace, for a single project, or for the current package or diagram.
   - In the **Include** section, select artifacts to be included in the generated output. Note that you can include audit results
   - If you wish to open the generated report immediately, check **Open in Viewer** option.
3. Click **Finish** to generate documentation.

Related Procedures

Configuring the Documentation Generation Facility

Documentation Templates Procedures

This section provides how-to information on creating and editing custom documentation templates using the Documentation Template Designer.

A Typical Scenario of Creating a Custom Documentation Template	This topic outlines general steps involved in creating a custom documentation template.
A Typical Scenario of Creating a Template for Multi-Frame Documentation	This part discusses how to use the Documentation Template Designer to create the templates for multiframe HTML documentation.
Creating Controls	This topic describes how to create controls in a section of a documentation template.
Creating Custom Documentation Template	In this section you will learn how to create and edit formatting styles.
Creating Formatting Styles for Documentation Templates	In this section you will learn how to create and edit formatting styles
Creating Hypertext Links (Advanced)	How to create hypertext links for multi-frame documentation.
Creating Javadoc Link References (Advanced)	How to convert Javadoc tags into link references
Creating Sections	This topic describes how to create sections in a documentation template, using the Template Designer.
Creating Stock Sections	How to create and delete a stock sections.

Defining Frameset Structure	How to define the frameset structure of a multi-frame HTML document, which describes how the frames are organized within the browser window.
Hyperlinking Controls to Element Documentation	How to hyperlink controls with the model elements' documentation.
Hyperlinking Documentation	How to create hypertext links in the ordinary and multi-frame documentation.
Image Mapping (Advanced)	How to create image maps from the diagram images.
Moving, Resizing and Aligning Controls	This topic describes how to change size and location of controls in the sections of a documentation template.
OCL In Documentation Templates	How to use OCL expressions in the templates for generating project documentation.
Setting Area Properties	This topic describes how to define area properties of a static section.
Setting Call to Template Section Properties	How to set output properties of a call to template section.
Setting frame and Frameset Properties	How to define properties of a frameset or a frame within a Frameset Template.
Setting Section Properties	Once a section is created, define its properties.
Setting Template Properties	Once a documentation template is created, you can define its properties.
Using Word Documents in Documentation Templates	It is possible to use styles, headers and footers of the Word documents in the custom documentation templates. In this section you will learn how to attach and detach a Word document.

A Typical Scenario of Creating a Custom Documentation Template

Creating a custom documentation template and populating it with sections and controls involves the following general steps:

Step	Link
Create a stub template	Creating a Custom Documentatin Template
Create template structure	Creating Sections
Provide reusable elements	Creting Stock Sections
Define sections properties	Setting Section Properties
Define area properties	Setting Area Properties
Define template properties	Setting Template Properties
Add controls to the Sections	Creating Controls
Customize controls	Moving, Resizing and Aligning Controls

A Typical Scenario of Creating a Template for Multi-Frame

Documentation

This part discusses how to use the Documentation Template Designer to create the templates for multi-frame HTML documentation.

To create a template for multi-frame documentation:

Step	Link
Create a stub multi-frame template	Creating Custom Documentation Template
Create the frameset structure, to describe how the frames are organized within the browser window:	Defining Frameset Structure
Design the body of a frameset template, keeping in mind that for the multi-frame templates static sections and headers and footers for the folder sections and iterators are prohibited. Create controls. Set section, area and template properties as described in the section	A Typical Scenario of Creating a Custom Documentation Template
Set call to template section properties:	Setting Call to Template Section Properties
Create hypertext links, including image maps and Javadoc link references:	Hyperlinking Documentation

Creating Controls

To create a new control in a static section, header, or footer

1. Select a section in the details pain.
2. Click the type of control you want to insert on the template designer toolbar.

   TIP: Alternatively you can right-click the section in the details pane, point to **Insert Control** and select the type of control from the menu.

3. On the context menu of a control, choose **Properties** and fill in the control properties in the tabbed **<Control> Properties** dialog box.
4. Click **OK**.

   NOTE: **Insert Control** command is also available on the context menu of a panel control.

Creating Custom Documentation Template

To create a documentation template

1. On the main menu, choose **File > New > Other**
2. In the **New** dialog that opens, expand the **Modeling** node.

3 Choose **Documentation Template** and click **Next**.

4 In the **GenDoc Template Designer File** dialog, specify the following:

- In the **File Name** field, enter the fully qualified name of the template file.
- In the **Template type** field, choose the desired template type from the drop-down list (documentation template or frameset template).

5 Use the Template Designer toolbox or context menus to create the desired template structure.

NOTE: Once a documentation template is created, you can view and modify its properties using the **Template Properties** dialog.

Creating Formatting Styles for Documentation Templates

In this section you will learn how to create and edit formatting styles.

TIP: Create as many Formatting Style types as you would like, then assign the desired Formatting Style to a Control.Once the Formatting Style has been assigned to controls, to change any style properties for these controls,change the property in the appropriate Formatting Style. Once the Formatting Style is updated, the change will show immediately in the controls.

To create a new formatting style

1 On the tool palette of the **Template Designer**, click **Show Template Properties** button. **Template Properties** dialog opens.

2 In the **Formatting Styles** tab, click the **New** button. **Style** dialog opens.

3 Select **Main** tab of the dialog, and specify the following parameters: In the **Name** field, specify the style name. As you enter the name, it displays in the title bar of the **Style** dialog. In the **Type** field, choose whether the style applies to a paragraph, or to a character. In the **Level** field, choose the nesting level of the style.

4 In the Font, Color and Border tabs, define the respective parameters of the style.

5 Click **OK**.

To edit formatting style

1 On the tool palette of the **Template Designer**, click **Show Template Properties** button. **Template Properties** dialog opens.

2 In the **Formatting Styles** tab, select the desired style and click the **Edit** button. **Style** dialog opens.

3 Repeat steps 3 to 5 of the previous task.

Creating Hypertext Links (Advanced)

Multi-frame HTML documentation requires hypertext links. A hypertext link connects a link reference (source) and a link destination (target). The link reference is a piece of text or an image; it is the property of a control. The link destination is a file or an anchor in a file; it is the area property of a static section, header or footer.

By default, if no target frame for a hyper-reference is specified, the referenced document is loaded into the same frame window as the page that contains the link reference. Target frame parameter alters this behavior to load the target file into a named frame, so the source document is not replaced.

To assign a target frame to a link reference

1. On the context menu of the desired control, choose **Properties**.
2. Select the **Hyperlink to Elements** tab
3. Click the **Edit Expression** button next to the **Target Frame Name Expression** field.
4. Select the desired notation and enter an expression for the name of a frame window defined in the frameset structure. The expression should return the name of one of the frame windows defined in the FrameSet Structure.

To create a link reference for a control

1. Create a target from a section as described in the section *Hyperlinking Controls to Element Documentation*
2. Right-click the desired control and choose **Properties** on the context menu.
3. In the **Hyperlinks to Elements** tab, click radio button **Link to Element's Specific Doc**.
4. Click the **Edit Expression** button next to the **Expression for Model Element** field to determine which elements' documentation is the link target.
5. If the target area is marked as a Documentation Subject Selector, click the **Edit Expression** button next to the **Expression for Documentation Subject Selector** field to match the expression for template area described above.

Creating Javadoc Link References (Advanced)

The Documentation Template Designer provides conversions for Javadoc References represented in the following forms:

— inside {@link} tags

— as the value of some Javadoc element's properties

You need to specify the conversion in the properties of the control.

To convert a {@link} tag

1. On the context menu of the desired image control, choose **Properties**.
2. Select **Other** tab.
3. Check the option **Render Embedded Javadoc Tags**.

   TIP: Only a text control (label control, data control, or formula control) can generate documentation text.

Converting a value of an element's property to a hyperlink is more complicated than converting an {@link} tag. Such conversions require using one of these documentation generation functions:

getJDRefDisplayName()

getJDRefElement()

The following procedure gives a general outline of actions required to perform conversion. Most often this kind of conversion is used for the see property.

To convert the value of an element property

1. Create a property iterator that will go through the instances of the property.
2. Create one or more static sections that correspond to the Javadoc references of the desired type.
3. Provide enable condition for each section, which activates it for the appropriate JDRef.
   getJDRefType(getDGVariable("curPropertyInstance")) == "<JDRef type>" where "<JDRef type>" is "element", "URL" or "text".
4. Create formula control in each section.
5. On the context menu of a formula control, choose **Properties**.
6. In the **Formula** tab of the **Formula Control** dialog, enter the following expression:
   getJDRefDisplayName(getDGVariable("curPropertyInstance"))

   NOTE: This expression is common for all types of Javadoc references. The value of the expression is the text that will be displayed in the documentation.

7. In the **Hyperlinks to Elements** tab of the **Formula Control** dialog, select the hyperlink type:
   - For the JDRefs of the "element" type, click the radio button **Link to Element's Specific Docs**, and enter the following expression in the **Expression for model element** field: getJDRefDisplayName(getDGVariable ("curPropertyInstance"))
   - For the JDRefs of the "URL" type, click the radio button **URL Link** , and enter the following expression in the **Expression for URL** field: getJDRefURL(getDGVariable("curPropertyInstance"))
   - For the JDRefs of the "text" type hyperlink definition is not defined. Since such a JDRef does not refer to any element, the function getJDRefElement() always returns null, producing no hyperlinks.

Creating Sections

A nested section can be created for an existing folder or iterator; sibling section can be created for any existing section. For these sections **Insert Nested Section** or **Insert Sibling Section** toolbar buttons and menu commands are enabled, unlike the report header and footer, which lie outside the template body.

To create a new section in a documentation template

1. Select an existing section in the template.
2. Right-click an existing section in the scope pane, point to **Insert Sibling Section** or **Insert Nested Section** on the context menu and select the type of new section.

   TIP: Alternately use the toolbar buttons of the Documentation Template Designer.

3. When necessary, determine the essential template information for the new section.
   - **Static section**: None required at creation.
   - **Element iterator**: Select the element metatype from the list of available types.
   - **Element property iterator**: Select the scope.
   - **Folder section**: None required at creation.
   - **Call to stock section**: Select the stock section from the list of existing sections. Refer to the section *Creating Stock Sections*.
   - **Call to template section**: None required at creation.

4. Click **OK** to complete the insertion.
5. Set section properties as required. Iterators and folder sections can also contain headers and footers. If such section does not contain a header or a footer, its context menu provides **Add Header** or **Add Footer** commands. If a header or a footer exists, the context menu provides **Delete Header** or **Delete Footer** commands.

To add a header or a footer to an iterator

1. Select an existing element iterator or folder section.
2. On the context menu, choose **Add Header** or **Add Footer**.

Creating Stock Sections

Stock sections are reusable folders or iterators that reside in the template's collection of stock sections. Each stock section displays in a separate named tab in a documentation template.

In this section you will learn how to:

- Create stock sections from scratch
- Create stock sections from an existing section
- Delete stock sections
- Show stock section

To create a stock section

1. On the toolbar of the **Template Designer**, click the **New Stock Section (Element Iterator)** button or **New stock section (Folder) button**. Dialog **New Stock Section** opens.
2. In the **Name** field, enter the name of the new stock section.
3. For the element iterators, select the desired metatype from the list of available metatypes.
4. Check the option **Intrinsic to Property Iterator**, if necessary. If this flag is checked, the scope of stock section root iterator or folder section depends on the Properties Iterator this section is called from: the only available iteration scopes for an element iterator are *customized* and *programmed*. For folder sections, it means that the metatype chooser tab is absent.
5. Click **OK**.

To create a stock section from an existing section

1. In the scope pane of a template, select the desired element iterator or folder section, from which you want to create a stock section.
2. Right-click the section and choose **Copy into Stock** on the context menu.
3. In the **New stock section (Folder)** dialog, enter the stock section name.
4. Click **OK**.

To delete a stock section

1. Right-click on the stock section tab be deleted.
2. On the context menu of the tab, choose **Remove Stock Section**.

To show a stock section for a call to stock section

1. Right-click on the call stock section.
2. On the context menu of the section, choose **Show Stock Section**.

Defining Frameset Structure

The frameset structure of a multi-frame HTML document describes how the frames are organized within the browser window. Once a frameset template is created, you can define its structure through the template properties.

To define structure of a frameset template

1. On the toolbar of the Template Designer, click the **Show Properties** button. **Template Properties** dialog opens, with the root frameset highlighted.
2. Define template properties in the **General** and **Template Parameter** tabs, as described in the section

   [Setting Template Properties]
3. In the **Frameset Structure** tab, choose layout of the root frameset template, clicking one of the radio-buttons **Columns** or **Rows**.
4. Add a frame or a frameset to the root frameset. To add a frame, click **Add Frame** button. To add a frameset, click **Add Frameset** button. Repeat this step to create the desired structure.
5. For each frame or frameset node, define its properties, as described in the section

   [Setting frame and Frameset Properties]
6. Click **OK** when ready.

Hyperlinking Controls to Element Documentation

Any generated output that contains an anchor or a bookmark, can be a link target. Documentation templates have facilities for inserting anchors at the “main documentation” of model elements. You can insert anchors for static sections, headers, and footers.

When you define the location of a model element main documentation, you can specify hyperlink references to it for any control that is not a panel. The control can be in the same template as the main documentation or it can be in a different template.

In this section you will learn how to:

- Make a target from a static section, footer, or header
- Link a control to a target

To make a target from a static section, footer, or header

1. Right click the details pane of the desired section and choose **Area Properties**
2. Click the **Hypertext Target** tab.
3. Specify **Expression for the Target Bookmark Selector**. When specified, this option inserts a bookmark into a file used as File Link target. Click the **Edit Expression** button to create the expression in OCL or legacy notation.

4. Define **Start of the Current Element's Specific Documentation.** If this option is checked, the output of this section is identified as the "main documentation" for the current element. This option is used for hyperlinks of Link to Element's specific Doc type.
5. Specify **Expression for the Documentation Subject Selector**. This option is only enabled if **Start of the Current Element's Specific Documentation** option is checked. It marks the location of the elements' specific documentation with the appropriate Documentation Subject Selector, allowing users to create hyper-references to different documentation locations generated by the same model element. Click the Edit Expression button to create the expression in OCL or legacy notation.

To link a control to a target

1. In the details pane of a section, select the desired label, image or formula control.
2. On the context menu of the control, choose **Properties**.
3. Click the **Hyperlinks to Element** tab.
4. Define the ink type:
   - **Link to Element's specific Doc:** You must identify the element whose documentation is to be the target in the
   - text field for Expression for RWI-Element.
   - **File Link:** You must fill in the path to the file. If the hyperlink target is a bookmark in the file, you must supply
   - that as well.
   - **URL Link:** You must supply the URL.
5. Specify the link settings, depending on the selected link type.
6. Optionally, provide compound hyperlinks, clicking the **Add Hyperlink <n>** button. This adds a new **Hyperlinks to Element** tab to the dialog.

Hyperlinking Documentation

HTML documentation requires hypertext links. A hypertext link connects a link reference (source) and a link destination (target). The link reference is a piece of text or an image. The link destination is a file or an anchor in a file.

To create hypertext links, refer to the following sections

1. Creating hyperlinks in the ordinary documentation:

   Hyperlinking Controls to Element Documentation
2. Creating hyperlinks in the multi-frame documentation:

   Creating Hypertext Links (Advanced)
3. Creating image mapping for the model elements in diagrams:

   Image Mapping (Advanced)
4. Converting Javadoc link references to hyperlinks:

   Creating Javadoc Link References (Advanced)

Image Mapping (Advanced)

When an image control is for the whole diagram in the model, the reference definition creates link references for all model elements depicted on the diagram. To create the image map, you must enter all expressions in the hyperreference definition relative to the element returned by the call context OclAny getDGRwiElement ('diagramMapElement')

When Documentation Generator generates the image of a diagram, it creates the image map, which includes all model elements depicted on the diagram. While doing this, it iterates through diagram elements, substituting the variable diagramMapElement with every diagram element, calculating a hyper-reference for it, and then, and inserting it into the image map. For example: context uml::kernel::Elementif getDGRwiElement ('diagramMapElement').isDiagram() then getDGRwiElement('diagramMapElement') else getDGRwiElement('') endif

To create an image map

1. On the context menu of the desired image control, choose **Properties**.
2. In the **Image** tab, select Diagram as an image type.
3. Select the **Hyperlink to Elements** tab
4. Click **Link to Element's Specific Doc** radio button.
5. Click the **Edit Expression** button next to the **Expression for Model Element** field and enter all expressions in the hyper-reference definition relative to the element returned by the call:

   context OclAny

   getDGRwiElement('diagramMapElement')

Moving, Resizing and Aligning Controls

The Documentation Template Designer displays a newly created control as a rectangle positioned at the insertion point in the details pane. You can move the control to change where its output appears in the generated documentation. You can also modify the size of the rectangle to determine the approximate size of the region for the output. Increasing the rectangle size is especially important for a label for which the default size is not large enough to allow its entire text to be displayed in the report.

If you select two or more controls in a section, you can use the context menu of the selection to uniformly align controls within the section. You should take care when applying multiple alignments since it is possible for controls to overlap.

TIP: Precise positioning and sizing is not possible.

CAUTION: There is no simple undo for changes in alignment or size. When you change alignment or resize controls, you must manually readjust the controls to return them to the former status.

To move a control within a section

1. Select the desired control. Notice that the mouse pointer changes to a cross with double ended arrows.
2. With the control selected, drag and drop the control to the desired position within the section.

To copy or move a control to another section

1. Select the desired control.
2. On the context menu of the selected control, choose **Copy** or **Cut**.

3 Right-click on the target section and choose **Paste** on the context menu.

To resize a control

1 Place the mouse pointer on either the right or left edge of the rectangle. Notice that the mouse pointer changes to a double-ended arrow.

2 Drag the edge of the rectangle to a desired size.

To align controls

1 Select two or more controls within a section.

2 Right-click the selection and choose **Alignment** on the context menu.

3 On the submenu, choose one of the following options:

- Left Side
- Right Side
- Top Side
- Bottom Side
- Make same width
- Make same height
- Make same size

OCL In Documentation Templates

You can compose model queries and define enable conditions using OCL syntax, and use them in a template for generating documentation. OCL or Legacy type expressions can be entered in the template element's properties dialog box using the provided Expression Editor. Where applicable the editor is either opened in the tab or you can use the Edit Expression button to open the editor.

The standard OCL operations and the special native OCL extensions are provided for the functions that are specific for Documentation Generation. Native OCL extensions mostly have the same signature and meaning as the legacy Documentation Generation functions have. Code sense suggests these operations along with the standard OCL ones.

To add an expression to your template

1 Open a template where you want to add OCL expression.

2 In the **Properties** dialog box, open the tab where you want to type the expression. If the Expression Editor is not opened in the tab, click the **Edit Expression** button.

3 Specify the context for your expression in the **Context** field.

4 In the **Body** area you can type the expression text. The code sense, syntax highlighting and validating are available.

5 Click **OK** to save the expression in the template.

Setting Area Properties

Area properties apply to static sections, headers and footers. They are defined in the **Area Properties** dialog, which is common for static sections, headers and footers, and is invoked from the details pane. Refer to the dialog descriptions for details.

To set area properties

1. Select the desired static section, header or footer of a template.
2. On the context menu of the details pane, choose **Area Properties**.
3. In the **Area Properties** dialog, specify the desired settings and click **OK**.

Setting Call to Template Section Properties

The section properties of a call to template determine how the output for a template call can be used. With multiframe HTML documentation, call to template sections typically generate separate files that can be loaded into a frame of the resulting HTML project documentation.

To access the properties of a call to template section, select **Properties** from the section’s right-click menu. Refer to the **Call to Template Properties** dialog description for details.

To define properties of a call to template section

1. In the **Template** field of the **General** tab, click the **Browse** button, and select the desired template.
2. Select the type of generated output. If the output generated from the template is to be loaded into a frame, you should select Separate File from the radio buttons.
3. Define the name of the generated document. Click the **Edit Expression** button to create the expression.

   NOTE: If a particular call of a template is to be iterated many times to produce multiple documents, you should derive the output document name from the properties of the current model element. You can use the expression getProperty("$name") to get the name of the current model element.
4. Define the name of the output directory. Click the **Edit Expression** button to create the expression.
5. Define output image subdirectory for the images files.

Setting frame and Frameset Properties

In this section you will learn how to define properties of each frame and frameset that comprise a multi-frame template.

To define properties of a frame

1. In the **Frameset Structure** tab of the **Template Properties** dialog, select the desired frame.
2. Specify the frame name, percent size and scrolling mode.
3. Click the **Edit Expression** button in the **Source File Name Expression** field. In the **Edit Expression** dialog, select expression type and enter the expression body.

4 Click the **Edit Expression** button in the **Enable Condition** field. In the **Edit Expression** dialog, select expression type and enter the expression body.

To define properties of a frameset

1 In the **Frameset Structure** tab of the **Template Properties** dialog, select the desired frameset.

2 Choose layout of a frameset, clicking one of the

3 Click the **Edit Expression** button in the **Enable Condition** field. In the **Edit Expression** dialog, select expression type and enter the expression body.

Setting Section Properties

Once a section is created, define its properties. Section properties are defined in the **Properties** dialogs, which are specific for each kind of sections.

You can invoke the Properties dialog for the iterators and folder sections from the scope pane or from the details pane of a section. For the static sections the Properties dialog is invoked from the scope pane only.

To set properties of a section

1 In the scope section of the **Template designer**, select the desired template section.

2 On the context menu, choose **Properties**.

3 Define properties as required and click **OK**.

NOTE: Properties dialogs are specific to each section type. Refer to the dialog descriptions for details.

Setting Template Properties

To set properties of a documentation template

1 On the toolbox of the **Template Designer**, click the **Show Template properties** button.

2 In the **General** tab, you can change the following:

- Enter template description
- Define the report title expression, clicking the editor button to open the **Edit Expression** dialog.
- Select Root Object Metatype from the list of available metatypes.
- Attach a Word document as a formatting template. Refer to the section *Using Word Documents in*
- *Documentation Templates* for details.
- Check or clear options to generate headers and footers, as required.

3 In the **Page Settings** tab, you can specify page size, margins, and landscape or portrait orientation.

4 In the **Formatting Styles** tab, you can change formatting styles. Refer to *Creating Formatting Styles* section for details.

5 In the **Template Parameters** tab, specify the formal parameters that will be used for calling this template from another template.

Using Word Documents in Documentation Templates

The **Template Designer** provides a way to use styles, headers and footers of the *.rtf, *.dot and *.doc Word files in your custom documentation templates. When the path to the desired Word file is specified, the styles of the referenced Word document appear in the list of formatting styles; the header and footer of the referenced Word document appear on each page of the generated report.

In this section you will learn how to:

- attach a Word document to a documentation template
- detach a Word document from a documentation templateTo attach a Word document to a documentation template

1. Open a documentation template in the **Template Designer**, or create a new one.
2. On the tool palette of the **Template Designer**, click **Show Template Properties** button. **Template Properties** dialog opens.
3. In the **Formatting Template** field, specify the path to the desired Word file.

   TIP: Alternately, click the **Browse** button and navigate to the desired Word file. The styles of the referenced document display in the **Formatting Styles** tab of the **Template Properties** dialog.

   TIP: **Delete** and **Edit** buttons do not work for the styles of the attached document.

4. Click **OK**.

CAUTION: Only simple text Word *.rtf, *.dot and *.doc templates headers/footers are supported. The headers and footers with embedded images, objects, complex formatted text and fields are not processed.

To detach a Word document from a documentation template

1. Open a documentation template in the **Template Designer**
2. On the tool palette of the **Template Designer**, click **Show Template Properties** button. **Template Properties** dialog opens.
3. Remove the path from the field **Formatting Template**
4. Click **OK**.

Interoperability and Migration

This section provides how-to information on exchanging model information between the various products of the ER/Studio Software Architect product line.

Importing a Project in XMI Format	How to import XMI data.
XMI Export and Import of the Models with Cross-Project References	You can import and export multi-root projects using XMI. Note that XMI import and export is implemented differently for UML 2.0 projects.

Importing a Project in XMI Format

You can import projects or sections of projects created in other modeling tools and saved in XMI format.

To import a project from XMI file

1 Select **File Import** on the main menu. The **Import** dialog opens.

2 Select **XMI File** and click **Next**.

3 In the **Import Project from XMI File** dialog, specify the following:

- Project to which your XMI data will be imported in the **Select destination project**.
- Full path to .xml, .xmi, or .uml2 file you want to import in the **Select source .xmi file**.

4 Click **Finish**.

NOTE: .uml2 file can be imported to UML 2.0 projects.

After you are notified that the import process is complete, you can view the results in the **Model Navigator**.

If there are any warnings produced during XMI import, the XMI Import dialog notifies you to refer to the **Task** view. To open the Task view, from the main menu, select **Window > Show View > Other >Basic > Tasks**.

Related Concepts

Model Import and Export Overview

Related Procedures

XMI Export and Import of the Models with Cross-Project References

XMI Export and Import of the Models with Cross-Project References

You can import and export multi-root projects using XMI. Note that XMI import and export is implemented differently for UML 2.0 projects.

- **UML 2.0 projects:** When a project that contains cross-project references is exported to XMI file, special files *.imports.uml2 are created for each referenced root. The exported XMI file contains references to these files. When an XMI file is imported, the resulting project contains the main model only. If the referenced roots still exist in the workspace, then the resulting UML 2.0 model recognizes them. References to the elements from these roots may be resolved only if their UINs have not been changed since export. Note that UIN may change when an element is moved.

To export a UML 2.0 project with cross-project references

1 On the main menu, choose **File Export**.

2 On the first page of the **Export Wizard**, select XMI file and click **Next**.

3 On the second page of the wizard:

- Select UML 2.0 project to be exported;
- Specify export destination;

4 Click **Finish**.

Related Concepts

Interoperability and Migration

Model Import and Export Overview

Related Procedures

Importing a Project in XMI Format

Exporting a Project to XMI Format

Keyboard Shortcuts

You can perform many diagram actions without using the mouse. You can navigate between diagrams and diagram elements, create diagram elements, use drag-and-drop operation, and more, using the keyboard only.

Navigational Shortcut Keys

Keyboard shortcuts for navigation and browsing:

Action	Shortcut	Notes
Navigate between open diagrams in the Diagram Editor	CTRL+TAB	The title of the diagram that has focus is in bold text.
Navigate between elements on a diagram	Arrow keys	
Select elements	SHIFT + arrow key	
Expand node in Model Navigator	RIGHT ARROW	
Collapse node in Model Navigator	LEFT ARROW	
Open the Properties View	F4, or ALT + ENTER	
Close current diagram	CTRL+F4	
Toggle between a selected container node and its members	PGDOW /PGUP	
Navigate between nodes or node members	Arrow keys, SHIFT + arrow keys	
In the Diagram Editor , toggle focus between selected element and diagram.	CTRL+SPACE	
Select on Diagram	CTRL+F3	

Shortcut Keys for Editing

Keyboard shortcuts for editing:

Action	Shortcut
Cut, Copy, or Paste model elements or members.	CTRL+X, CTRL+C, CTRL+V
Activate the in-place editor for a diagram element to edit, rename a member.	F2
Undo	CTRL+Z
Redo	CTRL+Y, CTRL+SHIFT+Z
Select all elements on the diagram	CTRL+A
Close the Overview window	ESC

Action	Shortcut
Add a new package to a diagram	CTRL+E
Add a new class to a diagram	CTRL+L
Add new method (operation) to a class or interface	CTRL+M
Add a new field (attribute) to a class	CTRL+W
Add a new interface to diagram	CTRL+SHIFT+L
Add shortcuts	CTRL+SHIFT+N
Add a new diagram from the Model Navigator	CTRL+SHIFT+D
Invoke **Content Assist** in the OCL editor	CTRL+SPACE

Zoom Shortcut Keys

Keyboard shortcuts for zooming the diagram image:

Action	Shortcut	Notes
Zoom in	+	Use the numeric keypad
Zoom out	-	Use the numeric keypad
Fit the entire diagram in the Diagram Editor	*	Use the numeric keypad
Display the actual size	/	Use the numeric keypad

Other Shortcut Keys

Other keyboard shortcuts:

Action	Shortcut	Notes
Open the Print Diagram dialog	CTRL+P	
Diagram update	F5	
Drag-and-drop operation	>	While the focus in on the necessary element, press this key until the move handle appears. Move the element using the arrow keys and press ENTER to drop the element.

Preferences

Threre are three different ways to gain access to the preferences available in the ER/Studio Software Architect application:

Preferences	Description
Windows Preferences	These preferences modify the basic settings for the application. These preferences also include the Diagram and UML Profile Preferences.
Diagram Preferences	You can gain access to this specific set of preferences by clicking **Diagram > Preferences**. You can set properties that pertain specifically to diagram including: layout, printing, and view management
UML Profile Preferences	You can gain access to this specific set of preferences by clicking **Model > Profile > Preferences**. You can set properties for UML 2.0 profiles.

Windows Preferences

The following pages of the **Preference** dialog enable you to modify the basic settings

TIP: Some preference pages allow you to use keyboard shortcuts to set preferences. If you see a label with an underlined letter, you can use ALT + the underlined letter (for example, ALT + K) to quickly set the preference.

Preference	Description
General	Set preferences for appearances, compare/patch, content types, editors, keys, network connections, perspectives, security, startup and shutdown, web browser and workspace settings
Embarcadero Licensing	Set preferences for automatic license updates.
Generate Documentation	Use **Generate Documentation** node to define options for generated documentation.
Help	Specify how you want the help information to be displayed
Modeling	On the **Modeling** page you can change deletion, ignored folders, and team sharing preferences.
Team	Set general team preferences, and perspectives.

Restore/Apply

All of the Preference dialogs have the following options:

Item	Description
Restore Defaults	Restores default selections for this dialog.
Apply	Applies selections made in this dialog.

General

Option	Description
Always run in background	Turn this option on to perform long running operations in the background without blocking you from doing other work.
Keep next/previous editor, view and perspectives dialog open	If this option is turned on then the editor and view cycle dialogs will remain open when their activation key is let go. Normally the dialog closes as soon as the key combination is release
Show heap status	When this preference is selected, a new item will appear within the window status bar which shows the amount of heap space currently being used.
Open Mode	When enabled, a double click (or a single click depending on your Open Mode strategy set in Preferences > General) will open a compare editor. When it is disabled, clicking on a revision will open that revision.

Appearance

Option	Description
Current Presentation	Specify the currently active presentation (look and feel). You have three choices: • Default (current) • Classic Presentation • Eclipse 2.1 Style Presentation
Override presentation settings	If you select this option you can change the following settings: • Editor tab positions • View tab positions • Perspective switches positions
Show text on the perspective bar	Specify whether labels should be shown in the perspective bar as well as icons

Option	Description
Current Theme	Specify the currently active theme (color and font set) Default (current) Reduced Palette Classic Theme
Description	
Show traditional style tabs	Specify whether traditional (square) tabs should be used in place of the curved tabs.
Enable colored labels	Enables the support for colored labels in the proposal popup.

Color and Fonts

Option	Description
Color and Fonts	You can change colors for the following: Active hyperlink text color Content Assist background color Content Assist foreground color Counter color Decoration color Error text color Hyperlink text color Qualifier information color You can change the following font settings: Banner font Dialog Font Header Font Text Font
Label Decorations	Select this option to show extra information about an item on its label or icon. Each item has a specific set of decorations. **CVS**: Shows CVS specific information on resources in projects under CVS control. Information includes the revision number, branch or version name, etc. To customize the CVS decorator go to Team>CVS>Label Decorations. **File Icons Based on Content Analysis**: Displays an icon based on the examination of the contents of a file. This yields a more precise answer than one derived simply from the file name. **Linked Resources**: Adds an icon decoration to linked resources.

Compare/Patch

Option	Description
General tab	
Open Structure compare automatically	This option controls whether a structure compare is automatically performed whenever a content compare is done. Turn this option off if you don't want to see the structural differences.
Show structure compare in Outline view when possible	If this option is on, structure compare will be displayed in the Outline view whenever it is possible.
Show additional compare information in the status line	If this option is on, additional information about a change is shown in the status line. Turn this option on if you are interested in additional information about a change.
Ignore white space	This option controls whether or not whitespace change are shown in the compare viewer. Turn this option on if you want to see changes in whitespace.
Automatically save dirty editors before browsing patches	This option controls whether any unsaved changes are automatically saved before a patch is applied. Turn this option on if you want to save changes automatically.
Enter regular expressions used to identify added or removed lines in a patch	These options control which lines should be counted as added and removed lines when applying a patch. Both options are based on regular expressions.
Enter member names that should be excluded from “Compare with Each other”.	This option allows you to filter members that should be excluded from 'Compare With Each Other'.
Filtered Members	This option allows you to filter members that should be excluded from 'Compare With Each Other'
Text Compare tab	
Synchronize scrolling between panes in compare viewers.	The two comparison viewers will "lock scroll" along with one another in order to keep identical and corresponding portions of the code in each pane side-by-side. Turn this option off if you do not want the compare viewers to lock scroll.

Option	Description
Initially show ancestor pane	Sometimes you want to compare two versions of a resource with the previous version from which they were both derived. This is called their *common ancestor*, and it appears in its own comparison pane during a three way compare. Turn this option on if you want the ancestor pane to always appear at the start of a comparison.
Show pseudo conflicts	Displays pseudo conflicts, which occur when two developers make the same change (for example, both add or remove the exact same line of code or comment). Turn this option on if you want pseudo conflicts to appear in compare browsers.
Connect ranges with single line	Controls whether differing ranges are visually connected by a single line or a range delimited by two lines. Selected by default
Highlight individual changes	Controls whether the individual changes inside conflicts are highlighted.
When the end/beginning is reached while navigating an element	Use this option to configure what occurs when the end/beginning is reached while navigating an element. **Prompt:** If this option is on and you selected to compare a single element you will be asked whether you want to go to the beginning/end of the element after the end/beginning is reached. If you are comparing two or more elements you will be asked whether you want to go to the beginning/end of the current element or to go to the next/previous element. Moreover, if you choose to remember your decision, this option will be changed to one of the below respectively. **Loop back to the beginning/end:** When this option is on, the selection will me moved back to the beginning/end after you reach the end/beginning of an element. **Go to the next/previous element:** If you are comparing two or more elements and this option is on after you reach the end/beginning of an element the next/previous element will be opened.
Preview	Local Remote

Content Types

The **General > Content Types** preference page enables you to edit content types and their associated file names and character sets. You can also associate arbitrary file names or file extensions with content types. A *content type* acts as a description of a certain class of files (for instance, XML files). Eclipse uses this description in various scenarios, such as editor look-ups and file comparisons.

Editors

Option	Description
Show multiple editor tabs	Specifies whether you wish to show multiple editor tabs. If off, editor workbooks have one large tab and all non-visible editors are accessible only from the chevron.
Allow in-place system editors	
Restore editor state on startup	Specify if a prompt dialog should be opened or if a new editor should be opened when all editors are "dirty" (have unsaved changes).
Prompt to save on close even if still open elsewhere	Opens a prompt to save even if the file is opened.
Close editors automatically	You can configure the number of editors that can be opened prior to editors being reused and what should occur when all editors have unsaved changes

File Associations

Option	Description
File Types	• **Add...**: Adds a new file or file type (extension) to the predefined list. In the resulting New File Type dialog, type the name of a file or a file extension. If you are adding a file extension, you must type either a dot or a "*." before the file type (e.g., ".xml" or "*.xml" as opposed to simply "xml"). • **Remove**: Removes the selected file type from the list

Option	Description
Associated editors	**Associated editors list** • **Add...**: Adds a new editor to the list of editors associated with the file type selected above. In the resulting Editor Selection dialog, you can choose an editor to launch either inside the Workbench (internal) or outside the Workbench (external); click **Browse** to locate an editor yourself if the editor you want is not displayed in the list. • **Remove**: Removes the association between an editor and the file type selected above. **Note:** Any editor that is bound by content type may not be removed from this list. Currently, there is no mechanism available to remove these editors. • **Default**: Sets the selected editor as the default editor for the file type selected above. The editor moves to the top of the Associated Editors list to indicate that it is the default editor for that file type.

Keys

Option	Description
Scheme	A 'scheme' is a set of bindings. Eclipse includes two schemes: • Default • Emacs (extends Default)
Type filter text	In this text field, enter the information you want to use as the filter.
Commands	Copy Command Unbind Command Restore Command
Name	Enter the name of the shortcut key
Description	A default description appears when the command is selected.
Binding	With multi-stroke key sequences, schemes, and contexts, there are a lot of things to keep in mind when customizing key bindings.
When	Displays when the command is to be used. The default can be changed.

Option	Description
Conflict	

Network Connections

Option	Description
System proxy configuration (if available)	Specifies the settings profile to be used when opening connections. Choosing the **Direct** provider causes all the connections to be opened without the use of a proxy server. Selecting **Manual** causes settings defined in Eclipse to be used. On some platforms there is also a **Native** provider available, selecting this one causes settings that were discovered in the OS to be used
Direct connection to the internet	The table displays entries that are available for all providers. Checkboxes in the first column of the table indicate entries to be used for the currently selected provider. Selected by default
Manual proxy configuration	Use this table to specify, either by name or pattern, which hosts should not use any proxy. A direct connection will always be used for matching hosts. Check boxes in the first column of the table indicate entries to be used for the currently selected provider.

Perspectives

Option	Description
Open in New perspective	Use this option to set what happens when you open a new perspective. Do you want the perspective opened within the current Workbench window or opened in a new window?
Open a new view	Use this option to specify what happens when a new view is opened. It is either opened to its default position within the current perspective or it is opened as a fast view and docked to the side of the current perspective.

Option	Description
Open the associated perspective when creating a new project	Use this option to specify the perspective behavior when a new project is created. You can set it to switch the current perspective to be the one associated with the project type and open the perspective in the same Workbench window as the current one, switch the perspective and open it in a new Workbench window, or not to switch perspectives at all.
Available Perspective	Make Default Reset Delete

Security

Click on the Secure Storage node to gain access to the following preferences:

Option	Description
Password tab	
Cached password	Option to clear passwords
Master password providers	The **Master password providers** section contains a list of currently available password providers. By default, the enabled provider with the highest priority is used to encrypt data added to secure storage. The priority range is from 0 to 10, with 10 being the highest. A password provider can be disabled it if malfunctions, or if you prefer a lower priority password provider.
Details	Displays information about the selected master password providers.
Contents tab	
Default Secure Storage	Displyas the eclipse plug-in of the default secure storage
Values associated with the selected node:	Shows the identification and value of the selected node (if available)
Advanced Tab	
Select encryption algorithm to use in new storages	Changes in the encryption algorithm are only applied to data stored after the change. If you have already created a secure storage it would have to be deleted and re-created to use the newly selected encryption algorithm.

Startup and Shutdown

Option	Description
Prompt for workspace on startup	When starting the application, a prompt always opens asking for workspace information.
Refresh workspace on startup	If this option is turned on then the workbench will synchronize its contents with the file system on startup
Confirm exit when closing last window	If this option is turned on then the workbench will ask if you wish to exit when closing the last window
Plug-ins activated on startup	This option allows you to select which available plug-ins should be activated on startup.

Web Browser

The selected Web browser is used by default when Web pages are opened, although some applications may always use the external browser.

Option	Description
Use internal Web browser	Always use an internal Web browser.
Use external web browser	Always use an external Web browser.
External Web browers	Lists the available external web browsers. You can add, edit or remove browers.

Workspace

Option	Description
Refresh automatically	If this option is turned on then the workbench will prompt you each time it is started for what workspace to use.
Workspace save interval (in minutes)	Set the interval of how frequently you want the workspace to be saved automatically.
Open referenced project when a project is open	You have three options: always, never, or prompt.
Text file encoding	

Linked Resources

Option	Description
Enable linked resources	Used to globally enable or disable the linked resource feature for the entire workspace. By default, linked resources are enabled.
Defined path variables	Defining **patth variables** that are used when creating linked resources.

Local History

Option	Description
Days to keep files	Indicates for how many days you want to maintain changes in the local history. History state older than this value will be lost.
Maximum entries per file	Indicates how many history states per file you want to maintain in the local history. If you exceed this value, you will lose older history to make room for new history.
Maximum file size (MB)	Indicates the maximum size of individual states in the history store. If a file is over this size, it will not be stored.

Embarcadero Licensing

Option	Description
Enable automatic license updates	Licenses automatically updated as available.
Automatic License Update Schedule	You have two options: look for license updates each time the product is started; or look for license updates every (select interval from drop-down menu)

Generate Documentation

Window > Preferences > Generate Documentation

This node includes the following groups of options:

- **Generate HTML**

- **HTML Output Options**

  RTF Output Options

Generate HTML

Window > Preferences > Generate Documentation > Generate HTML

This node includes the following options:

Option	**Description**
Javadoc style	If this option is checked, the generated output corresponds to the Javadoc style (without navigation tree and diagrams).
Process line breaks	If this option is checked, line breaks are preserved.

Include Classes and Members options

Option	Description
public	If this option is checked, elements with the public visibility modifier are included in the generated documentation.
protected	If this option is checked, elements with the protected visibility modifier are included in the generated documentation.
private	If this option is checked, elements with the private visibility modifier are included in the generated documentation.
package	If this option is checked, elements with the package visibility modifier are included in the generated documentation.
deprecated	If this option is checked, the deprecated elements are included in the generated documentation.

Include Tags

@author	If this option is checked, the @author tag is included in the generated documentation.
@version	If this option is checked, the @version tag is included in the generated documentation.
@param	If this option is checked, the @param tag is included in the generated documentation.

@return	If this option is checked, the @return tag is included in the generated documentation.
@see	If this option is checked, the @see tag is included in the generated documentation.
@since	If this option is checked, the @since tag is included in the generated documentation.
@throws	If this option is checked, the @throws tag is included in the generated documentation.
User-defined tags	If this option is checked, the user-defined tags are included in the generated documentation.

Javadoc options

Option	**Description**
Window Title	Enter the window title.
Doc Title	Enter the title of the generated documentation.
Header	Enter the string that will be displayed in the header of each page of the generated output.
Footer	Enter the string that will be displayed in the footer of each page of the generated output.
Bottom	Enter the string that will be displayed at the bottom of each page.
Generate Tree	If this option is checked, the navigation tree is generated.
Generate Index	If this option is checked, index is generated.
Split index	If this option is checked, the index file is split into parts in alphabetical order.
Generate	Use If this option is checked, a table that shows usages of each class is included in the generated output.
Generate Deprecated List	If this option is checked, the list of deprecated elements is included in the generated output.
Generate Help	If this option is checked, a page that will be displayed on pressing the Help hyperlink is generated.
Generate Navigation bar	If this option is checked, a navigation bar is generated.
Stylesheet File	Specify stylesheet file.
Overview File	Specify path to a file that will be included in the generated documentation. A link to the overview file is placed at the top of the JavaDoc frame.

HTML Output Options

Window > Preferences > Generate Documentation > HTML Output Options

This node includes the following options:

Option	Description
Process line breaks	If this option is checked, line breaks are preserved.

RTF Output Options

Window> Preferences > Generate Documentation > RTF Output Options

This node includes the following options:

Option	Description
Formatting template	Enter the fully-qualified name of a formatting template (such as a Microsoft Word *.dot file), or pick a template with the Browse.
Render HTML tags	Check to translate HTML tags into appropriately formatted text in the printed documentation. The following tags are supported: <b>,<i>,<u>,<h1> — <h6>, <code>, <tt>, <em>, <font color>, <pre>, <p>, <br>, <ol>, <ul>, <li>
RTF Process line breaks	Check to preserve line breaks even if the Render HTML tags is checked on.
Store graphics in RTF	Check to embed all the graphics in a single RTF document.
Diagram image format	Select EMF or GIF format.
Color representation	Select RGB color or 16-bit color.
Included text formatting	Select whether to preserve the original formatting or apply the one from the Formatting template that is specified in the first option.

Help

Option	Description
Specify how help information is displayed	This option allows you to determine whether the documents selected in the help view will be opened in-place or in the editor area.

Option	Description
Use external browser	If embedded web browser is supported on your system, help window uses an embedded help browser to display help contents, whenever possible, and this option is available. Select it, to force help to use external browsers. Use "Web Browser" preference page to select browser to use.
Open Modes	This option allows you to determine whether the window context help will be opened in a dynamic help view or in an infopop.
Included text formatting	Select whether to preserve the original formatting or apply the one from the Formatting template that is specified in the first option.

Content

Option	Description
Include help content from a remote infocenter	If checked, this option enables the use of remote help content. The rest of the fields on the page are only enabled if this option is checked.
Remote Infocenters	View the properties for this remote data source

Modeling

Window> Preferences > Modeling

Use these preferences to change startup, deletion, error reporting, ignored folders, and team sharing options.

Deletion Tab

Option	Description
Show confirmation when element is about to be deleted	Prompts for a confirmation before an element is deleted.
On pressing 'Delete' key always delete from:	Model Element is deleted from both model and view.
View only	Element is deleted from view, but remains in model.

Ignored Folders Tab

Use this tab to specify the folders you want to ignore. Usually this list contains **CVS, bin, lib** and **doc** directories. Ignored folders are not parsed, so no diagram will be generated for them.

Button	Description
Add	Opens the name field so you may enter a new folder.
Remove	Removes the selected folder.

Team/Compare Tab

Use this tab to indicate whether you want to include diagram folders in the Team/Compare actions.

Diagram

Window > Preferences > Modeling Diagram

Use these preferences to change how your diagrams are displayed:

Option	Description
Diagram	
Font	Lists the current diagram font and size settings.
Change	Click to open the system font dialog and change the font.
Diagram toolbar visible	If selected, the diagram toolbar appears in the Diagram Editor
Diagram background	Select to change the diagram background color.
Antialiased graphics	If selected, the diagram elements and graphics on the diagrams appear antialiased.
Antialiased text	If selected, the text on the diagrams appear antialiased.
Show detailed feedback	If selected, a translucent trace of the elements displays on drag-and-drop.
Rulers, Grid and Snapping	
Show grid	Displays a background grid in the Diagram Editor.
Snap to grid	Diagram elements, when dragged, will lock in place at the nearest grid point.
Snap to geometry	Snaps selected objects for precise placement by aligning them with guides and other objects.
Show snap feedback	Highlights guides and grids when an object aligns with them.
Grid width	Determines width, in pixels, between grid points in a row.
Grid height	Determines the height, in pixels, between grid points in a column.
Grid color	Click the icon to define the grid color.
Grid type	Determines the grid type (line or dotted).
Show rulers	Displays rulers with selected units.
Store ruler guides	Saves created guides in your project upon exiting.
Ruler units	Sets units of the rulers. You can select from centimeters, inches or pixels.
Palette - Show Imported Categories	Enables you to show the imported categories if required, or hide them, if your diagram becomes overloaded.
Other	
Show projection bars	If this option is checked, projection bars display for the elements that can display their projections (Activity Partitions, Objects or Lifelines on Sequence diagrams, BPMN pool)
Show chooser when link target is invalid	If this option is checked, **Select target for the link** dialog displays, when you drop the link on an invalid target location (for example, on the diagram background).

Option	Description
Copy image along with diagram elements	If this option is checked, images of the copied diagram elements are also placed to the clipboard. You can paste these images to the other applications. In the large diagrams this may slow down performance.

TIP: These selections become active on the current diagram when you click **Apply**.

Related Concepts

Diagram Overview

Related Procedures

Diagrams

EMF Model Compare Preferences

Window > Preferences > Modeling > EMF Model Compare

Use these preferences to change how models are compared.

EMF Model Compare

Use the **EMF Model Compare** tab to set global model comparison options.

Item	Description
Initially show containment references in difference overview	Specifies if you want to display references to containment features into the **Structure Compare** area of the **Compare Editor**.

ID Features

Use the **ID Features** tab to choose ID (key) EMF features for model comparison. By default, name feature is set as ID throughout all metamodels (where applicable).

Item	Description
Metamodels	Displays the hierarchy of EMF metamodel classes which features will be used as IDs during model comparison.
Features of class	Displays the list of features for the selected class. The checked features will be used as IDs during model comparison.
Reset local preferences	Resets ID features of the currently selected class to the default state.
Go To “Defined in”	Navigates to the feature where the selected containment feature is defined.

Ignored Features

Use the **Ignored Features** tab to choose EMF features which you want to ignore during model comparison.

Item	Description
Metamodels	Displays the hierarchy of EMF metamodel classes which features will be ignored during model comparison.
Features of class	Displays the list of features for the selected class. The checked features will be ignored during model comparison.
Go To “Defined in”	Navigates to the feature where the selected containment feature is defined.

Model File Compare

Use the **EMF Model File Compare** tab to set model file comparison options.

Item	Description
Compare logical structure of model files	Specifies if you want to enable the Model File compare feature.

Related Procedures

Comparing Models

Interaction Diagrams 2.0

Window > Preferences > Modeling > Interaction Diagrams 2.0

Use these preferences to customize view and editing options for the sequence and communication diagrams UML 2.0.

Option	Description
Show	
Sequence Numbers	If selected, the sequence numbers display in the Diagram Editor.
Invocations	If selected, method invocations display in the Diagram Editor.
Edit	
Delete preserves content of frame	If selected, the deletion of a diagram element does not affect the contents of the deleted frame.
Color Combined Fragments	If selected, combined fragments display in color.

Option	Description
Show “Select signature” dialog on message creation	If this option is selected, **Select signature** dialog displays when a message is drawn to a lifeline. If the lifeline does not have any associated type, or this type does not contain any operation, the dialog doesn't appear.
Allow asynchronous call message to be straight	If this option is selected, the asynchronous call messages are drawn at the right angle to the target lifeline.

Related Concepts

About OCL Support

Layout

Window > Preferences > Modeling > Layout

Use these preferences to define the alignment of diagram elements.

Option	Description
Links Layout	Determines shape of the links (direct or rectilinear).
Algorithm	Click the drop-down arrow to select a layout algorithm. UML diagrams can be thought of as graphs (with vertices and edges). Therefore, graph data structures (algorithms) can be applied to the UML diagrams for diagram layout. The various algorithms and their optional settings are described below. The algorithm that you specify executes when you lay out your diagram. See detailed description in the section Algorithm specific options below.
Recursive	This option is available for all layout algorithms. Selecting this option allows you to lay out all subelements within containers while laying out diagram nodes.

Algorithm specific options are described in the following subsections.

<Autoselect>

Option	Description
<autoselect>	Each of the layout algorithms contains internal information about the types of diagrams it will work with and the numeric characteristics for the final quality of the produced layout when applied to each applicable diagram type. Several algorithms can be available for the same diagram type. The <autoselect> option uses such internal information and picks the best layout algorithm for the current diagram type.

Hierarchical

The Hierarchical algorithm originates from the Sugiyama algorithm. The algorithm draws the UML diagram hierarchically according to the preferences that you select.

Option	Description
Vertical and **Horizontal**	Minimal distance between elements in pixels. Here you can specify Vertical and Horizontal distance options.
Justification	This option defines the alignment of classes. The Justification setting is dependent on the Inheritance setting. Select from the following: Top: If the Inheritance option is set as Vertical, then all nodes in a column are aligned at the left of the column. If the Inheritance option is set as Horizontal, then all nodes in a row are aligned at the top of the row. Center: If the Inheritance option is set as Vertical, then all nodes in a column are aligned at the center of the column. If the Inheritance option is set as Horizontal, then all nodes in a row are aligned at the center of the row. Bottom: If the Inheritance option is set as Vertical, then all nodes in a column are aligned at the right of the column. If the Inheritance option is set as Horizontal, then all nodes in a row are aligned at the bottom of the row.

Option	Description
Layer ordering	The heuristics are used to sort nodes within each layer to minimize edge-crossings. Barycenter: The Barycenter heuristic reorders the nodes on node N according to the barycenter weight. The weight of node N is calculated as a simple average of all its successors/predecessors relative coordinates. Median: The Median heuristic reorders the nodes on node N according to the median weight. The weight of node N is calculated as a simple average of this nodes' relative positions dealing only with two central successors/ predecessors coordinate Hybrid: The Hybrid heuristic combines the Median and Barycenter heuristics with the Proportion (see below) setting.
Inheritance	This option defines how classes are aligned with each other if they are connected by an inheritance link. Horizontal: Classes connected by inheritance are aligned horizontally Vertical: Classes connected by inheritance are aligned vertically
Proportion	Used in conjunction with the **Hybrid ordering** option. The optimal setting for this value is 0.7.

Tree

The algorithm draws the given graph in a tree layout according to its maximum spanning-tree.

Option	Description
Process non tree edges	If this option is selected, non-tree edges are bent to fit into the diagram layout.
Horizontal and Vertical	Minimal distance between elements in pixels. Here you can specify Vertical and Horizontal distance options.

Option	Description
Justification	This option defines the alignment of elements. The Justification setting is dependent on the Hierarchy direction setting. Select from the following: Top: If the Hierarchy direction option is set to Vertical, then all nodes in a column are aligned at the left of the column. If the Hierarchy direction option is set to Horizontal, then all nodes in a row are aligned at the top of the row. Center: If the Hierarchy direction option is set to Vertical, then all nodes in a column are aligned at the center of the column. If the Hierarchy direction option is set to Horizontal, then all nodes in a row are aligned at the center of the row. Bottom: If the Hierarchy direction option is set to Vertical, then all nodes in a column are aligned at the right of the column. If the Hierarchy direction option is set to Horizontal, then all nodes in a row are aligned at the bottom of the row.
Hierarchy direction	This option defines the hierarchy direction of the elements Horizontal: Elements are aligned horizontally Vertical: Elements are aligned vertically
Reverse hierarchy	Last in the hierarchy elements are laid out first in the diagram.

Orthogonal

The Orthogonal algorithm uses heuristics to distribute diagram nodes within a lattice.

Option	Description
Node placement strategy: There are three strategies for node placement: Tree, Balanced, and Smart.	
Tree	The Tree node placement strategy creates a spanning-tree diagram layout. The spanning-tree for the given graph is calculated and diagram nodes are placed on the lattice to minimize the tree edges length; thus, minimizing the distance between nodes that are linked with a tree-edge.

Option	Description
Balanced	The balanced node placement strategy uses a balanced ordering of the vertices of the graph as a starting point. Balanced means that the neighbors of each vertex V are as evenly distributed to the left and right of V as possible.
Smart	The Smart node placement strategy sorts all vertices according to the in/out degrees for each vertex and fills the lattice starting from the center with the vertices with the greatest degree.
Distance between elements	Specifies the minimum distance between diagram elements. Distance is in pixels.

Spring Embedder

Spring Embedder are force directed layout algorithms that model the input graph as a system of forces and try to find a minimum energy configuration of this system. All edges are drawn as straight lines. When you lay out the graph according to the **Spring Embedder** layout algorithm, the program will simulate the graph as a physical model (masses and springs) and subject it to physical forces. The unnecessarily-long edges will be the most tense, and will try to contract the most. When the nodes and edges have pushed and pulled themselves to equilibrium, you will have a geometric representation of the graph.

Option	Description
Movement	Specify the nodes movement factor. The more value you specify, the more distance will be between the nodes in the final graph. If you specify 0 as the movement factor, you will get random layout of the nodes
Spring force	Specify the rigidity of the springs. The greater value you specify, the less will be the length of edges in the final graph. **Tip:** Lay out your graph with the default spring settings first and then edit the spring options if necessary.

Embarcadero

Layout options used in the legacy versions.

Option	Description
Layout inheritance	This option defines how classes are aligned with each other if they are connected by an inheritance link. Select either: From left to right: Classes connected by inheritance are aligned horizontally from left to right. From top to bottom: Classes connected by inheritance are aligned vertically from top to bottom. From right to left: Classes connected by inheritance are aligned horizontally from right to left. From bottom to top: Classes connected by inheritance are aligned vertically from bottom to top.
Layout justification	This option defines the alignment of classes. The Justification setting depends on the Inheritance setting. The elements are aligned as summarized in the following table.

Inheritance	Justification	
Left-right	Top	Right of the column
	Center	Center of the column
	Bottom	Left of the column
Right-left	Top	Left of the column
	Center	Center of the column
	Bottom	Right of the column
Top-bottom	Top	Bottom of the row
	Center	Center of the row
	Bottom	Top of the row
Bottom-top	Top	Top of the row
	Center	Center of the row
	Bottom	Bottom of the row

OCL

Window > Preferences > Modeling > OCL

Use the **OCL** preferences to specify auxiliary OCL operations and other OCL settings.

OCL Metamodels

Use the **OCL Metamodels** tab of the **OCL** preferences to choose which metamodels you want to use with OCL. The tab contains the list of all available metamodels; the required metamodels are listed under:

- Architect - UML 2.0 Project

CAUTION: Do not deselect the required metamodels.

You can select any metamodel and thus make it visible to the OCL processor. It means that metaclasses of the selected metamodel can be used as the contexts for the OCL operations, created in the **OCL Operations** tab.

It is not possible to use the arbitrary metamodels as the contexts for the audits, metrics, documentation generation expressions and search, but it is allowed to use the features and operations of these metaclasses in the bodies of the audits and metrics.

For example, you can create a metamodel for a profile. This metamodel displays in the list of available metamodels. If this metamodel is selected, it becomes possible to create operations with the respective context and create audits that will evaluate element properties, specific for the selected profile.

OCL Operations Options

Use the **OCL Operations Options tab** of the **OCL** preferences to define custom OCL operations which you want to use in OCL queries throughout - in audits, metrics, search expressions, gendoc templates, etc. As an example, you can use the predefined set of OCL operations created for audits. OCL operations are defined in the standard way, using ocl def: expressions.

Option	Description
Name	Displays names of auxiliary OCL operations defined upon a metamodel.
New	Opens the **Edit Operation** dialog which allows to choose the context and provide the body of a new operation.
Remove	Removes the selected operation from the list.
Edit	Opens the **Edit Operation** dialog which allows to choose the context and edit the body of the selected operation.
Import...	Imports a text file containing auxiliary OCL operations. Use the **Export...** command to create the file. **Important:** During the import, OCL operations defined in the file replace the current list of operations.
Export...	Exports the current list of auxiliary OCL operations to the text file with .oclOperations extension.

OCL Library Operations

Use the **OCL Library Operations** tab of the **OCL** preferences to view (not edit) the list of signatures of library operations which have been implemented as native extensions to OCL. The list includes a number of powerful String operations which are not defined in the OCL specification.

Option	Description
Select All	Checks all operations in the list.
Clear All	Unchecks all operations in the list.

Model Names Mappings

Use the **Model Names Mappings** tab of the **OCL** preferences to define mappings for keywords and other illegal symbols or words which you cannot use in OCL expressions directly.

Option	Description
Whole name	Specifies if you want to apply the mapping rule only to whole words matching the find substring. Thus, if the substring is "body" and replacement is "_body", all occurrences of the word "body" will be replaced by "_body", but e.g. "mybody" - will not.
Substring	Defines a find substring. Before validating an OCL expression, if the substring is found in the expression body, it will be replaced by one defined in the Replacement field. To avoid naming conflict, provide replacements for all OCL keywords which can appear in your model elements names
Replacement	Defines a replace substring.
New	Adds a new empty mapping rule to the list.
Remove	Removes the selected mapping rule from the list.

Related Concepts

OCL Support

Related Procedures

Working with Custom OCL Operations

Patterns

Window > Preferences > Modeling > Patterns

Use the **Patterns** preferences to define pattern conversion profiles.

Option	Description
Pattern conversion profiles	Lists available pattern conversion profiles.
Edit	Opens the **Edit Transformation Profile** dialog which allows to change the selected pattern conversion profile.

Option	Description
Remove	Removes the selected pattern conversion profile from the list.
Import...	Opens the **Import Pattern Conversion Profiles** dialog which allows to import one or more profiles stored in the .xml file.
Export...	Opens the **Export Pattern Conversion Profiles** dialog which allows to export one or more profiles to the .xml file.

Print

Window > Preferences > Modeling > Print

Use this dialog to define settings of the printed output.

Option	Description	
Size and Orientation	Paper size	Determines the size of the paper to be used. If you set the option to custom, you can use the Custom Paper Size tab to define the size. Default size is A4.
	Orientation	Sets the orientation of the paper. Default orientation is Portrait.
	Width (in.)	Determines the width, in inches, of the custom paper size. Default value is 8.5.
	Height (in.)	Determines the height, in inches, of the custom paper size. Default value is 11.
Margins (inches)	Top (in.)	Determines the size of the top margin in inches. Default value is 1
	Bottom (in.)	Determines the size of the bottom margin in inches. Default value is 1
	Left (in.)	Determines the size of the left margin in inches. Default value is 1
	Right (in.)	Determines the size of the right margin in inches. Default value is 1
Header and Footer	Print page header	Prints the header on each page. The header is a combination of the diagram's name and the page number.Use the adjacent field to specify the header text. Default value is On, %PAGE%, %ELEMENT%
	Print page footer	Prints the footer on each page. The footer is a combination of the diagram's name and the page number. Use the adjacent field to specify the footer text. Default value is On, %PAGE%, %ELEMENT%

Option	Description	
Diagram Print Options	Print zoom	Sets the zoom size of the document. This zoom size and its effect can be seen in the Preview dialog. This zoom option overrides any zoom size that may have been set with Zoom In or Out. The default value is 0.7.
	Fit to page	Fits the printed material on the paper size selected.
	Print border	Switches the border on or off for printing.
	Print empty pages	Determines whether empty pages are printed or ignored.
	Synchronize with Default Printer	Options Click to set print options to match your default printer options.

Related Concepts

Diagram Overview

Related Procedures

Printing Diagrams

QA Model

Window > Preferences > Modeling > QA Model

Use the **QA Model** preferences to define model audits and metrics.

Audits

Option	Description
Name	Displays names of the defined QA model audits. Select the checkbox next to the audit name to activate it.
Description	Provides an audit description.
Severity	Specifies the audit severity.
Select All	Selects all categories and all elements within categories. Selected categories will be included in the current QA set.
Clear All	Deselects all categories and all elements. Deselected categories will not be included in the current QA set.
New	Opens the **Edit Audit** dialog which allows to create a new audit.
Remove	Removes the selected audit from the list
Edit	Opens the **Edit Audit** dialog which allows to change the selected audit.

Option	Description
Import...	Allows you to import a previously saved set of audits.
Export...	Allows you to export the current set of audits as a file.
Clone	Opens the **Edit Audit** dialog which allows to create a new audit identical to the audit which is currently selected in the list.

Metrics

Option	Description
Name	Displays names of the defined QA model metrics. Select the checkbox next to the metric to activate it. Description Provides a metric description.
Severity	Specifies the metric severity
Select All	Selects all categories and all elements within categories. Selected categories will be included in the current QA set
Clear All	Deselects all categories and all elements. Deselected categories will not be included in the current QA set
New	Opens the **Edit Metric** dialog which allows to create a new metric.
Remove	Removes the selected metric from the list.
Edit	Opens the **Edit Metric** dialog which allows to change the selected metric.
Import...	Allows you to import a previously saved set of metrics.
Export...	Allows you to export the current set of metrics as a file.
Clone	Opens the **Edit Metric** dialog which allows to create a new metric identical to the metric which is currently selected in the list.

Related Concepts

Quality Assurance

UML Profiles

Window > Preferences > Modeling > UML Profiles

Use these preferences to define a default set of profiles available in your projects.

UML20 Tab

Profile	Description
UML in Color	If selected, the UML in Color is included into the default profile set.

Related Concepts

UML Profiles Basics

UML Profiles Preferences Constraints

Window > Preferences > Modeling > UML Profiles > Constraints

Use this page to manage constraints to be run on elements if appropriate profiles are applied to the project.

Item	Description
UML 2.0	Each tab corresponds to one of the supported metamodels and contains the list of profiles with their constraints. If a constraint is checked, it will be applied to the model elements by the **Run Profile Constraints** command, after applying the parent profile. Checking or clearing a profile node results in checking or clearing all the nested constraints.
Select All	Checks all profile and constraint nodes in the current tab.
Deselect All	Clears all profile and constraint nodes in the current tab.

Related Concepts

UML Profiles Basics

UML Profiles Preferences. View Management

Window > Preferences > Modeling >UML Profiles > View Management

Use this page to manage showing and hiding stereotypes after applying profiles to the projects.

Item	Description
UML 2.0	Each tab corresponds to one of the supported metamodels and contains the list of profiles with their stereotypes. If a stereotype is checked, it will be hidden on diagrams after applying the parent profile. If a stereotype is not checked, it will show up on diagrams after applying the parent profile. Checking or clearing a profile node results in checking or clearing all the nested stereotypes.
Select All	Checks all profile and stereotype nodes in the current tab
Deselect All	Clears all profile and stereotype nodes in the current tab.

TIP: To manage the stereotype decoration in diagrams, refer to Show Stereotype option (**Preferences Modeling > View Management > Text Decorations > Show Stereotype**)

Related Concepts

UML Profiles Basics

Related Procedures

Working with Required Stereotypes

View Management Preferences

Window > Preferences > Modeling > View Management

Use these preferences to determine which elements are visible in the Diagram Editor.

Details Tab

Detail Level	Analysis	View names only (no visibility signs are shown).
	Design	View names and types (visibility signs are shown). Default value.
	Implementation	View names and types, parameters for operations, and initial values of attributes (visibility signs are shown).

Show Icons for	This option enables you to use icons for better distinguishing metaclasses of elements with similar look. Icons help recognize metaclasses in the following cases:	
	Element shown as a label inside another element	For example, class members; package or diagram children; internal transition and deferred event of a State element.
	Diagrams	An element that represents a diagram on diagram.
	Classifier shown as Class	Classifier element whose notation is the same as class notation (for example, interfaces, components etc.)
Make Classifiers look like Classes on diagram	If selected, the classifiers display on the diagram in a boxlike manner (like classes), regardless of their original notation. Default value: Off	
Default Activity Partition orientation	Choose whether the activity partitions will have vertical or horizontal orientation by default.	
Maximum auto-width of class element	If textual name of an element or a member exceeds the specified width in pixels, it will be truncated. This option becomes effective if the user has not resized the element manually.	
Always show “Attributes” and “Operations” compartments	If selected, these compartments are always visible, even they are empty. Otherwise compartments only show when respective members exist. Default value: Off	
Sort members alphabetically	Default value: Off	
Mark link to member with a dot	Default value: On	
Show shortcut sign	Default value: On	
3D look	Default value: On	

Text Decorations Tab

These options enable you to control text that displays on diagram elements.

Option	Description	
Show stereotypes	If selected, the following options become enabled:	
	Show members stereotypes	
	Show <<communicate>> stereotype on Association Link on Use Case Diagram	
Show Diagram Types	If selected, a type of the diagram is displayed above the diagram name in the Diagram Editor. Default value: On	
Show name of referenced Class in extending Class icon	If selected, the name of the base class displays in the icon of the extending class. Default value: Off	
Use Fully Qualified Names in Shortcuts to classes from different packages	If selected, shortcut names on diagrams will contain both package name and class name. Default value: On	
Show link names	If selected, the default non-empty link names display on diagrams. Default value: On	

Option	Description	
Show names of Decision/Merge element	Default value: Off	

Show/Hide Elements Tab

All options in this dialog are selected by default except **Elements marked as hidden**.

Option	Description	
Elements	Packages	Show all packages
	Interfaces	Show all interfaces
	Classes	Show all classes
	Notes	Show note elements
	Elements marked as hidden	Shows elements marked individually as hidden on the diagram. By default, this filter is off so that the Hide from View menu functions.
	Non-public Classes and Interfaces	Shows all package local classes and interfaces, including inner classes.
Members	Shows all attributes, operations and inner classes/interfaces. Either Interfaces or Classes must be selected for this check box to be active.	
	Attributes	Shows all attributes. Either Interfaces or Classes must be selected for this check box to be active. Members must also be selected for this check box to be active.
	Operations	Shows all operations. Either Interfaces or Classes must be selected for this check box to be active. Members must also be selected for this check box to be active.
	Non-public Members	Shows all private, local and protected members (members include attributes, operations, inner classes and inner interfaces). Either Interfaces or Classes must be selected for this check box to be active. Members must also be selected for this check box to be active.
Links	Associations	Shows all associations links.
	Generalizations	Shows all generalization links.
	Implementations	Shows all implementation links
	Dependencies	Shows all dependency links.

Related Concepts

Diagram Overview

Related Procedures

Diagrams

Modeling Resources Team Preferences

Window > Preferences > Team Modeling > Resources

Use these preferences to change the team sharing options for modeling resources.

Option	Description
AutoCheckout modeling resource on edit	Checks out model files on attempt of to change corresponding model entities. The checkout is performed when the edited resource is saved to disk, that is the actual checkout might happen after you performed several modifications. This mode requires VCS provider to mark files that are not checked out as read-only.
Checkout before model modification	Performs the checkout before any modification is performed.
Ignore default package diagrams	Does not check out or check in the default package diagrams.

Related Procedures

Diagrams

Team

References

This section covers reference information.

Glossary	This glossary contains the basic terminology of ER/Studio Software Architect
Keyboard Shortcuts	Describes keyboard shortcuts.
Components of the User Interface	This section describes GUI components of the user interface you use for modeling, quality assurance, requirements management and more
Projects	This part contains reference information about the supported project types and formats and project properties.
Preferences	This part contains reference information about Preferences.
Profiles Reference	Contains reference information about profiles and profiles API.
UML 2.0 Reference	This section contains reference material about UML 2.0 diagrams.
Requirements Management	This part contains reference information about requirements management facilities.
Quality Assurance	This part contains reference information about audits and metrics.
Project Documentation	This part contains reference information about project documentation: command and syntax of the documentation generation utility, and reference information of the documentation template designer.
Model Import and Export	This part contains reference information about exchanging model information between ER/Studio Software Architect and another applications.
Version Control	This part contains reference information about the VCS.
Dialogs	This part contains reference information about the various dialogs.

Components of the User Interface

This section describes GUI components of the user interface you use for modeling, quality assurance, requirements management and more.

Menus	This part contains reference information about the various menus.
Model Bookmarks View	This view lists available bookmarks and allows to navigate directly to a bookmarked model element.
Compare Editor	Use the **Compare Editor** to review and merge differences in the structure and properties of the models which you have compared.

Menus	This part contains reference information about the various menus.
Tool Palette	
Diagram View	
Model Navigator	
OCL Expressions View	Use the **OCL Expressions** view to quickly evaluate OCL expressions in the explicitly specified context (EMF model element), or in the context of the current selection.
Properties View	This view shows properties of the selected element.
XSL Editor	Use the **XSL Editor** to write your XSL transformation scripts.
Last Validation Results View	
Patterns GUI Components	This part describes GUI components of the interface you use for the Pattern features
Quality Assurance GUI Components	Describes GUI components of the interface which you use for Quality Assurance features.

Available Menus

- Menus
- Model Navigator Context Menu
- Common Diagram Context Commands
- Common Diagram Context Commands
- Package Context Menu
- Common Element Context Commands
- Common Link Context Commands

Menus

Item	**Description**
File menu	You can use the File menu to export diagrams to image files, and print diagrams.
Edit menu	Use the Edit menu to cut, copy, paste, and delete diagrams and diagram elements, select all items on a diagram, and undo/redo actions.
View menu	The View menu contains the command for opening the Model View.
Project menu	Use the Project menu to enable or disable support for specific projects currently open in the workspace.
Tools menu	Use the Tools menu to generate documentation, open the pattern registry and pattern organizer, and set specific options.

Item	Description
Diagram context menu	Use the Diagram context menu to add new elements, manage the layout, zoom in and out, show or hide diagram elements, synchronize your diagram with the Model view, and edit hyperlinks.
Model View context menu	Context menus of the various elements of the Model View are context-sensitive. The list of elements that can be added to a diagram from the Model View depends on the element and diagram type.
Element context menus	You can add or delete members (or delete the element itself), cut/copy/paste, show element information and more. Explore the context menus of the different elements as you encounter them to see what is available for each one.

Model Navigator Context Menu

The Model Navigator has several different context menus, each specific to the resource selected. This section discusses the various context menus for the following resource levels in the Model Navigator:

- Project Level
- Package Level
- Diagram Level
- Element Level, Class
- Element Level, Operation
- Element Level, Link

NOTE: Depending on your development platform, some of the menu commands described may not be applicable.For more information, refer to the documentation set provided with your IDE: select Help Contents on the Help menu.

Project Level

Model Navigator projects offer the following context menu commands. Some of the commands are the same across all the context menus. For commands that are not mentioned here, refer to the Common Diagram Context Commands section.

Option	Description
New	Displays a submenu with all basic elements that can be added to the project
New Diagram	Creates a new diagram: Activity, Collaboration, Class, Component, Deployment, State, Use Case, Sequence.
Open in New Tab	Open the project level diagram in Diagram editor in a new tab.
Open	Open the project level diagram in Diagram editor. It will replace the currently shown diagram with this one.

Option	Description
Open Type Hierarchy	Highlights the node selected in the Hierarchy view. The Hierarchy view will expand and highlight that element in the tree-view. If closed, the Hierarchy view will open. For more information, refer to the documentation set provided with your IDE. Select Help Contents on the Help menu.
Delete	Deletes the selection. You are prompted for confirmation.
Refactor	Begins the refactoring process to restructure your code without changing its observable behavior. For more information, refer to the documentation set provided with your IDE. Select Help Contents on the Help menu
Refactor - Rename	Starts the Rename refactoring dialog. Renames the selection and (if enabled) corrects all references to the elements (also in other files).
Refactor - Move	Starts the Move refactoring wizard: Moves the selection and (if enabled) corrects all references to the elements (also in other files).
Import	Opens the Import wizard for a number of import options. Follow this link for more information about the XMI File option. TEC also offers the ability to import Control Center projects. For details on other import options, refer to the documentation set provided with your IDE. Select Help Contents on the Help menu.
Export	Opens the Export wizard for a number of export options, some of which are specific: UML Documentation and XMI File, for instance. For details on other export options, refer to the documentation set provided with your IDE. Select Help Contents on the Help menu.
Refresh	Refreshes the current view.
Properties	Displays the properties for the selection in the Properties view. While the Properties command accessed through the Model Navigator opens the Properties view for the selected resource, you should be aware that this differs from the Properties command accessed through the Navigator and Packages views. The Navigator view Properties command, for instance, opens a project-specific Properties dialog. Among other things, it allows you to turn templates and pattern-recognition on or off per project. These selections override those made in the global Preferences dialogs.
Update	Using the update command will update the diagram.

Package Level

Within the Model Navigator at the package level of a project, the context menu displays many of the same commands as at the project level, however, there are some additional options available described below.

Option	Description
Select on Diagram	Using the Select on Diagram command, you can open the corresponding diagram that the element belongs to in the Diagram editor and highlight the element on the diagram.

Option	Description
Show in Packages View	The Show in Packages View command highlights the node selected in the Packages tree-view. The Packages view will expand and highlight that element in the tree-view. If closed, the Packages view will open.

Diagram Level

The context menu of the diagram level of a project displays the same commands as at the package level.

Element Level, Class

The context menu for class and interface elements has the following specific command:

Option	Description
Select on Diagram	The Select on Diagram command highlights the node selected in the Diagram editor. If the diagram is closed, it opens to display the node.

Element Level, Operation

The context menu for an operation has the following specific command:

Option	Description
Select on Diagram	The Select on Diagram command highlights the operation selected in the Diagram editor. If the diagram is closed, it opens to display the operation.

Element Level Link

For information on the link level context menu, see Common Link Context Commands.

Common Diagram Context Commands

The context menus of the various diagrams provide functions specific to each diagram. Explore the context menus of the different diagrams as you encounter them to see the commands available for each one.

However, for the most part, all of the UML diagrams do share common context menu commands. To use the context menu for a diagram, simply right click the background of the diagram in the Diagram editor.

From the Diagram editor, each diagram has the following common context menu commands.

New

Each diagram has the New command. Each diagram has a submenu specific to the New command containing each diagram's specific elements.

Select in Model Tree

The Select in Model Tree command highlights the node selected in the Model Navigator tree-view. The Model Navigator will expand and highlight that element in the tree-view. If closed, the Model Navigator will open. For more information, see Using Select in Model Tree.

Cut

This action removes the diagram. You can then choose to Paste the element into the desired location.

Copy

This action copies the selected diagram. After copying an element, choose to Paste it into a new location.

Clone

The Clone command lets you quickly create a new diagram or element with the same content as the existing one. An element that can be cut/copied and pasted can also be cloned by using the Clone command. Cloning is basically a one-step copy-and-paste. For more information, see Using Clone.

Paste

Using the Paste command, you can paste a diagram that has been cut or copied.

Paste element

Use this command to paste a copied diagram or element as a shortcut to another diagram.

Rename

The Rename dialog renames the element and refactors the change throughout the project.

Delete

The Delete command will delete the element from the project. A Confirmation dialog opens to confirm the deletion.

Export

Opens the **Select** dialog with available export destinations.

Import

Opens the **Select** dialog with available import sources.

Add Linked

Provides search options for references, implementations, and inheritance according to the specified types and scopes. For more information, see Add Linked.

Refactor

Move/Rename and Refactor a project or element. Other options appear in the submenu based on the selection. Dialogs are opened in which you can enter the needed information or locations.

Model Bookmark

Bookmarks allow you to navigate to resources that are frequently used. You can set, remove, and view bookmarks using the Bookmarks view.

Hyperlinks

The Hyperlinks command offers a submenu with the following options:

Option	Description
Edit	Using Edit, you can view, add, and remove hyperlinks to a project. For more information on hyperlinks, see Working with Hyperlinks.
<Hyperlink>	<Hyperlink> represents an actual hyperlinked element. In the example shown above, Activity is a hyperlinked element. If there are no hyperlinks for the diagram, then only the Edit option displays.

Layout

The Layout command offers a submenu with options for laying out your diagram elements. For more information, see Using the Automated Layout Features.

Hide / Show

Note that individual elements on a diagram can be hidden using the element context menu. If you do not see an element in a diagram, choose Hide/Show from the diagram context menu and check the hidden elements list in the Show Hidden dialog. For more information, see Hiding Elements.

Team

Use the Team command to add a project to the repository. Once the project is added, you can create patches, commit, synchronize with repository, and so on. Refer to the documentation set provided with your IDE for complete information. From the menubar, choose Help > Help Contents.

Properties

Using the Properties command opens the Properties view for the current element or diagram. For more information, see Using the Properties View.

Related Procedures

Common Diagrams Procedures

Package Context Menu

All of the UML diagram types share common context menu commands. To use the context menu for a diagram, simply right click in the Diagram Editor .

The context menu for a package element residing on a diagram differs slightly from the package diagram context menu. The package context menu shares the common context menu commands as well as commands specific to it: The following table lists the commands available in each context menu.

Package Diagram Context Menu	Package Context Menu
New	New
New Diagram (for package elements)	**New Diagram** (for package elements)
Select in Model Tree	Select in Model Tree

Package Diagram Context Menu	Package Context Menu
Select Metaclass in Model Tree	Select Metaclass in Model Tree
Referred Projects	Referred Projects
	Open
	Open in Active Editor
Edit commands: cut, copy, past, paste element	Edit commands: cut, copy, paste, delete, rename
Import	Import
Export	Export
Compare With	Compare Wtih
Model Bookmrks > Add Bookmarks/Remove Bookmarks	Model Bookmrks > Add Bookmarks/Remove Bookmarks
Team > Navigate to Resources	Team > Navilgate to Resources
Hyperlinks	Hyperlinks
	Optimize size, layout, hide
Properties	Properties
Layout	
Hide/Show	

The unique commands are explained in the following topics.

New

The New command for the package element offers a submenu with the following options:

- Selecting Class from the submenu adds a class element to the diagram.
- Selecting Interface from the submenu adds an interface element to the diagram.
- Selecting Package from the submenu adds a package element to the diagram.
- Selecting Object from the submenu adds an object element to the diagram.
- Selecting Note from the submenu adds a note element to the diagram.
- To refer to an element located outside of the current diagram, or to another diagram, you can use shortcuts. Invoking the Shortcut command displays a selection dialog, where you can choose the desired element (or diagram) from the appropriate location.

New Diagram

The New Diagram command for the package element offers a submenu allowing you to create new diagrams. The new diagrams are created in the current package.

Select in Model Tree

The diagram node or element node is highlighted in the Model Navigator tree.

Select Metaclass in Model Tree

The selected element is highlighted in the metamodels node in the Model Navigator.

Referred Projects

When you select this command, the pull-right menu displays all referred projects.

Open

When selecting a package element on a diagram, you can open the package diagram in the Diagram editor by using the Open command. Using this command keeps the current diagram open, while the newly opened diagram opens in its own tabbed page in the Diagram editor. This command reside on the context menu for package elements on diagrams.

Open in Active Editor

When selecting a package element on a diagram, you can open the package diagram in Diagram editor by using the Open in Active Editor command. Using the Open in Active Editor command will replace the currently opened diagram with the newly opened diagram. This command resides on the context menu for package elements on diagrams.

Compare Wtih

Compares two or three selected model elements against each other and shows differences in a separate view.

Model Bookmarks

You can bookmark model elements. These bookmarked elements appear in the Model bookmarks View. You can add or remove these bookmarks.

Team > Navigate to Resources

If a read-only file is located inside a package, and this package has been moved or renamed, you are not allowed to check out the files in this situation. Use the Team >Navigate to Resource command to select the files for locking and then try to move or rename the package.

Hyperlink

You can edit an existing hyperlink, or hyperlink the diagram or element to a diagram you create.

Layout

The Layout command offers a submenu with options for laying out your diagram elements. For more information, see Using the Automated Layout Features.

Hide/Show

Hides individual elements on a diagram. If you do not see an element in a diagram, choose Show hidden from the diagram context menu and check the hidden elements list. For more information, see Hiding Elements.

Common Element Context Commands

Select element > Right-click

The context menus of the various elements provide functions specific to each element. For example, you can add or delete members from a class, cut-copy-paste, hide and show elements, and more. Explore the context menus of the different elements as you encounter them to see the commands available for each one.

However, for the most part, all of the UML elements do share common context menu commands. To use the context menu for an element, simply right click on the element in the Diagram Editor .

Each element has the following common context menu commands:

Select in Model Tree

The Select in Model Tree command highlights the node selected in the Model Navigator tree-view. The Model Navigator will expand and highlight that element in the tree-view. If closed, the Model Navigator will open.

Cut

One of the usual edit operations. This action deletes the source element from the current Diagram Editor . You can then choose to Paste the element into the desired location.

Copy

One of the usual edit operations. This action copies the selected element. After copying an element, choose to Paste it into a new location.

Clone

The Clone command lets you quickly create a new element with the same content as the existing one. An element that can be cut/copied and pasted can also be cloned by using the Clone command. Cloning is basically a one-step copy-and-paste. For more information, see Using Clone.

Paste

One of the usual edit operations. Using the Paste command, you can paste an element that has been cut or copied.

Paste shortcut

Use this command to paste a copied diagram element as a short cut on another diagram.

Rename

The Rename dialog renames the element and refactors the change throughout the project.

Delete

The Delete command will delete the element from the project. A Confirmation dialog opens to confirm the deletion.

Add Linked

Provides search options for references, implementations, and inheritance according to the specified types and scopes. For more information, see Add Linked.

Model Bookmark

Bookmarks allow you to navigate to resources that are frequently used. You can set, remove, and view bookmarks using the Bookmarks view.

Hyperlinks

The Hyperlinks command offers a submenu for managing and viewing hyperlinks:

Option	Description
Edit	Using Edit, you can view, add, and remove hyperlinks to a project. For more information on hyperlinks, see Working with Hyperlinks.

Option	Description
<Hyperlink>	<Hyperlink> represents an actual hyperlinked element. In the example shown above, Activity is a hyperlinked element. If there are no hyperlinks for the diagram, then only the Edit option displays.

Print

Use the Print command to print a single diagram element. Use CTRL + CLICK to select multiple diagram elements for printing. For more information, see Printing Diagrams, and Diagram Elements.

Optimize Size

Use this command to resize an element to its default size. For elements that contain subelements, the default size will respect the position of the subelements and remain large enough to show them.

Hide

Individual elements can be hidden in diagrams using the Hide command. If you do not see an element in a diagram, choose **Hide/Show** from the Diagram context menu and check the hidden elements list.

Properties

Using the Properties command opens the Properties View for the current element. For more information, see Using the Properties View.

Related Procedures

Common Diagrams Procedures

Common Link Context Commands

The context menus of the various link elements provide functions specific to each link. Explore the context menus of the different link elements as you encounter them to see the commands available for each one.

Shared Commands

For the most part, all of the link elements share common context menu commands. To use the context menu for a link, simply right click the link in the Diagram editor. Each link has the following common context menu commands:

Option	Description
Scroll to Source	If the required end of the link is out of reach, choose Scroll to Source to scroll to the client end of the link.
Scroll to Destination	If the required end of the link is out of reach, choose Scroll to Destination to scroll to the supplier end of the link

Option	Description
Select in Model Tree	The Select in Model Tree command highlights the node selected in the Model Navigator tree-view. The Model Navigator will expand and highlight that element in the tree-view. If closed, the Model Navigator will open. For more information, see Using Select in Model Tree.
Delete	The Delete command will delete the element from the project. A Confirmation dialog opens to confirm the deletion.
Add Linked	Provides search options for references, implementations, and inheritance according to the specified types and scopes. For more information, see Add Linked.
Hyperlinks	The Hyperlinks command offers a submenu with the following commands: **Edit** Using Edit, you can view, add, and remove hyperlinks to a project. For more iinformation on hyperlinks, see Working with Hyperlinks. **<Hyperlink>** <Hyperlink> represents an actual hyperlinked element. In the example shown above, Activity is a hyperlinked element. If there are no hyperlinks for the diagram, then only the Edit option displays.
Hide	Individual elements can be hidden in diagrams using the Hide command. If you do not see an element in a diagram, choose Hide / Show from the Diagram context menu and check the hidden elements list.
Properties	Using the Properties command opens the Properties view for the current element. For more information, see Using the Properties View.

Widely Encountered Commands

Perhaps not available on context menus for all diagram elements, the following commands are often encountered.

Option	Description
Add Linked	Provides search options for references, implementations, and inheritance according to the specified types and scopes. For more information, see Add Linked.
Generate Class Diagram	The Generate Class Diagram command creates an exact copy of the diagram as a new class diagram. Active only where applicable. For more information, see Creating Diagrams.
Link Type	Use the submenu list of the link types to specify an association, aggregation, or composition link.

Option	Description
Client Cardinality	Choose the appropriate cardinality from the drop down list. Available cardinality choices are: 687 0..1 1 0..* 1..*
Supplier Cardinality	Choose the appropriate cardinality from the list. Available cardinality choices are: 0..1 1 0..* 1..*
Add client qualifier	Use this command to designate a qualified association to reduce multiplicity for associations.
Add supplier qualifier	Use this command to designate a qualified association to reduce multiplicity for associations.

Model Bookmarks View

The Model Bookmarks view lists bookmarked model elements. Using the context menu you can navigate to the bookmarked element on the diagram or remove a bookmark from the list.

Context Menu Commands

Show in Model Navigator	Highlights the element in the Model Navigator
Select All	Selects all bookmarks
Remove Bookmark	Removes one or more selected bookmarks
Select on Diagram	Highlights the selected bookmark on the diagram. The diagram opens in the Diagram Editor, if necessary.

Compare Editor

Model > Compare With Each >Other (As Models or Model Elements)

Use the **Compare Editor** to review and merge differences in the structure and properties of the models which you have compared.

Structure Compare

Use the **Structure Compare** area to view the differences found during model comparison. Double-click the difference to open details in the **Changed Properties** and **Substructure/Properties Merge** areas.

Show Containment References	Toggles plain and structured view of the **Structure Compare** area. The structured view displays references to containment features (if any) for a particular difference.
Expand	All Expands all nodes in the **Structure Compare** area.
Collapse	All Collapses all nodes in the **Structure Compare** area.
Export Result	Opens the **Export Result** wizard which allows to export the comparison results to a text file.

Substructure/Properties Merge

Use the **Substructure/Properties Merge** area to review the differences found in the structure and between properties of the compared models.

Undo	Steps one step back in the history of the applied commands..
Redo	Steps one step forth in the history of the applied commands
Copy to the left	Moves the selected change from the right window (remote model) to the left window (local model).
Copy to the right	Copy to the right Moves the selected change from the left window (local model) to the right window (remote model).
Navigate	Shows the selected element in the **Model Navigator** view

Problems

Use the **Problems** area to view problems found during model comparison.

Description	Displays the problem description.
Element	Displays the name of the problematic model element.

Source	Specifies in which model the problem occurs: **Left** (local model) or **Right** (remote model).

Related Concepts

Model Compare and Merge

Tool Palette

Window > Show View

ER/Studio Software Architect extends Palette of the platform of by adding model elements to it.

The diagram Palette displays special icons for the supported UML diagrams. When a diagram opens in the Diagram Editor , the appropriate icons appear in the Palette.

In the diagram Palette you see the top level model elements that can be placed on the current diagram.

NOTE: The set of available model elements depends on the type of the current diagram and active profiles.

Related Procedures

Creating a Simple Link

Diagram View

Context menu (in the Model View) > Open Diagram

The **Diagram View** displays model diagrams. Each diagram is presented in its own tab. To open the **Diagram View**, choose a diagram, namespace or a package in the **Model View**, right-click it and choose **Open Diagram** on the context menu.

Most manipulations with diagram elements and links involve drag-and-drop operations or executing right-click (or context) menu commands on the selected elements.

Some of the actions provided by the context menus are:

- Add or delete diagram elements and links
- Add or delete members in the elements
- Create elements by pattern
- Cut, copy, and paste the selected items
- Hyperlink diagrams
- Zoom in and out

Item	Description
Working area	The main part of the **Diagram View** shows the current diagram.

Item	Description
Context menu	The context menus of the **Diagram View** are context-sensitive. Right-clicking model elements, including class members, provides access to element-specific operations on the respective context menu. Right-clicking the diagram background opens the context menu for the diagram.
Overview	Opens the Overview pane (see below).

Overview pane

The overview feature of the **Diagram View** provides a thumbnail view of the current diagram. The Overview button is located in the bottom right corner of every diagram.

OCL Editor

The OCL Editor is used to enter and edit OCL expressions. Any changes to the names of your model components (classes, operations, attributes, and so on) used in these expressions are automatically updated. This guarantees that your OCL constraints always stay up-to-date.

Related Concepts

Diagram Overview

OCL Support

Related Procedures

Creating a Diagram

Navigating Between the Tree View and Diagram

Model Navigator

Window > Show View > Other > Modeling > Model Navigator

The Model Navigator provides the logical representation of the model of your project: namespaces (packages) anddiagram nodes. Using this view, you can add new elements to the model; cut, copy, paste and delete elements, and more. Context menu commands of the Model Navigator are specific to each node. Explore these commands as you encounter them.

In the Model Navigator, only the nodes and their respective subnodes shown in theDiagram Editor , are listed under the corresponding diagram node. For example, if you have a package containing a class, both the package and class are shown under the diagram node in the Model Navigator. However, any members of the class are not shown under the diagram node as they are displayed under the namespace (package) node only.

Model Navigator is a dockable window. The docking areas are any of the four borders of the window. You can position the Model Navigator according to your preferences.

The following options are applicable to the Model Navigator:

- In the **Show diagram nodes expandable** field (**Options > Model View**) choose *True* to show, or *False* to hide expandable diagram nodes. By default, the **Model View** displays expandable diagram nodes with the elements contained therein. You can hide expandable diagram nodes to further simplify the visual presentation of the project.
- In the **Show links** field (**Options > Model View**) choose *True* to show, or *False* to hide expandable diagram nodes. By default, the Model View does not display links between nodes. You can opt to show the links to make the visual presentation of the project more detailed.
- In the **Sorting type** field (**Options > Model View**) choose *Metaclass*, *Alphabetical*, or *None* to sort elements in the **Model View**. By default, diagram nodes are sorted by metaclass. You can sort elements in the **Model View** by metaclass, alphabetically, or none.e
- In the **View type** field (**Options > Model View**) choose *Diagram-centric* or *Model-centic* from the list box. For the sake of better presentation you can opt to show your model in diagram-centric or in modelcentric modes. The Diagram-centic mode assumes that the design elements are shown under their respective diagrams. The Model-centric mode assumes that all elements are shown under the namespaces.

Item	**Description**
Root project node	The topmost item of a project structure.
Nodes	Namespaces (packages), diagrams, then model elements of the current model.
Refresh Model View	..

Related Concepts

UML Modeling Overview

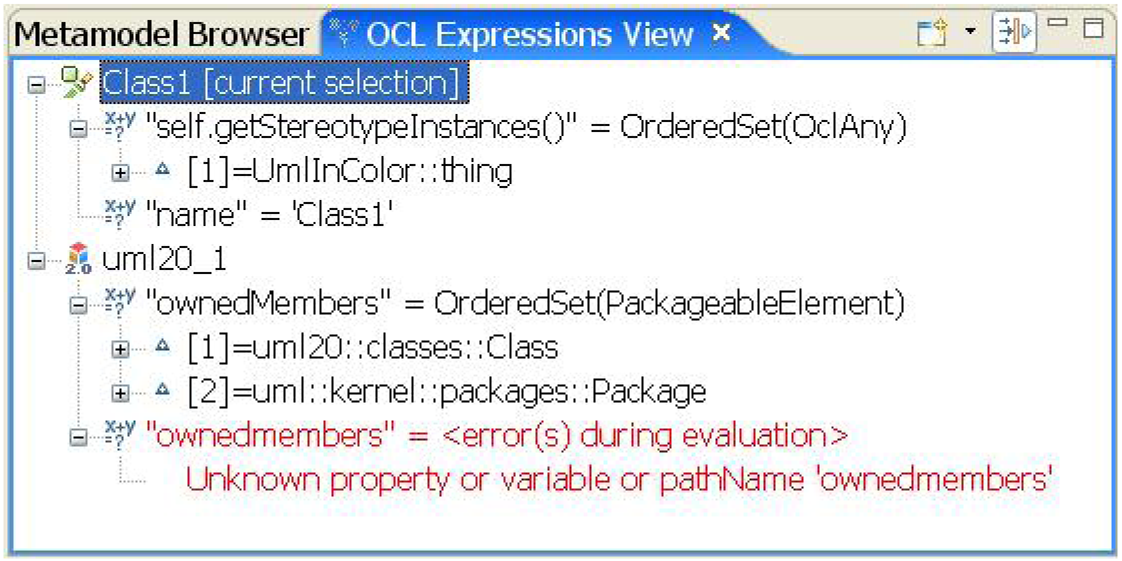
Related Procedures

Creating a Diagram

Navigating Between the Tree View and Diagram

OCL Expressions View

Use the **OCL Expressions** view to quickly evaluate OCL expressions in the explicitly specified context (an EMF model element), or in the context of the current selection. The screen shot below shows Class1 from UML 2.0 model in the current selection, and the model itself explicitly added as the second context.



```
Class1 [current selection]
  "self.getStereotypeInstances()" = OrderedSet(OclAny)
    [1]=UmlInColor::thing
  "name" = 'Class1'
uml20_1
  "ownedMembers" = OrderedSet(PackageableElement)
    [1]=uml20::classes::Class
    [2]=uml::kernel::packages::Package
  "ownedmembers" = <error(s) during evaluation>
    Unknown property or variable or pathName 'ownedmembers'
```

The children of Class1 element are expressions in uml20::classes::Class OCL context. The children of the second element are expressions in uml::xxx::Model OCL context . The **OCL Expressions** view displays a tree structure with several levels. The first level contains contexts, with the predefined **Current selection** context. The second level contains OCL expressions.

A context element specifies the model element against which ER/Studio Software Architect evaluates child expressions. The OCL context of child expressions is defined by the type of the context element. For example, children of UML 2.0 class context element will be evaluated in uml20::classes::Class OCL context.

The **OCL Expressions** view preserves manually added contexts and OCL expressions after the Workbench restart. The **OCL Expressions** view evaluates your OCL expressions in real time as you work on your model. Evaluation results display either in expression labels (for simple results), or as expression children (for model- or collection-like results). Evaluation errors display in red with <errors during evaluation> label, error messages display as children of the expression node.

NOTE: To use OCL in EMF models you should enable the corresponding metamodels in the **OCL Preferences** dialog.

Toolbar

Item	Description
Add Model Element...	Opens the **Model Elements** dialog which allows to choose a context model or model element from your workspace and add it to the current list of contexts.
Add EMF Model Element...	Opens the **Workspace Contents** dialog which allows to choose a context EMF model or model element from your workspace and add it to the current list of contexts.
Add URI...	Opens the **Add model** dialog which allows to specify the URI of a context model or model element which you want to add to the current list of contexts.
Hide/Unhide Empty Nodes	Hides/Displays empty expression nodes
Refresh	Reevaluates the selected OCL expressions and reloads the selected contexts. Note, that the **OCL Expressions** view does not automatically refresh contexts and expressions when the referenced models change.

Context Menu

Item	Description
Add Model Element...	Opens the **Model Elements** dialog which allows to choose a context model or model element from your workspace and add it to the current list of contexts.
Add EMF Model Element...	Opens the **Workspace Contents** dialog which allows to choose a context EMF model or model element from your workspace and add it to the current list of contexts.
Add URI...	Opens the **Add model** dialog which allows to specify the URI of a context model or model element which you want to add to the current list of contexts.
Add Expression...	Opens the **Edit...** dialog which allows to compose a new OCL expression and add it as a direct child of the selected context model or model element. The dialog supports OCL error reporting and code completion options.
Edit...	Opens the **Edit...** dialog which allows to edit the selected OCL expression. Enable Expression Enables real-time evaluation of the selected OCL expression node and all its children.
Disable Expression	Disables real-time evaluation of the selected OCL expression node and all its children.

Item	Description
Delete	Removes the selected expression or context model element from the view.
Refresh	Reevaluates OCL expressions currently in the **OCL Expressions** view.

Related Concepts

About OCL Support

Properties View

Context menu > Properties

Every diagram and element has a general Properties View. The composition of the Properties View changes depending on the element or diagram selected in the Diagram Editor or Model Navigator. Use the Properties View to set properties for all diagrams and elements.

The Properties View displays properties in two columns: **Property** and **Value**. The **Property** column displays the names of the properties of a selected resource. The **Value** value column displays the values of the properties of a selected resource. Double-click on a value to edit.

Groups of properties display on the left of the Properties View

Properties

This section displays the common properties of the selected object. The number of fields in this section varies depending on the selected diagram or element. See each element description for details.

Custom

This section displays the custom properties and their values specified for the selected object.

Add	Creates a new entry in the list of properties.
Remove	Deletes the selected entry from the list of properties.

Description

Use this field to add description text for a diagram or element. You can provide descriptions of the model elements in the Edit tab, using the in rich text formatting, and view the resulting text in the Browser tab.

Edit Tab Item	Description
B	Applies bold style.

Edit Tab Item	Description
I	Applies italic
U	Applies underlining
S	Applies strikeout
Change font color	Opens the color chooser dialog for the font of the selected elements.
Change background color	Opens the color chooser dialog for the background of the selected elements.

Brower Tab Item	Description
Stop loading the current page	
Refresh the current page	
Load the initial content	

Hyperlinks

Use this field to add hyperlinks to the element.

Item	Description
Add hyperlink	Opens the Hyperlinks dialog, where you can select elements to be linked with the current element.
Remove	Deletes the selected hyperlink.
Remove all	Removes all hyperlinks from the element.

Requirements

You can track various requirements properties including type, priority, and difficulty for diagrams and individual elements. You can specify a requirements document for the diagram or elements.

traces	Displays the number of traces associated with the selected element.
author	Use this field to add author properties. Click ellipse to add values.
difficulty	Use the drop down list to set difficulty to High or Low. Medium represents the default value.
document	Use the document field to link to a specific document.

number	Use the number field to assign a requirement number.
priority	Use the drop down list to set priority to High or Low. Medium represents the default value.
req. description	Use this field to add a requirements description. Click ellipse to add text.
testcase	Use the text field to designate a testcase requirement.
type	Use the drop down list to set the appropriate type. You can choose from Business Rule, Feature, Performance, Product Requirement, or User Need.

View

Diagrams do not have view properties. Each element (class, interface, object, and so on) will have the view properties listed below:

2D look	The property value can be set as Yes or No. Yes represents the default value.
background color	Use this field to add author properties. Click ellipse to add values.
foreground color	This field sets the RGB background color for the element. {255, 255, 255} represents the default color value. Use the drop down list to choose a color.

Properties View

Show/Hide Categories:	This button groups lines under their appropriate categories
Filter Properties	This button determines whether advanced properties are displayed in this view. Basic properties are always shown.
foreground color	This field sets the RGB background color for the element. {255, 255, 255} represents the default color value. Use the drop down list to choose a color.
Restore Default Value	If you make changes to a value, this button restores the selected property to its default value.
Menu	Displays the Show/Hide Categories and Filter Properties commands.
Minimize	Minimizes current properties view to the view title. To show the entire view, click Restore.

Maximize	Maximizes the current view to the entire window. To restore the view, click Restore.
Information	Available for certain properties. May display a small text editor for larger text entries, a selection wizard, or a file chooser dialog.
Browse	Displays the Selection dialog: Available for certain properties. Displays the Selection dialog, enabling you to select an element from the Model.
Drop down	Available for certain properties. Displays a list of available options to choose from.

XSL Editor

Use the **XSL Editor** to write your XSL transformation scripts.

The **XSL Editor** provides XSL code sense and auto-complete (CTRL+SPACE) options. The XSL-specific **Outline** view displays an outline of the structure of the currently-active XSL file in the editor area.

Context Menu

Item	Description
Show Source	Navigates to the element which is the source of the selected transformation step
Show Target	Navigates to the element which is the result of the selected transformation step.

Trace Synchronizer View

Window > Show View > Other... > Requirements > Trace Synchronizer

This topic provides information about the **Trace Synchronizer** view. You can use this view to find and fix desynchronized traces to CaliberRM or RequisitePro requirements.

Toolbar Icons and Context Menu Items

Synchronize Traces	Opens the Trace Synchronizer dialog.
Refresh trace synchronization information	Refreshes the trace information displayed in the Trace Synchronizer view.
Save as HTML	Opens the Save As dialog, where you can export the current content of the Trace Synchronizer view to an HTML file.

Update Trace	Discards local changes and updates the selected traces from the repository.
Restore Trace	Discards changes in the repository and restores the requirement information stored in the model.
Delete Trace	Deletes the trace.
Navigate to Trace Source	Opens the trace source (requirement) in the CaliberRM or RequisitePro Navigator depending on the requirement type.
Navigate to Trace Target	Opens the trace target (model element) in the appropriate editor.

Columns

Status	Displays the status of the trace source.
Trace from	Displays the name of the trace source.
Trace from project	Displays the name of the source CaliberRM project.
Status	Displays the status of the trace target.
Trace to	Displays the name of the trace source.
Trace to project	Displays the name of the target CaliberRM project.
Status summary	Displays the summary information about the current trace status.

Status items

Not Found	Information about the object is not found..
Current	Information about the object is up to date.
Missed	Information about the object is missing.
New	The object is new.
Modified	The object has been modified.
Outdated	The object becomes outdated.

Last Validation Results View

Window > Show View > Other > Patterns >Last Validation Results

The **Last Validation** view displays results of the latest validation of a pattern definition. This view opens automatically, when validation process reports about errors.

UML 2.0 Reference

This section contains reference material about UML 2.0 diagrams.

In This Section

UML 2.0 Class Diagrams

This section describes the elements of UML 2.0 Class diagrams.

In This Section

Topic	Description
UML Class Diagram Elements	Gives the list UML 2.0 class diagram elements.
Class Diagram Relationships	Describes class diagram relationships for UML 2.0 specifications.
Class Diagram Properties Association Class and N-ary Association	Describes association class and n-ary association
Dependency Link Properties Generalization/Implementation Link Properties	
Operation Context Menu	

UML Class Diagram Elements

The table below lists the elements of UML 2.0 class diagrams that are available using the Palette. Note that availability of the elements depends on the project type and profiles.

Package	Node
Class	Node
Interface	Node
Enumeration	Node
Data type	Node. Available in design projects only.

Association class	Node
Port	Node. Available in design projects only.
Instance specification	Available in the source code projects
Generalization/Implementation	Link
Required interface	Link
Provided interface	Link
Association	Link
Association end	Link
Dependency	Link
Instantiates	Link
OCL constraint	Node
Constraint link	Link
Template signature	Node
Template binding	Link
Note	Annotation
Note link	Annotation link

Class Diagram Relationships

There are several kinds of relationships for UML 1.4 and UML 2.0 Class diagrams.

Types of Relationships

Association	A relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other to perform its work. In a diagram, an association is a link connecting two classes. Associations can be directed or undirected. A directed link points to the supplier class (the target). An association has two ends. An end may have a role name to clarify the nature of the association. A navigation arrow on an association shows which direction the association can be traversed or queried. A class can be queried about its Item, but not the other way around. The arrow also lets you know who "owns" the implementation of the association. Associations with no navigation arrows are bi-directional. Aggregation: An association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole.
Generalization/Implementation	An inheritance link indicating that a class implements an interface. An implementation has a triangle pointing to the interface.
Dependency	
Required interface	Available in UML 2.0 class diagram. Applying a provided interface link to a port on the client class creates a link in ball-and-socket notation.
Provided interface	Available in UML 2.0 class diagram. Applying a provided interface link between a class and an interface creates a regular generalization/implementation link.
Instantiates	Available in UML 2.0 class diagram. This link can be drawn between an instance specification and its instantiated class

Multiplicities

Every class diagram has classes and associations. Navigability, roles, and multiplicities are optional items placed in a diagram to provide clarity.

The multiplicity of an association end is the number of possible instances of the class associated with a single instance of the other end. Multiplicities are single numbers or ranges of numbers.

- 0..1 Zero or one instance. The notation n . . m indicates n to m occurrences
- 0..* or No limit on the number of occurrences (including none)
- 1 Exactly one occurrence
- 1..* At least one occurrence

Class Diagram Properties

This section describes the properties specific to attributes of classes, inner classes, and interfaces. Every element has general properties as well as specific properties. For more information, see General Properties. The composition of the Properties view changes depending on the element selected in the Diagram Editor or Model Navigator view. You can view and modify values of properties through theProperties View.

diagram type	Shows the current diagram type.
name	The name of the class diagram.
stereotype	Choose the appropriate stereotype from the drop down list, or add your own stereotype. The available stereotypes are: ▪ data management ▪ facade ▪ framework ▪ human interaction ▪ problem domain ▪ stub ▪ subsystem ▪ system ▪ system interaction

Association Class and N-ary Association

Association classes appear in diagrams as three related elements:

- Association class itself (represented by a class icon)
- N-ary association class link (represented by a diamond)
- Association connector (represented by a link between both)

Association classes can connect to as many association end classes (participants) as required.

TheProperties View of an association class, association link, and connector contains an additional Association tab. This tab displays the only label property, its value being synchronized with the name of the association class. For the association classes and association end links, the Custom node of the Properties View displays additional properties that correspond to the role of this part of n-ary association (associationClass and associationEnd respectively).

You can delete each of the association end links or participant classes without destroying the entire n-ary association. However, deleting the association class results in deleting all the components of the n-ary association.

Dependency Link Properties

This section describes the dependency links' specific properties. Every element has common properties as well as specific properties. For more information, see General Properties. The composition of the Properties View changes depending on the element selected in theDiagram Editor or Model Navigator view. You can view and modify values of properties through the Properties View.

Property	Description
client	This property field indicates the client for the link.
client role	Use client role to add a label to the dependency link. The label appears on the UML diagram towards the client side of the link
label	Use label to add a label to the dependency link. The label appears on the UML diagram between the client and the supplier.
stereotype	Use this field to add your own stereotype property.
supplier	This property field indicates the supplier for the link.
supplier role	Use supplier role to add a label to the dependency link. The label appears on the UML diagram towards the supplier side of the link.

Generalization/Implementation Link Properties

This section describes the properties specific to generalization/implementation links. Every element has general properties as well as specific properties. For more information, see General Properties. The composition of the Properties view changes depending on the element selected in the Diagram or Model Navigator view. You can view and modify values of properties through the Properties view.

Item	Description
client	This property field indicates the client for the link.
supplier	This property field indicates the supplier for the link.

Operation Context Menu

All of the UML diagram elements share common context menu commands. To use the context menu for an element, simply right click on the element in the Diagram editor. To view the common context menu commands, see Common Element Context Commands.

The context menu for an operation shares the common element context commands as well as the following commands specific to it:

Open

Selecting Open from the context menu, opens the selected class containing the operation in the text editor.

Show in Packages View

The Show in Packages View command highlights the node selected in the Packages tree-view. The Packages view will expand and highlight that element in the tree-view. If closed, the Packages view will open.

Show in Model Package Explorer

The Show in Model Package Explorer View command highlights the node selected in the UML Explorer tree-view. The Model Package Explorer view will expand and highlight that element in the tree-view. If closed, this view will open.

Modifiers

The Modifiers command for the operation offers a submenu with the following options

Static	Selecting Static from the context menu sets the static property for the operation.
Abstract	Selecting Abstract from the context menu sets the abstract property for the operation.
Public	Selecting Public from the submenu sets the visibility property for the operation to public.
Protected	Selecting Protected from the submenu sets the visibility property for the operation to protected
Private	Selecting Private from the submenu sets the visibility property for the operation to private.
Package Local	Selecting Package Local from the submenu sets the visibility property for the operation to package local

NOTE: The visibility options are not available for Interface members.

Add Javadoc comment

Using this command, you are able to add Javadoc comments for the operation.

UML 2.0 Use Case Diagrams

This section describes the elements of UML 2.0 Use Case Diagrams.

In This Section

UML 2.0 Use Case Diagram Elements	Describes UML 2.0 use case diagram elements.
Extension Point	Describes an extension point (Use Case diagrams).

UML 2.0 Use Case Diagram Elements

The table below lists the elements of UML 2.0 Use Case diagrams that are available using the Palette.

Name	Type
Actor	node
Subject	node
Use Case	node
Extension point	node Creates extension points in the use cases in order to specify a point in the behavior of a use case, where this behavior can be extended by the behaviot of some other use case. This element is available on the context menu of a use case.
Extends	link
Includes	link
Generalization	link
Association	link
Note	annotation
Note Link	annotation link

Extension Point

An **extension point** refers to a location within a use case where you can insert action sequences from other use cases..

An extension point consists of a unique name within a use case and a description of the location within the behavior of the use case.

In a use case diagram, extension points are listed in the use case with the heading "Extension Points" (appears as bold text in the Diagram View).

UML 2.0 Interaction Diagrams

This section describes the elements of UML 2.0 Communication and Sequence diagrams.

In This Section

UML 2.0 Sequence Diagram Elements	Describes UML 2.0 sequence diagram elements.
UML 2.0 Communication Diagram Elements	Describes UML 2.0 communication diagram elements.
Interaction	Describes Interaction
UML 2.0 Message	Describes UML 2.0 messages (Interaction diagrams).
Execution Specification and Invocation Specification	Describes an execution specification and invocation specification.

Operator and Operand for a Combined Fragment	About operator and operand for a combined fragment.
Clipboard operations with execution and invocation specifications	Provides information about clipboard operations with execution and invocation specifications.

UML 2.0 Sequence Diagram Elements

The table below lists the elements of UML 2.0 sequence diagrams that are available using the Palette.

Name	Type
Lifeline	Draws an object with its lifeline in an interaction. For each lifeline its projection bar displays on top of the diagram. When scrolling down, these projection bars are always visible.
Interaction	Draws an interaction node in diagram.
Message	Draws a message link between the source and target lifelines.
Found Execution	Draws a message link to a target lifeline. The source of such message is unknown.
State Invariant	Draws a state invariant node on a lifeline.
Action Execution	Draws an action execution node on a lifeline.
Combined fragment	Draws a combined fragment node on a lifeline.
Interaction use	Draws an interaction use node on a lifeline.
OCL constraint	node
Constraint link	link
Template Signature	node
Template Binding	link
Note	annotation
Note Link	annotation link

CAUTION: Sequence diagram can contain shortcuts to the other diagram elements. However, shortcuts to the elements that reside in the other interaction diagrams are not supported.

NOTE: Interaction diagrams, represented in the Model Navigator, display a number of auxiliary elements that are not visible in the Diagram Editor . These elements play supplementary role for representation of the diagram structure. Actually, these elements are editable, but it is strongly advised to leave them untouched, to preserve integrity of the interaction diagrams.

UML 2.0 Communication Diagram Elements

The table below lists the elements of UML 2.0 communication diagrams that are available using the Palette.

Name	Type
Lifeline	node
Interaction	node
Message	link
OCL	constraint node
Constraint link	link
Template Signature	node
Template Binding	link
Note	annotation
Note Link	annotation link

NOTE: Interaction diagrams, represented in the **Model View**, display a number of auxiliary elements that are not visible in the **Diagram View**. These elements play supplementary role for representation of the diagram structure. Actually, these elements are editable, but it is strongly advised to leave them untouched, to preserve integrity of the interaction diagrams.

Interaction

By using ER/Studio Software Architect, you can create interactions for the detailed description and analysis of inter-process communications..

Interactions can be visually represented in your projects by means of the two most common interaction diagrams: Sequence and Communication. On the other hand, interactions can exist in projects without visual representation.

Interaction use

Within an interaction, you can refer to the other interactions described in your project. So called “Interaction use” elements serve this purpose. Note that referenced interaction can be explicitly defined from the model, or just specified as a text string.

Each interaction use is attached to its lifeline with a black dot. This dot is an individual diagram element. If an interaction use is expanded over several lifelines, you can delete the attachment dots from all lifelines but one. An interaction use should be connected with at least one lifeline.

Lifeline

A lifeline defines an individual participant of the interaction. A lifeline is shown in a sequence diagram as a rectangle followed by a vertical-dashed line.

Lifelines of an interaction can represent the parts defined in the class or composite structure diagrams. If the referenced element is multivalued, then the lifeline should have a selector that specifies which particular part is represented by this lifeline.

If a lifeline represents a connectable element, has type specified, or refers to another interaction, the Select menu becomes enabled on the context menu of this lifeline. Using this menu, you can navigate to the part, type or decomposition associated with the lifeline. These properties are defined by using the Properties View. If the **represents** property is set, the **type** and **name** properties are disabled.

You can define these properties manually by typing the values in the corresponding fields of theProperties View If the specified values are not found in the model, they are displayed in single quotes. Such references are not related to any actual elements and the Select menu is not available for them. If the specified values can be resolved in the model, they are shown without quotes, and the Select menu is available for them.

State invariant

A state invariant is a constraint placed on a lifeline. This constraint is evaluated at runtime prior to execution of the next execution specification. State invariants are represented in the interaction diagrams in two ways: as OCL expressions or as references to the state diagrams. You can use the state invariants to provide comments to your interaction diagrams and to connect interactions with states.

It is important to note that ER/Studio Software Architect provides validation of the state invariants represented as OCL expressions. If the syntax is wrong, or there is no valid context, the constraint is displayed red. For example, to be a valid context, a lifeline should have **type** and **represents** properties defined.

UML 2.0 Message

Call messages are always visible in diagrams; reply messages normally are not displayed. However, you can visualize the reply message.

Messages on different diagram types

- **Messages in communication diagrams:** When you draw a message between lifelines, a generic link line displays between the lifelines and a list of messages is created under it. The link line is present as long as there is at least one message between the lifelines.
- **Messages in sequence diagrams:** Messages in sequence diagrams have the same properties as those in communication diagrams but allow you to perform more actions. The further discussion refers mainly to the sequence diagram messages.

Properties of the messages for both types of interaction diagrams can be edited in the Properties View.

Properties of the message links

Call messages have the following properties:

Property	Description
Signature	Use this field to specify the name of an operation or signal associated with the message. Note that changing the signature of a message call results in changing the signature of the corresponding reply.

Property	Description
Sort	Use this field to select the type of synchronization from the drop-down list. The possible values are: **asynchCall, synchCall, asynchSignal**. The message link changes its appearance accordingly. There are certain limitations related to the asynchronous calls: Sometimes it is impossible to create or paste an asynchronous call because of the frame limitations. Execution specification for an asynchronous call must always be located on a lifeline.
Name	Displays the link name. This field is editable.
Full name	Displays the fully qualified link name. This field is not editable.
Visibility	Use this field to select the visibility modifier from the drop-down list.
Stereotype	Use this field to define the message stereotype. The stereotype name displays above the link.
Metaclass	This read-only field displays the message metaclass.
Label	Use this field to define the link label.
Attribute	Use this field to define the link attribute.
Arguments	Displays actual arguments of an operation associated with a message call. This field is editable.
Return value	Use this field to enter the return value.
Commentary	Use this textual field to enter comments for a message link
Show reply message	Use this Boolean option to define whether to draw a dashed return arrow.
Sequence number	Use this field to view and edit the sequential number of a message. When the message number changes, the message call changes respectively.
no duration	Use this Boolean option to make the invocation and execution specifications invisible in diagram. This option can be only specified for the link that has no operation.

Property	Description
creation	Use this Boolean option to define creation message. If this option is true, the message link points to the lifeline object node.
destruction	Use this Boolean option to define destruction message. If this option is true, the message link points to the execution specification marked with a cross sign.

Reply messages have the following properties:

Property	Description
Stereotype	Use this field to define the message stereotype
Attribute	Use this field to define an attribute to which the return value of the message will be assigned. This field can be edited.
Signature	Use this field to specify the name of an operation or signal associated with the message. Note that changing the signature of a message reply results in changing the signature of the corresponding call.
Arguments	Displays arguments of an operation associated with a message call. This field can be edited. Note that changing the list of arguments of a reply message results in changing the corresponding call.
Return value	Displays the return value of an operation associated with a message link. This field can be edited.
Sort	Use this field to select the type of synchronization from the drop-down list. The possible values are:**asynchCall,** **synchCall, asynchSignal.** The message link changes its appearance accordingly.
Commentary	Use this text field to comment the link.

NOTE: Such properties of the call and reply messages as arguments, attribute, qualified name, return value, signature, and sort pertain to the invocation specification. You can edit these properties in the invocation specification itself, in the call or in the reply messages. As a result, the corresponding properties of the counterpart message and the invocation specification will change accordingly. Stereotype and commentary properties are unique for the call and reply messages.

Execution Specification and Invocation Specification

In sequence diagrams, ER/Studio Software Architect automatically renders invocation specification and execution specification of a message that shows the period of time when the message is active. When you draw a message link from the source lifeline to the destination lifeline, the invocation and execution specification bars are created automatically. You can extend or reduce the period of time of a message by vertically dragging the top or bottom line of the invocation or execution specification as required.

For an invocation or execution specification you can define **no duration** property. If this property is checked for one specification, it will be automatically checked for the other one. Also, you can define this property for the message. If **no duration** property is set to **true**, the specification icons reduce to the minimal possible dimensions and become invisible. By default the execution specification is synchronized with the invocation specification. You can make invocation specification and execution specification asynchronous.

It is also possible to create an execution specification on a lifeline without creating an incoming message link. In this case a found message is created, that is a message that comes from an object that is not shown in diagram. Use theProperties View to hide or show the found messages.

Messages in sequence diagrams have their origin in an invocation specification. This is an area within an execution specification. Though this element is not defined in the UML 2.0 specification, it is a useful tool for modeling synchronous invocations with the reply messages. In particular, invocation specification marks a place where the reply messages (even if they are invisible) enter the execution context of a lifeline, and where sub-messages may reenter the lifeline.

Active and passive areas of the execution specification are rendered in different colors. The white execution specification bars denote active areas where you can create message links. The gray bars are passive and are not a valid source or target for the message links.

Operator and Operand for a Combined Fragment

In this section:

- About combined fragment
- Operator
- Operand

About combined fragment

A combined fragment can consist of one or more interaction operators and one or more interaction operands. Number

of interaction operands (just one, or more than one) depends on the last interaction operator of this combined fragment.

Use the Palette, or context menus to create these elements. The operator type shows up in the descriptor in the upper-left corner of the design element. Note that you can define multiple operators in a combined fragment. In this case, the descriptor contains the list of all operators, which is a shorthand for nested operators.

When an operator is created, add the allowed operands, using the combined fragment's context menu. A combined fragment can be expanded over several lifelines, detached from and reattached to lifelines. In the Properties View, use the **Operators** field to manage operators within the combined fragment.

Each combined fragment is attached to its lifeline with a mounting link that displays in diagram as a black dot. This mounting link is an individual diagram element, which can be selected or deleted. Deleting a mounting link means detaching a combined fragment from the lifeline. Note that a combined fragment cannot be detached from all lifelines and should have at least one attachment dot.

You can reattach a combined fragment later, using the anchor tool.

Operator

When a combined fragment is created, the operator displays in a descriptor pentagon in the upper left corner of the frame. You can change the operator type, using the **operator** field of the Properties View, which is immediately reflected in the descriptor.

The descriptor may contain several operators. UML 2.0 specification provides this notation for the nested combined fragments.You can use this notation, or create nested combined fragment nodes.

Operand

Operands are represented as rectangular areas within a combined fragment, separated by the dashed lines. When a combined fragment is initially created, the number of operands is defined by the pattern defaults. Further, you can create additional operands, or remove the existing ones.

Note that the uppermost area of the operator is empty and does not contain any operands. It is reserved for the descriptor. Clicking on this area selects the entire operator; clicking on one of the dotted rectangles selects the corresponding operand. If a combined fragment contains only one operand, the entire combined fragment and the single existing operand are still separately selectable.

Clipboard operations with execution and invocation specifications

Clipboard operations are supported for the execution and invocation specifications. Cut, Copy, and Paste commands are available on the context menu of an execution specification and invocation specification. It is possible to copy or move these elements within the same diagram or to another diagram.

When an execution or invocation specification is copied, it means that the entire branch of messages is copied also. Pasting the clipboard contents to a target lifeline results in changing the message numbers according to the numbering of messages in the target lifeline.

If you paste an invocation or execution specification to another diagram, the entire outgoing bunch of messages will be pasted also, with all the respective lifelines. If the target diagram does not contain lifelines for this execution specification, they will be created automatically.

It is also possible to move and copy message branches using the drag-and-drop technique. To move an execution or invocation specification, drag-and-drop it to the target location. To create a copy, drag-and-drop while holding the CTRL key down.

UML 2.0 State Machine Diagrams

This section describes the elements of UML 2.0 State Machine Diagrams.

In This Section

UML 2.0 State Machine Diagram Elements	Describes UML 2.0 state machine diagram elements.
State Machine Diagram Context Commands	
State Machine Diagram Elements Properties\	
Transition	Describes a transition (UML 1.4 Activity, UML 1.4 Statechart, UML 2.0 State Machine diagrams).
History Element (State Machine Diagrams)	Describes UML 2.0 history.

UML 2.0 State Machine Diagram Elements

The table below lists the elements of UML 2.0 State Machine diagrams that are available using the Palette.

Name	Type
State Machine	node A state machine describes the behavior of a part of a system. A state machine owns one or more regions.
State	node A state models a situation during which some invariant condition holds.
Entry point	node Execution of the state starts at this point. It is possible to create several entry points for one state, that makes sense if there are substates.
Exit point	node Execution of the state finishes at this point. It is possible to create several exit points for one state, that makes sense if there are substates.
Initial	node
Final	node
Terminate	node
Shallow history	node
Deep history	node
Region	node Use regions inside the states to group the substates. The regions may have different visibility settings and history elements. Each state has one region immediately after creation (though it can be deleted.) In the regions, you can create all the elements that are available for the State Machine diagram. only available on the state context menu
Fork	node
Join	node
Choice	node
Junction	node
Transition	link Draws a link from the exit point of source state (or the state without exit points) to the entry point of the destination (or the state without points).
Internal Transition	link Internal transition elements are only available on the state context menu. Note annotation

Name	Type
Note Link	annotation link

State Machine Diagram Context Commands

Diagram Editor > Right-click

All of the UML 2.0 diagrams share common context menu commands. To use the context menu for a diagram, right click in the Diagram Editor .

To view the common context menu commands, see Common Diagram Context Commands.

The State Machine diagrams offer the following context commands.

Diagram Context Menu

Option	Description
New	The **New** command for the State Machine diagram offers a submenu with the following options: • State Machine • Constraint • Note • Shortcut

State Machine Context Menu

Option	Description
New	The **New** command for the State Machine element offers a submenu with the following options: • Entry point • Exit point • Region

Region Context Menu

Option	Description
New	The **New** command for the Region element offers a submenu with the following option: • State • Initial • Final • Shallow history • Deep history • Terminate • Fork • Join • Choice • Junction • Note

State Context Menu

Option	Description
New	The **New** command for the State element offers a submenu with the following options: • Internal Transition • Reference To Entry Point • Reference To Exit Point • Region

State Machine Diagram Elements Properties\

This section describes the State Machine diagram elements specific properties. Every element has common properties as well as specific properties. For more information, see General Properties. The composition of the Properties View changes depending on the element selected in theDiagram Editor or Model Navigator view. You can view and modify values of properties through the Properties View.

State Machine element

- abstract
- context
- extends
- final
- full name
- is reentrant
- metaclass
- name
- parameters
- specification
- stereotype
- visibility

Region element

- full name
- metaclass
- name
- stereotype
- visibility

Entry Point element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Exit Point element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Reference to Entry Point element

- entry
- full name
- metaclass
- name
- stereotype
- visibility

Reference to Exit Point element

- exit
- full name
- metaclass
- name
- stereotype
- visibility

State element

- do activity link
- entry activity link
- exit activity link
- full name
- is composite
- is orthogonal
- is simple
- is submachine state
- metaclass
- name
- stereotype
- submachine
- visibility

Initial element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Final element

- do activity link
- entry activity link
- exit activity link
- full name
- is composite
- is orthogonal
- is simple
- is submachine state
- metaclass
- name
- stereotype
- submachine
- visibility

Shallow History element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Deep History element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Terminate element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Fork element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Join element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Choice element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Junction element

- full name
- kind
- metaclass
- name
- stereotype
- visibility

Transition link

- client
- effect
- full name
- kind
- label
- metaclass
- name
- stereotype
- supplier
- trigger
- visibility

Dependency link

- client named element
- full name
- label
- metaclass
- name
- stereotype
- supplier named element
- visibility

Transition

A single transition comes out of each state or activity, connecting it to the next state or activity.

A transition takes operation from one state to another and represents the response to a particular event. You can connect states with transitions and create internal transitions within states.

Internal transition

An internal transition is a way to handle events without leaving a state (or activity) and dispatching its exit or entry actions. You can add an internal transition to a state or activity element.

Self-transition

A self-transition flow leaves the state dispatching any exit action(s), then reenters the state dispatching any entry action(s).

Guard expressions

All transitions, including internal ones, are provided with the guard conditions (logical expressions) that define whether this transition should be performed. Also you can associate a transition with an effect, which is an optional activity performed when the transition fires. The guard condition is enclosed in the brackets (for example, "[false]") and displayed near the transition link on a diagram. Effect activity is displayed next to the guard condition. You can define the guard condition and effect using the **Properties Window**.

Guard expressions (inside [ ]) label the transitions coming out of a branch. The hollow diamond indicates a branch and its subsequent merge that indicates the end of the branch.

History Element (State Machine Diagrams)

The Shallow History and Deep History elements are placed on regions of the states.

There may be none or one Deep History, and none or one Shallow History elements in each region. If there is only one history element in a region, it may be switched from the Deep to Shallow type by changing its kind property. Refer to UML 2.0 Specification for more information.

UML 2.0 Activity Diagram

This section describes the elements of UML 2.0 Activity Diagrams.

In This Section

UML 2.0 Activity Diagram Elements	Describes UML 2.0 activity diagram elements.
UML 2.0 Activity Diagram Context Commands	Describes context menu commands of UML 2.0 activity diagram.

UML 2.0 Activity Diagram Elements

The table below lists the elements of UML 2.0 Activity diagrams that are available using the Palette.

Activity	Node Activities are action states in an activity diagram. Action states are states with outgoing transitions that are triggered by the completion of an action associated with the state
Activity parameter	Node component An activity parameter node is an object node for inputs and outputs to activities.
Activity partition	An activity partition is a kind of activity group for identifying actions that have some characteristic in common
Action	An action is an executable activity node
Initial	Node at which flow starts when the activity is invoked.
Activity final	Node at which a flow in an activity stops
Decision	Node A decision element indicates possible transitions relative to Boolean conditions of the owning object. The decision represents a branch in the control flow of an activity diagram
Merge	Node that brings together multiple alternate flows.
Flow final	Node that terminates a flow.

Control flow	Link that starts an activity node after the previous one is finished.
Input pin	Pin
Output pin	Pin that holds input values to be consumed by an action.
Value pin	Input pin that provides a value to an action that does not come from an incoming object flow
Object node	Node that is a part of defining object flow in an activity.
Value buffer	Node
Data store	Node
Object Flow	An object flow relationship can be drawn: from an Activity to an Object from SignalSending to an Object from an Object to SignalReceipt from/to an Object to/from a Fork/Join
Accept Event Action	Node
Accept Time Event Action	Node
Send Signal Action	Send Signal Action is an explicit symbol used on an activity diagram for certain kinds of information that can be specified on transitions.
Note	annotation
Note link	annotation link

UML 2.0 Activity Diagram Context Commands

All of the UML diagram types share common context menu commands. To use the context menu for a diagram, right click in the Diagram editor.

To view the common context menu commands, see Common Diagram Context Commands.

Diagram Context Menu

Activity	Adds activity element to the diagram.
Constraint	Adds a constraint element to the diagram.
Note	Adds a note element to the diagram.
Shortcut	To refer to an element located outside of the current diagram, or to another diagram, you can use shortcuts. Invoking the Shortcut command displays a selection dialog, where you can choose the desired element (or diagram) from the appropriate location.

Activity Context Menu

The activity element offers a special context command named New with a submenu for adding the following elements:

Activity parameter	Adds activity parameter element to the activity.
Activity partition	Adds a partition to the activity

Action	Adds an action element to the activity.
Initial	Adds an initial element to the activity.
Activity Final	Adds an activity final element to the activity.
Decision	Adds decision element to the activity.
Merge	Adds merge element to the activity.
Fork	Adds fork element to the activity.
Join	Adds join element to the activity.
Flow Final	Adds flow final element to the activity.
Object Node	Adds an object node element to the activity.
Central Buffer	Adds a central buffer element to the activity.
Data Store	Adds a data store element to the activity.
Accept Event Action	Adds accept event action element to the activity.
Accept Time Event Action	Adds accept time event action element to the activity.
Send Signal Action	Adds send signal action element to the activity.

Action Context Menu

The action element offers New context command with a submenu for adding the following elements:

Input Pin	Adds Input Pin to the action.
Output Pin	Adds Output Pin to the action.
Value Pin	Adds Value Pin to the action

Accept Event Action Context Menu

The Accept Event Action element offers New context command with a submenu for adding the following elements:

Output Pin	Adds Output Pin to the Accept Event Action.

Accept Time Event Action Context Menu

The Accept Event Action element offers New context command with a submenu for adding the following elements:

Output Pin	Adds Output Pin to the Accept Time Event Action

Send Signal Action Context Menu

The Accept Event Action element offers New context command with a submenu for adding the following elements:

Input Pin	Adds Input Pin to the Send Signal Action.
Output Pin	Adds Output Pin to the Send Signal Action

Value Pin	Adds Value Pin to the Send Signal Action.

UML 2.0 Component Diagrams

This section describes the elements of UML 2.0 Component diagrams.

In This Section

UML 2.0 Component Diagram Elements	Describes UML 2.0 component diagram elements.
Component Diagram Context Commands	.

UML 2.0 Component Diagram Elements

The table below lists the elements of UML 2.0 component diagrams that are available using the Palette.

Component	node
Port	node
Artifact	node
Interface	node
Instance specification	node
Delegation connector	link
Provided interface	link
Required interface	link
Association	link
Aggregation	link
Realization	link
Note	annotation
Note link	annotation link

Component Diagram Context Commands

Diagram Editor > Right-click

All of the UML diagram types share common context menu commands. To use the context menu for a diagram, right click in the Diagram editor. To view the common context menu commands, see Common Diagram Context Commands.

The component diagram context menu shares the common context menu commands as well as commands specific to it.

Diagram Context Menu

New	The **New** command for the component diagram offers a submenu with the following options: • Component • Artifact • Interface • Instance Specification • Constraint • Note • Shortcut

Component Context Menu

New	The **New** command for the component element offers a submenu for adding the following elements: • New Attribute • Operation • Port • Component • Artifact • Interface • Instance Specification

Artifact Context Menu

The Artifact element offers a special context command named New with a submenu for adding the following elements:

New	The **New** command for the artifact element offers a submenu for adding the following elements: • Attribute • Operation • Artifact • Deployment Specification

Interface Context Menu

New	The **New** command for the interface element offers a submenu for adding the following elements: • New • Attribute • Operation • Interaction

Instance Specification Context Menu

New	The **New** command for the Instance Specification element offers a submenu for adding the following elements: • Slot

UML 2.0 Deployment Diagrams

This section describes the elements of UML 2.0 Deployment diagrams.

In This Section

Deployment Diagram Context Commands	
UML 2.0 Deployment Diagram Elements	Describes UML 2.0 deployment diagram elements.

Deployment Diagram Context Commands

Diagram Editor > Right-click

All of the UML 2.0 diagrams share common context menu commands. To use the context menu for a diagram, right click in the Diagram Editor .

To view the common context menu commands, see Common Diagram Context Commands.

The deployment diagram offers the following context commands.

Diagram Context Menu

New	The New command for the deployment diagram offers a submenu with the following options • Node • Device • Execution Environment • Artifact • Deployment Specification • Constraint • Note • Shortcut

Node Context Menu

New	The **New** command for the Node element offers a submenu with the following options: • New • Attribute • Operation • Node • Device • Execution environment

Device Context Menu

New	The **New** command for the Device element offers a submenu with the following options: • New • Attribute • Operation • Node • Device • Execution environment

Execution Environment Context Menu

New	The **New** command for the Execution Environment element offers a submenu with the following options: • New • Attribute • Operation • Node • Device • Execution environment

Artifact Context Menu

New	The **New** command for the Artifact element offers a submenu with the following options: • New • Attribute • Operation • Artifact • Deployment Specification

Deployment Specification Context Menu

New	The **New** command for the Deployment Specification element offers a submenu with the following options: • New Attribute • Operation • Artifact • Deployment Specification

UML 2.0 Deployment Diagram Elements

The table below lists the elements of UML 2.0 deployment diagrams that are available using the Palette.

Name	Type
Node	A node is a computational resource upon which artifacts may be deployed for execution. Nodes can be interconnected through communication paths to define network structures.
Device	Node that represents a physical computational resource with processing capability upon which artifacts may be deployed for execution. Devices may be complex, i.e. they may consist of other devices.
Execution Environment	Node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts.
Artifact	node An artifact represents a physical entity and is depicted in diagram as a rectangle with the <<artifact>> stereotype. An artifact may have properties which define its features, and operations which can be performed on its instances. Physically the artifacts can be model files, source files, scripts, binary executable files, a table in a database system, a development deliverable, a word-processing document, or a mail message. A deployed artifact is one that has been deployed to a node used as a deployment target. Deployed artifacts are connected with the target node by deployment links. Artifacts can include operations. You can create complex artifacts, by nesting artifact icons.
Deployment specification	node. A deployment specification specifies a set of properties which determine execution parameters of a component artifact that is deployed on a node.
Deployment	link
Generalization	link
Association	link
Dependency	Link used to model general dependencies. In Deployment diagrams, this notation is used to depict the following metamodel associations: (i) the relationship between an Artifact and the model element(s) that it implements, and (ii) the deployment of an Artifact (instance) on a Node (instance)
Manifestation	Link. A manifestation is the concrete physical of one or more model elements by an artifact.
Communication path	Link. A communication path is an association between two Nodes, through which Nodes are able to exchange signals and messages.
Note	annotation
Note Link	annotation link

UML 2.0 Composite Structure Diagrams

This section describes the elements of UML 2.0 Composite Structure Diagrams.

In This Section

UML 2.0 Composite Structure Diagram Elements	Describes UML 2.0 composite structure diagram elements.

UML 2.0 Composite Structure Diagram Elements

The following is a list of UML 2.0 composite structure diagram elements.

Name	Type
Class	node
Interface	node
Collaboration	node
Collaboration Occurrence	node
Part	node
Referenced part	node
Port	node
Provided interface	link
Required interface	link
Connector	link
Collaboration role	link
Role binding	link
Note	annotation
Note Link	annotation link

Dialogs

The part contains reference information about various dialogs:

Print Diagram

Print Diagram

File > Print

This dialog box enables you to print selected diagrams to the specified printer. The dialog box is invoked by choosing File > Print from the main menu with a diagram open in the **Diagram View**.

Print diagrams	From this list box, choose the diagrams to be printed. The possible options are: Active diagram Active with neighbors (all diagrams within the same namespace) All opened diagrams All diagrams in the model
Print zoom	Enter a zoom factor for the printout. By default, the zoom factor is set to 1.
Fit to page	Check this option if you want to print the diagram on a single page. If checked, the Print zoom field is disabled.
Preview	Click the down arrow to show the print preview page.
Preview zoom	Use the slider to set up the preview zoom. The current value of the zoom factor is displayed to the left of the slider.
Auto preview zoom	Check this option to fit the image to the preview window.
Print	Press this button to send the selected diagrams to the default printer. Use the down arrow to choose the Print dialog box command, which enables you to configure the printer options.

Glossary

This topic contains a dictionary of specific terms used in the user interface and documentation. This dictionary is sorted alphabetically.

B

Behavior: A predefined response to a threshold violation.

C

Cardinality: The number of elements in a set. See also **multiplicity**.

Classifier: In general, a classifier is a classification of instances. It describes a set of instances that have features in common. A classifier is a group of the following model elements: class, interface, association class, structure, delegate, enumeration, module, interaction. In UML 2.0 projects this group includes the data type element as well. Some of that elements can include members or other classifiers. A classifier inserted into another classifier is called an inner classifier.

Compartment: Some of the model elements (basically, classes) are represented by rectangles with several **compartments** inside. You can change appearance of the compartments.

Container: A container is a **classifier** that can include one or more model elements, or **members**.

D

Design Project: One of the two basic project types supported by **design** and **implementation**. A design project is language-neutral. It does not contain source code.

Diagram: A graphical presentation of a collection of shortcuts to **model elements** from one or more **packages** or **namespaces**. Most often a diagram is rendered as a connected graph of arcs (relationship links) and vertices (nodes).

The set of available diagram for a project depend on the project type.

I

Inner classifier: An inner classifier is a **classifier** inserted into another classifier

Invocation specification : An area on a UML 2.0 Sequence Diagram. Most often an **invocation specification** is located within an **execution specification**. This element is not defined in the UML 2.0 specification, but introduced in ER/Studio Software Architect. It is a useful tool for modeling synchronous invocations with the reply messages. A message in UML 2.0 Sequence Diagrams has its origin in an invocation specification.

M

Member: A member is a **model element** inserted into a classifier, or a **container**. If a member is a classifier, it is called **inner classifier**.

Model element : Model element is any component of your model that you can put into a package or a namespace. Model elements include nodes and links between them.

Multiplicity : A specification of the range of allowable cardinalities that a set may assume, for example:0..*. Multiplicity specifications can be given for association ends, parts within composites, and other purposes. A multiplicity is a subset of the non-negative integers. See also cardinality.

N

N-ary association: An association among three or more nodes. an **association class** implements this functionality.

P

Package: An element for storing diagrams, model elements, and other packages. Every project consists of one or more packages or namespaces. You cannot delete the default package (namespace).

Pattern instance: An oval model element that represents a pattern with a special pre-defined behavior.

S

Shortcut: A presentation of a model element, a diagram, a namespace, a package, or some external artifact, placed on a diagram.

V

View filter : A mechanism to show or hide a specific kind of model elements. When dealing with large projects, the amount of information shown on a diagram can become overwhelming. You can selectively show or hide information.