SYBASE®

Working with the Repository

PowerDesigner® 15.3

Windows

DOCUMENT ID: DC38090-01-1530-01
LAST REVISED: November 2010

Sybase, Inc., One Sybase Drive, Dublin, CA 94568

Contents

CHAPTER 1 Getting Started with the Repository

The PowerDesigner® repository is a tool for storing and sharing versioned models and other files used in your modeling process. It provides robust, granular security, and supports full access from within PowerDesigner and viewing and some administration via a Web browser.

Note: Before you can access the repository, an administrator must install and configure it and provide you with a user name and password. For information about installing the repository, see the *Installation Guide*.

The majority of repository users:

- Browse and search the repository (see *Browsing the Repository* on page 30 and *Chapter 3, Browsing the Repository Via the Web* on page 61)
- Add and update documents in the repository (see *Checking Documents into the Repository* on page 11)
- Check documents out of the repository to view or edit them (see *Checking Documents out of the Repository* on page 21), and may also:
  - Create versions to retain a history of the development process (see *Freezing Document Versions* on page 40)
  - Lock document versions to temporarily protect them from modification by other users (see *Locking Document Versions* on page 42)

More advanced users will have additional rights to:

- Create groups of document versions to mark particular milestones in the development process (see *Grouping Document Versions in a Configuration* on page 44)
- Creating branches in the repository version tree (see *Branching Version Trees* on page 48)

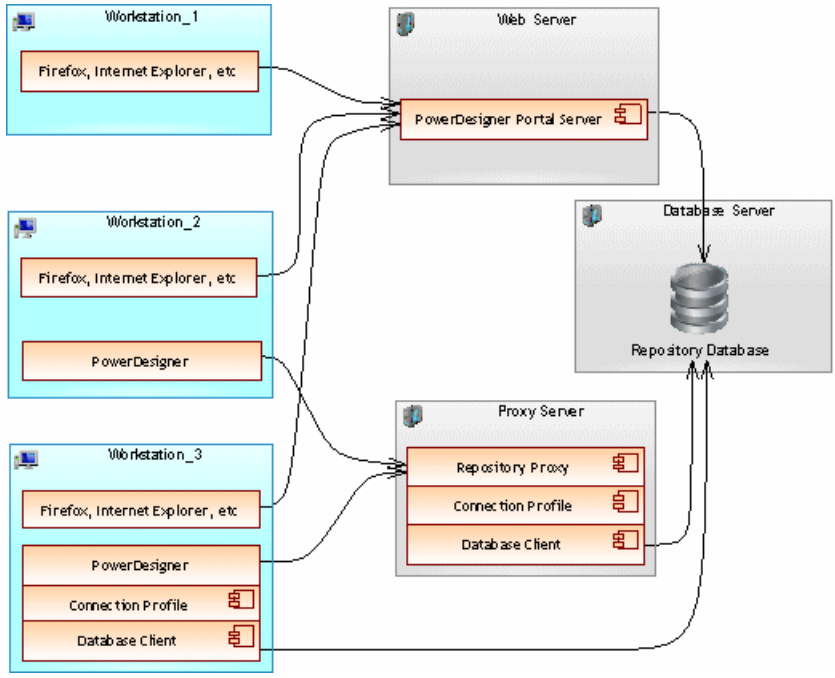
Administrators take responsibility for:

- Installing and controlling access to the repository (see *Creating, Upgrading, and Dropping the Repository Database* on page 83 and *Controlling Repository Access with LDAP* on page 85)
- Adding resource files to the repository and managing their sharing (see *Sharing Resources in the Repository* on page 54)
- Managing rights and permissions in the repository (see *Controlling Rights for Repository Users and Groups* on page 90 and *Controlling Permissions for Repository Items* on page 96

Setting up PowerDesigner to Access the Repository

You can access the repository directly, via the repository proxy, or through a web browser (via the PowerDesigner Portal). Your administrator will tell you which methods are available to you.

In the following diagram, the workstations connect to the Repository database in various ways:



These connection types are further explained in the following sections:

- Web connection [if the PowerDesigner Portal is installed] – see *Chapter 3, Browsing the Repository Via the Web* on page 61
- Direct connection – see *Setting up PowerDesigner for Direct Access* on page 3
- Proxy connection [if the repository proxy is installed] - see *Setting up PowerDesigner for Access via the Proxy* on page 5

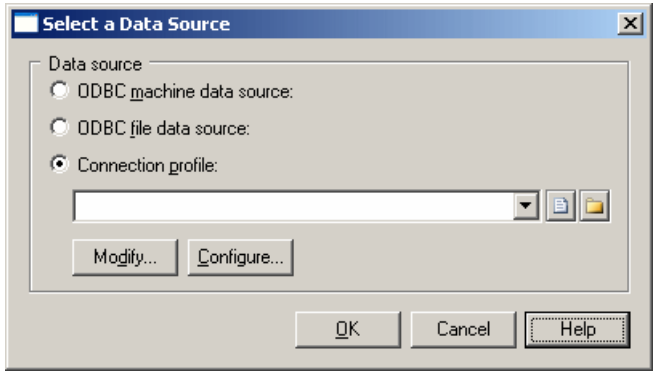
Note: Before you can access the repository, an administrator must install and configure it and provide you with a user name and password. For information about installing the repository, see the *Installation Guide*.

Setting up PowerDesigner for Direct Access

To create a direct repository definition, you must have a login name and password for the repository AND a user name and password for the database where it is stored. You may also need to install a database client or driver in order to connect to the repository database. Your repository administrator will supply you with this information.

If your administrator has installed the repository proxy, see *Setting Up PowerDesigner for Access via the Proxy* on page 5.

1. Select **Repository** > **Repository Definitions** to open the List of Repository Definitions.
2. Click the **Add a Row** tool and type a name in the **Repository Name** column to identify the definition for future use.
3. Type your user name in the **Repository User** column. If you are an administrator and are creating the first repository definition that will be used to initialize the repository (see *Chapter 4, Repository Administration* on page 83), you must use the `ADMIN` repository user name.
4. Press F4 or click the Ellipsis button in the **Data Source Name** column to open the Select a Data Source dialog:

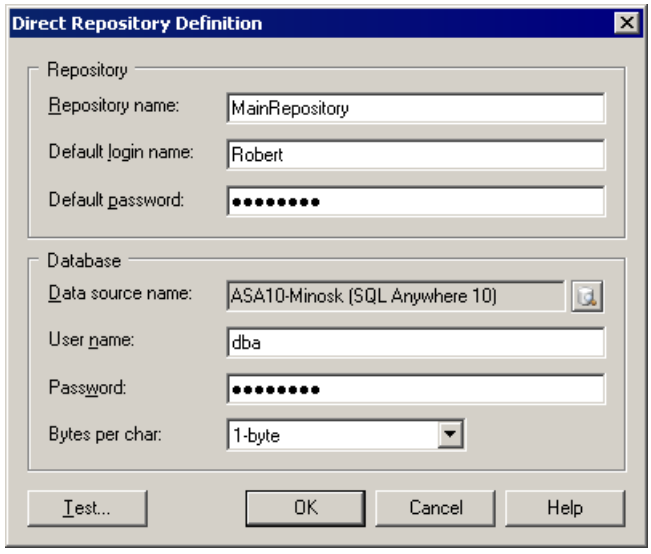


5. Select one of the following options:
   - ODBC machine data source
   - ODBC file data source
   - Connection profile (for native, JDBC, ADO.NET, OLE DB or DirectConnect connections)

   Use the tools to the right of the data source field to browse to a new connection profile file or directory, and the **Modify** and **Configure** buttons to change your existing data source connection.

   For detailed information about creating, configuring, and using connection profiles, see "Connecting to a Database" in *Chapter 1, Getting Started with PowerDesigner* of the *Core Features Guide* .

6. Select a data source and click **OK** to return to the List of Repository Definitions, where it will be displayed in the **Data Source Name** column.
7. Click **Apply** to apply your changes, and then click the **Properties** tool to open the Direct Repository Definition dialog:



This window has two group boxes:

- **Repository** – specifies the name of the repository definition, and the login name and password that you will use to connect to it.
- **Database** – specifies the name of the data source that hosts the repository database, and the user name and password that you will use to connect to it. These are case-sensitive and optional, but if you do not specify them here you will need to enter them each time you connect to the repository. The password box always display stars, whether or not a password has been entered.

  For non-Oracle unicode or multi-byte character set databases, you should also specify the bytes per character used by the database:
  - 1-byte - [default] For SBCS (Single-Byte Character Set)
  - 2-byte - For DBCS (Double-Byte Character Set)
  - 3-byte - For Unicode or MBCS (Multi-Byte Character Set)

8. Test the definition by clicking the **Test** button. A message box will state whether your connection succeeds or fails.
9. Click **OK** to return to the List of Repository Definitions.

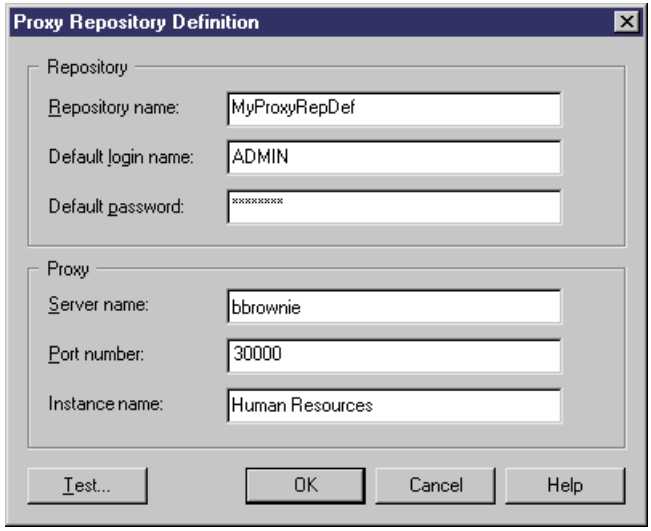
Note: For more information about creating a data source for Adaptive Server® Anywhere, see the *Installation Guide*.

Setting up PowerDesigner for Access Via the Proxy

To create a proxy repository definition, you must have a login name and password for the repository and know the server on which the proxy is installed. Your repository administrator will supply you with this information.

If you are connecting directly, see *Setting up PowerDesigner for direct access* on page 3.

1. Select **Repository > Repository Definitions** to open the List of Repository Definitions.
2. Click the **Add a Row** tool and type a name in the **Repository Name** column to identify the definition for future use.
3. Type your user name in the **Repository User** column, and select **Proxy** in the **Connection Type** column (you may need to widen the window to see this column).
4. Click **Apply** to apply your changes, and then click the **Properties** tool to open the Proxy Repository Definition window:



This window has two group boxes:

- **Repository** – specifies the name of the repository definition, and the login name and password that you will use to connect to it.
- **Proxy** – specifies the name of the server where the repository proxy is installed, the port number, and the name of the repository instance that you will connect to. These are case-sensitive and optional, but if you do not specify them here you will need to enter them each time you connect to the repository. The password box always display stars, whether or not a password has been entered.

5. Test the definition by clicking the **Test** button. A message box will state whether your connection succeeds or fails.

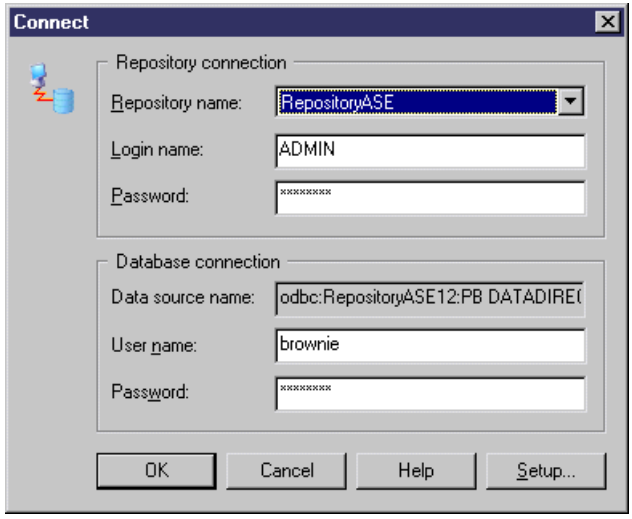
6. Click **OK** to return to the List of Repository Definitions.

Connecting to a Repository

Before you can connect to a repository, it must be created on a server (by your administrator), and you must have created a repository definition.

For information about creating a repository definition, see *Setting up PowerDesigner to Access the Repository* on page 2.

1. Select **Repository > Connect** to open the Connect dialog.
2. Select a repository name to display its connection parameters.



3. Click **OK** to connect.

   The contents of the repository are displayed in the **Repository** tab of the Browser. The root node displays the name of the repository, the user login and the current branch (see *Branching Version Trees* on page 48.

Connecting Automatically to the Default Repository

You can specify that you want to connect automatically to a repository each time you open PowerDesigner. On opening PowerDesigner will try to connect to the repository to which you last connected successfully.

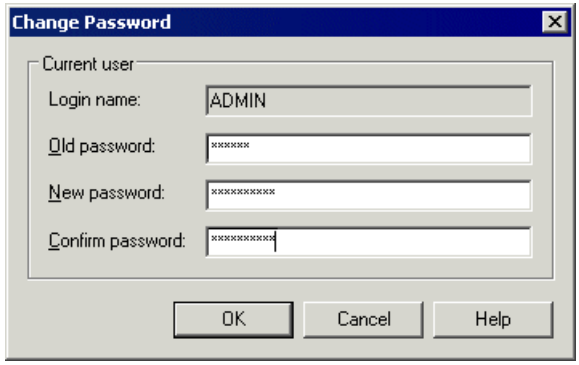
1. Select **Tools > General Options** to open the General Options dialog and click the Repository category.
2. Select the **Auto-connect** check box in the Connection group box, and then click **OK**.

For information about the other options on this tab, see *Repository General Options* on page 8.

Changing Your Repository Password

You can change the password that you use to connect to the repository at any time.

1. Select **Repository** > **Change Password** to open the Change Password dialog box.
2. Enter your old and new password and then confirm the new password.



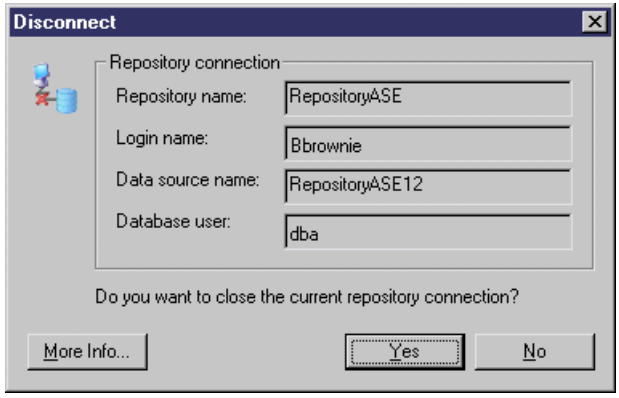
3. Click **OK** to complete the change.

Note: An administrator can change a password on behalf of a user. Select **Repository** > **Administration** > **Users** and double-click a user in the list to open its property sheet and modify its password.

Closing a Repository Connection

You can close a repository connection if you need to change repository or to work in a standalone environment.

1. Select **Repository** > **Disconnect** to open the Disconnect dialog.



2. Click **Yes** to close the connection.

Repository General Options

To set repository options, select **Tools > General Options** to open the General Options window, and then select the Repository category.

Connection

The following options are available in the Connection group box:

Option	Description
Auto-connect	Automatically opens the last connected repository when you start a PowerDesigner session.
Default bytes per char	Specifies the default number of bytes per character used by the repository database. This value is used to initialize the Bytes per char parameter in Direct Repository Definition dialog (see *Setting up PowerDesigner for Direct Access* on page 3).

Browser

The following options are available in the Browser group box:

Option	Description
Auto-refresh	Refreshes the Repository Browser at the interval specified in the Refresh every box.
Display	Specifies whether the name or code of items is displayed in the Repository Browser.
Display model contents	Enables the display of model diagrams and objects in the Repository Browser.
Display version numbers	Enables the display of version numbers for items in the Repository Browser.
Display status icons in Local Browser	Enables the display of status icons (which track whether items are locked and if changes have been made locally and/or in the repository since check out) in the Local Browser (see *Checking the State of a Document* on page 28).

Check Out

The following options are available in the Check Out group box:

Option	Description
Merge document	Specifies that the Merge document check box in the Check Out window is selected by default.
Add to workspace	Specifies that the Add to workspace check box in the Check Out window is selected by default.
Check out dependencies	Specifies that the Check out dependencies check box in the Check Out dialog box is selected by default.
Automatic resources update	Enables the automatic checking out and updating of shared resource files (see *Sharing Resources in the Repository* on page 54).

Check in

The following options are available in the Check In group box:

Option	Description
Batch update	Specifies that the Batch update check box in the check in options window is selected by default.
Freeze after check in	Specifies that the Freeze after check in check box in the Check In window is selected by default.
Save diagrams for PowerDesigner Portal	Specifies that the Save diagrams for PowerDesigner Portal check box in the Check In window is selected by default.

CHAPTER 2 Working with Repository Documents

Depending on the rights and permissions provided to them, repository users can check in and out project, model, and file documents and may be able to create configurations, branches, and perform other operations on the contents of the repository.

Note: You must have at least the `Read` permission to check a document out, and at least the Write permission to check in a document (see *Controlling Permissions for Repository Items* on page 96).

Checking Documents into the Repository

You check documents into the repository to share them with others. You must have at least the `Write` permission on a document (or on one of its packages) to be able to check it in.

1. Select the documents to check in on the **Local** tab of the Browser, then right-click and select **Check In** to open the:
   - Check In Project window - if you selected a project (see *Checking In a Project* on page 11).
   - Check In Document window - if you selected a single model or file (see *Checking In a Document* on page 12).
   - Check In Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see *Checking In Multiple Documents* on page 14).
2. Select the appropriate parameters (see *Check In Parameters* on page 16), and click **OK** to check in the document.
3. If you are checking in a document that is already stored in the repository, the Merge Changes to Repository Document dialog (see *Resolving Conflicts During Check In* on page 18) may open to let you review the differences between your local document and the version in the repository. When you are satisfied, click **OK** to complete the check in.

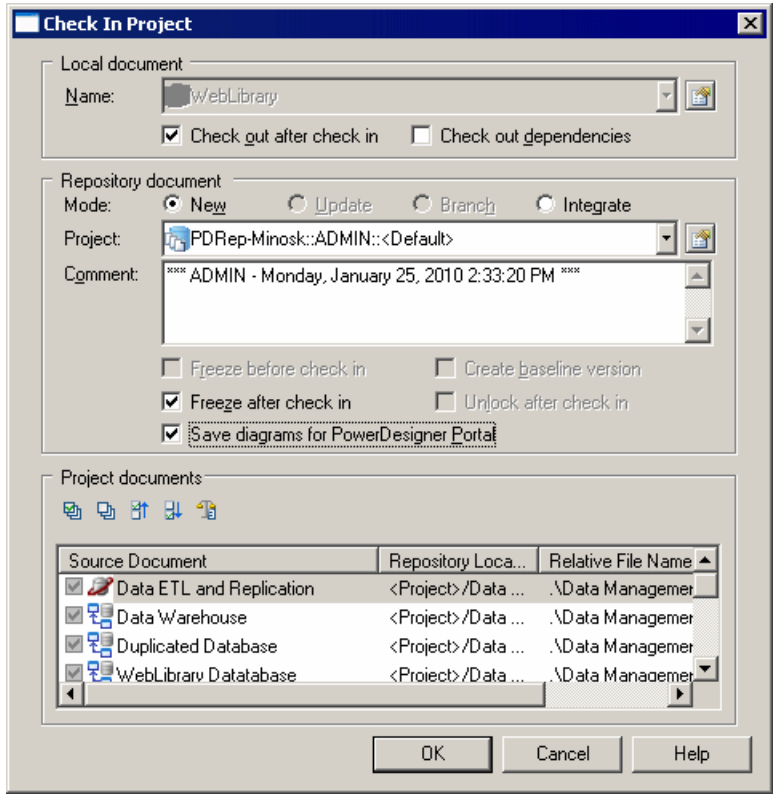
Note: If you change the target (DBMS, OO language, etc) outside of the target family (such as moving from AS IQ 12.6 to 12.7), you will no longer be able to update the existing repository document and will need to save the model in a new location in the repository.

Checking in a Project

The Check In Project dialog guides you through checking in a PowerDesigner modeling project along with all the models and other documents it contains.

You arrive here by right-clicking a Project in the **Local** tab of the Browser, and selecting **Check In**. The dialog is populated with any project documents that have changes. The first

time you check in a project, you must check in all its documents; subsequently, only documents with changes are listed and can be deselected if desired.



For information about the parameters available in this dialog, see *Check In Parameters* on page 16.

Checking in a Document

The Check In Document dialog guides you through checking in a PowerDesigner model or other document.

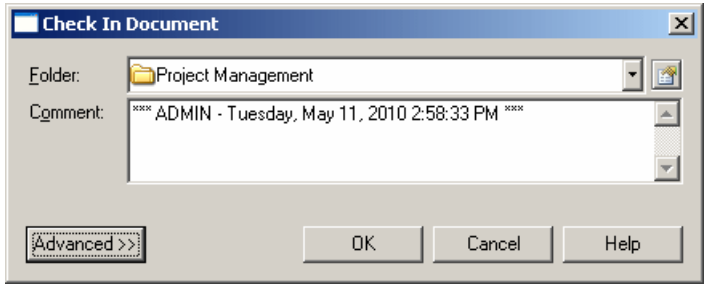
You can arrive here by:

- Right-clicking a model or file in the **Local** tab of the Browser, and selecting **Check In**.
- Selecting a model or file in the Check In Multiple Documents dialog (see *Checking In Multiple Documents* on page 14), and clicking the **Settings** tool.
- Clicking the **Add Document** tool in the Check In Multiple Documents window (see *Checking In Multiple Documents* on page 14). You must specify the model or file to check in in the **Local Document** group box.

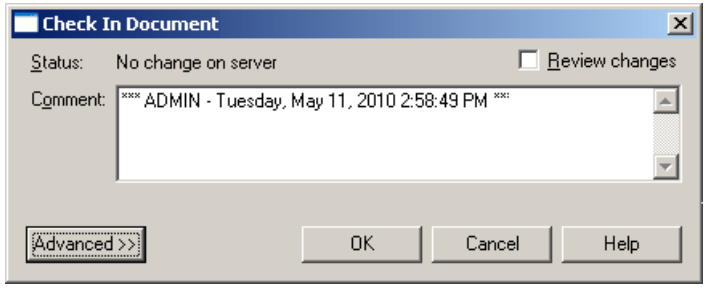
- Selecting a resource file, and clicking the **Check In** tool in the a resource file list (see *Checking Resources to Share into the Repository* on page 56).

Depending on the context, this window can take various forms:

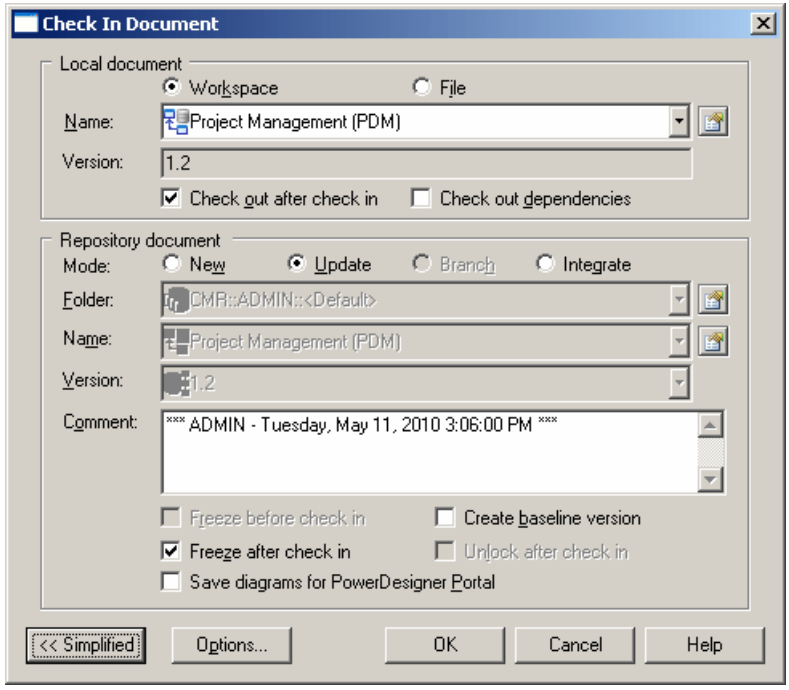
- Simplified - Checking in a document for the first time (select a folder to check into, enter a comment, and click **OK**):



- Simplified - Checking in updates to a document already in the repository (choose whether to review your changes (see *Resolving Conflicts During Check In* on page 18), enter a comment, and click **OK**):



- Advanced - When you click the **Advanced** button on either of the simplified dialogs:



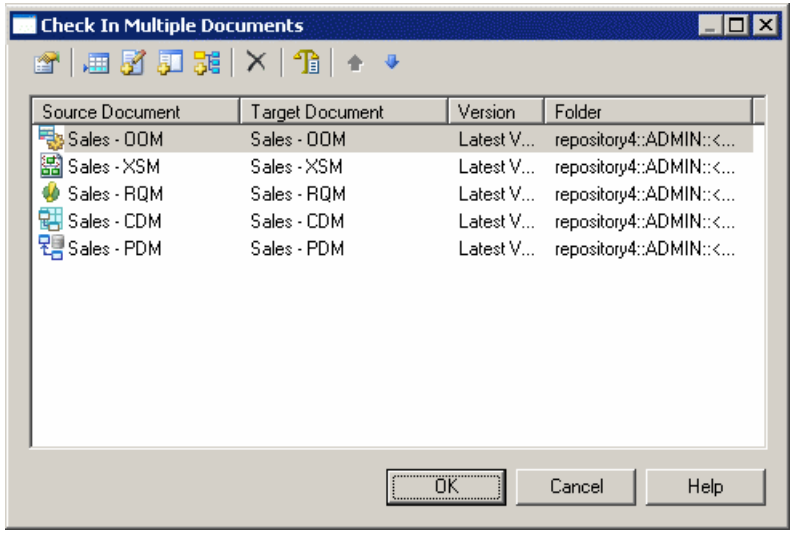
For information about the parameters available in this dialog, see *Check In Parameters* on page 16.

Checking in Multiple Documents

The Check In Multiple Documents dialog is used when you select more than one document for check in.

You can arrive here by:

- Right-clicking the workspace root or a folder containing multiple documents (or manually selecting multiple documents) in the **Local** tab of the Browser, and selecting **Check In**.
- Selecting **Repository** > **Check In** (or pressing **Ctrl+Alt+C**).



The following tools are available to manage the documents in the list:

Tool	Description
	Settings – Opens the Check In Document window to let you specify check in parameters for the selected document (see *Checking in a Document* on page 12).
	Add Document - Opens the Check In Document window to let you to specify a document to add to the list (see *Checking in a Document* on page 12).
	Add Modified Documents in Workspace - Adds all the documents modified in the workspace during the current session to the list.
	Add All Documents in Workspace - Adds all the documents in the workspace (including closed documents) to the list.
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation or other links to the list. When you check in PowerDesigner models related by shortcuts, the models being referenced are checked in before the models containing the shortcuts.
	Delete – Removes the selected document from the list.
	Check In Options – Opens the Check In Options window to allow you to specify the update mode for changes (see *Resolving Conflicts During Check In* on page 18).
	Move Up – Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

Check in Parameters

The parameters on the Check In Document and Check In Project windows let you specify details of the documents being checked in.

Local Document

The Local Document group box contains the following parameters:

Parameter	Description
Workspace or File	[not for project] Select a radio button to specify and view the document name by its workspace name or file name and path.
Name	Specifies the name of the document to check in. Depending on whether you have selected Workspace or File, this will be a: • Model name - click the **Properties** tool to open its property sheet • File name and path - click the **Browse** tool to open a file selection dialog.
Version	[not for project – read-only] Specifies the current version of the model or file in the repository, if available
Check out after check in	[models and resources only] Updates the local version after check in by checking out the repository version to ensure that you have the most up-to-date version where other users may be working concurrently.
Check out dependencies	[models only] Checks out any models on which the model (contained in a project or not) is dependent (through generation links or shortcuts) after check in.

Repository Document

The Repository Document group box contains the following parameters:

Parameter	Description

Mode	Specifies the type of check in. You can choose between:
	• **New -** [always available] The document is not yet in the repository, or not yet in the current branch (see *Branching Version Trees* on page 48), or you want to check it into a new location in the current branch.
	• **Update**/**Branch** – These modes are mutually exclusive and one is always selected by default depending on whether the document already exists in the current branch or one of its base branches. The Project, Folder, Name, and Version fields are read-only:
	  • **Update** - Updates the document in the current branch or creates a new version in the current branch if the document was checked out from a base branch of the current branch.
	  • **Branch** - Creates the first version of the document in the current branch.
	• **Integrate** – [always available] You want to merge the document into another version in a parallel branch or into a different document.
	Note: Each check in is processed as one transaction. If you check in large models using the New mode, the server will create large rollback logs, and use a large number of locks; you might want to configure your server to allow such large transactions. The other modes use much smaller transactions.
	For example of the use of check in modes, see *Checking in a Document Using the Update, Branch, and Integrate Modes* on page 51)

Folder/Project	[read-only except in New mode] Specifies the location where the document will be checked in. Click the **Properties** tool to open the property sheet of the folder. If you do not specify a target folder, the document is checked in at the root of the browser. You should grant individual access permissions to this document.
Name	[not for project – read-only except in Integrate mode] Specifies the name of the document to apply your changes to. Click the **Properties** tool to open the property sheet of the document.
Version	[not for project – read-only except in Integrate mode] Specifies the version of the document to apply your changes to.
Comment	Specifies a comment to describe the purpose of the check in.
Freeze before check in	Freezes the previous versions of any modified objects in the repository, and creates a new version to accept the changes to be checked in (see *Freezing Document Versions* on page 40).

Parameter	Description
Freeze after check in	Freezes all modified object versions after check in to ensure that no further changes can be made to them. Future changes to objects will be made to new versions (see *Freezing Document Versions* on page 40).
Create baseline version	Creates a completely new version of the model, instead of simply versioning the objects that have changed. This option is only available if you have `Full` permission on the document. The Merge Changes to Repository Document dialog will not open. Baseline versions may be quicker to check in and out, but they require considerably more disk space than standard versions. The server will create large rollback logs, and use a large number of locks, and might need to be configured to allow large transactions. Occasionally the repository is unable to process a standard version, and in this situation you should try to create a baseline version. This can be used as starting point for a new development step.
Unlock after check in	[locked documents only] Unlocks the document version after check in (see *Locking Document Versions* on page 42).
Save diagrams for PowerDesigner Portal	[not for resources] Prepares snapshots of the model diagrams that can be viewed in the PowerDesigner Portal (see *Chapter 3, Browsing the Repository Via the Web* on page 61).

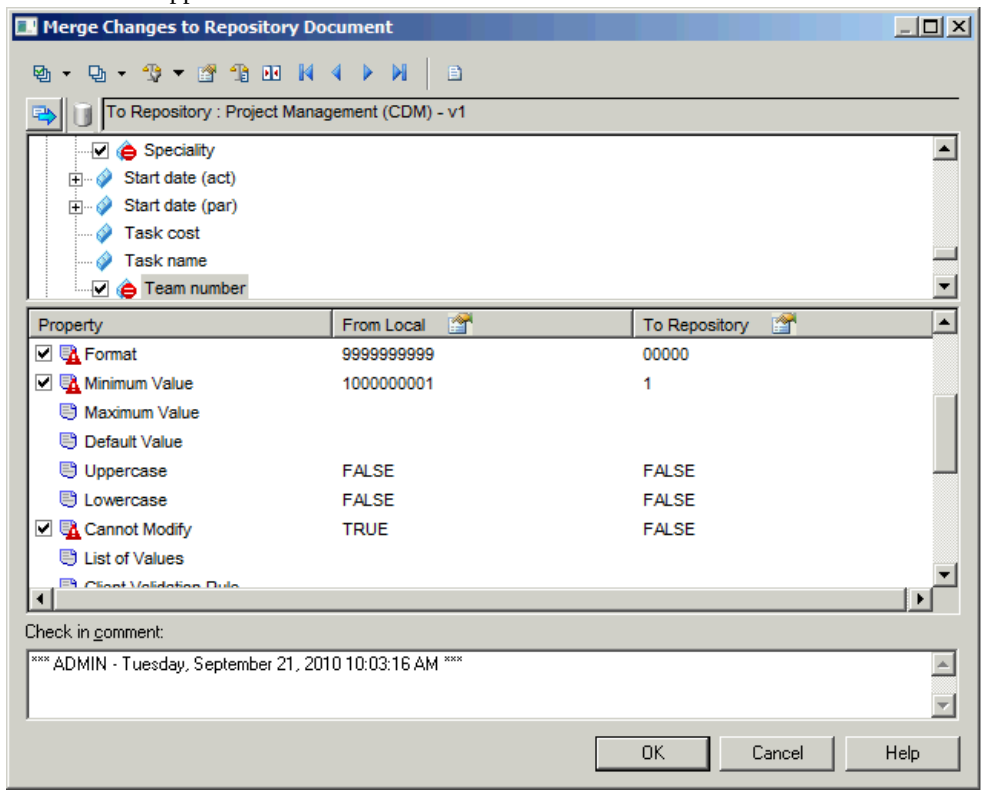
Project Documents

The Project Documents group box lists the model and file documents associated with the project, and lets you select them for check in with the project. Note that only project documents that are inside the project are listed here. Documents that are outside the project and only attached to it must be checked in separately.

Resolving Conflicts During Check In

The Merge Changes to Repository Document dialog lets you inspect and approve (or reject) the changes you are checking into the repository. To filter the tree and show only the changes,

click the **Change Filter** tool and select **Show All Changes**. By default, all your changes are selected to be applied.



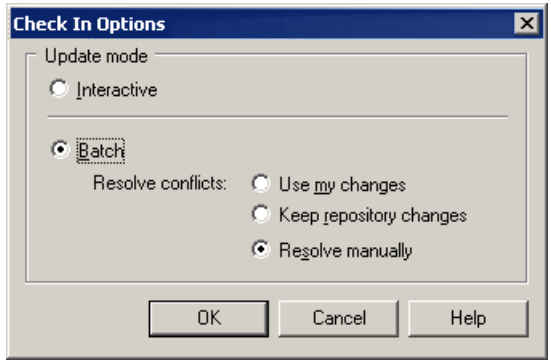
The window is divided in three parts:

- The *Object comparison pane* in the upper part of the window displays the anticipated result of the merge on the target model. Click the **Show Source Model** button at the top left of the pane to split the pane between the source and target models.
- The *Property comparison pane* in the middle of the window displays the properties of the nodes selected in the model tree, listing the name of the property and its values in the From and To models.
- The *Check in comment* pane lets you edit the comment that will be associated with this check in.

For detailed information about using this dialog, see *Chapter 7, Comparing and Merging Models* in the *Core Features Guide* .

You can instruct PowerDesigner to handle all or some of these conflicts (and thus suppress the display of the Merge dialog) by deselecting the **Review changes** option in the simplified Check In Document dialog or from the Check In Options dialog, which you open by:

- Clicking the **Options** button at the bottom of the advanced Check In Document dialog.
- Clicking the **Check In Options** tool in the Check In Project or Check In Multiple Documents dialog. The options specified will apply to all the models and files being checked in.



The following modes are available:

- Interactive - [default] Always opens the Merge Changes to Repository Document dialog to let you review and approve or reject each change before check in.
- Batch - Automates the merging of changes and treats conflicts (where the same object has been modified both by you and by another user) using one of the following rules:
  - Use my changes – applies your changes and overwrites changes in the repository version (even if they are to different attributes).
  - Keep repository changes – retains the changes in the repository and rejects your changes (even if they are to different attributes).
  - Resolve manually – opens the Merge Changes to Repository Document dialog to let you review and resolve conflicts.

Checking in Packages

If you check a model into the repository, you also check in the packages it contains.

While, generally, you need the Write or Full permission to check a document into the repository, you can check in a model on which you only have Read permission if it contains at least one package for which you have Write or Full permission. However, note that any edits made to packages for which you do not have Write or Full permission will be lost during check in.

For information about checking out package, see *Checking out packages* on page 27.

Checking in External Application Documents

The link between the local document and the repository document is saved in the current workspace. This link allows you to update a document, model or external application file, in the repository.

To be able to update an external application document that does not belong to the current workspace, you have to check the document out from the repository, select the Add to Workspace check box and then save the workspace. This allows you to update the document the next time you check it in.

Undoing a Check in

If you want to undo a check in, you can only delete the document version created by the check in.

Since a check in to the repository does not necessarily create a new document version (see *Document Versions* on page 40), such a deletion may not always be possible.

Checking Documents out of the Repository

You check out documents from the Repository in order to work on them in your local workspace. You must have at least the `Read` permission on a document (or on one of its packages) to be able to check it out.

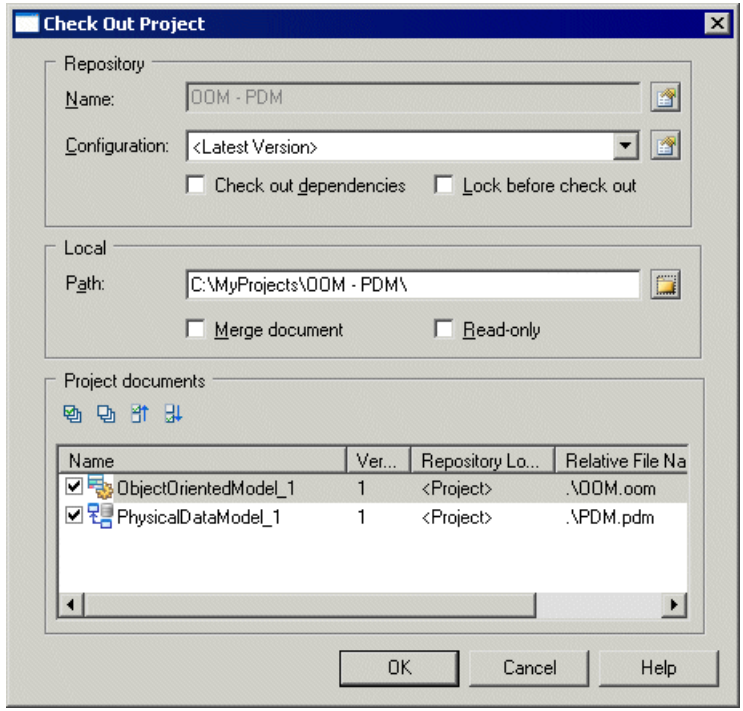
1. To check out a document not currently in your workspace, select the **Repository** tab of the Browser. If you are updating an existing file, select the **Local** tab.
2. Select the documents to check out, then right-click and select **Check Out** to open the:
   - Check Out Project window - if you selected a project (see *Using the Check Out Project Window* on page 22).
   - Check Out Document window - if you selected a single model or file (see *Using the Check Out Document Window* on page 22).
   - Check Out Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see *Using the Check Out Multiple Documents Window* on page 23).
3. Select the appropriate parameters (see *Check Out Parameters* on page 24), and click **OK** to check out the document.
4. If you are checking out a document that is already present in your workspace (and you have selected the **Merge document** option), the Check Out Document window (see *Resolving Conflicts During Check Out* on page 26) will open to allow you to review the differences between the version in the repository and your local version. When you are satisfied, click **OK** to complete the check out.

Checking out a Project

The Check Out Project dialog guides you through checking out a PowerDesigner modeling project along with all the models and other documents it contains.

You can arrive here by:

- Right-clicking a project in the **Repository** tab of the Browser, and selecting **Check Out**.
- Right-clicking a project in the **Local** tab of the Browser, and selecting **Update from Repository**.



For information about the parameters available in this dialog, see *Check Out Parameters* on page 24.

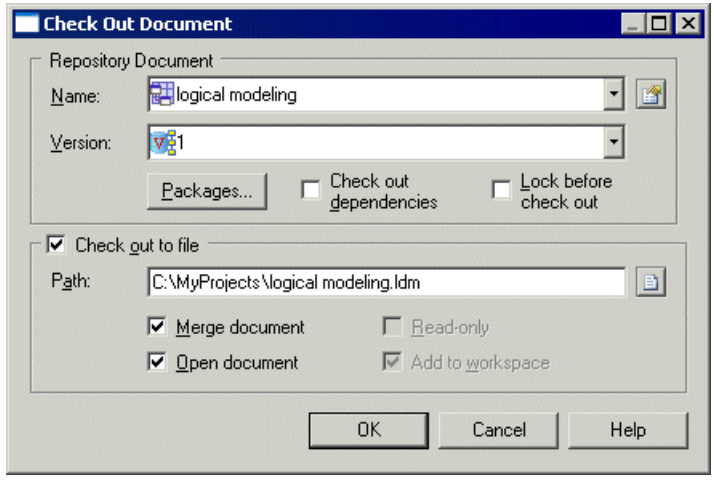
Checking Out a Document

The Check Out Document dialog guides you through checking out a PowerDesigner model or other document.

You can arrive at this window in any of the following ways:

- Right-clicking a model or file in the **Repository** tab of the Browser and selecting **Check Out**.

- Right-clicking a model or file in the **Local** tab of the Browser and selecting **Update from Repository**.
- Selecting a model or file in the Check Out Multiple Documents window (see *Checking out Multiple Documents* on page 23) and click the **Settings** tool.
- Clicking the **Add Document** tool in the Check Out Multiple Documents window (see *Checking out Multiple Documents* on page 23). You must specify the model or file to check out in the **Local Document** group box.
- Select a resource file, and click the **Update from Repository** tool in a resource file list.



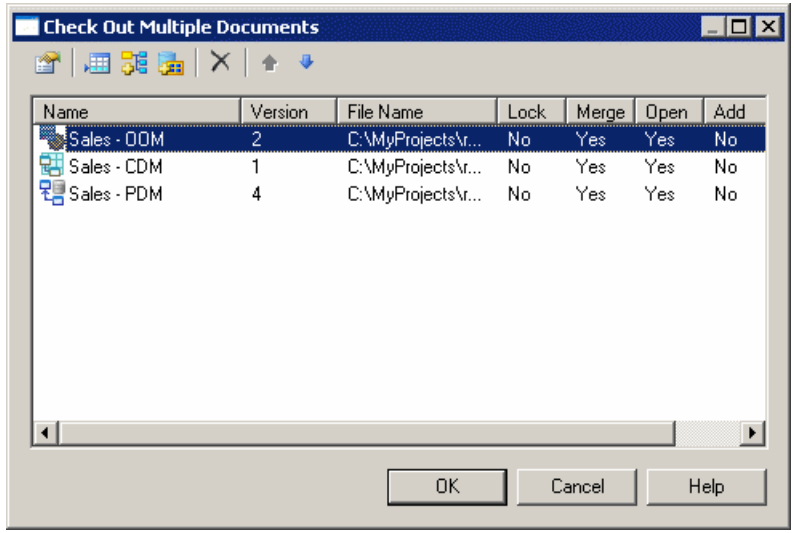
For information about the parameters available in this dialog, see *Check Out Parameters* on page 24.

Checking out Multiple Documents

The Check Out Multiple Documents dialog is used when you select more than one document for check out.

You can arrive here by:

- Right-clicking the workspace root or a folder containing multiple documents (or manually selecting multiple documents in the **Repository** tab of the Browser, and selecting **Check Out**.
- Right-clicking the workspace root or a folder containing multiple documents (or manually selecting multiple documents in the **Local** tab of the Browser, and selecting **Update from Repository**.
- Selecting **Repository** > **Check Out**(or pressing **Ctrl+Alt+E**).



The following tools are available in this window:

Tool	Description
	Settings – Opens the Check Out Document window to let you specify check out parameters for the selected document (see *Checking out a Document* on page 22).
	Add Document Version - Opens the Check Out Document window to let you to specify a document to add to the list (see *Checking out a Document* on page 22).
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation links to the check out list.
	Add Document Versions from a Configuration – Opens a selection window allowing you to add all the document versions included in a configuration to the list (see *Grouping Document Versions in a Configuration* on page 44).
	Delete - Removes the selected document from the list.
	Move Up - Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

Check Out Parameters

The following check out parameters are available from the Check Out Document and Check Out Project windows to let you specify details of the check out:

Repository (Document)

This group box contains the following parameters:

Parameter	Description
Name	Specifies the name of the document to be checked out from the repository.
Version	[not for projects] Specifies the version of the document to check out.
Configuration	[projects only] Specifies the configuration containing the project models and files to check out or (if no configuration is specified) the latest project models and files versions held in the repository. You create a project configuration by right-clicking a project node in the Repository tab of the Browser. See *Creating a configuration* on page 44.
Packages	[models only] Click the button to open a package selection dialog box in which you can select packages rather than checking out the entire model.
Check out dependencies	[projects and models only] Specifies that external shortcut and generation link dependencies information is checked out. If you do not select this option then information on the Dependencies tabs of objects in the model may not be complete. For more information on the check out of dependencies, see *Impact Analysis* on page 37.
Lock before check out	Locks the selected version of the document before checking out, in order to prevent other users from modifying it. This option is only available to users with Lock Versions rights.

Check Out To File/Local

This group box contains the following parameters:

Parameter	Description
Check out to file	[not for projects] Specifies that the document will be checked out to a file on your local machine. This option can only be deselected for PowerDesigner models, which can be loaded in memory and added to your workspace without creating a file.
Path	Specifies the path of the file or (projects only) folder to which the document will be checked out. Click the Select File tool to the right of this field to browse for a location.

Parameter	Description
Merge document	Opens the Check Out Model window (see *Using the Check Out Model window* on page 26) before checking out, to allow you to compare the local and repository versions of a document, and to review and approve or reject differences for checking out. If you check-out multiple documents, they automatically replace the existing document versions, and are not merged with them. This option is only available if you have specified a path to an existing file or folder on your machine.
Read-only	Sets the read-only flag on the file or folder after checking out.
Open document	[not for projects] Opens the document after checking out, either in the current PowerDesigner workspace or in an external application.
Add to workspace	[models and files only] Adds the checked out document to the current workspace.

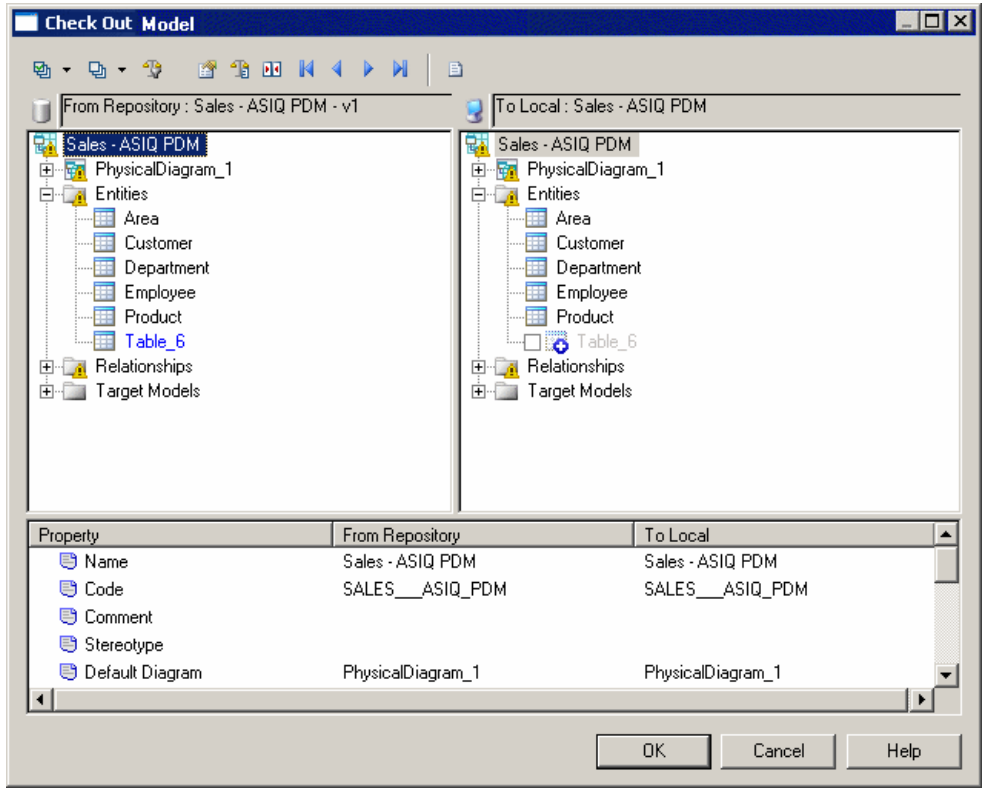
Project Documents

The Project Documents group box lists the model and file documents associated with the project, and lets you select them for check out with the project. Note that only project documents that are inside the projects are listed here. Documents that are outside the project and only attached to it are not listed and must be checked out separately.

Resolving Conflicts During Check Out

The Merge Changes to Local Document dialog lets you inspect and approve (or reject) all the changes you are checking out of the repository.

This dialog will open if you are checking out a PowerDesigner model to an existing file on your computer and you have selected the **Merge model** option.



The repository version of the model that you are checking out is displayed in the left pane, and the local version is in the right pane. To filter the tree and show only the changes, click the **Change Filter** tool and select **Show All Changes**. By default, all the changes from the repository are selected to be applied. For detailed information about using this dialog, see *Chapter 7, Comparing and Merging Models* in the *Core Features Guide* .

Checking Out Packages

By default, when you check out a PowerDesigner model, all its packages on which you have Read, Write, or Full permissions are checked out with the model. Packages with List or no permissions cannot be checked out.

If you do not want to check out a complete model, you can select one or more packages to check out from the Select Packages dialog box, which displays the packages on which you have Read, Write, or Full permissions.

If you select a parent package, PowerDesigner automatically extends the selection to sub-packages. If you deselect some sub-packages in the package tree, the selection of the parent packages is grayed to indicate this. If you select a sub-package, PowerDesigner automatically extends the selection to parent packages to preserve the context of the sub-package.

Action	Selection	Impact
Select modelingproject		Package 1, 2, and 3 are automatically selected
Deselect Package 1		Package 2, and 3 are automatically deselected
Select Package 3		Package modelingproject, 1, and 2 are automatically selected
Deselect Package 3		Selection check boxes in package modelingproject, 1, and 2 are grayed

Note: When you check out a package containing shortcuts, the package selection is extended to include the target packages referenced by shortcuts even if you do not have the Read permission on these packages.

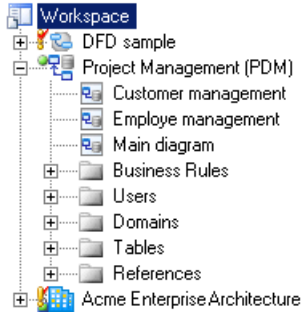
Checking the State of a Document

PowerDesigner monitors the state of your local document versions against those stored in the repository. A small icon is displayed before the document name in the Local tab of the Browser and, if you select the document, its status is displayed in the status bar.

Note: You can control the display of status icons with the **Display status icons in Local Browser** general option (see *Repository General Options* on page 8).

In the following example:

- the document "DFD sample" has been changed by you and by another user
- the document "Project Management (PDM)" is up-to-date
- the document "Acme Enterprise Architecture" has been changed and locked by you and changed by another user



The following table lists the possible states of a document:

Icon	Description
	Up to date (green circle) – Your local version is the same as the one in the repository.
	Changed by you (red checkmark) – Your local version has changes not saved to the repository.
	Changed by another (yellow exclamation point) – The repository version has changes that are not reflected in your local version.
	Up to date and locked by you (green circle and green padlock). For information about locks, see *Document Locks* on page 42.
	Up to date but locked by another (green circle and red padlock).
	Changed by you and changed by another (red checkmark and yellow exclamation point).
	Changed and locked by you (red checkmark and green padlock).
	Changed by you but locked by another (red checkmark and red padlock).
	Changed by another but locked by you (yellow exclamation point and green padlock).
	Changed and locked by another (yellow exclamation point and red padlock).
	Changed and locked by you and changed by another (red checkmark, yellow exclamation point, and green padlock).

Icon	Description
	Changed by you but locked and changed by another (red checkmark, yellow exclamation point, and red padlock).

Browsing the Repository

The PowerDesigner repository can contain various types of node.

Node	Description
Root	[when connection is established] Displays the repository definition name, the user login, and the current branch.
Folder	Used to organize documents in the repository.
Project	Container for models and other files.
Document	Models, multi-model reports, resource files (see *Sharing Resources in the Repository* on page 54), and external application files such as MS Office files, or graphics.
Package	Packages in models.
Object category	Object categories in models (if the Show Objects check box is selected in the General Options).
Diagram	Diagrams in models (if the Show Objects check box is selected in the General Options).
Object	Objects in models (if the Show Objects check box is selected in the General Options).

Repository Root Properties

Right-click the root of the Repository tab of the Browser, and select Properties to open its property sheet.

The Connection tab contains the following properties:

Property	Description
Repository name	Specifies the name of the repository to which you are connected.
Login name	Specifies the login name through which you are connected.
Current branch	Specifies the branch that you are viewing.

In addition, the following tabs are available:

- LDAP - lists LDAP configuration parameters to enable remote LDAP users to access the repository, see *Controlling Repository Access with LDAP* on page 85.
- Permissions – lists those users and groups with permissions on the entire repository. For more information about permissions, see *Controlling Permissions for Repository Items* on page 96.
- Version Info – specifies when the repository was created and last modified, see *Repository Document and Object Properties* on page 63.

Repository Folders

You can use folders to organize the contents of the repository by, for example, creating separate folders for each subject area, or each team working on a project. You must have write permission on the location where you want to create the folder. Documents checked in inside a folder inherit its permissions.

Note: A PowerDesigner project (see *Chapter 2, Projects and Frameworks* in the *Core Features Guide*) acts as both a repository document and a folder. Use the project properties to control permissions for working with the project file itself (project diagram and any framework or dependency matrices) and the folder properties to grant access permissions to the models contained within the project, as you would for a repository folder.

You create a folder by right-clicking the root node, a project, or a folder, selecting **New > Folder**, and entering a name for the folder in the property sheet. Click the **Permissions** tab to add permissions for all appropriate users and groups for the folder and its contents (see *Controlling Permissions for Repository Items* on page 96).

You can rename a folder from its property sheet or by selecting its node in the Browser and pressing the **F2** key.

You can move folders and documents in the repository by drag and drop. You cannot move a single document version nor can you move objects from one model to another. To move a folder or a model you must have Full permission on the folder or model to be moved, and Write permission on the target folder. You cannot move a project inside another project.

To delete a folder, first delete or move any contents and then right-click it and select **Delete**.

Repository Document and Object Properties

You can double click any folder, document or model object in the repository to see its property sheet.

General Tab

The General tab displays read-only information about the object.

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users

Property	Description
Code	[not available for folders] The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	[not available for folders] Additional information about the object.
Type	[not available for folders] Type of object: PowerDesigner model or package, other types of documents and object.
Location	[folders, documents, and model objects] Full path of the folder or document. [model packages]: hierarchy of packages in the model.
Status	[not available for folders] Indicates the status of the document: Updatable or frozen Locked and author of lock

Version Info Tab

The Version Info tab displays information about the origins of the object.

Property	Description
Creation	Displays the full name of the user who created the item in the repository, and the date of creation.
Last modification	Displays the full name of the user who last modified the item, and the date of this modification.
Generated From Origin Object	[model objects only] This box is displayed when the current object has been generated from another object, and displays the name of the origin object. Click the Properties button to the right of this field to open the origin object property sheet.
Version number	[not available for folders] Number of the version.
Version branch	[not available for folders] Name of the branch to which the version belongs.
Comment	[not available for folders] Version comment.

Versions Tab

The Versions tab displays a tree made of the different branches containing versions of the document. You can expand the branch nodes to display versions of the document. The following management tools are available on the Version tab of a document:

Tool	Description
	Properties - Displays the property sheet of the document

Tool	Description
	Check Out - Checks out the selected version. See *Checking Documents out of the Repository* on page 21.
	Compare - Compares two versions. See *Comparing Models in the Repository* on page 36.
	Freeze - Freezes the selected version. See *Document Versions* on page 40.
	Unfreeze - Unfreezes the selected version. See *Document Versions* on page 40.
	Lock - Locks the selected version. See *Document Locks* on page 42.
	Unlock - Unlocks the selected locked version. See *Document Locks* on page 42.
	Delete - Deletes the selected version

Note: When you check in a document, the local date is converted to GMT. It is converted again to local time when you check the document out. This is to let you to compare document versions stored in the repository when you are working in different time zones.

Other Tabs

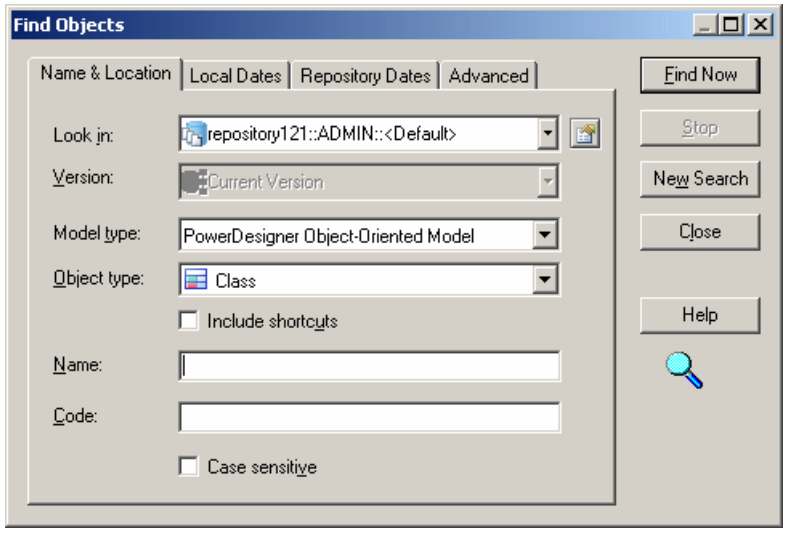
The other tabs are as follows:

- Notes [models, packages, and model objects only] - contains two sub-tabs:
  - Description - provides detailed information about an object.
  - Annotation - contains your notes regarding implementation.
- Attributes [models, packages, and model objects only] - displays additional properties set for these items.
- Collections [models, packages, and model objects only] - displays a list of collections or related objects (but not sub-objects) for the item. For example, the Collections tab in the property sheet of a CDM entity may display the business rules attached to the entity, but will not display entity attributes.
- Permissions [models, packages, folders and other documents only] – lists those users and groups with permissions on the document. For more information about permissions, see *Controlling Permissions for Repository Items* on page 96.

Finding PowerDesigner Model Objects

You can search for PowerDesigner model objects in the repository. You do not need special user rights and the List permission is sufficient for searching.

1. Press **Ctrl+Alt+F**, select **Repository > Find Objects**, or right-click the repository root, a folder, or model in the Repository tab of the Browser and select **Find Objects** to open the Find Objects dialog:



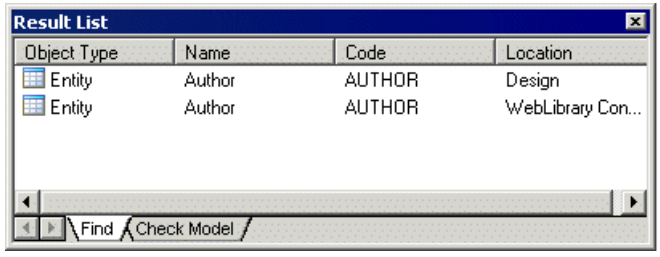
2. Specify the appropriate parameters in the different tabs of the dialog box, and then click the **Find Now** button.

   You can stop the process at any time by clicking the **Stop** button in the Find Object dialog box.

   The Find process in the lower part of the PowerDesigner window.

Note:

Progress is reported in the Output window, and the dialog remains open until the process completes and the Result List displays the result.



If you right-click an object in the Result list you can select:

- **Properties** - to display the properties of the found objects
- **Check Out Model** - to check out the model containing the found object

Find Objects Parameters

The four tabs in the Find Objects window allow you to specify many parameters for your search.

- Name & Location – restricts the search by any or all of:
  - location within the repository
  - version number of the object
  - type of model
  - type of object, and whether shortcuts should be included
  - the name or code of the object including wildcards
- Local Dates – restricts the search by creation or modification date and user on the local machine
- Repository Dates – restricts the search by creation or modification (check in) date and user in the repository
- Advanced – restricts the search by object attribute. The attributes available on this tab change depending on the object selected on the Name & Location tab and whether shortcuts are included. Select the checkbox in the U column to search on that attribute, and enter a string to search on in the expression column.

Special Characters

The following special characters allow you to use basic regular expressions when searching for the name or code of an object or attribute values:

- * - none to any number of characters
- ? – exactly one character
- * - * is a normal character
- \? - ? is a normal character
- \\ - \ is a normal character
- true/false - Boolean value (True is when the check box is selected in the interface)

For example:

W* finds "Work" and "Washington"

*IST finds "List" and "Specialist"

*96 finds "01/11/96" and "26/08/96"

????ER finds "Writer" and "Seller" but not "inner"

COLN? finds "COLN1" and "COLN2"

*_emp_??? finds "Div_emp_fun" and "Div_emp_idn"

For full documentation of the PowerDesigner Find Objects window, see "Finding Objects" in the Objects chapter of the *Core Features Guide*.

Comparing Models in the Repository

You can compare two PowerDesigner models to obtain a detailed list of the differences between them.

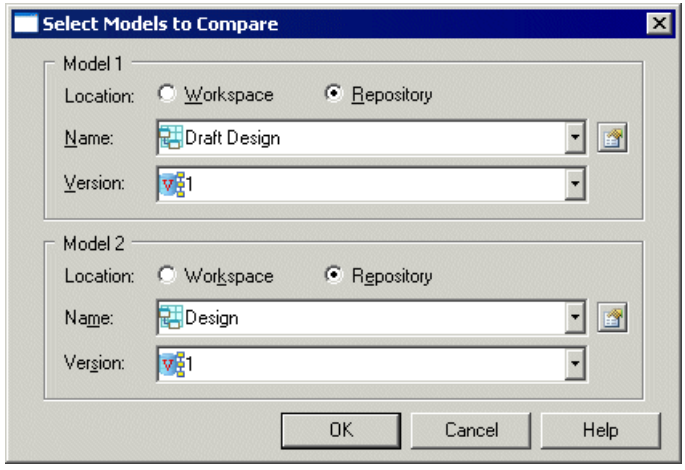
You can compare:

- Two different models in the repository
- Two different versions of the same model in the repository
- One model on your local machine with another model in the repository

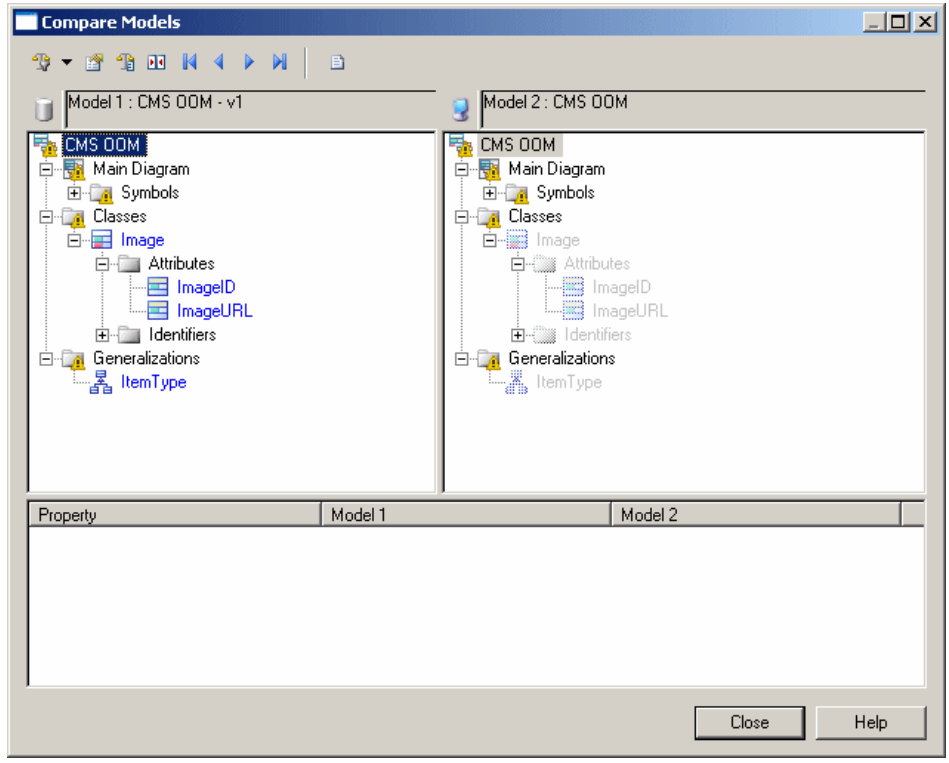
Warning! You have to select models of the same type, PDM, OOM, etc. You cannot compare models of different types.

1. Press **Ctrl+Alt+M**, select **Repository > Compare**, or right-click a model in the Repository tab of the Browser, and select Compare.

   The Select Models to Compare window opens:



2. For each of Model 1 and Model 2, select:
   - A location to specify whether the model to compare is located in your local workspace or in the repository.
   - The name of the model to compare – click the Properties tool to the right of the list to open the property sheet of the selected model. Select the same model in both lists if you want to compare different versions.
   - The version of the model to compare – the most recent version is selected by default.
3. Click OK to open the Compare Models window:



For full documentation of the Compare Models window, see the Comparing and Merging Models chapter in the *Core Features Guide*.

Impact Analysis

When you create an external shortcut, or when you attach a business rule to an object, a dependency link is created between models or objects.

Dependencies are used to verify the use of an object or model, they can be of two types:

- *Internal*, when the links are within a model. These dependencies are saved in the model and appear in the Dependencies tab of an object property sheet
- *External*, when the links exist between models. These dependencies are created during intermodel generation or external shortcut creation, they appear in the Dependencies tab if the related model is opened in the workspace. If the related model is not available, you can use the repository to retrieve external dependencies

Understanding Impact Analysis

The repository computes and saves external dependencies information in the checked in models. You can check out this information, and display external dependencies in the Dependencies tab of an object property sheet, even when the related models are not available.

This information can be used when you need to analyze the impacts of changes performed on a model or an object in the following way:

- Generation link, you can check if a model has been generated, and the name of the model(s) generated from the current model (column Generated As in Dependencies tab)
- External shortcut link, you can verify if objects in the selected model have shortcuts in external models

Checking Dependencies Out from the Repository

There are various ways to check out information about external dependencies in a model only if the related model is also checked in in the repository.

- Select the Check Out Dependencies check box in the Check Out dialog box when you check out a document.
- Select the Check Out Dependencies check box in the Repository tab in the General Options dialog box to systematically check out external dependencies.
- Select the Check Out Dependencies check box in the Check In dialog box if you choose to check out the model after check in.

Updating Model Dependencies from the Repository

This feature is only available for models checked in in the repository.

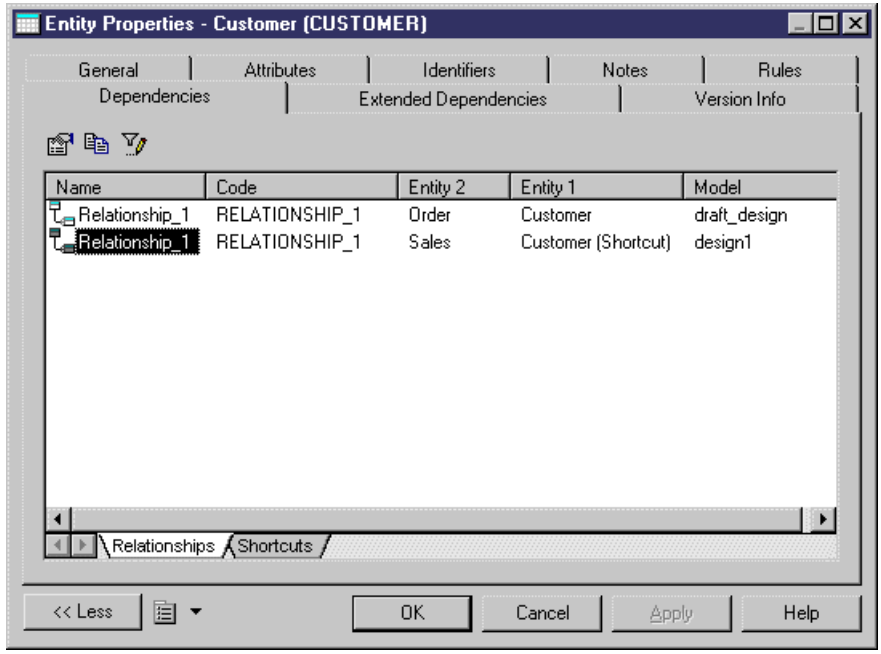
The external dependencies retrieved from the repository reflect the state of a model as it is stored in the repository. They are likely to be out of date if the local model has been modified, or not yet checked in.

Right-click a model node in the local Browser and select Update Dependencies from Repository in the node contextual menu.

For example, the model "draft_design" contains entity Customer. You create an external shortcut of Customer in model "design1". In model "design1", you create entity Sales and create a relationship between the shortcut of Customer and Sales.

You check in both CDM and make sure the Check Out Dependencies check box is selected in the Check in dialog box.

When you open the Dependencies tab of entity Customer in the target model, the following information appear to indicate that the entity has a shortcut and this shortcut is used by a relationship in another model:



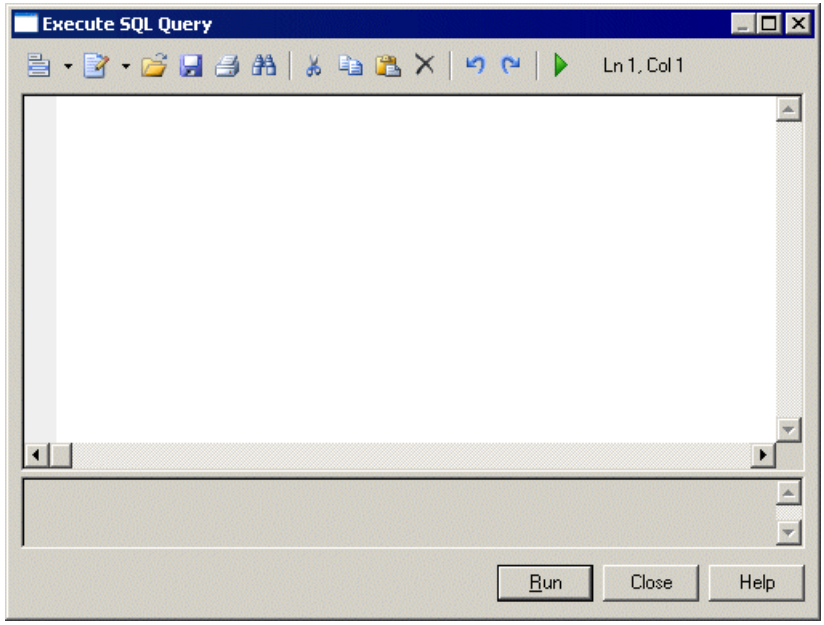
Querying the Repository Using SQL

You can run standard SQL SELECT queries against the repository through the Execute Query window. If you want to execute more complex queries, you should use your DBMS query editor.

Note: You cannot execute SQL queries while using the repository proxy.

1. Select **Repository > Administration > Execute SQL** to open the Execute Query window.

   Note: The Connect dialog box opens if you are not connected to the repository. Enter the connection parameters and click OK to connect.



2. Enter one or more SQL queries in the appropriate syntax for your DBMS in the window and click the Execute button.

   The results of your query are displayed in the Results pane.
3. Click Close to exit the Execute Query dialog box.

Freezing Document Versions

Unlike many repository systems, the PowerDesigner repository does not require you to create a new version of a document each time you check it in. You create a version only when you want to, by *freezing* the document.

Freezing a document allows you to go back and view its state at the point in time when it was frozen. You could choose to freeze a document (and thus create a new version) each time you check it in, once a day, once a week, or at any other interval.

Note: To freeze a document, you must have the `Freeze versions` right (see *Specifying User Rights* on page 92) and the appropriate permissions on the document.

Details of the current version of the document are displayed on the **Version Info** tab of its repository object property sheet, and all the previous versions appear as a tree on the **Versions** tab.

You can freeze a document:

- During check in (see *Checking Documents into the Repository* on page 11) by selecting one or both of:

    - the **Freeze before check in** option - to preserve the state of the model before your changes
    - the **Freeze after check in** option - to preserve the state of the model including your changes (this option can be selected by default if you select the **Freeze after check in** general option (see *Repository General Options* on page 8.
- At any time, by right-clicking the document, folder, project, or root node and selecting the **Freeze** command. Performing a freeze on a folder, project, or root node will freeze all its contents.
- At any time, by clicking the **Freeze** tool on the property sheet of:
    - a document - on the **Versions** tab
    - a branch - on the **Members** tab
    - a configuration - on the **Members** tab

Note: All repository documents and model objects can be versioned, but repository folders, projects, users, and groups are not versioned. Versions other than the first or baseline version are stored as differences from the previous version. As a rule of thumb, the baseline version of a model stored in the repository takes twice the size as it does when saved as an XML file.

You cannot individually control the freezing and versioning of PowerDesigner model objects. Each changed object has its version frozen and version number increased by one each time you freeze its parent model's version.

Unfreezing Document Versions

If you have the `Write` or `Full` permission on a document that is frozen, you can unfreeze it. Unfreezing a document version allows you to make further changes to the existing version, but means that you will no longer be able to go back and view the previously frozen state.

You can unfreeze a frozen document at any time by:

- Right-clicking the document, folder, project, or root node and selecting the **Unfreeze** command. Performing an unfreeze on a folder, project, or root node will unfreeze all its contents.
- Clicking the **Unfreeze** tool on the property sheet of:
    - a document - on the **Versions** tab
    - a branch - on the **Members** tab
    - a configuration - on the **Members** tab

Note: You can only unfreeze the latest version of a document.

Deleting Document Versions

If you have the `Full` permission on a document, you can delete the most recent version of the document or delete the document (and all its versions in all branches). Deleting a document version allows you to go back to a previously frozen state.

Warning! Deleting a version or a document cannot be undone.

When you delete a version or a document, you also delete all the objects it contains, and may also delete the target objects of shortcuts in other models. However these target objects are likely to exist in the versions following the deleted version. This has no impact in the repository, links between shortcuts and target objects are re-build after a check out, in the local workspace.

You can delete a frozen document version by:

- Right-clicking the document and selecting the **Delete Version** command.
- Clicking the **Delete** tool on the property sheet of:
  - a document - on the **Versions** tab
  - a branch - on the **Members** tab
  - a configuration - on the **Members** tab

You can delete a document and all its versions in all branches by right-clicking the document and selecting the **Delete Document** command.

Locking Document Versions

You can lock a repository document to prevent other users from making changes to it. Only you or a user with the `Full` permission on the document can check in changes to the document until you unlock it. Generally, you will release the lock when you check in the document.

Locked documents display a lock icon in the repository browser:

Icon	Description
	Blue icon if the connected user has set the lock
	Red icon if the connected user has not set the lock

Note: To lock a document, you must have the `Lock versions` right (see *Specifying User Rights* on page 92) and the appropriate permissions on the document.

You can lock a document:

- During check out (see *Checking Documents out of the Repository* on page 21) by selecting the **Lock before check out** option.
- At any time, by right-clicking the document, folder, project, or root node and selecting the **Lock** command. Performing a lock on a folder, project, or root node will lock all its contents.
- At any time, by clicking the **Lock** tool on the property sheet of:
  - a document - on the **Versions** tab
  - a branch - on the **Members** tab

    - a configuration - on the **Members** tab
- Using the List of Locks (see *Managing Document Locks in the List of Locks* on page 43).

Unlocking Documents

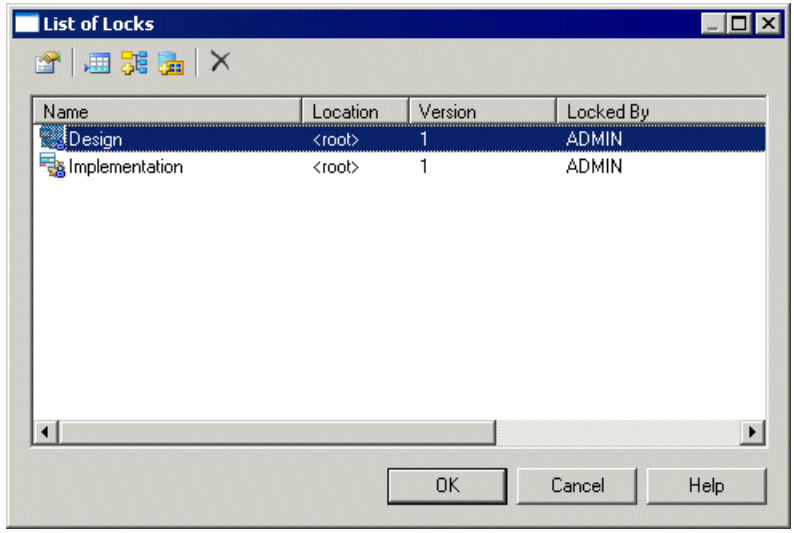
If you have locked a document (or have the `Full` permission on a document locked by another user, you can unlock it. Unlocking a document version allows any user with the appropriate permissions to check in changes to it.

You can unlock a locked document:

- During check in (see *Checking Documents into the Repository* on page 11) by selecting the **Unlock after check in** option.
- At any time, by right-clicking the document, folder, project, or root node and selecting the **Unlock** command. Performing an unlock on a folder, project, or root node will unlock all its contents.
- At any time, by clicking the **Unlock** tool on the property sheet of:
    - a document - on the **Versions** tab
    - a branch - on the **Members** tab
    - a configuration - on the **Members** tab
- Using the List of Locks (see *Managing Document Locks in the List of Locks* on page 43).

Managing Document Locks in the List of Locks

You can review the documents that are currently locked, and lock or unlock on the List of Locks, which is available by selecting **Repository** > **Administration** > **Locks**.



The following tools are available on the List of Locks:

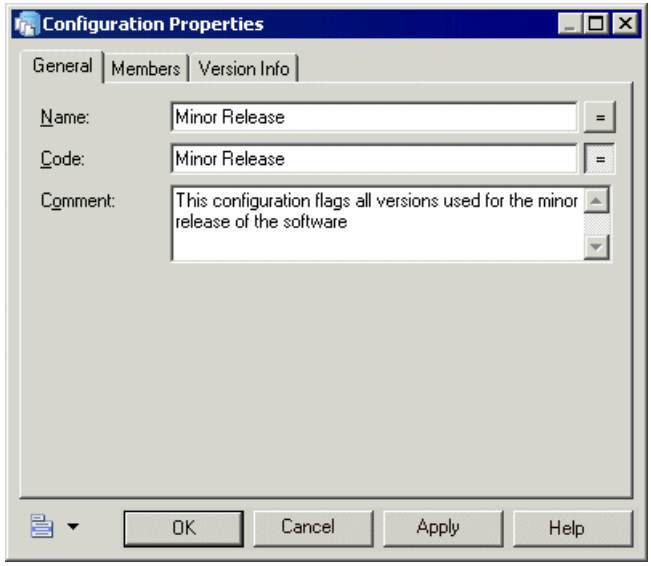
Tool	Description
	Settings – Opens the Lock Document dialog for the selected document, which shows details of the lock upon it.
	Add Lock – Opens the Lock Document dialog. Select a document to lock in the Name list and a comment, and click OK to lock it and return to the List of Locks.
	Add Related Document Versions - [models only] Locks any models that the selected model references through external shortcuts. The related models are automatically added to the List of Locks.
	Add Document Versions from a Configuration – Opens the Select Configuration dialog box, which allows you to select a configuration for locking. Document versions contained within the configuration are locked and added to the List of Locks.
	Delete – Unlocks the selected document, and deletes it from the List of Locks. Note that this tool does not delete the document version.

Note: You can lock all the document versions in the current branch by right-clicking the root node and selecting Lock from the context menu. This command is available only if you have the Lock versions right and the appropriate permissions on the documents.

Grouping Document Versions in a Configuration

A *configuration* is a container for grouping a set of repository documents versions together to mark a particular point in a project, such as a release or other project milestone. You must have the `Manage Configurations` right to create configurations.

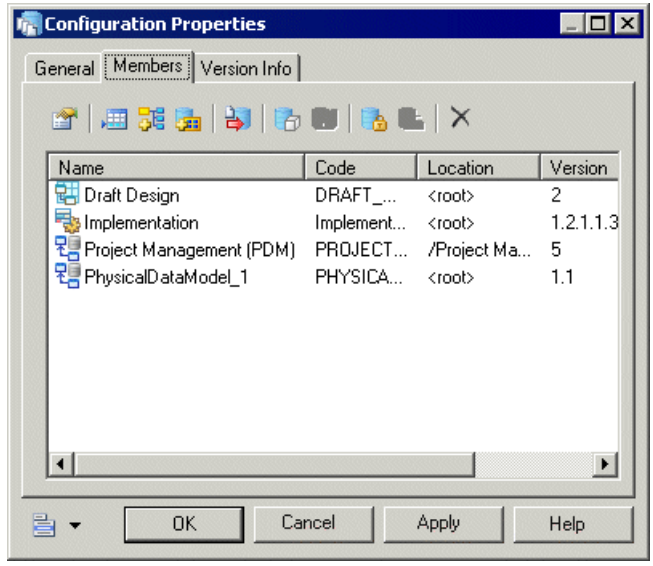
1. Select **Repository** > **Configurations** to open the List of Configurations.
2. Click the **Add a Row** tool to create a new configuration, and then click the **Properties** tool to open its property sheet.
3. Enter an appropriate Name, Code, and Comment to describe the configuration.



4. Click the **Members** tab to add document versions to the configuration, using the following tools.

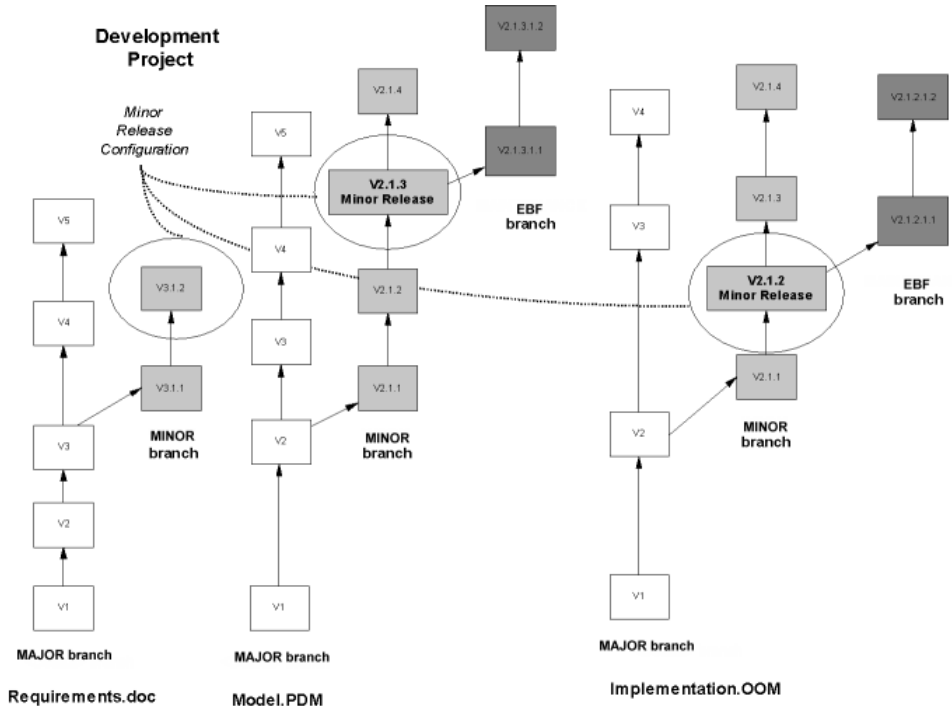
Tool	Description
	Properties – Opens the selected repository document property sheet.
	Add Document Version – Opens the Add Document to Configuration dialog, which allows you to specify the version of a document to add to the configuration.
	Add Related Document Versions – [models only] Adds any models that the selected model references through external shortcuts. The related models are automatically added to the configuration.
	Add Document Versions From a Configuration – Opens the Select Configuration dialog, which lets you select a configuration, whose members will be added to the configuration.
	Check Out – Checks out the selected repository document version (see *Checking Documents out of the Repository* on page 21).
	Freeze – Freezes the selected repository document version (see *Freezing Document Versions* on page 40).
	Unfreeze – Unfreezes the selected repository document version (see *Unfreezing Document Versions* on page 41).
	Lock – Locks the selected repository document version (see *Locking Document Versions* on page 42).

Tool	Description
	Unlock – Unlocks the selected repository document version (see *Unlocking Documents* on page 43).
	Delete – Deletes the selected document version from the configuration (see *Deleting Document Versions* on page 41).



5. Click **OK** to return to the List of Configurations.

In the following example, the versions used for a minor release are included in a configuration called Minor Release:



Creating a Configuration from a Project or Folder

You can quickly create a configuration from the current versions of all the documents in a project or repository folder by right-clicking it and selecting **Create Configuration**.

The property sheet of the new configuration opens. You can modify the default name and add a comment on the **General** tab and modify the document versions it contains on the **Members** tab (see *Grouping Document Versions in a Configuration* on page 44).

Checking Out a Configuration

When you check out a configuration, you check out all the document versions it contains.

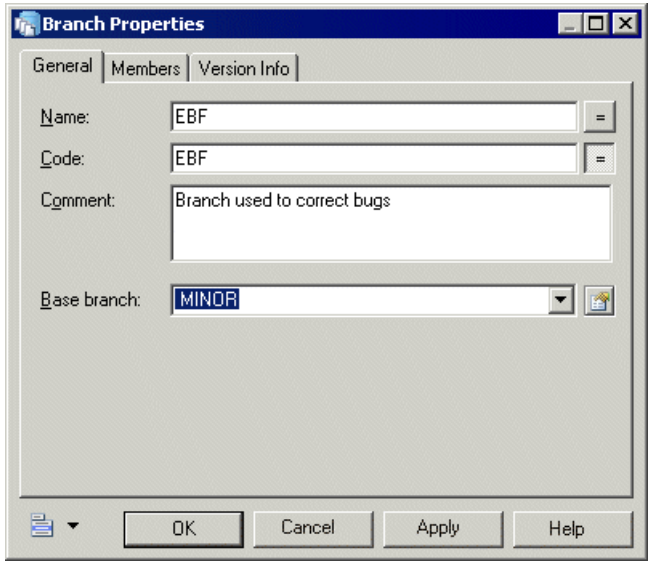
1. Select **Repository > Check Out** to open the Check Out Multiple Documents dialog (see *Using the Check Out Multiple Documents Window* on page 23).
2. Click the **Add Document Versions from a Configuration** tool, select the configuration to check out, and click **OK** to add the document versions it contains to the list.
3. Click **OK** to check out the documents.

Note: When you check out a configuration containing resource files, these files may overwrite any modifications in your local resource files. If you need to preserve your changes, modify the path to which the configuration will be checked out.

Branching Version Trees

When you connect to the repository for the first time, the <Default> branch is created to act as the trunk of the version tree. You can create branches that diverge from the trunk to support concurrent development of multiple versions of repository documents. You must have the Manage Branches right to create branches.

1. Select **Repository** > **Branches** to open the List of Branches.
2. Click the **Add a Row** tool to create a new branch, and then click the Properties tool to open its property sheet:



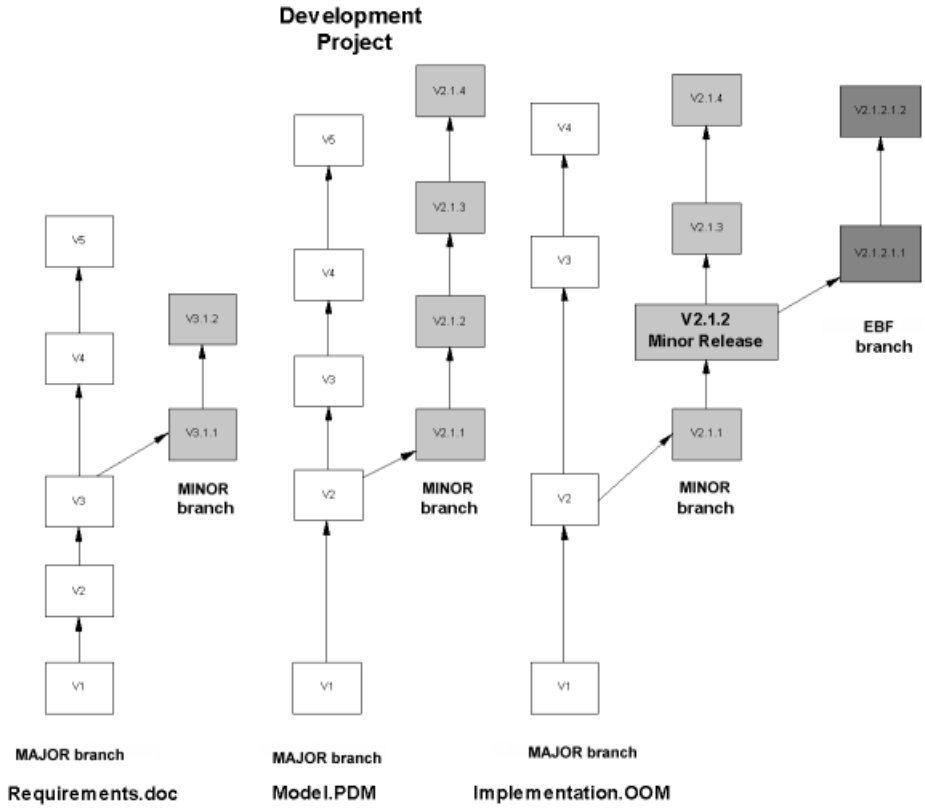
3. Enter an appropriate name, code, and comment to describe the branch.
4. Select the base branch upon which you want to base the new branch. If this is your first branch, the base branch will be Default. If you have already created other branches, it can be any of them.

   On creation a branch inherits all the document versions in its base branch. If you select <None>, the new branch will be a second trunk in parallel with Default, and will be empty upon creation.
5. Click **OK** to return to the List of Branches.

The following example shows three documents being used by a software development team:

- Requirements.doc - to specify the software architecture.
- Model.PDM - to design the database.

- Implementation.OOM - to develop the software implementation.



Depending on the type of release on which they are working, each engineer will choose a different branch to connect to in the repository, and will have access to different versions of the documents:

Branch	Document versions
MAJOR - used to work on the major release of the software, specifying new requirements for, designing, and implementing major features.	Requirements.doc V5 Model.PDM V5 Implementation.OOM V4
MINOR - (based on MAJOR)used to work on a minor release to correct bugs and add small features.	Requirements.doc V3.1.2 Model.PDM V2.1.4 Implementation.OOM V2.1.4

Branch	Document versions
EBF (Emergency Bug Fix) - (based on MINOR) created after the minor release to let engineers correct bugs on a released software version.	Requirements.doc V3.1.2 Model.PDM V2.1.4 Implementation.OOM V2.1.2.1.2

Checking a Document into a Specific Branch

Whenever you connect to the repository, you are connecting to a specific branch. If you want to check your document into another branch, you must change branch before beginning your check in.

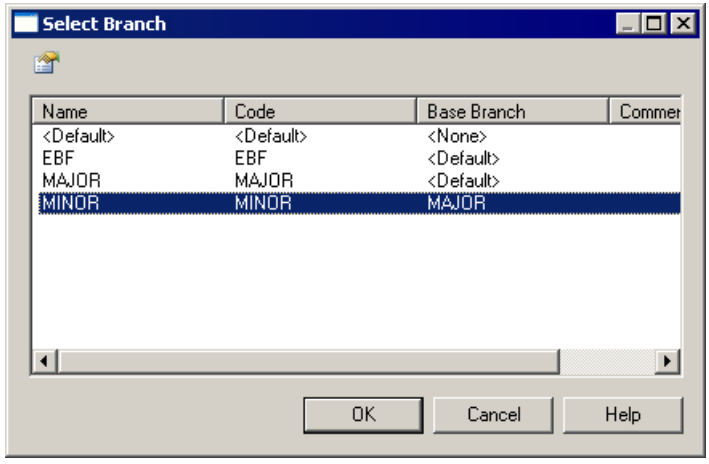
Your current branch is displayed at the end of the root node in the **Repository** tab of the Browser as follows:

```
repository::username::branch
```

For example:

```
Main Repository::John::MAJOR
```

1. Select **Repository** > **Change Branch** to open the Select Branch dialog box.



2. Select a branch in the list and click **OK** to return to the repository browser, which now displays the document versions contained by the selected branch.
3. Proceed with your check in as usual (see *Checking Documents into the Repository* on page 11)

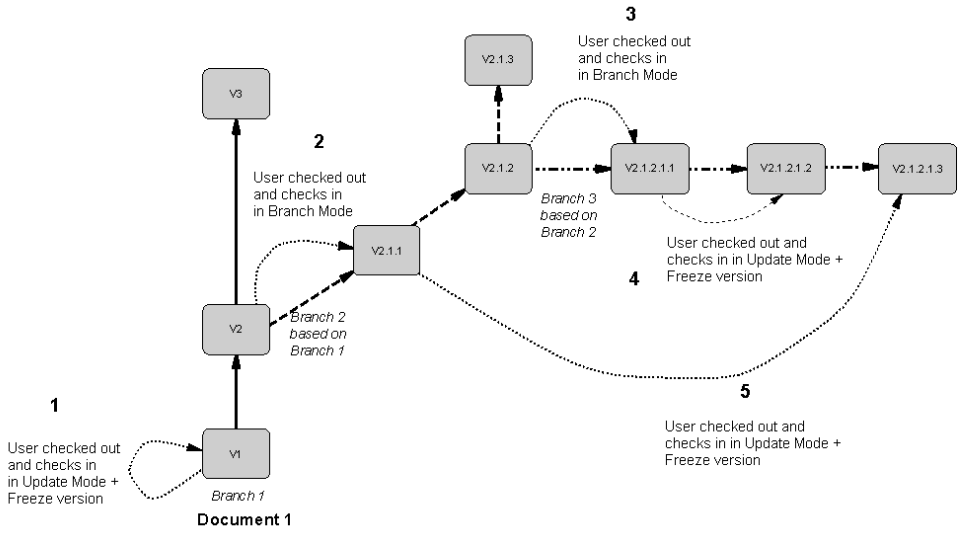
   When you have completed your check in, you can confirm that your document version has been added to the correct branch by opening the property sheet of the branch and clicking its **Members** tab (see *Managing Branch Members* on page 52).

Checking in a Document Using the Update, Branch, and Integrate Modes

The check in modes allow you to check your document into the same branch (update), a new branch (branch), or to merge changes from one branch into another parallel branch (integrate).

Remember that you should always select the correct target branch before check in as you cannot change branch from the Check In dialog.

The following example shows several situations involving the Update and Branch check in modes:

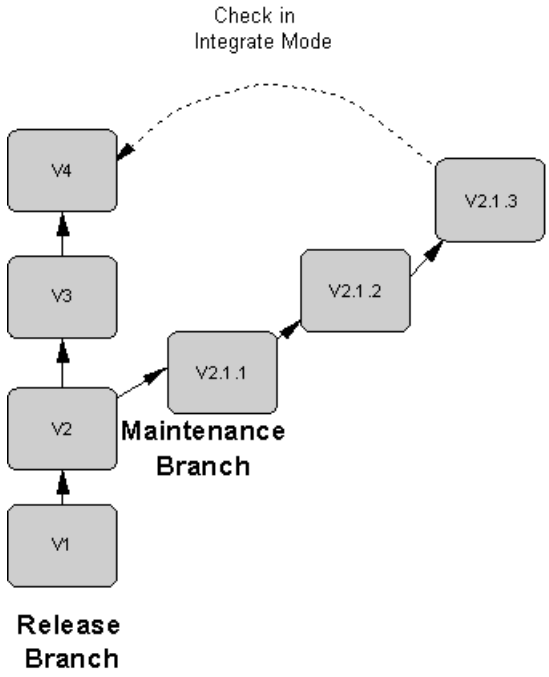


Case	Check out branch	Current branch	Check in Mode
1	Branch 1	Branch 1	*Update* mode to update V1
2	Branch 1	Branch 2	*Branch* mode to create the version in the current branch
3	Branch 2	Branch 3	*Branch* mode to create the version in the current branch
4	Branch 3	Branch 3	*Update* mode, as the document already exists in the current branch, and freeze previous version.
5	Branch 2	Branch 3	*Update* mode, as the document already exists in the current branch (V 2.1.1.1), and freeze previous version.

Note: The repository uses the GUID (Global Unique ID) of documents to identify them during check in and check out. If you use the Save As command to create a copy of the document, the

GUID is modified and the new document cannot be checked in in Update mode. However, if you copy or move the file in Windows Explorer, the GUID will not be affected.

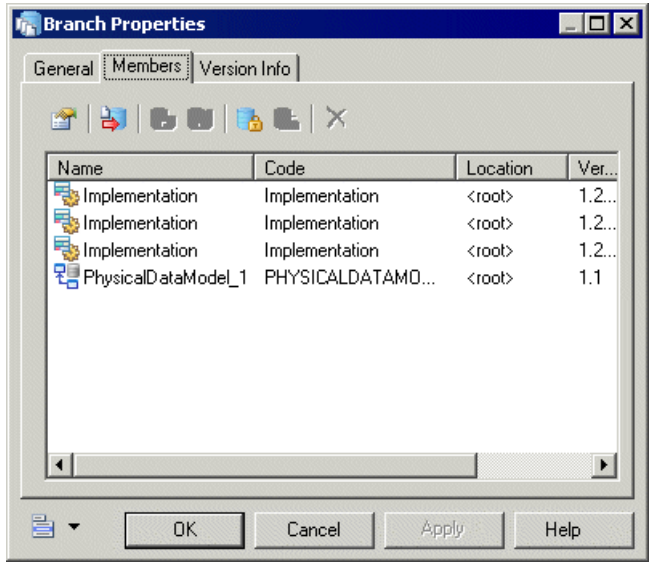
The following example shows a maintenance engineer working on a bug fix in the Maintenance branch wants to integrate this correction to the Release branch:



Managing Branch Members

When you create a new branch, it has no version *members* of its own, but it does have access to all the document versions of its base branch and all its ancestor branches.

You can add versions to a branch by checking them in. Each document version is displayed as a *member* of the branch on the Members tab of its property sheet. If you check the same document in several times in a branch, each version becomes a member of the branch.



The following tools are available on the Members tab:

Tool	Description
	Properties – Opens the selected repository document property sheet.
	Check Out – Checks out the selected repository document version (see *Checking Documents out of the Repository* on page 21).
	Freeze – Freezes the selected repository document version (see *Freezing Document Versions* on page 40).
	Unfreeze – Unfreezes the selected repository document version (see *Unfreezing Document Versions* on page 41).
	Lock – Locks the selected repository document version (see *Locking Document Versions* on page 42).
	Unlock – Unlocks the selected repository document version (see *Unlocking Documents* on page 43).
	Delete – Deletes the selected document version from the branch (see *Deleting Document Versions* on page 41).

Deleting a Branch

If you have the Manage Branches right, you can remove unused branches by deleting them from the list of branches.

Warning! Deleting a branch cannot be undone. You cannot delete the current branch, and you cannot delete a branch if it contains any members.

1. Select **Repository** > **Branches** to open the List of Branches.
2. Select a branch in the list and click the **Delete** tool.
3. Click **OK** to confirm deletion.

Sharing Resources in the Repository

You can store PowerDesigner resource files (DBMSs, target languages, and extensions) in the repository and automate their deployment to your team. Resource file sharing helps you to ensure that all your team are using the same DBMS or other target with any appropriate extensions.

Resource files are not shared by default, but once you have chosen which resource files you want to share, checked them into the repository, and put in place policies for sharing, your team will automatically receive updates to their resources each time they open a model. You can also use templates to enforce the use of shared resources when team members create models.

Automatic updates to shared resources can be disabled for individual team members if, for example, they are working on designing extensions to a DBMS, target language, or extension file.

Preparing to Share Resources in the Repository

No resource files are shared by default. The resource manager must decide which resources to share (generally those DBMSs, languages, and extensions with which the team works or those that have been modified for your development), and set up her machine as a template for the environment that will be deployed to all client machines.

Creating Named Paths to Locate Shared Resources

Models that refer to shared resources must know where to locate them. The resource manager must create named paths to point to a standard folder outside of your PowerDesigner installation (and, preferably, one which will be available and writeable on all client machines) to which all shared resources will be deployed.

The named paths you create will be deployed to users via a user profile and will enable all users to point to the appropriate resources.

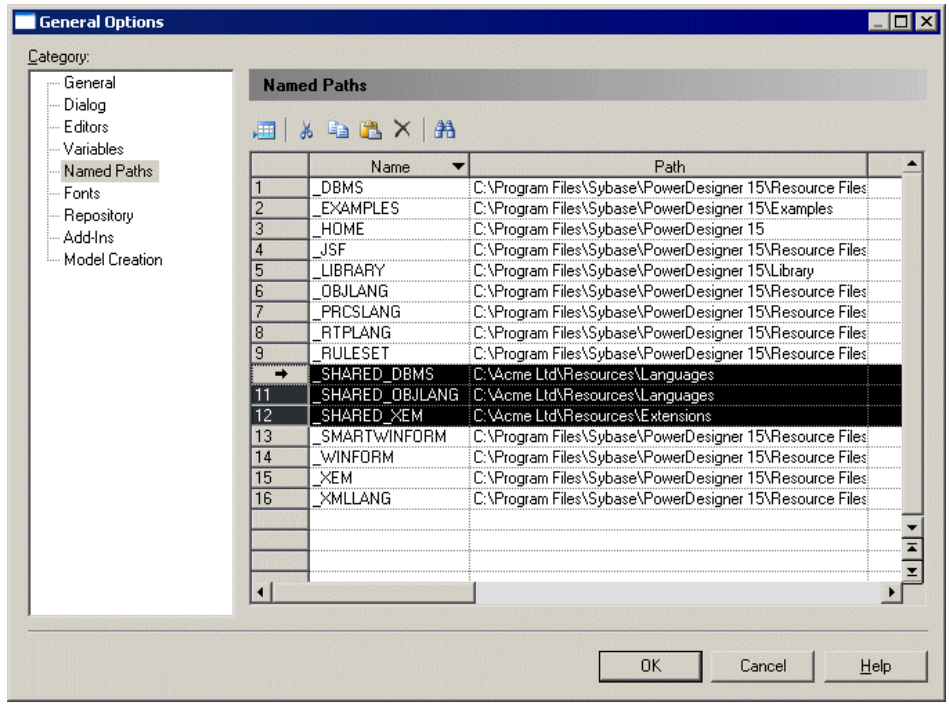
1. Select **Tools > General Options**, and then click the **Named Paths** category to display the list of named paths.
2. Enter a name (for example, `_SHARED_DBMS`) and a default path (for example `C:\Acme\Resources\DBMS`) for each shared resource folder that you have created.

   You should create a named path for each type of resource that you intend to share. These may include:

- DBMSs (*.xdb) and object (*.xol), business process (*.xpl), and XML (*.xsl) languages
- Extensions (*.xem)
- Model category sets (*.mcc) - to enforce the use of shared resources at model creation time (see *Making Shared Resources Available in the New Model Dialog* on page 57)
- Model templates - to be referenced by your model category sets
- User profiles (*.upf) - to configure each of these elements

3. Click **OK** to close the dialog.

In the following example, the Acme resource manager has created separate named paths for DBMSs, object languages, and extensions:



For more information about named paths, see "Naming Conventions" in *Chapter 8, Customizing Your Modeling Environment* of the *Core Features Guide* .

Selecting and Preparing Resources to Share

The resource manager selects the resource files to share and places them in appropriate folders in preparation for checking them into the repository.

The resource manager is responsible for collecting and organizing the resources to share. She may obtain them from other users or by copying them from her own installation (see *Chapter 1, Resource Files and the Public Metamodel* in *Customizing and Extending PowerDesigner*).

1. Copy all the resource files to share into the folders for which you have created named paths.
2. For each type of resource file that you intende to share, select **Tools > Resources > List of *ResourceType*** to open the appropriate resource list.
3. Click the Path tool, and browse to and select the folder in which you have placed the resource files of this type to populate the resource list with its contents.

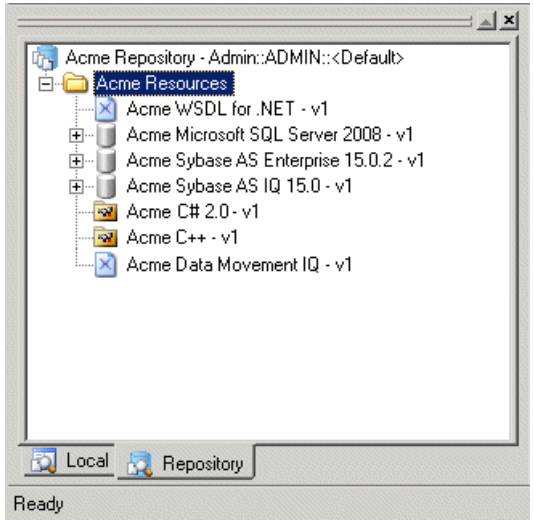
Checking Resources to Share into the Repository

The resource manager must check the selected resource files into the repository in order to make them available to the team as shared resources.

1. Connect to your repository, and create a folder in which to store your shared resources (see *Repository Folders* on page 31). You should ensure that all users have at least Read permission for this folder and all the documents in it.
2. Open the appropriate list of resources (**Tools > Resources >** *ResourceType*), select the resource you want to share, and click the **Check In** tool to open the Check In Document window.
3. In the Folder list, select your shared resources folder and click **OK** to check in the resource and return to the list.

   When you have checked in all the resource files of this type that you want to share, close the list and continue with another list of resource files.

   In the following example, the Acme resource manager has checked her shared DBMSs, object languages, and extensions into the Acme Resources folder:



Making Shared Resources Available in the New Model Dialog

Once you have set your named paths, grouped your resources to share into the external folders, and pointed your resource lists to them, you can create model creation categories and model templates to enforce the use of these shared resources for model creation.

Model category sets contain categories which in turn contain templates for creating models and which point to predefined target resources. Once you have created a category set, you can make it appear in the New Model dialog by selecting it in the General Options dialog.

Since model category sets are themselves PowerDesigner resource files, and you will need to deploy the category set to your users, you will need to store them with your other resources to share and specify a named path to point to them.

1. If you have not already done so, create a named path to point to the folder where you will store your model category sets, ready to upload them to the repository.
2. Select **Tools > Resources > Model Category Sets** to open the List of Model Category Sets, click the Path tool and browse to and select this path.
3. Click the **New** tool to create a new category set, give it a name, and click **OK**. Create the categories that you need, and as many diagram templates as you need to point to each of your shared resources.

   Note: For detailed information about creating model category sets, see "Guiding Model Creation through Categories and Templates" in the Customizing your Modeling Environment chapter of the *Core Features Guide*.
4. Once you have created your category set, select **Tools > General Options**, and click the **Model Creation** category in the left pane.
5. Make sure that the **Enable categories** checkbox is selected, and then select your category set as the default.

   Note: Your category set will only appear in this list if it is correctly defined and the list path is set to point to your shared resources directory.
6. [optional] Deselect the **Enable model types** and **Enable model template files** checkboxes to hide these options in the New Model dialog.
7. Click **OK** to save your changes, and then select **File > New Model** to open the New Model dialog and review your category set.
8. Once you are happy with your category set, check it in to the shared resources folder in your repository (see *Checking Resources to Share into the Repository* on page 56).

Creating a User Profile to Enable Resource Sharing

The resource manager creates a user profile to deploy the settings required to push shared resources to her team. New users will select this user profile when installing PowerDesigner. Existing users will check this user profile out of the repository and apply it.

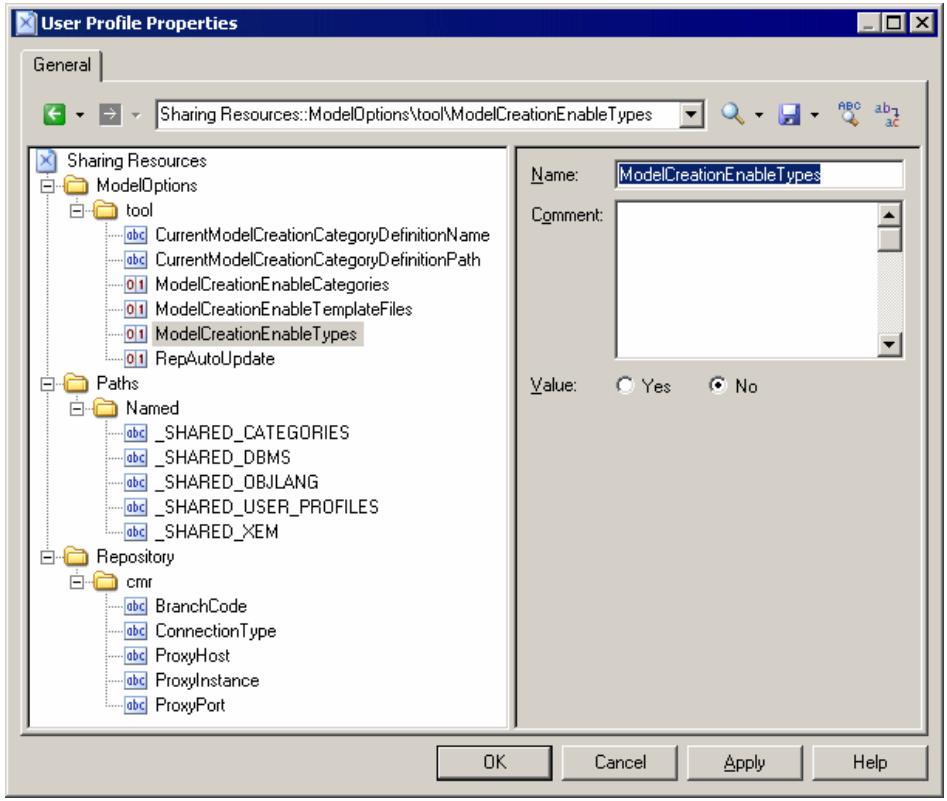
Since user profiles are themselves PowerDesigner resource files, and you will need to deploy the user profile to your users, you will need to store them with your other resources to share and specify a named path to point to them.

Before creating the user profile, you should ensure that all the necessary settings are correctly set on your machine, and that you have a directory structure like the following (and equivalent named paths), which contains all the resources you want to share:

```
C:\Shared Resources
    \DBMS
    \Extensions
    \Object Languages
    \Model Category Sets
    \User Profiles
```

1. Select **Tools** > **Resources** > **User Profiles** to open the List of User Profiles.
2. Click the **New** tool to open the New User Profile dialog, enter a name for your profile and select **<registry>** in the **Copy from** list to copy your PowerDesigner defaults from your Windows registry.
3. Specify a name and a location to create the profile and click **Save**.
   The user profile is created and opened for review in the resource editor.
4. Delete all the items except the following:
   - RepAutoUpdate general option (in ModelOptions/tool) - which automates the update of resources from the repository.
   - The ModelCreation options (in ModelOptions/tool) - which control the use of categories and templates in the New Model dialog.
   - Named paths (in Paths/Named) - for each of the shared resource folders you have created.
   - Repository definition (in Repository/*Repository Name*) - for the repository in which the shared resources are stored.

   In the following example, the Acme Shared Resources user profile contains values for the Auto-Update and ModelCreation options, shared resource named paths, and repository connection information via the repository proxy:



5. Click **OK** to close the editor.
6. Check the user profile in to the shared resources folder in your repository (see *Checking Resources to Share into the Repository* on page 56).

   For more information about user profiles, see "User Profiles" in *Chapter 8, Customizing Your Modeling Environment* of the *Core Features Guide*.

Communicating Resource Sharing Policies to Your Users

When the resource manager has set up the environment to share resources, she must communicate the appropriate sharing policies to her team.

To allow your team to begin using the shared resources, you need to provide them with the following information:

- Repository login and password - to enable them to connect to the repository where you have stored the shared resources, the user profile and any model templates and categories for use in model creation.
- User profile - to customize the setup of their PowerDesigner environments for sharing resources. Existing users will check the user profile out into their User Profiles folder, and then apply it. New PowerDesigner users will select the appropriate user profile when

installing PowerDesigner. You can also deploy the user profile to new installs during silent installation (see Installing PowerDesigner in Silent Mode in the Installation Guide).

- Model templates and categories - to check out to enforce the use of shared resources when creating new models.

Working with Shared Resources

For the majority of users, once their PowerDesigner environments are correctly configured, the use of shared resource files is completely automatic and invisible. However, the resource manager and some advanced users who edit resource files will want to disable the automatic updating of resources.

To disable the automatic updating of resource files, select **Tools > General Options**, click the Repository category, and then deselect the **Automatic resources update** option.

Editing Shared Resource Files

Advanced users with the appropriate rights can check in changes to shared resources through the standard Check In Document window. These changes will be made available to team members next time they open a model that is attached to the shared resource.

1. Select **Tools > Resources >** *ResourceType* to open the appropriate list of resources.
2. Select a resource and click the **Check In** tool to open the Check In Document window.
3. Select the appropriate parameters (see *Check in Parameters* on page 16), and click **OK** to check in your changes to the repository.

Comparing Shared Resources Files

Anyone with at least `Read` permission on selected resource files can perform a comparison to obtain a detailed list of the differences between them.

From the **List of *ResourceType*** , select a resource, and click the **Compare with Repository** tool to compare:

- Two different resources in the repository
- Two different versions of the same resource in the repository
- One resource on your local machine with another resource in the repository

Note: You can only compare resources of the same type: DBMS, object languages, XEM, etc. For more information, see *Comparing Models in the Repository* on page 36.

CHAPTER 3 Browsing the Repository Via the Web

The PowerDesigner Portal allows you to view the contents of your PowerDesigner repository in your web browser. You can share your models with a wide audience, including those who are not PowerDesigner users.

Note: An administrator must install and configure the PowerDesigner Portal server before you can access the repository via the web. For installation details, see "Installing the PowerDesigner Portal" in the Installing the Repository chapter of the *Installation Guide*.

Logging into the PowerDesigner Portal

In order to log into the PowerDesigner Portal, you must have a repository user account, and obtain the appropriate address for the site from your administrator. Your administrator must also advise you which connection profile to use to connect to your repository.

1. In your web browser navigate to the web site address provided by your administrator (for example http://*server*:3030/cmr):

2. Select the selection profile recommended by your administrator and enter your repository user name and password.

You can click the Show profile information link to display details of the selected profile or the Advanced options link to manually enter a host name and port instead of using a profile.

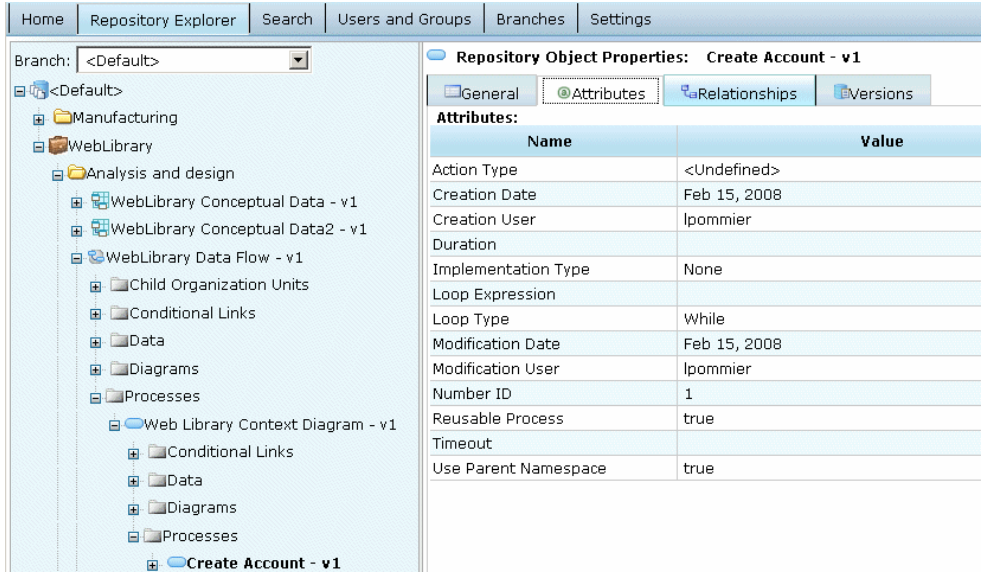
3. Click Logon to go to the PowerDesigner Portal home page.

   Each PowerDesigner Portal page includes the following links in the top-right corner of the screen:

   - Help – to access this help document
   - Language – to change the interface language
   - Preferences – to review your connection parameters and change your password
   - Refresh – to refresh the display
   - Logout - to logout

PowerDesigner Portal Repository Explorer

The Repository Explorer page is the main page of the PowerDesigner Portal, which contains a tree view of the contents of the repository and, on the right, the properties of the selected item.



The explorer can contain the following types of node.

- Root - displays the current branch
- Folder - used to subdivide and organize the documents in the repository
- Project – a container for models and other documents

- Document – projects, models, multi-model reports, and external application files such as MS Office files, or graphic files
- Package - model packages
- Object category - model object categories
- Diagram - model diagrams. You can click on object symbols in the diagrams to go to the relevant object.
- Object - model objects

Repository Document and Object Properties

Click any document or model object in the repository explorer to view its property sheet.

The following tabs are available:

- **General** - displays basic, read-only, information about the object. For shortcuts, click the entry in the **Target object** field to open the object that the shortcut references.
- **Properties** - [models, packages, and model objects] displays the properties of the object.
- ***Sub-objects*** - [objects] displays a list of this type of sub-object (for example, table columns, keys, or indexes, or class attributes or operations)
- **Relationships** - [objects] displays lists of the other objects with which the object is related
- **Versions** - displays the various versions of the document stored in the repository, along with information about the current version of the document, its creation and last modification. For external application files, click the icon in the **Download** column to download the file.
- **Permissions** - [models, packages and other documents] lists those users and groups with permissions on the document (see *Adding User or Group Permissions for a Repository Item* on page 63).
- **Compare** - [models and objects] Allows you to compare different versions of models and objects (see *Comparing Model and Object Versions* on page 64).
- **Subscriptions** - [projects, folders, and models] Allows you to subscribe to email notifications of changes to the object (see *Subscribing to Change Notifications* on page 64).

Adding User or Group Permissions for a Repository Item

You can change access permissions on an item from the PowerDesigner Portal.

1. Click the object in the repository explorer to display its property sheet, and then click the Permissions tab.
2. Click the Add tool to open a selection box listing all the available users and groups.
3. Select one or more users and groups, selecting the appropriate permission for each from the drop down list box in the Granted Permission column. The following permissions are available:

- List - User or group has minimal permissions to view the item in the browser; display read-only property sheets and search for models.
- Read - User or group has all the List permissions, and can also compare documents, and check the document (or folder contents) out from the repository.
- Write - User or group has all the Read permissions, and can also check in, freeze and lock document versions.
- Full - User or group has all the Write permissions, and can also manage permissions granted to users or groups and remove locks on documents.

Note that users who do not have any permissions on a document or folder cannot even see these items in the browser.

4. [optional – for projects, folders, models] Select the Cascade permission changes to folder contents checkbox to cascade your changes to children of the node.
5. Click the Save button to save your changes and return to the Permissions tab.

   The Granted Permission column shows the permissions explicitly granted to groups and users, and Effective Permission column displays their actual permissions, the most extensive permission granted either explicitly or by way of membership in a group.

Subscribing to Change Notifications

You can subscribe to be notified of changes made to any project, model, external application document, or folder on which you have at least Read permission.

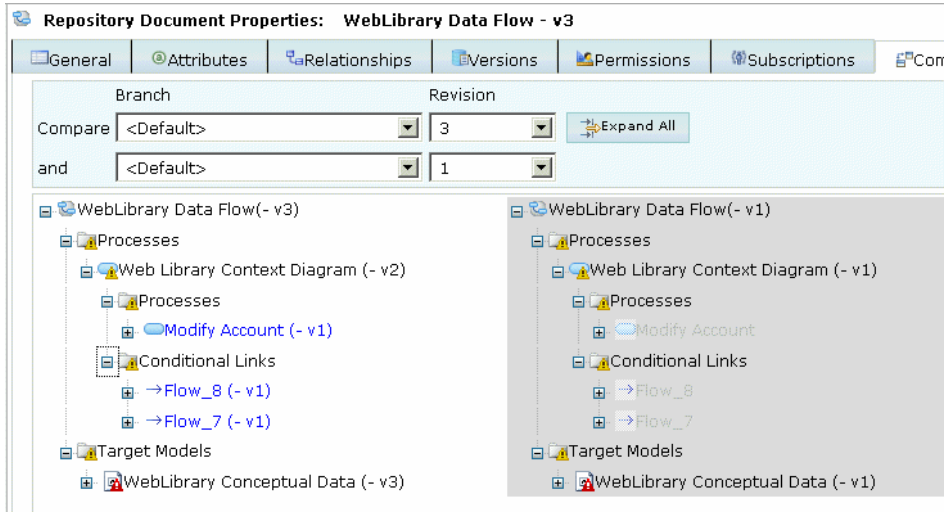
1. Navigate to the project, model, or folder for which you want to subscribe in the Repository Explorer, and then click on the Subscriptions tab.
2. Select the Email checkbox against your name in the Notification Method column, and then click Save.

Note: If you have the Manage users right, click the New button on the Subscriptions tab to open a list of users and groups, select all those whom you want to subscribe to change notifications, and then click Save.

Comparing Model and Object Versions

You can compare two versions of a PowerDesigner model or object in the PowerDesigner Portal to obtain a detailed list of the differences between them.

1. Click the object in the repository explorer to display its property sheet, and then click the Compare tab.

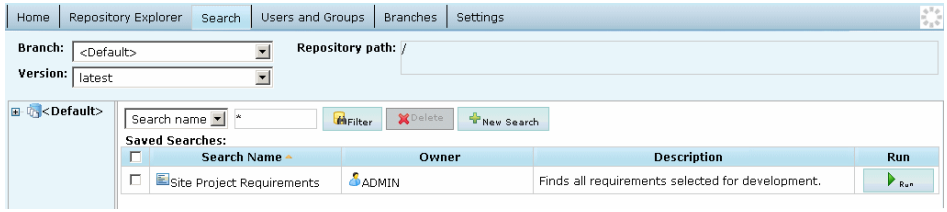


2. Select the branches and versions numbers of the two versions of the model or object that you want to compare.
3. [optional] Click the Expand All button to expand all the nodes.
4. [optional] Click any of the child nodes to view changes to their properties.

PowerDesigner Portal Search

You can search for PowerDesigner model objects in the PowerDesigner Portal by clicking the Search tab. You cannot search external application documents. You must have at least the `Read` permission to find objects.

1. Click the **Search** tab to access the Search screen:



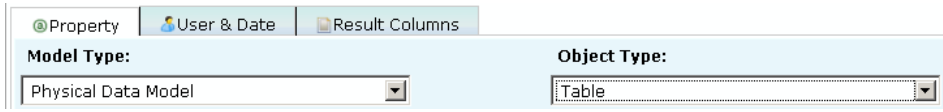
2. Select a saved search from the list and click the **Run** button or click the **New Search** button to create a new search.
3. [optional] Select a branch, model (by clicking it in the Explorer), and model version at the top of the screen to restrict your search. By default, you search the entire repository.

Note: Searching the entire repository may take a long time. You should, wherever possible, try to restrict the scope of your search.

In this example I select to search version 4 of the `Project Management (PDM)`:

4. [optional] Select a model type and an object type to search. By default, you search all model types and all object types.

   In this example, I select `Physical Data Model` `Table`:



5. Select a property, operator and value to search against. Depending on the property that you select, the choice of operator may be restricted and/or values may be suggested. Note that the case sensitivity of your searches depends on your DBMS.

   When using the `LIKE` operator, the following special characters allow you to use basic regular expressions for text searches:

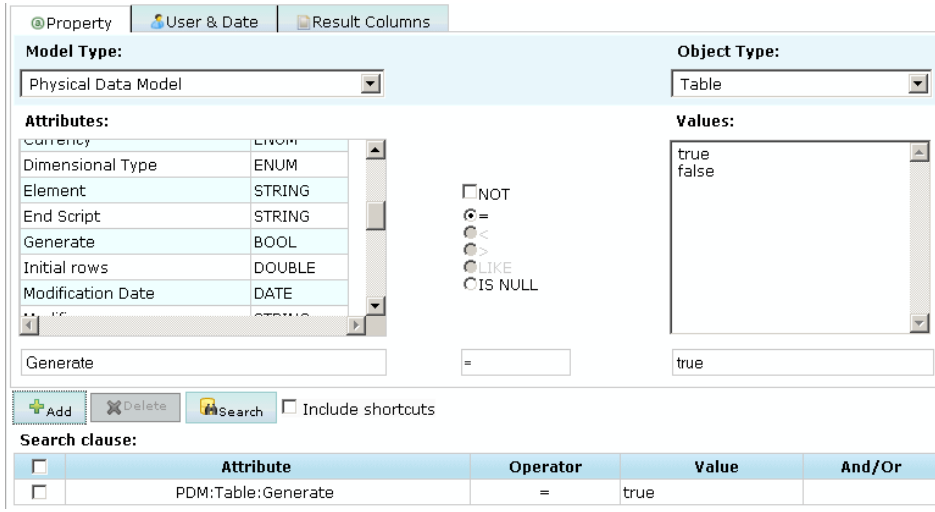
   - * - none to any number of characters

     For example: W* finds "Work" and "Washington"
   - ? – exactly one character

     For example: ????ER finds "Writer" and "Seller" but not "inner"
   - \ - escapes *, ?, or \
   - true/false - Boolean value (True is when the check box is selected in the interface)

6. Click the **Add** button to add the complete search clause to the **Search clause** table.

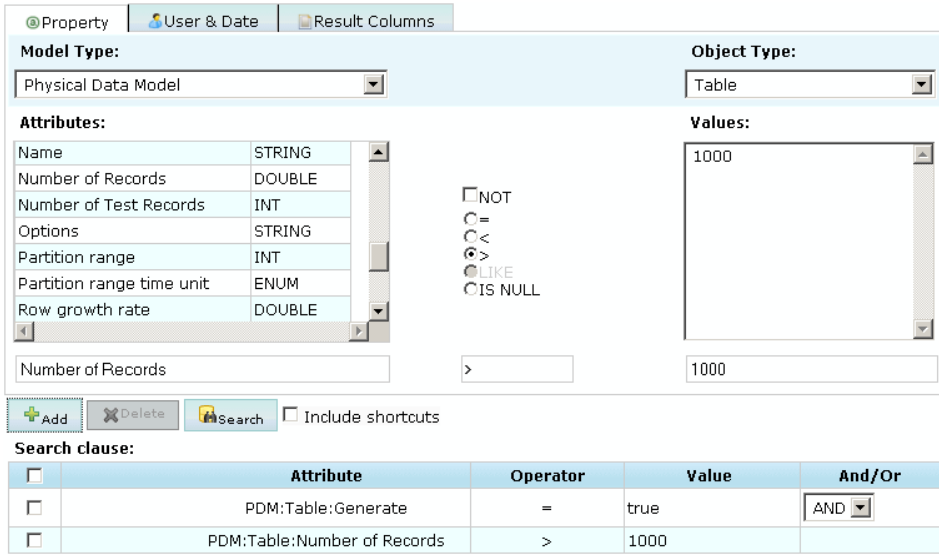
   In this example, I choose to search for tables that are selected for generation:



7. [optional] Create additional search clauses and add them to the **Search clause** table, choosing one of the following conjunctions:

- AND – results must satisfy all clauses
- OR – results must satisfy one of the clauses

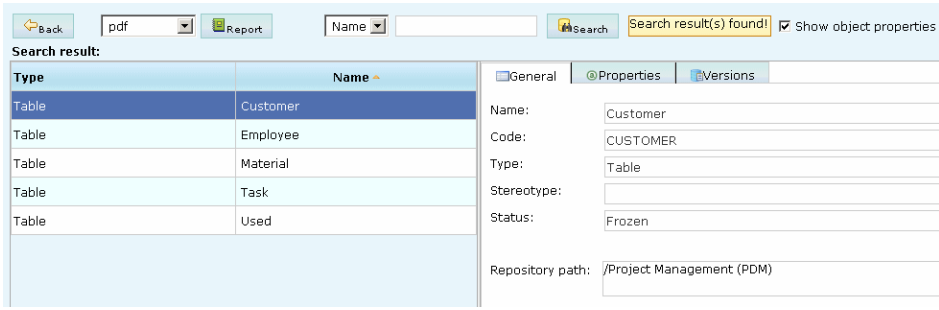
In this example, I refine my search to look only for tables that are estimated to have more than 1,000 records:



8. [optional] Click the **User & Date** tab and specify additional search clauses based on user names and creation and modification dates.
9. [optional] Click the **Result Columns** tab and select the columns that you want to display on the result page. By default, only the object type and name are shown.
10. [optional] Select the **Include Shortcuts** check box to include shortcuts to objects in your search results.
11. Click the **Search** button to start your search.

    Your search results are displayed. Click on an object to display all its available properties.

    In my example, my search returns five tables that meet my criteria:

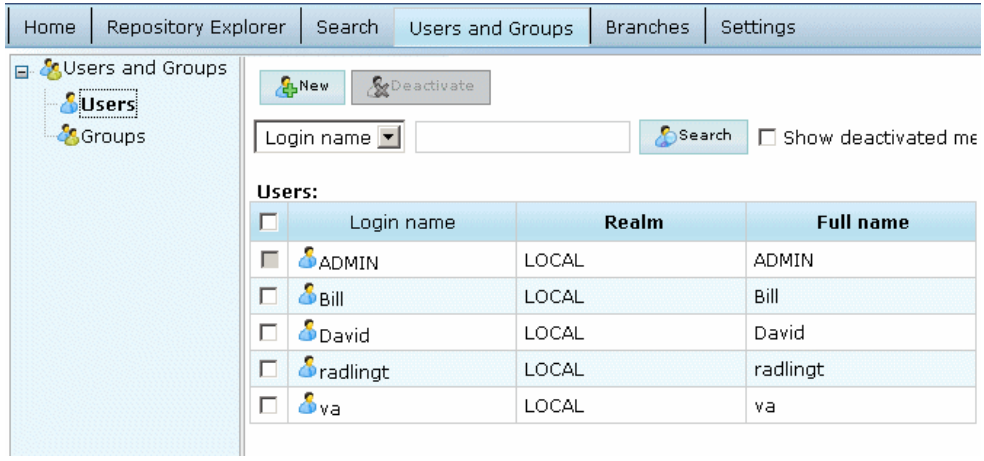


12. [optional] After performing a search, you can:

- Select either pdf or xls and click the **Report** button to generate an Excel or PDF version of your results.
- Toggle between displaying and hiding all the properties of the selected object with the **Show object properties** option.
- Select an attribute, enter a value in the search box at the top of the screen and click the **Search** button to search within your results.
- Click the **Back** button and:
  - Modify your search string.
  - Enter a Saved search name and description and then click the **Save** button to make your search available for reuse on the Search page.

PowerDesigner Portal Users and Groups

Repository users with the Manage users right can view, create, modify, and deactivate users and groups, and manipulate their assignment to groups via the PowerDesigner Portal.



Creating a Repository User

You can create users from the Users and Groups screen.

1. Click the Users and Groups tab and click on Users on the left of the screen to display the list of users.
2. Click the New button, select to create a local or remote user, and enter a login name.

   Note: Remote users and groups are those for whom authentication is delegated to an LDAP server. Note that remote users and groups can be added to local groups, but local users and groups cannot be added to remote groups.

3. [local users only] Enter a full name, password, email, and description for the user. An email address is required if the user is to receive change notifications.

4. [optional] Click the Parent Groups tab and add the user to any appropriate groups (see *Adding a member to a group from the member's property sheet* on page 70)
5. [optional] Click the Rights tab and assign the user any appropriate rights (see *Granting Rights to Users and Groups* on page 69)
6. Click the Save button to create the user and return to the list of users.

Note: When you create a user, make sure you grant him/her access rights on repository documents, either by inserting this user into a group, or directly. If a user connects to the repository without access rights, he cannot see any documents in the browser.

Creating a Repository Group

You can create groups from the Users and Groups screen.

1. Click the Users and Groups tab and click on Groups on the left of the screen to display the list of groups.
2. Click the New button, select to create a local or remote group, and enter a group and full name.
3. [optional] Enter an email address and/or a description for the group.
4. [optional] Click the Members tab and add any appropriate users to the group (see *Adding a member to a group from the group's property sheet* on page 71)
5. [optional] Click the Rights tab and assign the group any appropriate rights (see *Granting Rights to Users and Groups* on page 69)
6. Click the Save button to create the group and return to the list of groups.

Note: You have to grant access rights to a new group in order for user members to inherit these permissions. If a user connects to the repository without access rights, he cannot see any document in the browser.

Granting Rights to Users and Groups

Each repository user and group has a set of *rights* that govern its interactions with the repository. A new user has only the Connect right assigned by default, and a new group has no rights. You can assign rights to users and groups on the Rights tab of their property sheets.

User and group rights can be associated with permissions on documents to define the actions a user or group can effectively perform on a document.

Some rights are automatically associated with permissions for reasons of consistency:

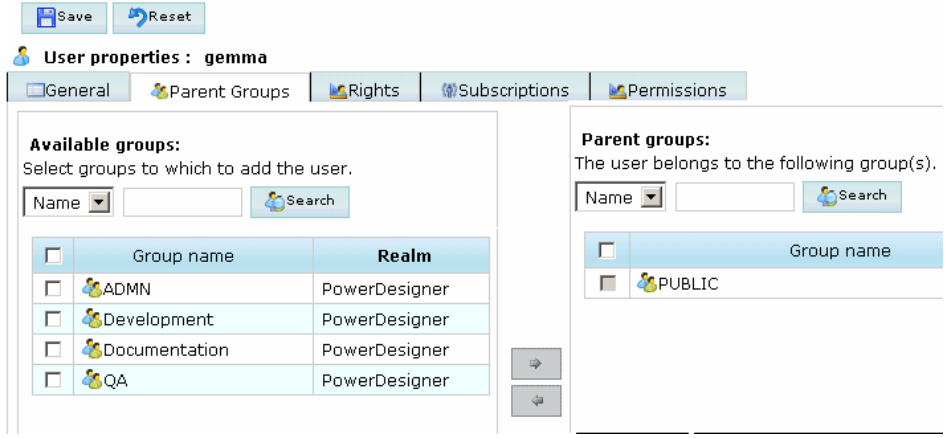
- Lock and Freeze Versions are associated with the Write permission. A user with the Lock or Freeze Versions right must have a Write permission on the repository document.
- A user with the Full permission on a document can unlock or unfreeze a version he did not lock or freeze.

1. Click the Users and Groups tab, select Users or Groups on the left of the screen, and click the appropriate entry in the list to go to its property sheet.
2. Click the Rights tab and select the check boxes corresponding to the rights you want to assign. The following rights are available:
   - Lock – to be able to lock document versions
   - Freeze – to be able to freeze document versions
   - Manage repository - to be able to upgrade a repository and drop the repository database
   - Connect - to be able to connect to the repository
   - Manage users - to be able to create, modify and delete repository users and groups, grant rights to users and groups, and add users or groups to a group
   - Manage configurations – to be able to create, modify,and delete configurations
   - Manage branches - to be able to create, modify, and delete branches
   - Manage all objects - to be able to create, check in, check out, unlock, unfreeze, define permissions and delete any document version.
3. Click the Save button to save your changes and return to the list of users or groups.

Adding a Member to a Group from the Member's Property Sheet

You can add a user or a group to a group from the prospective member's property sheet. When the member is added to the group he will benefit from the rights associated with that group.

1. Click the Users and Groups tab, select Users or Groups on the left of the screen, and click the appropriate entry in the list to go to its property sheet.
2. Click the Groups tab to display the list of available groups and the list of groups to which the user or group belongs:

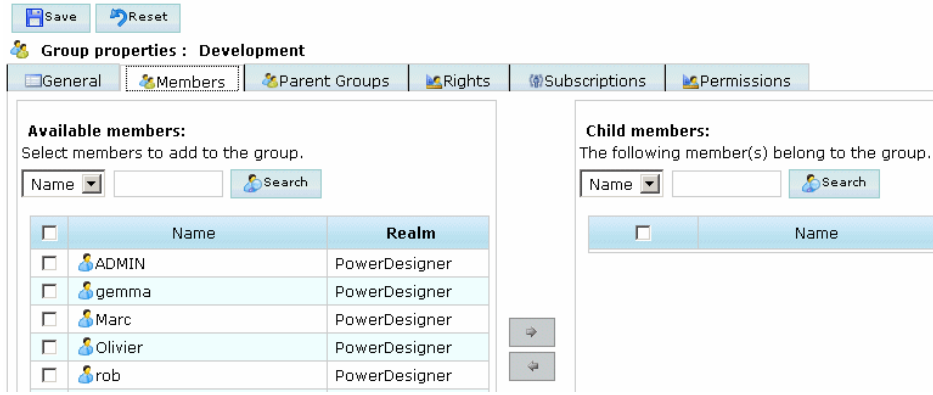


3. Select one or more groups from the Available groups list, and then click the right-pointing arrow to join them and move them to the Parent groups list.
4. Click the Save button to save your changes and return to the list of users or groups.

Adding a Member to a Group from the Group's Property Sheet

You can add a user or a group to a group from the receiving group property sheet. When the member is added to the group he will benefit from the rights associated with that group.

1. Click the Users and Groups tab, select Users or Groups on the left of the screen, and click the appropriate entry in the list to go to its property sheet.
2. Click the Members tab to display the list of available members and the list of child members:



3. Select one or more members from the Available members list, and then click the right-pointing arrow to add them and move them to the Child members list.
4. Click the Save button to save your changes and return to the list of groups.

Deactivating a User

The repository administrator or a user with the Manage Users right can deactivate users, canceling their rights and permissions in the repository. A user cannot deactivate himself.

1. Click the **Users and Groups** tab, and select **Users** on the left of the screen to display the list of users.
2. Select a user in the list, click the **Deactivate** button, and then confirm the deletion by clicking **OK**.

   The user remains in the List of Users but his rights and permissions are canceled. Select the **Show deactivated members** checkbox above the list of users to see users who are no longer active. You cannot add a new user with the same username as a deactivated member.

Deleting a Group

The repository administrator or a user with the Manage Users right can delete groups from the repository. When you delete a group you do not delete the members (either users or groups) of the group.

Deleting a group has the following effects:

- The users who belonged to the group lose the group rights
- All permissions on documents granted by the group and subscriptions via the group are canceled
- The group is removed from the groups to which it belonged
- The group is no longer displayed in the list of groups

1. Click the **Users and Groups** tab, and select **Groups** on the left of the screen to display the list of groups.
2. Select a group in the list, click the **Delete** button, and then confirm the deletion by clicking **OK**.

   The group is removed from the list.

PowerDesigner Portal Branches

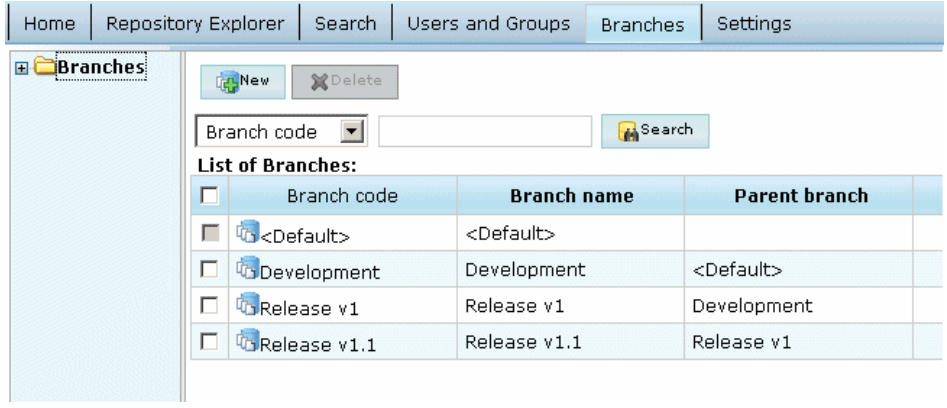
Branches allow you to split the version tree in order to support parallel development on documents.

You can easily switch between branches in the PowerDesigner Portal, view the branch tree, and also create new branches.

Creating a Branch

If you have the Manage Branches right, you can create branches on the Branches screen.

1. Click the Branches tab to display the List of Branches:



2. Click the New button, select the existing branch under which you want to create the branch in the Parent branch list, and enter a branch name and code.
3. Click the Save button to save your changes and return to the list of branches.

Changing Branch

The explorer only shows one branch at a time. You can easily change the branch to view by selecting it in the Branch list at the top left of the Explorer screen.

Deleting a Branch

If you have the Manage Branches right, you can delete branches. However, you cannot delete the current branch, and you cannot delete a branch if it contains any members.

Warning! Deleting a branch cannot be undone.

1. Click the Branches tab to display the list of branches.
2. Select a branch in the list, click the Delete button, and then confirm the deletion by clicking the OK button.

   The branch is removed from the list.

PowerDesigner Portal Settings

The Settings tab lists various parameters that administrators can use to control the server and client components of the PowerDesigner Portal.

PowerDesigner Portal Client Settings

The PowerDesigner Portal provides a number of client settings that you can modify on the **Settings** page.

Logging

The following settings allow you to control logging:

Setting	Description
Log file name	Specifies the name to give to the log file.
Log level	Specifies the minimum level of importance for the messages to be written to the log file.
Rollover by	Specifies how the log file should be rolled over. You can choose between: • Time/Date – the file rolls over after a certain time. Specify the duration and the format of the log file name in the date pattern field below. • File size – the file rolls over when it reaches a certain size. Specify the maximum file size in the field below.

Setting	Description
Truncate the log file after the server starts	Specifies to create a new log file each time the server is restarted.

To view the log file directly in the browser, select the level of message that you want to view and, optionally, a keyword to search on, and then click the Search button. Click the View button to view the file in a separate window.

SSL Connection

The following settings allow you to control SSL connections. You must log out and log back in again for changes to take effect:

Setting	Description
Keystore file name	Specifies the name of the keystore file, a database used by the security provider to store public and private key certificates, used by the security socket connection. A default client.keystore file is provided for the test environment.
KeyStore file path	Specifies the path to the keystore file. The default is CMR_HOME/keystore.
TrustStore password	Specifies the TrustStore password, used to access the keystore file. The default for the test environment file is "changeit". We recommend that you change the TrustStore password and regenerate the certificates and keystore files.

Theme

The following settings allow you to control the look of the site:

Setting	Description
Color theme	Specifies the color scheme used for the site pages.
Logo picture	Specifies the logo image used in the site title bar. Use the **Browse** button to locate an image to use, and the **Upload** button to upload it to the site.
Logo title	Specifies the title used in the site title bar.

Creating a New Color Theme

You can create new color themes for the PowerDesigner Portal. To do so, you will need to be comfortable editing image and css files.

1. Go to *`install_dir`*`\Tomcat\webapps\cmr\theme` and make a copy of one of the existing color theme directories to use as the basis of your new theme.
2. Rename the copied directory to the name that you want to see in the Color theme list on the Settings page.

3. Go to *new_theme_dir*\images\css-images\dgm to view all the component images that are used to construct the PowerDesigner Portal site.
4. Open each of the images in turn in your image editor, make appropriate, consistent changes to their color schemes, and then save them back into the folder.
5. Go to *new_theme_dir*\css and open cmr.css in your text editor to view the css code that controls the look of other elements of the PowerDesigner Portal site.
6. Make appropriate changes to the background-color and color statements and then save your changes.
7. Navigate to the PowerDesigner Portal site in your Web browser and click the Settings tab. Click **Refresh** to ensure that your changes are taken into account by the server.
8. Click **Settings > Client settings > Theme** and select your new color theme from the list.
9. [optional] Select an image to use as your site logo and enter a new logo title for your site.
10. Click **Save** to apply your changes.

PowerDesigner Portal Server Settings

The PowerDesigner Portal provides a number of server settings that you can modify on the **Settings** page.

Database

The following database settings are available. You must restart the PowerDesigner Portal for changes to take effect:

Setting	Description
Database type	Specifies the type of DBMS that hosts the repository.
Database driver	Specifies the driver used to connect to the repository database.
Host	[required] Specifies the name of the host machine for the repository database.
Port	[required] Specifies the port number of the host machine through which the repository database is available.
Database name	Specifies the name of the repository database.
User name	Specifies the database user name that the repository uses to access the DBMS.
Password	Specifies the database password that the repository uses to access the DBMS.
Isolation level	Specifies the isolation level used to isolate transactions in a multi-user environment. By default, level 1 is used for ASA databases and level 2 for ASE. See your DBMS documentation for information about the behavior of each level in your environment.
Initial pool size	Specifies the initial number of connections in the connection pool. The default is 1.

Setting	Description
Min idle	Specifies the minimum connection idle time.
Max idle	Specifies the maximum connection idle time.
Max wait (msec)	Specifies the maximum connection wait time.
Max active	Specifies the maximum permitted number of connections to the database.
Charset	[ASE only] Specifies the character set used by the database.

General

The following general settings are available. You must restart the PowerDesigner Portal for changes to take effect:

Setting	Description
Name	Specifies the name of the repository.
Comment	Describes the repository.
Session timeout (min)	Specifies the amount of time that the browser session is permitted to be idle before it is automatically logged out.
Max users	Specifies the maximum number of users that may connect to the PowerDesigner Portal at any one time.
Object cache size (MB)	Specifies the amount of memory assigned to object caching. The default is 128MB. Increasing this value will improve performance.

LDAP

The following LDAP settings are available. You must restart the PowerDesigner Portal for changes to take effect:

Setting	Description
Provider URL	[required] Specifies the URL for the LDAP provider in the format `ldap://ldapserverhost:port` , or as an IP address.
Security protocol	[optional] Specifies the protocol to be used when connecting to the LDAP server. If you are using SSL (which is the only protocol currently supported), then you should set this parameter to `ssl`. We recommend that you configure LDAP access at first without SSL, and only implement the protocol once you have access working.

Setting	Description
Default search base	[required] Specifies the level at which the query begins its search for users in the LDAP tree. As a minimum this should include the DCs of the LDAP server. For example if your ldap url is `http://ldap.sybase.com` then your DC would be `dc=sybase, dc=com`. Your default search base can include the location of the User directory such as `OU=Users, dc=devpd, dc=local`. The values you enter here affect what you put in the Authentication search base. If you do not put the location of the Users in the default search base then you must include them in the Authentication Search Base.
Server type	Specifies the type of the LDAP server. Selecting a server type sets silent defaults for the authentication and role filters. The following types are available: • none - [recommended] • sunone5 - for SunOne 5.x OR iPlanet 5.x • msad2k - for Microsoft Active Directory, Windows 2000 • nsds4 - for Netscape Directory Server 4.x • openldap - for OpenLDAP Directory Server 2.x Since every LDAP configuration is different and these defaults may not be appropriate for your installation, we recommend that you select `none`.
Trusted server	[required] Specifies that the LDAP server can be trusted.
Anonymous bind	[optional] Specifies that the server supports anonymous access to the LDAP tree. If this parameter is not selected, you must specify a bind DN and password. Note that Active Directory does not support anonymous binding out of the box.
Bind DN	[required unless Anonymous bind is selected] Specifies the LDAP account that has permissions to query the Active Directory. If the Bind DN is in the same DN as the Authentication search base then the BIND DN can be just the user id for the search. Otherwise, you will need the account login and password as well as the full Distinguished Name (DN) for that account. For example If the DefaultSearchBase is `ou=people,dc=Onebridge,dc=qa`, and you have a user `cn=csitest,cn=users,dc=Onebridge,dc=qa`, then the Bind DN cannot just be `csitest`, but must be `cn=csitest,cn=users,dc=Onebridge,dc=qa`.
Bind password	[required unless Anonymous bind is selected] Specifies the password to bind with when building the initial LDAP connection.

Setting	Description
Filter	[required] Specifies the LDAP query that looks up the user information. To determine the LDAP filter you will use, you must know the properties of the users defined in the Active Directory. The property that is being used as the login could be `name`, `samAccountName` or another property. In the following example we use the `samAccountName` as the login (which PowerDesigner captures in the variable `{uid}`: `(&(samAccountName={uid})(objectclass=user))`
Scope	[required] Specifies the scope of the authentication search. You can choose between: • onelevel [default] - only the level specified in the the Search base is searched • subtree - the search begins at the level of the Search base, but also searches any subnodes.
Method	[required] Specifies the method to use for authentication requests. You can choose between: • simple - clear text authentication. • DIGEST-MD5 - hashed password authentication, which requires that the server use plain text password storage.
Digest MD5 format	[required] Specifies the DIGEST-MD5 bind authentication identity format. The default is DN.
Search base	[optional] If the default search base specified in the General group box does not include the location of the User list in your Active Directory, you must specify it here. Users may be in a common node such as `cn=Users` or an organization unit such as `OU=Users`. To determine the correct search base, you should use an LDAP browser to look at the full distinguished name of a user. Note that your Bind DN may be a user in a different node in the tree than general users so it is very important that you have the correct information for each.
Filter	Specifies the role search filter, which, when combined with the search base and scope, returns a complete list of roles within the LDAP server. There are several default values depending on the chosen server type. If the server type is not chosen or this property is not initialized, no roles will be available.
Scope	Specifies the role search scope. You can choose between: • onelevel [default] • subtree

Setting	Description
Referral	Specifies the treatment of referrals. You can choose between: • ignore [default] • follow • throw
Name attribute	Specifies the attribute for retrieved roles that is the common name of the role. If this value is "dn" it is interpreted specially as the entire dn of the role as the role name. The default is "cn", the common name.
Search base	Specifies the role search base.

Logging

For information about the Logging settings, see *Client configuration* on page 73.

Notification

The following change notification settings are available:

Setting	Description
SMTP host	Specifies the host name of the SMTP server used to send mail.
SMTP port	Specifies the port number of the SMTP server used to send mail.
Sender's email address	Specifies the email address from which to send the change notification mails.
Server requires a secure connection	Specifies that the SMTP mail server requires a secure connection.
Server requires authentication	Specifies that the SMTP server requires authentication.
Log on using Secure Password Authentication (SPA)	Specifies to use SPA for authentication.
User name	Specifies a user name for SPA.
Password	Specifies a password for SPA.
Mail template	Specifies the content of the notification emails. Choose an item in the list to specify its content in the field below.

Enabling SSL

This topic provides guidance for setting up SSL security for the PowerDesigner Portal environment. For detailed information, please see your application server's documentation.

Using the Default Tomcat Server

You will need to edit the Tomcat `server.xml` file to set the `keystoreFile` attribute to use the default keystore file for SSL connections and have clients connect using the default `client.keystore` file.

For example, if you have installed the PowerDesigner Portal at `c:\sybase\cmr`:

```
<Connector port="8443" maxHttpHeaderSize="8192"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
  keystoreFile="C:\sybase\cmr\keystore\server.keystore">
</Connector>
```

For more information , see *http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html*

Using an Existing SSL Implementation

If you deploy the PowerDesigner Portal to an existing Tomcat server running SSL, you will need to export the certificate from your existing keystore file by using your keystore tool. If the keystore file contains many certificates, make sure you export the certificate which the Tomcat server uses. For example, using the JDK's keytool:

```
keytool –export –alias tomcat –file
  tomcat.cert –keystore <keystorefile>
```

Once the certificate is exported, you must import it to the `%CMR_HOME%\keystore\client.keystore` file using the JDK's keytool. For example:

```
keytool –import -alias tomcat –file tomcat.cert
  –keystore %CMR_HOME%\keystore\client.keystore
```

The password to the `client.keystore` is `changeit`.

Using an Existing Client Keystore

If you already have a client keystore file to connect to the server and don't want to use the the supplied default client.keystore file, you can either manually edit the `web-inf/config/cmrclient.xml` file or login to the PowerDesigner Portal as an administrator and change the following parameters:

```
<bean class="java.lang.Object" id="cmr.client.config.ssl">
        <param name="keyStoreFile" value="client.keystore"/>
        <param name="keyStoreFilePath" value=""/>
        <param name="keystorePass" value="changeit"/>
        <param name="truststorePass" value="changeit"/>
    </bean>
```

PowerDesigner Portal Server Status Settings

The PowerDesigner Portal provides a number of server status settings that you can modify on the **Settings** page.

Setting	Description
Server Information	Displays the host name, port, version, and status of the server, along with the number of users currently logged in. Click the Restart button to restart the PowerDesigner Portal service.
Change Notification Information	Displays the SMTP host, port, and status of the notification service. Click the Suspend button to suspend the notification service. Click the Resume button to resume the notification service. When resuming, you can optionally select the Include suspended notification items checkbox to send all notifications that would have been sent during the period since the suspension.

Troubleshooting the PowerDesigner Portal

These sections contain information to help you troubleshoot problems with the PowerDesigner Portal.

Unicode Encoding with Sybase SQL Anywhere V10

If you are using a SQL Anywhere® v10 database to store your repository information in a unicode format, you must create the database with the -z option set to an appropriate value.

For example, to create a database to correctly store Japanese characters, you should use the following option:

```
-z 932JPN -zn UTF8BIN
```

For a list of the -z options for all the supported languages, see the SQL Anywhere documentation, available at: *http://sybooks.sybase.com/nav/detail.do?docset=1333*

Increasing Tomcat Memory

By default 256MB of memory is allocated to Tomcat. If you will be browsing large models (20MB or more) or if you anticipate many concurrent users, you may need to increase this amount to 512M or 1GB. See your Tomcat documentation for information about how to modify this allocation.

CHAPTER 4 Repository Administration

The repository administrator is responsible for setting up the repository, and for controlling access to the documents stored within it.

Access can be controlled in the following two ways:

- User and group rights – provide global levels of access, which allow the user to view the repository and have a default level of control over various kinds of objects within it. You can also standardize the provision of user rights by creating groups. See *Controlling Rights for Repository Users and Groups* on page 90
- Document and folder permissions – provide access to individual documents, or to the contents of folders. See *Controlling Permissions for Repository Items* on page 96

Creating, Upgrading, and Dropping the Repository Database

The repository administrator is responsible for creating the repository database, and may need to subsequently upgrade or delete it.

For information about creating a repository, see the Installing the Repository chapter in the *Installation Guide*.

Upgrading the Repository

When you upgrade PowerDesigner, all modules installed on your machine are upgraded, including the repository. You cannot upgrade the repository through a repository proxy connection. You must use a direct ODBC connection instead.

Warning! Always back up your repository database before upgrading it. If you have installed the PowerDesigner Portal, you must stop the PowerDesigner Portal Service before performing an upgrade.

The first time you connect to the repository after upgrading PowerDesigner, you are asked to upgrade the database. You do not have to have installed all the PowerDesigner modules to upgrade the repository database. For example, if you have the Conceptual Data Model installed on your machine, the CDM part of the database is upgraded together with any parts shared by other modules. If you do not have the Object Oriented Model installed on your machine, that part of the database will not be upgraded. The next user of the database that has installed the OOM will upgrade that part of the database upon connection, and so on.

All clients using a particular repository database must upgrade simultaneously in order to continue to connect to it.

If you created your repository database with Microsoft SQL Server or Sybase ASE before PowerDesigner version 15.0, we recommend that you select the **Rebuild clustered indexes** check box in the Repository Update Script dialog box when upgrading your repository. Rebuilding cluster indexes improves repository performance, particularly for document check out.

If you do not select this option at upgrade time, or if you use a direct Microsoft SQL Server or Sybase ASE repository connection, you can select **Repository > Administration > Rebuild Clustered Indexes** at any time to perform the rebuild.

Repository Upgrade Failure

In case of upgrade failure due to network or permission problems, or insufficient disk space, you should restore the repository database from your backup version and retry an upgrade when problems are solved.

If the upgrade fails because of the data in your database, then you should contact technical support for assistance.

Dropping the Repository Tables

The Drop Repository Tables command removes all repository tables and their content from the database and cuts the connection with the database.

Warning! Dropping the repository database cannot be undone. If you have not backed up your models and other contents, they will be lost. You cannot drop the repository tables through a proxy connection. You must perform a direct ODBC connection instead.

1. Select **Repository > Administration > Drop Repository Tables**. You will be prompted to confirm that you want to continue. Click **Yes** to open the Repository Table Drop Script window.

   This window is read-only. You can only review the script, and not edit it. If you want to modify the script you should copy it to a script interpreter, after having selected the appropriate separator for your interpretor from the list.
2. Click **Execute** to begin the deletion.

Troubleshooting the Repository Installed on ASE

This section lists known issues encountered when installing the repository on an Sybase Adaptive Server Enterprise DBMS.

Commonly Required ASE Parameters

The following ASE parameters must frequently be configured higher than their default settings to optimize check in of models:

- `number of open objects`
- `number of open indexes`

- number of locks
- max memory
- procedure cache size

Must declare variable '@dr_t_a0'. SQLSTATE = ZZZZZ
If you see the above error and are connecting to an ASE server via ODBC, open the ODBC Data Source Administrator, select your data source, and select the **Enable Dynamic Prepare** option on the **Connection** tab.

Truncated data on extraction
If you are seeing data truncated on extraction from the repository and are connecting to an ASE server via ODBC, open the ODBC Data Source Administrator, select your data source, and enter the value `10000000` in the **Text size** field on the **Advanced** tab.

If characters such as the ellipsis are being truncated, you should verify that the character set of your driver and database are both set to `utf8`.

Troubleshooting the Repository Installed on DB2

This section lists known issues encountered when installing the repository on an IBM DB2 DBMS.

Exception During Search
If you are seeing exceptions when performing a search through the PowerDesigner Portal, you should consider doubling the size of the `applheapsz` parameter, which controls the application heap size.

Controlling Repository Access with LDAP

A repository administrator can delegate the authentication of repository users to an LDAP server by selecting **Repository > Administration > LDAP Parameters** and entering all the appropriate parameters for her environment.

Once the Repository has been configured to permit access to users authenticated by LDAP, any such user can connect to the repository without further intervention from the repository administrator. The first time that an LDAP user connects to the repository, an account is automatically created for him in the External users and Public groups.

PowerDesigner's Repository LDAP integration provides only authentication. Authorization is still managed via the permissions set within the PowerDesigner Repository environment. Initially, members of the External users group have only the Connect right granted, and members of Public have read access to everything in the repository. The administrator will grant other rights and permissions as appropriate.

You will probably want to retain finer control of write permission on repository documents. To have everything in place before your users connect to the repository, you can manually create

accounts for them and assign permissions for them before they connect (see *Pre-configuring LDAP User Permissions* on page 89).

Note: Before configuring the PowerDesigner Repository for Active Directory authentication you should contact your Active Directory administrator who will provide the information you need to complete the process, and may give you access to a tool such as an LDAP Browser utility.

To enter the LDAP configuration parameters, select **Repository** > **Administration** > **LDAP Parameters** (or right-click the root node, and select **Properties** to open the repository property sheet, and then click the **LDAP** tab).

General LDAP Parameters

Several of the parameters in the General group box are required:

Parameter	Description
Provider URL	[required] Specifies the URL for the LDAP provider in the format `ldap://ldapserverhost:port` , or as an IP address.
Security protocol	[optional] Specifies the protocol to be used when connecting to the LDAP server. If you are using SSL (which is the only protocol currently supported), then you should set this parameter to `ssl`. We recommend that you configure LDAP access at first without SSL, and only implement the protocol once you have access working.
Default search base	[required] Specifies the level at which the query begins its search for users in the LDAP tree. As a minimum this should include the DCs of the LDAP server. For example if your ldap url is `http://ldap.sybase.com` then your DC would be `dc=sybase, dc=com`. Your default search base can include the location of the User directory such as `OU=Users, dc=devpd, dc=local` . The values you enter here affect what you put in the Authentication search base. If you do not put the location of the Users in the default search base then you must include them in the Authentication Search Base.
Trusted server	[required] Specifies that the LDAP server can be trusted.

Parameter	Description
Server type	Specifies the type of the LDAP server. Selecting a server type sets silent defaults for the authentication and role filters. The following types are available: • none - [recommended] • sunone5 - for SunOne 5.x OR iPlanet 5.x • msad2k - for Microsoft Active Directory, Windows 2000 • nsds4 - for Netscape Directory Server 4.x • openldap - for OpenLDAP Directory Server 2.x Since every LDAP configuration is different and these defaults may not be appropriate for your installation, we recommend that you select `none`.
Anonymous bind	[optional] Specifies that the server supports anonymous access to the LDAP tree. If this parameter is not selected, you must specify a bind DN and password. Note that Active Directory does not support anonymous binding out of the box.
Bind DN	[required unless Anonymous bind is selected] Specifies the LDAP account that has permissions to query the Active Directory. If the Bind DN is in the same DN as the Authentication search base then the BIND DN can be just the user id for the search. Otherwise, you will need the account login and password as well as the full Distinguished Name (DN) for that account. For example If the DefaultSearchBase is `ou=people,dc=One-bridge,dc=qa`, and you have a user `cn=csit-est,cn=users,dc=Onebridge,dc=qa`, then the Bind DN cannot just be `csitest`, but must be `cn=csit-est,cn=users,dc=Onebridge,dc=qa`.
Bind password	[required unless Anonymous bind is selected] Specifies the password to bind with when building the initial LDAP connection.

Authentication Parameters

Most of the parameters in the Authentication group box are mandatory:

Parameter	Description
Filter	[required] Specifies the LDAP query that looks up the user information. To determine the LDAP filter you will use, you must know the properties of the users defined in the Active Directory. The property that is being used as the login could be `name`, `samAccountName` or another property. In the following example we use the `samAccountName` as the login (which PowerDesigner captures in the variable `{uid}`: `(&(samAccountName={uid})(objectclass=user))`
Scope	[required] Specifies the scope of the authentication search. You can choose between: • onelevel [default] - only the level specified in the the Search base is searched • subtree - the search begins at the level of the Search base, but also searches any subnodes.
Method	[required] Specifies the method to use for authentication requests. You can choose between: • simple - clear text authentication. • DIGEST-MD5 - hashed password authentication, which requires that the server use plain text password storage.
Digest MD5 format	[required] Specifies the DIGEST-MD5 bind authentication identity format. The default is DN.
Search base	[optional] If the default search base specified in the General group box does not include the location of the User list in your Active Directory, you must specify it here. Users may be in a common node such as `cn=Users` or an organization unit such as `OU=Users`. To determine the correct search base, you should use an LDAP browser to look at the full distinguished name of a user. Note that your Bind DN may be a user in a different node in the tree than general users so it is very important that you have the correct information for each.

Role Parameters

PowerDesigner does not currently support role-based authentication, and so any values you enter in the Role group box will not be taken into account:

Parameter	Description
Filter	Specifies the role search filter, which, when combined with the search base and scope, returns a complete list of roles within the LDAP server. There are several default values depending on the chosen server type. If the server type is not chosen or this property is not initialized, no roles will be available.
Scope	Specifies the role search scope. You can choose between: • onelevel [default] • subtree
Referral	Specifies the treatment of referrals. You can choose between: • ignore [default] • follow • throw
Name attribute	Specifies the attribute for retrieved roles that is the common name of the role. If this value is "dn" it is interpreted specially as the entire dn of the role as the role name. The default is "cn", the common name.
Search base	Specifies the role search base.

Pre-configuring LDAP User Permissions

You can manually create accounts for LDAP-authorized users and assign permissions for them, so that everything is in place for them before they connect for the first time

If LDAP-authenticated users have already connected to the repository, then they will have automatically created accounts that you can assign directly to groups (see *Inserting a User into a Group* on page 94).

1. Select **Repository > Administration > Users** to open the List of Users.
2. Click the Add a row tool, and type an appropriate name for the user in the Full Name column.
3. Enter the user's LDAP login in the Login Name field, and select the checkbox in the X column to specify that he is an externally-validated user.
4. Click the Properties tool to open the property sheet of the user and then click the Groups tab to add him to any appropriate groups (see *Inserting a User into a Group* on page 94).

Controlling Rights for Repository Users and Groups

The repository is designed to be used by different team members. Among the team members, a repository administrator has to manage the repository users.

The repository administrator is responsible for:

- Creating users and groups of users who will use the repository
- Managing the rights of the different users and groups in the repository

Note: The repository has separate groups and users from the DBMS

Groups are used to assign common rights to users in the repository. You can create a hierarchy of groups by inserting sub-groups into a group.

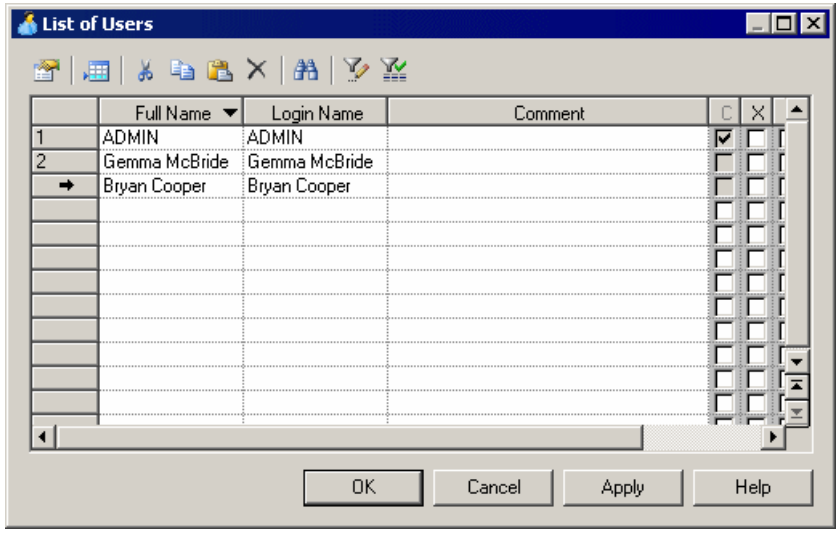
When you install the PowerDesigner repository, the following groups are created by default:

- Administrators, [ADMN] - has administrative rights over the repository.
- All users [PUBLIC] - has Connect rights. All users belong to this group.
- External users [EXTERNAL] - has Connect rights. Remote LDAP users are automatically added to this group when they connect for the first time.

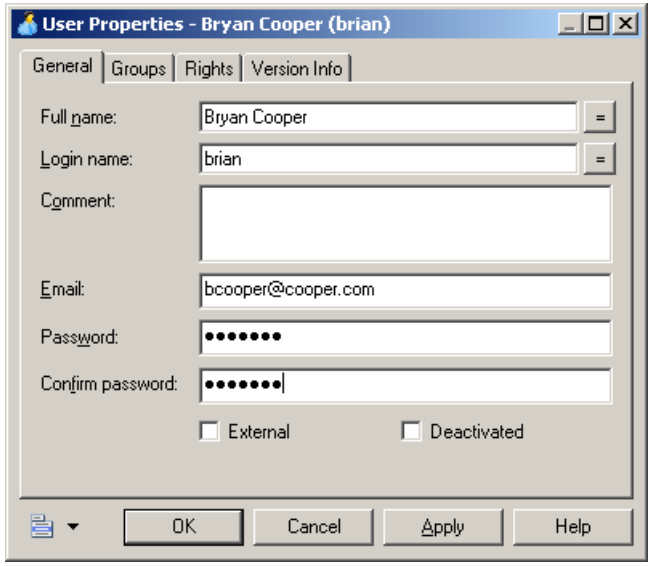
Creating a Repository User

The repository administrator is responsible for creating user accounts to enable users to connect to the repository and access the contents that they need.

1. Select **Repository** > **Administration** > **Users** to open the List of Users.



2. Click the **Add a row** tool, and type a name in the **Full Name** column.
3. Click the **Properties** button to open the property sheet of the user:



4. Enter the user login name and any other properties. The password is blank by default, and it is not necessary to set one at this time, but the password fields display eight stars.
5. Click OK.

Each repository user has the following properties:

Property	Description
Full Name	Specifies the real name of the user.
Login Name	Specifies the name used for logging in to the repository.
Comment	Specifies any additional information about the user
Email	Specifies the email address of the user
Password/ Confirm Password	Specifies the password of the user.
External	Specifies that authentication of the user is handled by an LDAP server (see *Controlling Repository Access with LDAP* on page 85). Selecting this check box disables the **Full name**, **Email**, and **Password** fields.
Deactivated	Specifies that the user is no longer active in the repository. The user remains in the List of Users and can be reactivated at any time. While he is deactivated, the user may not connect to the repository.

Note: When you create a user, make sure you grant him/her access rights on the repository document either by inserting her into a group (see *Inserting a user into a group* on page 94), or directly (see *Specifying a user's rights* on page 92). If a user connects to the repository without access rights, he cannot see any document in the browser.

Creating a Group

The repository administrator is responsible for creating the different groups of users in the repository.

1. Select **Repository > Administration > Groups** to open the List of Groups.
2. Click the **Add a row** tool and type a name in the Name column.
3. Click the **Properties** tool to open the property sheet of the new group.
4. Complete the appropriate properties and then click **OK.**

Each PowerDesigner repository group has the following general properties:

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Additional information about the group

Note: You have to grant access rights to a new group in order for user members to inherit these permissions (see *Specifying Group Rights* on page 94). If a user connects to the repository without access rights, he cannot see any document in the browser.

Specifying User Rights

A new user has no rights assigned by default and belongs only to the `PUBLIC` group, which has no rights. In order for the user to be able to do anything, the repository administrator must assign rights to the user. The same types of rights can be assigned to groups and users.

The user rights can be combined to create different kinds of user profiles. The following table shows typical project profiles. Note the difference between data administrator who manages data in the repository, and repository administrator who manages the repository and the users.

User profile	Required user rights
Database Administrator	Connect, Manage Repository, and Manage Users
Data Administrator	Connect and Manage All Documents

User profile	Required user rights
Team Leader	Connect, Lock Versions, Freeze Versions, Manage Branches, and Manage Configurations.
Designer	Connect, Lock Versions, and Freeze Versions

User rights can be associated with permissions on documents to define the actions a user can effectively perform on a document.

Some user rights are automatically associated with permissions for consistency reasons:

- Lock and Freeze Versions are associated with the Write permission. A user with the Lock or Freeze Versions right must have a Write permission on the repository document.
- A user with the Full permission on a document can unlock or unfreeze a version he did not lock or freeze.

For more information on document permissions, see *Controlling Permissions for Repository Items* on page 96.

1. Select **Repository** > **Administration** > **Users** to open the List of Users, select the appropriate user in the list, and click the **Properties** tool.
2. Click the **Rights** tab and select the check boxes corresponding to the rights you want to assign. The following rights are available:
   - Connect - User can connect to the repository.
   - Freeze Versions - User can freeze document versions. See *Document Versions* on page 40.
   - Lock Versions - User can lock document versions. See *Document Locks* on page 42.
   - Manage Branches - User can create, modify, and delete branches. See *Branching Version Trees* on page 48.
   - Manage Configurations - User can create, modify, and delete configurations. See *Grouping Document Versions in a Configuration* on page 44.
   - Manage All Documents - User can create, check in, check out, unlock, unfreeze, define permissions and delete any document version.
   - Manage Users - User can create, modify and delete repository users and groups, grant rights to users and groups, and add users or groups to a group.
   - Manage Repository - User can upgrade a repository and drop the repository database.

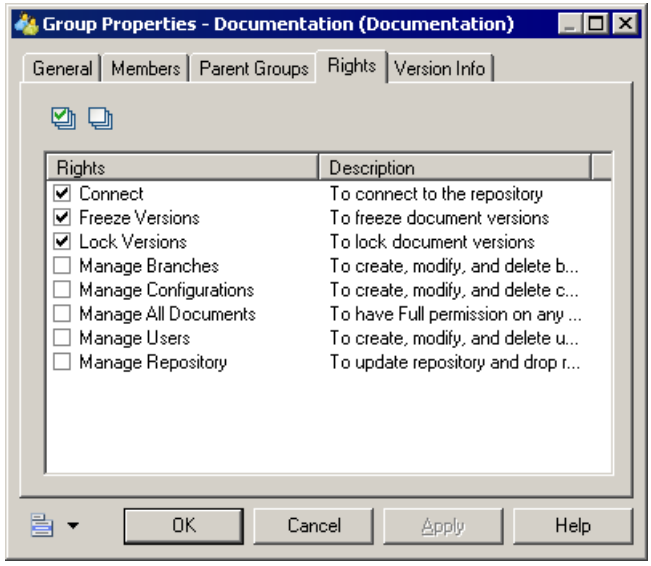
   Note: Users who have the `Manage All Documents` right (typically data administrators), are implicitly granted `Full` permission on all repository documents. Such users can check in, freeze, lock, and even delete documents for which they have been explicitly granted only Read permission.
3. Click **OK** to save you changes and close the use property sheet.

Specifying Group Rights

When you create a group, you have to define its rights, so that it can use the repository. The same kinds of rights can be given to groups and users.

Note: The rights a user or group receive when they become members of a group are added to their existing rights. For example, a user with the `Manage Branches` right will not lose this right when he becomes a member of a group that has only the `Freeze Versions` and `Lock Versions` rights.

1. Select **Repository** > **Administration** > **Groups** to open the List of Groups.
2. Select a group and click the **Properties** tool to display its property sheet.
3. Click the **Rights** tab to display the list of rights that the group has.
4. Select the check boxes corresponding to the rights you want to assign (for information about the various rights, see *Specifying User Rights* on page 92).



5. Click **OK** to close the property sheet.

Inserting a User into a Group

You can define groups of users in order to grant them rights and permissions on documents more easily. Each new user is automatically inserted into the PUBLIC group.

For more information about creating a group, see *Creating a group* on page 92.

1. Select **Repository** > **Administration** > **Users** to open the List of Users, select the appropriate user in the list, and then click the **Properties** tool.

2. Click the **Groups** tab to display the list of groups to which the user belongs.
3. [optional] Click the **Show All Parent Groups** tool to show all the parent groups of the selected group in the list.
4. Click the **Add Groups** tool to open a selection dialog listing all the available groups.
5. Select one or more groups from the list, and then click **OK** to add the user to them.

   The selected groups are added to the Groups tab of the user's property sheet.

Adding Users and Groups to a Group

You can add users and groups as members of a group on the **Members**tab of the group's property sheet.

For information about how to insert a user into a group from the user's property sheet, see *Inserting a user into a group* on page 94.

1. Select **Repository > Administration > Groups** to open the List of Groups, select the appropriate group, and click the **Properties** tool.
2. Click the **Members** tab to display the lists of users and groups who belong to the group.
3. [optional] Click the **Show All Parent Groups** tool to show all the parent groups of the selected group in the list.
4. Click the **Add** tool to open a selection box, which lists available users and groups on tabbed pages
5. Select one or more users and/or groups and click **OK** to add them to the group.

   The selected users and groups are added as members of the group and are listed on the **Members** tab of the group property sheet.
6. Click **OK** to close the group property sheet.

Inserting a Group into Another Group

You can insert a group into another group. For example, in a development team you have created several groups (Designers, Quality Assurance, Documentation). You can insert these groups into a larger group called R&D to which you can define special rights.

1. In the property sheet of the group, click the **Parent Groups** tab.
2. [optional] Click the **Show All Parent Groups** tool to show all the parent groups of the selected group in the list.
3. Click the **Add to Group** tool to open a selection box listing all the available groups.
4. Select one or more groups and then click **OK** to add the group to them.

Deactivating or Deleting Repository Users

The repository administrator or a user with the Manage Users right can deactivate or delete users.

Warning! A user cannot delete or deactivate himself, even if he has the Manage Users right. Deleting a user cannot be undone.

1. Select **Repository > Administration > Users** to display the List of Users dialog box.
2. [to deactivate the user] Select the user in the list and then select the [D]eactivate check box.

   The user remains in the list and can be reactivated at any time. While he is inactive, the user may not connect to the repository.
3. [to delete the user] Select the user in the list and click the **Delete** tool.

   If the user has created or modified documents, he is deactivated. Otherwise he is removed from the List of Users and completely removed from the repository.

Deleting a Group

When you delete a group from the repository you do not delete the members (either users or groups) of the group.

Deleting a group has the following effects:

- The users who belong to the group lose the group rights
- All permissions on documents granted by the group are canceled
- The group is removed from the groups it belongs
- The group is no longer displayed in the list of groups

1. Select **Repository > Administration > Groups** to display the List of Groups dialog box.
2. Select a group in the list and click the **Delete** tool.

   The group disappears from the list.
3. Click OK.

Controlling Permissions for Repository Items

Permissions are access rights granted to users or groups on the repository, folders, documents, and model packages.

Depending on the type of work being performed, users need different access permissions on the different documents contained in the repository. The permissions are to be combined with the rights granted to users or groups (see *Controlling Rights for Repository Users and Groups* on page 90).

You can grant access permissions to a user or a group on the following items:

- The repository root – these permissions apply to the entire contents of the repository.
- Folders – these permissions apply to the contents of the folder.
- Documents - both PowerDesigner models and external application files.
- Model Packages inside PowerDesigner models.

Note: You cannot grant permissions on individual PowerDesigner diagrams or model objects.

When you create a folder or add a document in the repository, the item has default access permissions:

- The user who created the item in the repository has Full access permission. This permission can be removed by an administrator.
- Other users or groups have no permissions except the permissions granted to the folders they belong to. If they do not have any permission, they cannot see the folder in the repository browser

When you define access permissions on a folder or the repository root, the permissions apply to all the items it contains. You can subsequently modify the permissions for each sub-item.

The same process applies to grant access permissions on the root, a folder, a document, or a package.

Permissions for Check in

To check in documents in the repository you need the following permissions:

Check in	Permission required
First check in: creating a document in the repository	Write on the target location (folder or root)
Update: updating an existing document in the repository	Read on the target location and Write on the document (including, for PowerDesigner models, all impacted packages)

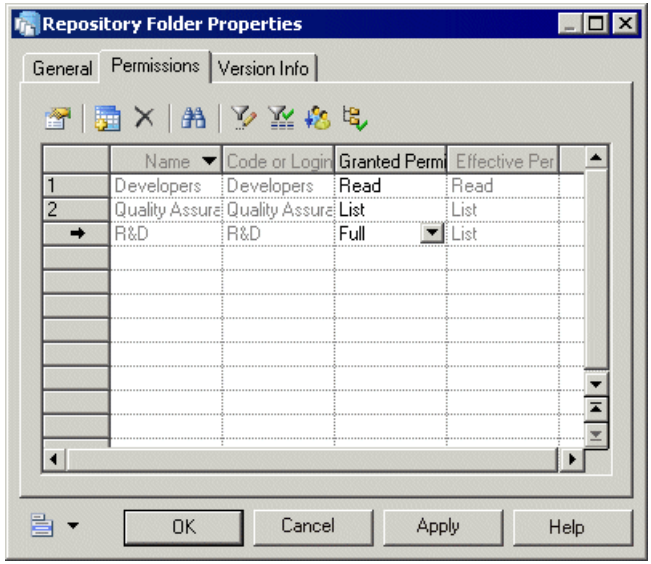
Defining Access Permissions on a Repository Item

To define access permissions on a repository item:

1. Right-click the item in the Repository Browser, and select Permissions from the contextual menu to open the item's property sheet at the Permissions tab.
2. Click the Add tool to open a selection box listing all the available users and groups.
3. Select one or more users and groups, and then click OK to add them to the item's Permissions list.
4. For each user or group, select the access permission from the drop down list box in the Granted Permission column. The following permissions are available:

- List - User or group has minimal permissions to view the item in the browser; display read-only property sheets and search for models.
- Read - User or group has all the List permissions, and can also compare documents, and check the document (or folder contents) out from the repository.
- Write - User or group has all the Read permissions, and can also check in, freeze and lock document versions.
- Full - User or group has all the Write permissions, and can also manage permissions granted to users or groups and remove locks on documents.

Note that users who do not have any permissions on a document or folder cannot even see these items in the browser.



5. Click OK to save your changes.

Understanding Granted and Effective Permissions

You can grant an access permission to a group or to a user. When you grant permissions to a group, the users belonging to this group inherit the group permissions.

In a property sheet, the list of permissions displays two columns:

- The Granted Permission column is used to grant permissions to groups and users using the permission dropdown list box
- The Effective Permission column displays the effective permissions of groups and users, it is not editable

By default, group members do not appear in the list of permissions, this is called the non-inherited display mode. However, you can display these users by clicking the Show All Authorized Users tool, this is called the inherited mode.

The behavior of the Permissions lists changes according to the display mode.

Non-inherited Mode
This is the default display mode, it displays the permissions actually granted to users or groups.

- Members groups are not displayed
- The permission <none> is not used
- The granted permission of a user or a group is always equal to the effective permission
- Deleting a group or a user is immediately applied

Inherited Mode

- Members of selected groups are displayed
- The Granted Permission column displays <none> for these users until you define a permission using the list
- The Effective Permission column displays the actual permission of the groups and users; in this column the most extensive permission selected among the group or the user granted permissions appear, as shown in the following example:

Group permission	User (from group) permission	Effective permission of user
List	Write	Write
Full	Read	Full

- When you delete a group and before you select the Apply button, the group remains visible in the list together with its users, the Granted Permission column displays <none>. When you apply the changes, the group and its members disappear from the list of permissions.

Copying Permissions to Child Members

When you add a folder or a document to a parent folder, access permissions automatically apply to the new child members of the folder.

If you subsequently modify the permissions of the parent, the children keep their original permissions and vice-versa. For example, you can modify the permissions on a package without affecting the permissions of its parent model.

If you want to restore the permissions of a parent over its child members, you have to use the Copy Permissions to All Children tool in the Permissions tab of the following property sheets:

Property sheet	Impact
Root	Copies permissions to all folders, documents, models and packages in the repository
Folder	Copies permissions to the members of the folder

Property sheet	Impact
PowerDesigner model	Copies permissions to model packages
PowerDesigner package	Copies permissions to sub-packages

Copying Parent Permissions to Child Members

To copy parent permissions to child members:

1. Right-click a parent node in the repository browser.
2. Select Permissions from the folder contextual menu.

   The node property sheet opens to the Permissions tab.

3. Click the Copy Permissions to All Children tool.
4. Click Apply.

   The permissions are copied to all child members, and the tool Copy Permissions to All Children tool is no longer pressed in the toolbar.

Permissions Examples

These examples illustrate the use of access permissions in the repository.

Example 1: Folders and Documents

The Y2K folder includes two sub-folders: Data and Specs. The Data sub-folder includes two models: Firstdraft (PDM) and Classes (OOM) models. The Specs sub-folder includes Overview.doc.

The following table lists the permissions assigned to the groups and users working on the project:

Folder	Project leader	Development leader	Developer	Design leader	Designer
Y2K	Full	Read	Read	Read	Read
Data	Full	Write	Read	Read	Read
Documents in Data	Full	Read	Write	Read	Read

Folder	Project leader	Development leader	Developer	Design leader	Designer
Specs	Full	Read	Read	Write	Read
Documents in Specs	Full	Read	Read	Read	Write

The following table lists the rights granted to the groups and users:

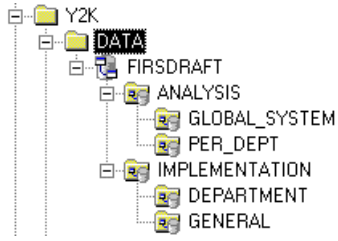
Groups or users	Right
Repository and Data Administrator	Connect, Manage All Documents, Manage Users, and Manage Repository
Project leader	Connect, Freeze Versions, Lock Versions, Manage Branches, and Manage Configurations
Developer team leader	Connect, Freeze Versions, and Lock Versions
Developers	Connect, Freeze Versions, and Lock Versions
Designers team leader	Connect, Freeze Versions, and Lock Versions
Designers	Connect, Freeze Versions, and Lock Versions

Note: Users who have the Manage All Documents right (typically data administrators), are implicitly granted Full permission on all repository documents. Such users can check in, freeze, lock, and even delete documents for which they have been explicitly granted only Read permission.

Example 2: Packages

The model FIRSTDRAFT is divided into 2 parent packages: ANALYSIS and IMPLEMENTATION, they correspond to the different tasks the group of developers has to perform.

Each parent package is divided in two packages, each package being under the responsibility of a developer.



The following permissions are assigned to the developers:

Package	Dev. leader	Dev. 1	Dev. 2	Dev. 3	Dev. 4
Analysis	Full	Read	Read	Read	Read
Global_ System	Full	Write	Read	Read	Read
Per_Dept	Full	Read	Write	Read	Read
Implementation	Full	Read	Read	Read	Read
Department	Full	Read	Read	Write	Read
General	Full	Read	Read	Read	Write

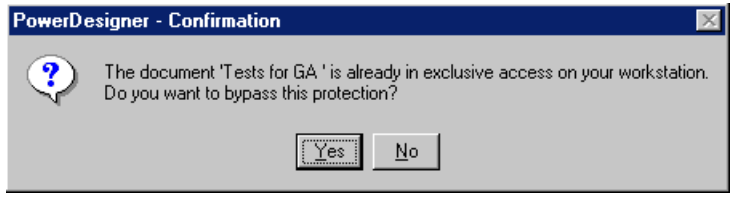
Temporary Protection for Updates to Repository Documents

When you check in, freeze, lock or delete a document in the repository, a temporary protection is applied to all the versions of the document to ensure the update is completed correctly, and an error message given if other updates are attempted.

For example, you cannot lock a version that another team member is currently checking in, or freeze a version that another team member is currently locking. This protection is released once the update is complete.

If your connection to the repository is interrupted during an update to a document, this temporary protection may remain on the document, and the next time you try to update the document, a message box is displayed to inform you that the document is already in exclusive access. To continue, you have to bypass the protection or cancel the action.

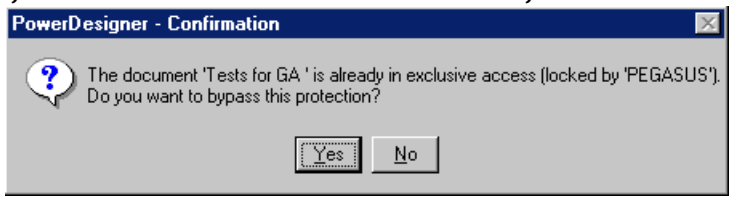
If the Document Was Protected and Blocked from Your Workstation



PowerDesigner verifies the machine name, not the connected user.

PowerDesigner	What you should do
Two sessions running on your workstation: for example, you are checking in a document in one session, and you want to lock a version of the same document from another PowerDesigner session	You must not bypass the temporary protection and wait for the check in to end
One session crashed when you were checking a document in	You can bypass the temporary protection to continue the operation

If the Document Was Protected and Blocked from Another Machine on the Network



You must ask a data administrator to unblock it. The data administrator should check that no other users are currently updating the document and then perform an action on the document (lock for example) to display the message box and click Yes to release the document.

Warning! Bypassing the temporary protection when you are not entitled to perform such a task may severely damage your document.

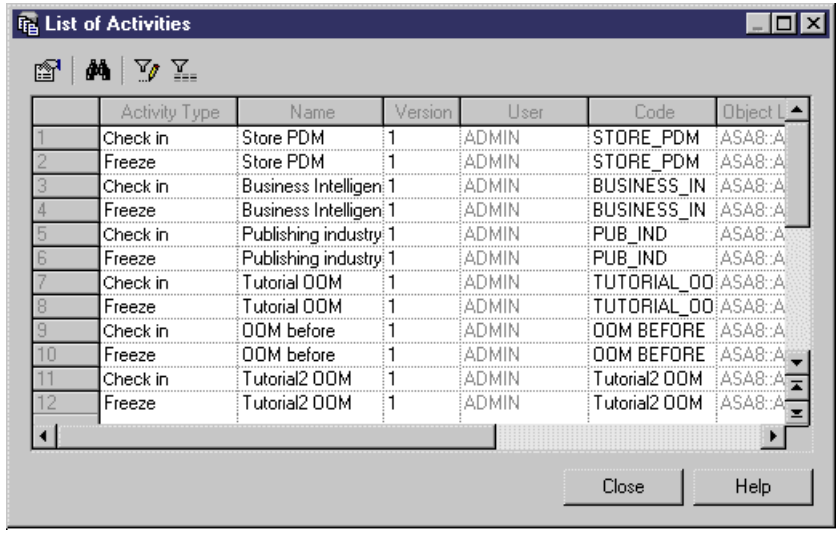
Auditing Repository Activities

Users with the Manage All Documents right can audit repository use through the List of Activities. Activities are actions that modify repository documents, such as check in, freezing, and deleting.

The list makes it possible to audit operations performed on repository documents, analyze user behavior patterns, and highlight activity sequences.

Consulting the List of Activities

To consult the list of activities:
Press Ctrl+Alt+V or select **Repository** > **Activities** to open the List of Activities.



Note: Select an item in the list and click the Properties tool to open the property sheet of the affected object. If necessary, you can use the Versions tab of the property sheet to check out or lock the document version.

The list displays all the operations that modified the documents stored in the repository, this list does not show object changes. You can customize the list by clicking the Customize Columns and Filter tool. The following information can be useful if you want to consult repository activities:

Column	Description
Activity type	Specifies the type of action performed on the document, such as check in, freeze or delete. Note that actions, such as lock or unlock, that do not modify the document are not reported in the list of activities.
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Location	Folder in which the document is stored
Version	Document version number
Comment	Activity comment
Activity Date	Date when operation was performed
User	Login of the user who performed operation

Index

A

B

C

U

V

W