

SYBASE®

Working with the Repository
PowerDesigner® 15.1

Windows

DOCUMENT ID: DC38090-01-1510-01

LAST REVISED: June 2009

Copyright © 2009 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. A ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

Getting Started with the Repository	1
Setting up PowerDesigner to Access the Repository	2
Setting up PowerDesigner for Direct Access	2
Setting up PowerDesigner for Access Via the Proxy	4
Connecting to a Repository	5
Connecting Automatically to the Default Repository	6
Changing Your Repository Password	7
Closing a Repository Connection	7
Working with Repository Documents	9
Checking the State of a Document	9
Adding Documents to the Repository	10
Adding a Project to the Repository	10
Adding a Single Model or File to the Repository	10
Adding Multiple Models and Files to the Repository	11
Checking in Changes to Documents Stored in the Repository	11
Checking in Changes to a Project	11
Checking in Changes to a Single Model or File	11
Checking in Changes to Multiple Models and Files	11
Using the Check In Project Window	11
Using the Check In Document Window	12
Check in Parameters	13
Using the Check In Multiple Documents Window	15
Using the Check In Options Window	16
Using the Check In Model Window	17
Check in Modes	18
Checking in Packages	20
Checking in External Application Documents	20
Undoing a Check in	20
Checking Documents Out From the Repository	20
Checking Out a Project not Present in Your Local Workspace	21
Updating a Project Present in Your Local Workspace	21
Checking Out a Model or File not Present in Your Local Workspace	21
Updating a Model or File Present in Your Local Workspace	21
Using the Check Out Project Window	21
Using the Check Out Document Window	22
Check Out Parameters	23
Using the Check Out Multiple Documents Window	24
Using the Check Out Model Window	25
Checking Out Packages	26
Browsing the Repository	26
Repository Root Properties	27

Repository Folders	27
Repository Document and Object Properties	28
Finding Objects in PowerDesigner Models	30
Comparing Models in the Repository	32
Impact Analysis	33
Querying the Repository Using SQL	35
Repository Options	35
Document Versions	36
Freezing Document Versions	37
Unfreezing Document Versions	37
Deleting Document Versions	38
Document Locks	38
Locking Documents	38
Managing Document Locks in the List of Locks	39
Unlocking Documents	40
Repository Branches	40
Creating a Branch	41
Changing Branch	42
Checking a Document into a Branch	43
Managing Branch Members	43
Deleting a Branch	44
Repository Configurations	44
Creating a Configuration	45
Managing Configuration Members	46
Checking Out a Configuration	47
Checking Out a Project Configuration	47
Sharing Resources in the Repository	47
Preparing to Share Resources in the Repository	47
Working with Shared Resources	51
Browsing the Repository Via the Web	53
Logging into the Repository Web Browser	53
Repository Explorer	54
Repository Document and Object Properties	54
Setting Permissions	55
Subscribing to Change Notifications	55
Comparing Model and Object Versions	55
Repository Web Browser Search	56
Repository Web Browser Users and Groups	58
Creating a Repository User	58
Creating a Repository Group	58
Granting Rights to Users and Groups	59
Adding a Member to a Group from the Member's Property Sheet	59
Adding a Member to a Group from the Group's Property Sheet	60
Deleting a User	61
Deleting a Group	61

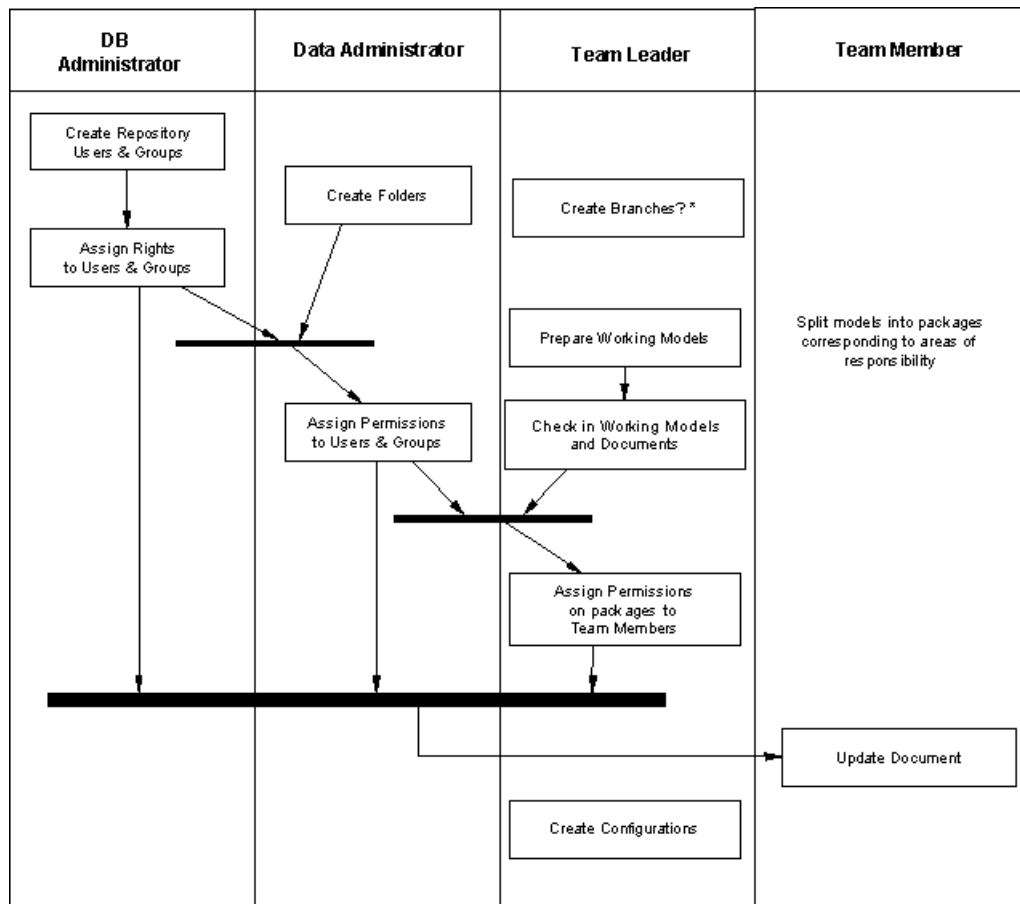
Repository Web Browser Branches	61
Creating a Branch	61
Changing Branch	62
Deleting a Branch	62
Repository Web Browser Settings	62
Client Settings	63
Server Settings	64
Server Status Settings	66
Troubleshooting the Repository Web Browser	67
Unicode Encoding with Sybase SQL Anywhere V10	67
Increasing Tomcat Memory	67
Repository Administration	69
Creating, Upgrading, and Dropping the Repository Database	69
Upgrading the Repository	69
Dropping the Repository Tables	69
Controlling Repository Access with LDAP	70
Pre-configuring LDAP User Permissions	71
Controlling Rights for Repository Users and Groups	72
Creating a Repository User	72
Creating a Group	73
Specifying a User's Rights	74
Specifying a Group's Rights	75
Inserting a User into a Group	76
Adding a User or a Group to a Group	77
Inserting the Current Group into a Group	78
Deleting a Repository User	78
Deleting a Group	78
Controlling Permissions for Repository Items	79
Defining Access Permissions on a Repository Item	79
Understanding Granted and Effective Permissions	80
Copying Permissions to Child Members	81
Permissions Examples	81
Temporary Protection for Updates to Repository Documents	83
Auditing Repository Activities	84
Consulting the List of Activities	84

Getting Started with the Repository

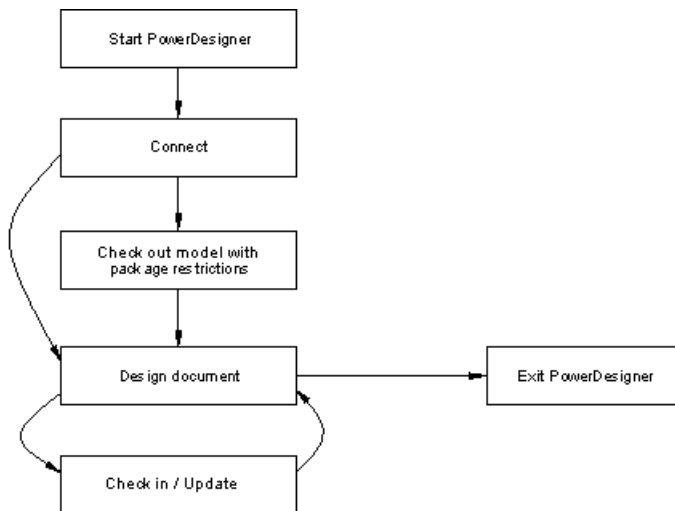
The PowerDesigner® repository is a tool for storing versioned documents. A *document* can be a model, a multi-model report, or any other kind of file.

The repository is installed as a database on a server, and is accessed by PowerDesigner users from their client workstations either directly through an ODBC or other connection, or via the repository proxy.

Different users will interact with the repository in different ways:

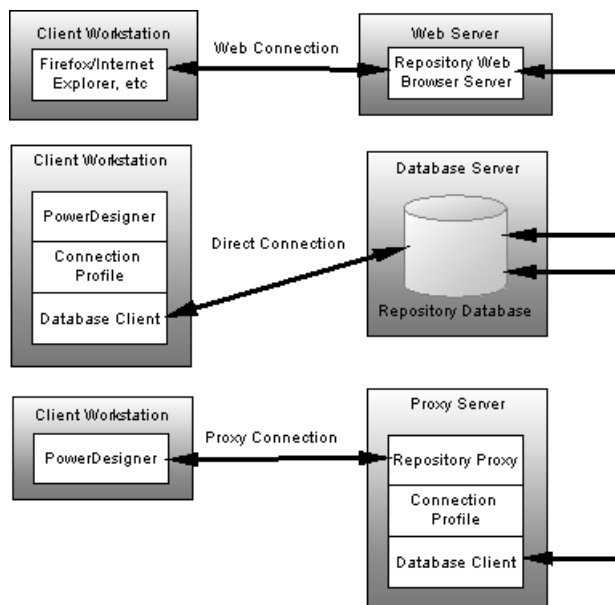


The "Update Document" item in the diagram above is the most common interaction with the repository, and can be broken down as follows:



Setting up PowerDesigner to Access the Repository

The following diagram illustrates the various ways in which you can access the repository:



These methods are further explained in the following sections:

- Web connection [if the repository web browser is installed] – see [Browsing the Repository Via the Web](#) on page 53
- Direct connection – see [Setting up PowerDesigner for direct access](#) on page 2
- Proxy connection [if the repository proxy is installed] - see [Setting up PowerDesigner for access via the proxy](#) on page 4

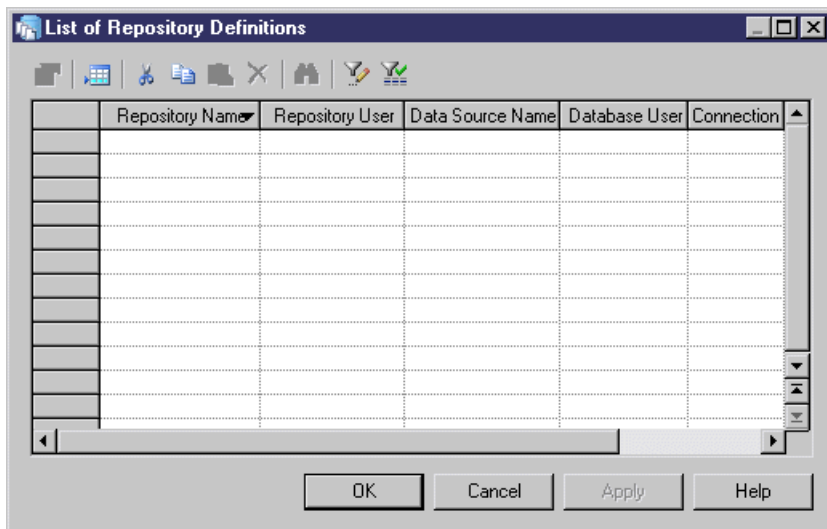
Setting up PowerDesigner for Direct Access

You create a direct repository definition when you are not connecting to the repository via the repository proxy.

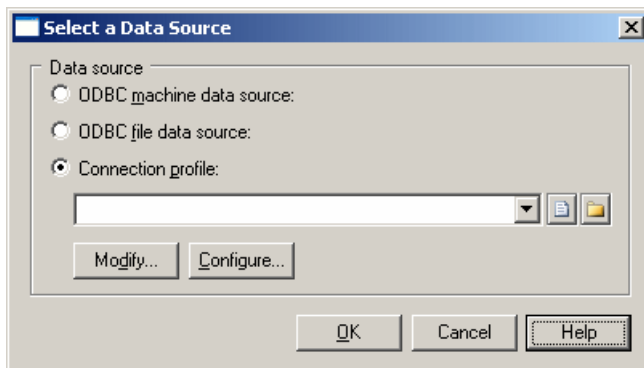
If you are using the proxy, see [Setting up PowerDesigner for access via the proxy](#) on page 4.

Warning! In order 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.

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. When you create the first repository definition that will be used to initialize the repository (see *Repository Administration* on page 69), 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 box:



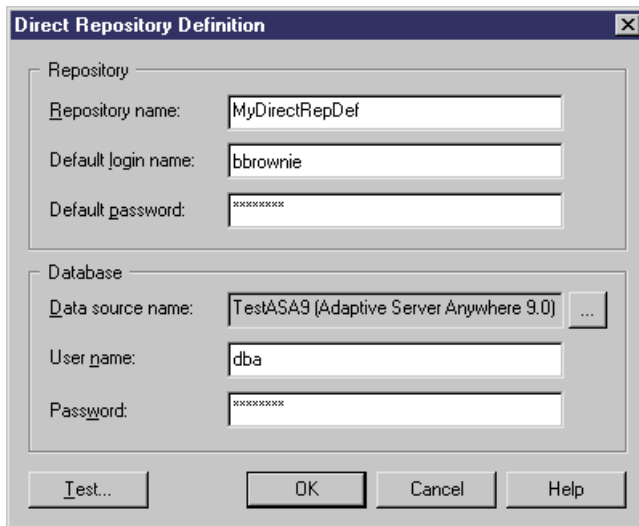
- 5.** Select one of the following radio buttons, depending on how you intend to connect to your database:

- ODBC machine data source
- ODBC file data source
- Connection profile (for native, JDBC, ADO.NET, OLE DB or DirectConnect connections)

You can 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 modify or configure your data source connection.

For detailed information about creating, configuring, and using connection profiles, see "Connecting to a Database" section in the Models Chapter 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 window:

The image shows a 'Direct Repository Definition' dialog box. It has two main sections: 'Repository' and 'Database'. The 'Repository' section contains three text boxes: 'Repository name' with the value 'MyDirectRepDef', 'Default login name' with the value 'bbrownie', and 'Default password' with a masked value 'XXXXXXXXXX'. The 'Database' section contains three text boxes: 'Data source name' with the value 'TestASA9 (Adaptive Server Anywhere 9.0)' and a browse button '...', 'User name' with the value 'dba', and 'Password' with a masked value 'XXXXXXXXXX'. At the bottom of the dialog are four buttons: 'Test...', 'OK', 'Cancel', and 'Help'.

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.

Both the user names and passwords are case-sensitive and optional. If you do not specify them here, you will need to enter them each time you connect to the repository. The password boxes always display stars, whether or not a password has been entered.

Note: You can test the definition by clicking the Test button. A message box will state whether your connection succeeds or fails.

8. Click OK to return to the List of Repository Definitions.

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

Setting up PowerDesigner for Access Via the Proxy

You create a proxy repository definition when you are connecting to the repository via the repository proxy.

If you are connecting directly, see [Setting up PowerDesigner for direct access](#) on page 2.

Warning! In order 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.

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:

Proxy Repository Definition

Repository

Repository name: MyProxyRepDef

Default login name: ADMIN

Default password: xxxxxxxxxxxx

Proxy

Server name: bbrownie

Port number: 30000

Instance name: Human Resources

Test... OK Cancel Help

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.

The user name and password are case-sensitive and optional. 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.

Note: You can test the definition by clicking the Test button. A message box will state whether your connection succeeds or fails.

5. Click OK to return to the List of Repository Definitions.

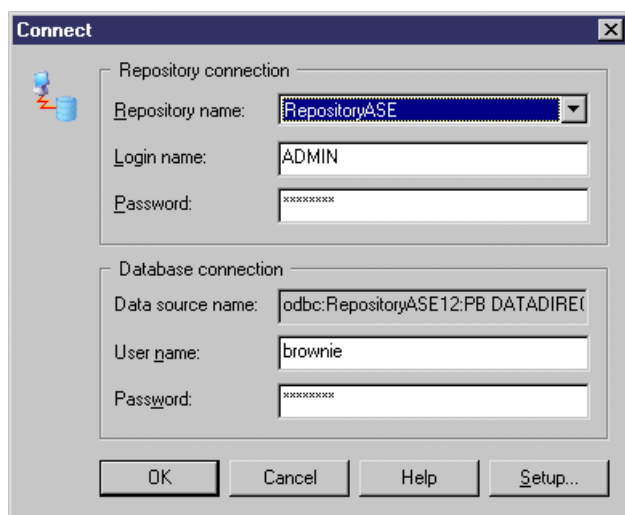
Connecting to a Repository

Once the repository has been set up by an administrator and you have created a repository definition, you can connect to the repository.

For information about setting up the repository, see the *Installation Guide*. For information about creating a repository definition, see [Setting up PowerDesigner to Access the Repository](#) on page 2.

1. Select **Repository > Connect** to display the Connect dialog box
2. Select a repository name in the list.

The connection parameters of the repository definition appear in the different fields of the dialog box.



3. Click OK to connect.

The root node displays the name of the repository, the user login and the current branch.

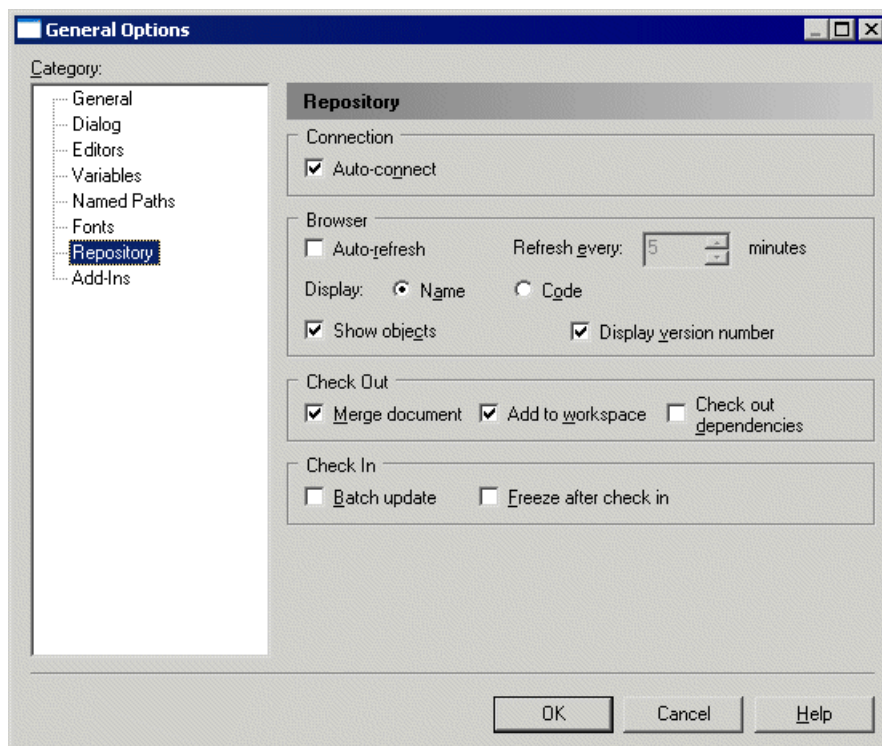
For more information about branches, see [Repository Branches](#) on page 40.

Connecting Automatically to the Default Repository

You can specify that you want automatically to connect to a repository each time you open PowerDesigner. The default repository to which you will connect is the last repository to which you connected successfully.

For more information about general options, see "Customizing your Modeling Environment" section in the Models chapter of the *Core Features Guide*.

1. Select **Tools > General Options** to open the General Options dialog box, and click the Repository category.

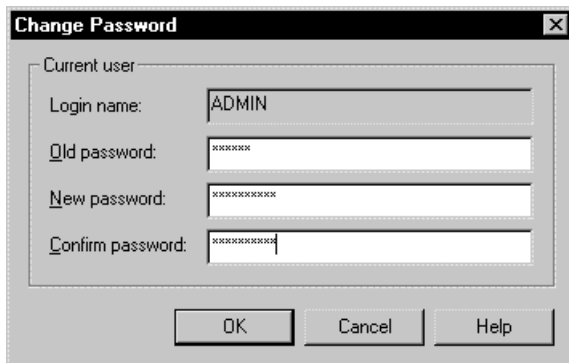


2. Select the Auto-connect check box in the Connection group box, and then click OK.

Changing Your Repository Password

You can modify the password that you use to connect to the repository.

1. Select **Repository > Change Password** to open the Change Password dialog box.
2. Type a new password in the New Password box and confirm this password in the Confirm Password box.



The 'Change Password' dialog box has a title bar with a close button. It contains a group box labeled 'Current user' with four text input fields: 'Login name' (containing 'ADMIN'), 'Old password' (masked with asterisks), 'New password' (masked with asterisks), and 'Confirm password' (masked with asterisks). At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

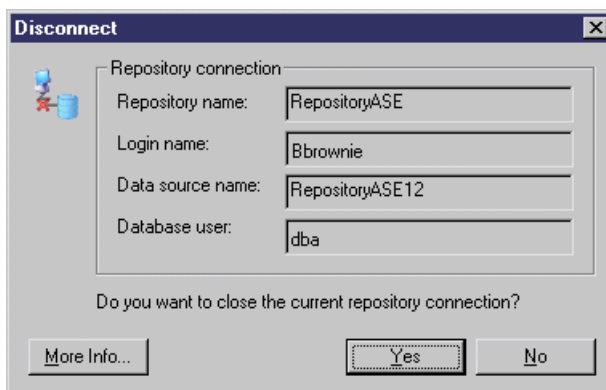
3. Click OK.

Note: Administrators can change the password of a user from the property sheet of the user. To do so, select **Repository > Administration > Users** and double-click a user in the list to display 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 box.



The 'Disconnect' dialog box has a title bar with a close button. It contains a group box labeled 'Repository connection' with four text input fields: 'Repository name' (containing 'RepositoryASE'), 'Login name' (containing 'Bbrownie'), 'Data source name' (containing 'RepositoryASE12'), and 'Database user' (containing 'dba'). Below these fields is the question 'Do you want to close the current repository connection?'. At the bottom are three buttons: 'More Info...', 'Yes', and 'No'.

2. Click Yes to close the connection.

Working with Repository Documents

Repository users can perform various operations on project, model or file documents stored in the repository.

The majority of repository users will be concerned primarily with:

- adding documents to the repository (see [Adding Documents to the Repository](#) on page 10)
- checking in changes made to these documents (see [Checking in Changes to Documents Stored in the Repository](#) on page 11)
- checking out documents (see [Checking Documents Out From the Repository](#) on page 20)
- browsing, searching for, and comparing documents (see [Browsing the Repository](#) on page 26)
- freezing and unfreezing documents (see [Document Versions](#) on page 36)
- locking and unlocking documents (see [Document Locks](#) on page 38)
- creating branches in the repository version tree (see [Repository Branches](#) on page 40)
- creating configurations, collections of related document versions (see [Repository Configurations](#) on page 44)

Note: You must have at least Write permission to add or check in a documents in the repository, and at least Read permission to check one out. For more information on permissions, see [Controlling Permissions for Repository Items](#) on page 79.

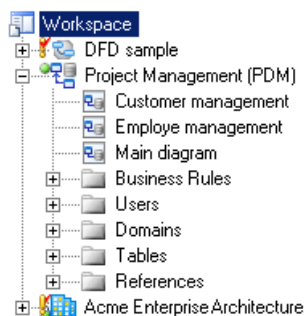
Checking the State of a Document

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 turn on and off the status icons with the "Display status icons in workspace" general option (see [Repository Options](#) on page 35).











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.

Icon	Description
	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 38.
	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).
	Changed by you but locked and changed by another (red checkmark, yellow exclamation point, and red padlock).

Adding Documents to the Repository

To add project, model, or file documents to the repository, you must have Write or Full permission on the target folder or root node.

Adding a Project to the Repository

You add a project to the repository through the Check In Project window.

When checking in a project for the first time, all associated models and files that are inside the project directory must be checked in to the repository.

1. Right-click the project in the Local tab of the Browser, and select Check In to open the Check In Project window. The New mode is selected automatically (see [Using the Check In Project window](#) on page 11).
2. Select the appropriate parameters (see [Check in parameters](#) on page 13), and click OK to add the project and its associated models and files to the repository.

Adding a Single Model or File to the Repository

You add a single model or file to the repository through the Check In Document window.

1. Right-click the model or file in the Local tab of the Browser, and select Check In to open the Check In Document window. The New mode is selected automatically (see [Using the Check In Document window](#) on page 12).

2. Select the appropriate parameters (see [Check in parameters](#) on page 13), and click OK to add the model or file to the repository.

Adding Multiple Models and Files to the Repository

You add multiple models and files to the repository through the Check In Multiple Documents window.

1. Right-click the workspace root or a folder containing multiple models and files, or Ctrl-select multiple models and files in the Local tab of the Browser, and select Check In to open the Check In Multiple Documents window.

You can add or remove models and files in the list using the tools provided (see [Using the Check In Multiple Documents Window](#) on page 15).

2. Click OK to add the models and files to the repository.

Checking in Changes to Documents Stored in the Repository

To check in changes to project, model, or file documents stored in the repository, you must have Write or Full permission on the document or (for a model) on one of its packages.

For more information on packages, see [Checking in PowerDesigner packages](#) on page 20.

Checking in Changes to a Project

You check in changes to a project through the Check In Project window.

1. Right-click the project in the Local tab of the Browser, and select Check In to open the Check In Project window. The Update mode is selected automatically (see [Using the Check In Project window](#) on page 11).
2. Select the appropriate parameters (see [Check in parameters](#) on page 13), and click OK to check in your changes with the version of the project held in the repository.

Checking in Changes to a Single Model or File

You check in changes to a single model or file through the Check In Document window.

1. Right-click the model or file in the Local tab of the Browser, and select Check In to open the Check In Document window. The Update mode is selected automatically (see [Using the Check In Document window](#) on page 12).
2. Select the appropriate parameters (see [Check in parameters](#) on page 13), and click OK to check in your changes with the version of the model or file held in the repository.

Checking in Changes to Multiple Models and Files

You check in changes to multiple models and files through the Check In Multiple Documents window.

1. Right-click the workspace root or a folder containing multiple models and files, or Ctrl-select multiple models and files in the Local tab of the Browser, and select Check In to open the Check In Multiple Documents window.

You can add or remove models and files in the list using the tools provided (see [Using the Check In Multiple Documents Window](#) on page 15).

2. Click OK to check in your changes with the versions of the models and files held in the repository.

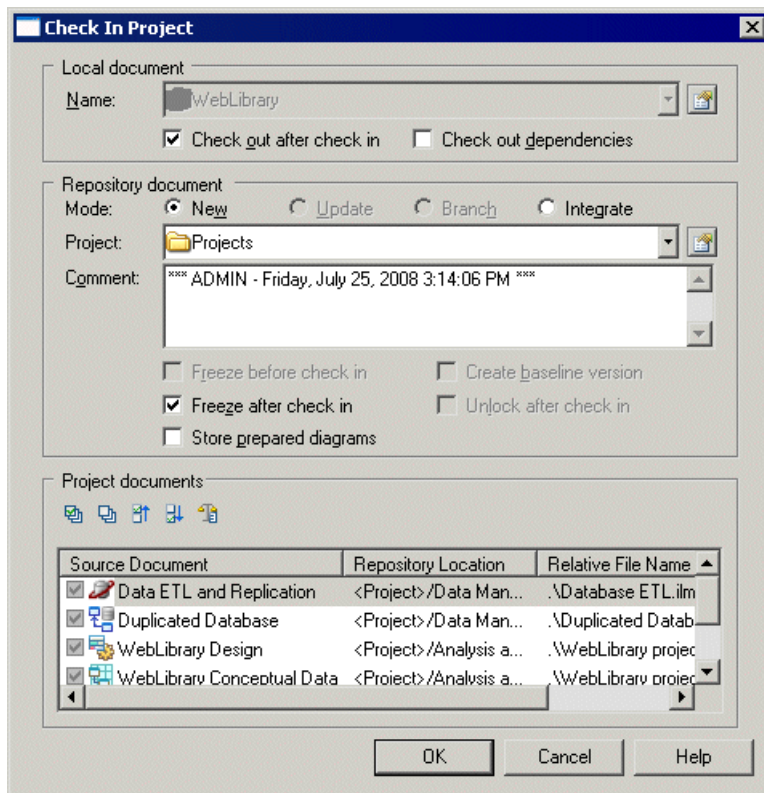
Note: If you change the DBMS or target language of a model that is in the repository (other than to another target within the same family, such as moving from ASIQ 12.6 to 12.7), you will no longer be able to check in future changes.

Using the Check In Project Window

The Check In Project window lets you specify the project and all its associated models and files to be checked in, and set check in parameters.

For more information on check in parameters, see [Check in parameters](#) on page 13.

You arrive at the Check In Project window by right-clicking the Project in the Local tab of the Browser, and selecting Check In. The window is pre-populated with the project documents, which are located in the project directory on your file system. These documents are all selected and must be checked in when you first check in a project. When you subsequently check in changes to your project, only those documents with changes are selected by default.



Using the Check In Document Window

The Check In Document window lets you specify the model or file to be checked in, and set check in parameters.

For more information on check in parameters, see [Check in parameters](#) on page 13.

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

- Right-click a model or file in the Local tab of the Browser, and select **Check In**. The window is pre-populated with details of the document.
- Select a model or file in the Check In Multiple Documents window (see [Using the Check In Multiple Documents Window](#) on page 15), and click the **Settings** tool. The window is pre-populated with details of the document.
- Click the **Add Document** tool in the Check In Multiple Documents window (see [Using the Check In Multiple Documents Window](#) on page 15). The Local Document details are blank, allowing you to specify a model or file to check in.
- Select a resource file, and click the **Check In** tool in the List of *Resource Type*. The window is pre-populated with details of the document (see [Checking Resources to Share into the Repository](#) on page 49).

Check in Parameters

The following check in parameters are available from the Check In Document and Check In Project windows to let you specify details of the local document, repository document, and project documents that will be affected by the check in:

Local Document

The Local Document group box contains the following parameters:

Parameter	Description
Workspace or File	[models and files only] Select one of these radio buttons 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. [models and files only] Depending on whether you have selected Workspace or File, this will be a: <ul style="list-style-type: none"> • Model name - click the Properties tool to the right of the field to open its property sheet • File name and path - click the Browse tool to the right 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 known.

Parameter	Description
Check out after check in	[models and resources only] Specifies that the local version will be updated to ensure that it reflects any changes in the repository version after the check in. This allows you to work with the most up-to-date version of a model or resource (contained in a project or not) or where several users may be editing objects concurrently. See Using the Check In Model window on page 17.
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 form of check in. You can choose between: <ul style="list-style-type: none"> • New - Creates the first version of a new document in the current branch, or the first version of a document into another folder in the current branch (see New mode on page 18). • Update – Updates an existing version of 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 (see Update/Branch modes on page 18). • Branch – Creates the first version of the document in the current branch (see Update/Branch modes on page 18). • Integrate – Integrates changes in another branch of the version tree of the document, or check ins a document under another name (see Integrate mode on page 19).
Folder/Project	[available in New mode] Specifies the root node or a sub-folder where the document will be checked in into the repository. Click the Properties tool to the right of this field to open the property sheet of the folder. If you do not select 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 – available in Integrate mode] Specifies the name of the repository document with which to integrate. Click the Properties tool to open the property sheet of the document.
Version	[not for project – available in Integrate mode] Specifies the version of the repository document with which to check in.
Comment	Specifies a comment to help future users identify 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.
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.

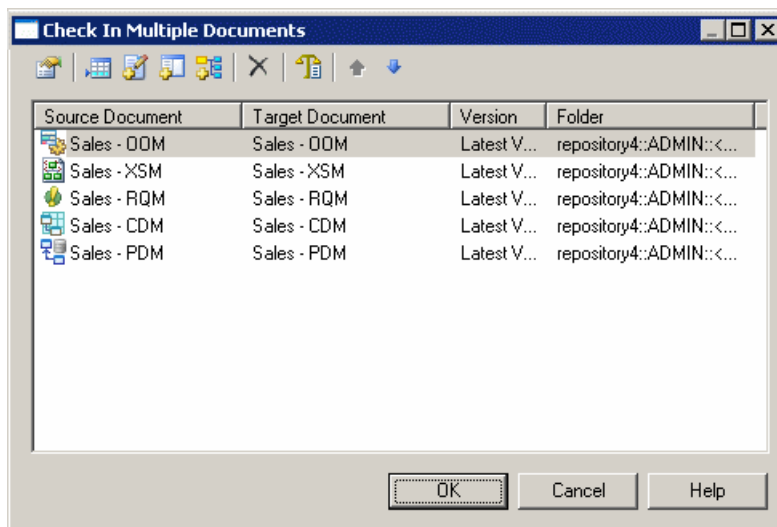
Parameter	Description
Create baseline version	<p>Creates a new version of all the objects in the entire model, whether or not they have been modified. This can be used as starting point for a new development step. This option is restricted to users with Full permission on selected document.</p> <p>A baseline version makes SQL queries faster because they do not have to deal with multiple object versions.</p> <p>The Create baseline version option does not use the merge interface since a new version is systematically created for each object in the model. This could help you bypass check in problems when dealing with objects that are not properly handled in the repository. However, this may be fairly space consuming - the server will create large rollback logs, and use a large number of locks, and you might want to configure your server to allow such large transactions.</p>
Unlock after check in	[locked documents only] Unlocks the document version after check in. If you do not select this option, the new or updated version is locked whereas the previous version is unlocked.
Store prepared diagrams	[not for resources] Prepares snapshots of the model diagrams that can be viewed in the PowerDesigner Repository Web Browser (see Browsing the Repository Via the Web on page 53).

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 projects are listed here. Documents that are outside the project and only attached to it are not listed and must be checked in separately.

Using the Check In Multiple Documents Window










The Check In Multiple Documents window allows you to add multiple models and files to the repository and/or to check in changes to multiple repository models and files.



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

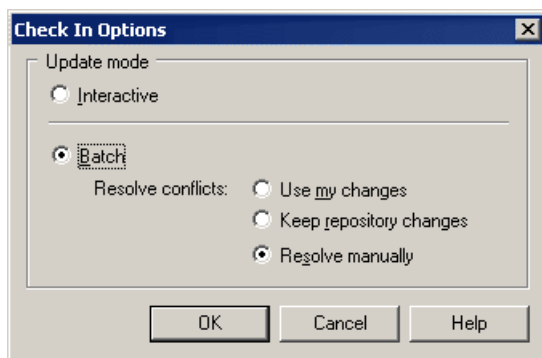
- Right-click the workspace root or a folder containing multiple models and files or Ctrl-select multiple models and files in the Local tab of the Browser, and select Check In.
- Select **Repository > Check In**.
- Press Ctrl+Alt+C

The following tools are available in this window:

Tool	Description
	Settings – Opens the Check In Document window (see Using the Check In Document window on page 12) for the selected document to allow you to specify how you want to check it in.
	Add Document - Opens the Check In Document window (see Using the Check In Document window on page 12) to allow you to select a document to add to the check in list.
	Add modified Documents in Workspace - adds all the documents modified in the workspace during the current session to the check in list.
	Add All Documents in Workspace - adds all the documents in the workspace (including closed documents) to the check in list.
	Add Related Document Versions – adds all documents related to the selected document through shortcuts or generation links to the check in list. This tool allows you to save the global environment of a document. 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 check in list.
	Check In Options – Opens the Check In Options window to allow you to specify the update mode for changes (see Using the Check In Options window on page 16).
	Move Up – Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

Using the Check In Options Window

The Check In Options window allows you to specify how differences between your local copy of a document and the repository version are resolved during check in.



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

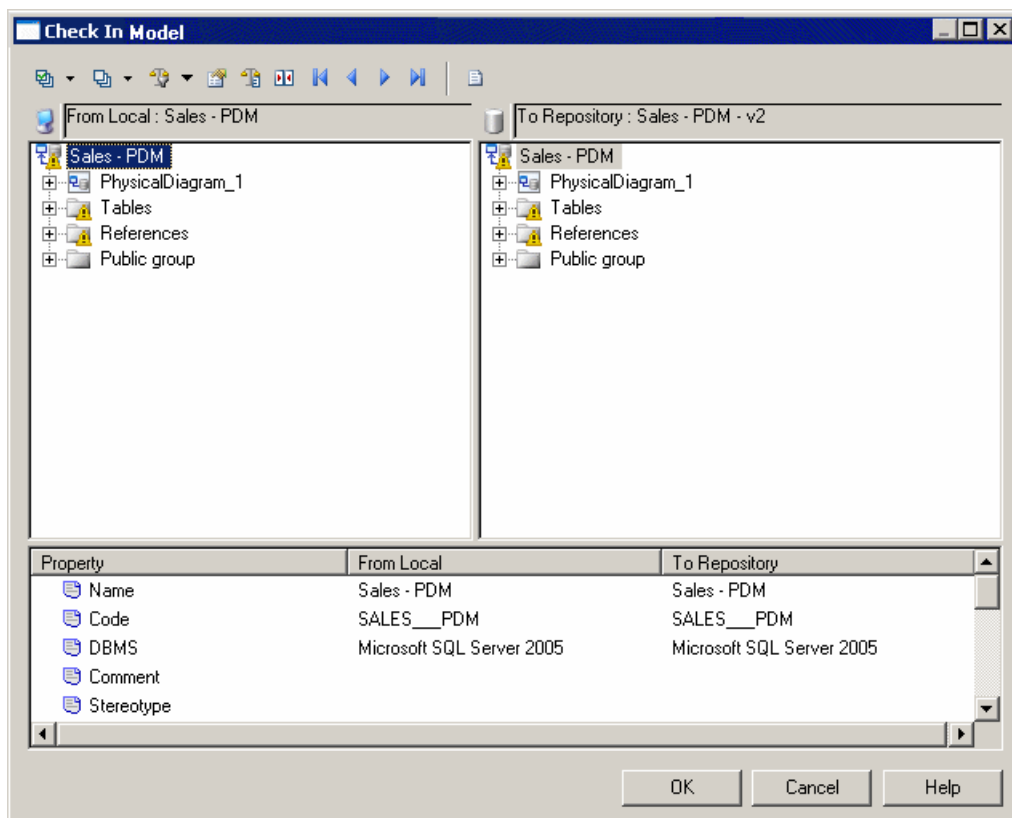
- Click the Check In Options tool in the Check In Project window. The options specified will apply to all the project models and files being checked in.
- Click the Options button at the bottom of the Check In Document window. Note that this button is only available if you are checking in a single document.
- Click the Check In Options tool in the Check In Multiple Documents window. The options specified will apply to all the documents being checked in.

You can choose either of the following check in modes in this window:

Check in Mode	Description
Interactive	[default] Opens the Check In Model window (see Using the Check In Model window on page 17) before the check in, to allow you to compare the local and repository versions of a document, and to review and approve or reject differences for check in.
Batch	<p>Automates check in of documents. If you select this mode, you must specify one of the following sub-modes for resolving conflicts:</p> <ul style="list-style-type: none"> • Use my changes – changes in the local document prevail against changes (made by other users) in the repository version. • Keep repository changes – changes in the repository version of the compartment prevail against changes in the local version. • Resolve manually – if conflicts exist, the Check In Model window (see Using the Check In Model window on page 17) opens to allow you to resolve conflicts. <p>You can set this mode as the default by selecting Tools > General Options, clicking Repository in the left-hand pane, and selecting the Batch update check box.</p>

Using the Check In Model Window

When you check in PowerDesigner models, the Check In Model window allows you to review all the differences between the local and repository versions and to apply or reject them as appropriate.



In the Check In Model window, the local version of the model is displayed on the left, and the repository version on the right. You can select or reject each of the differences for check in.

You can filter the content of the Check In Model window to display only conflicts, differences that occur when the same object has been modified in the local and repository versions, as in the following situations:

Conflict type	In the repository model	In the local model
Creation	Object removed	Object modified
Deletion	Object modified	Object removed
Attribute difference	Object properties modified (Name, Code, change in order of a list, addition of an object...)	Object properties modified

When you check a local model into the repository, a message is displayed in the Output window to explain each conflict and the proposed resolution.

A final message lets you confirm the resolutions proposed by PowerDesigner, you can accept them or cancel the merge.

For more information on merging models, see the Comparing and Merging Models chapter in the *Core Features Guide*.

Check in Modes

When you check in a document you have to select a check in mode, which helps you select the correct check in parameters.

The following modes are available:

- New – see [New mode](#) on page 18
- Update/Branch – [default] see [Update/Branch modes](#) on page 18
- Integrate – see [Integrate mode](#) on page 19

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.

New Mode

The New mode is always available in the check in dialog box.

This mode should be used in the following situations:

- The document version you want to check in does not exist in the repository, or does not exist in the current branch (see [Repository Branches](#) on page 40) or its base branches. If the document exists elsewhere in the repository, a message box informs you that the new version will share the version tree of that document.
- The document version you want to check in exists in the current branch, but you want to check it into another folder.

Update/Branch Modes

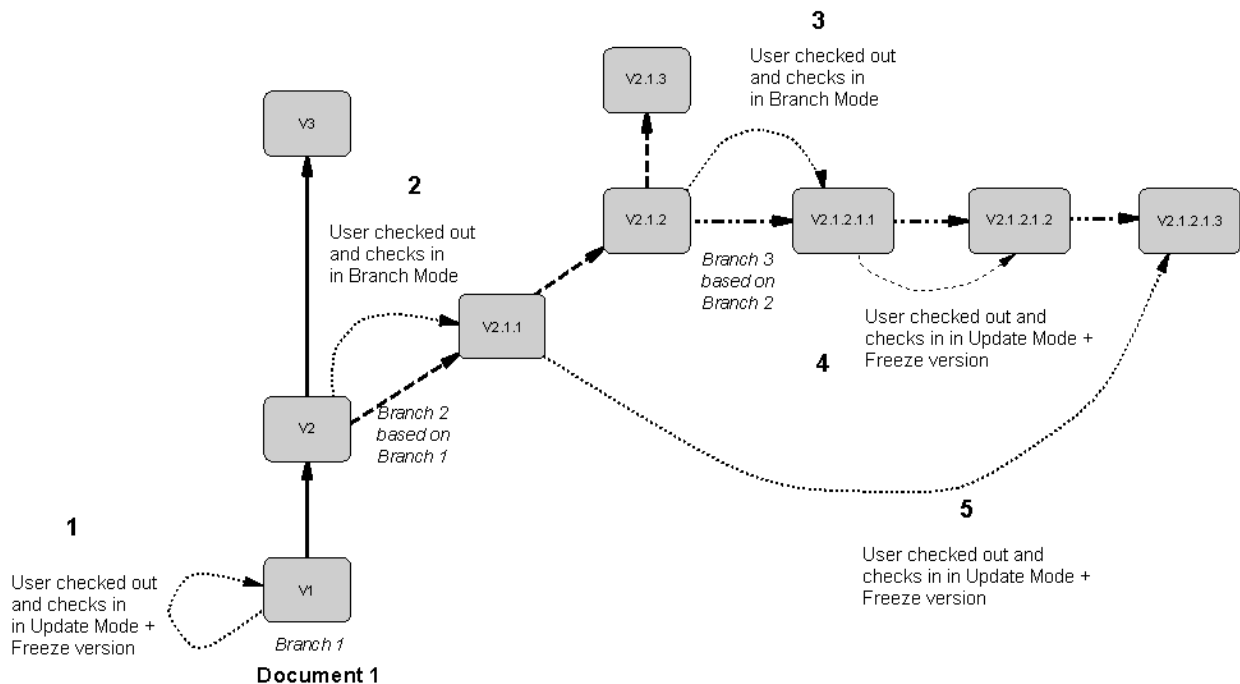
The Update and Branch modes are mutually exclusive. One of them is always the default mode, depending on whether the document you are checking in exists in the branch to which you are currently connected.

Update Mode	Branch Mode
Document exists in current branch: it has already been checked in	Document does not exist in the current branch
Checked out version of document comes from the current branch or its base branch(es)	Checked out version of document comes from the base branch of the current branch

For more information about branches, see [Repository Branches](#) on page 40.

When you select the Update or the Branch mode, you cannot modify the Project, Folder, Name, and Version boxes in the Check In Project and Check In Documents windows.

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.

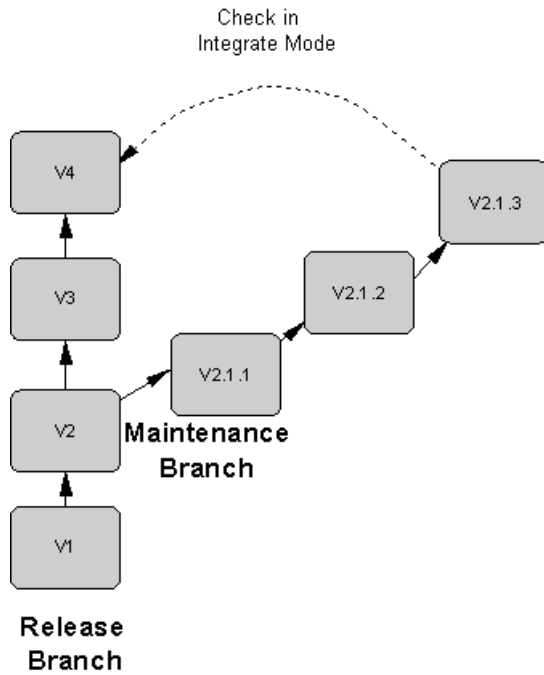
Integrate Mode

The Integrate mode is always available in the check in window.

This mode should be used in the following situations:

- You want to integrate a document version into a different document - You have to define the project or (models and files only) the folder, name, and version of the document.
- You want to integrate a version of a document into another version of the same document that you cannot reach through Update/Branch mode - You have to select a version available in the base branch of the current branch.

For example, a maintenance engineer working on a bug fix in the Maintenance branch wants to integrate this correction to the Release branch:



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 26.

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 36), such a deletion may not always be possible.

Checking Documents Out From the Repository

To check project, model, or file documents out from the repository, you must have read, write, or full permission on the document or (for a model), one of its packages. Documents for which you have only a List permission or no permission at all cannot be checked out.

You will want to check a document out from the repository either to add it to your local workspace, or to update your local version with changes made by other users.

Checking Out a Project not Present in Your Local Workspace

You can check out a project not present in your local workspace from the Repository tab of the Browser.

1. Select the Repository tab of the Browser to view the contents of the repository.
2. Right-click a project, and select Check Out to open the Check Out Project window (see [Using the Check Out Project window](#) on page 21).
3. Select the appropriate parameters (see [Check out parameters](#) on page 23), and then click OK to check out the project and (when appropriate) its associated models and files, and add them to your workspace.

Updating a Project Present in Your Local Workspace

You can update a project present in your local workspace from the Local tab of the Browser.

1. On the Local tab of the Browser, right click the project that you want to update and select Update from Repository to open the Check Out Project window (see [Using the Check Out Project window](#) on page 21).
2. Select the appropriate parameters (see [Check out parameters](#) on page 23), and then click OK to check out the project and (when appropriate) its associated models and files.

If you have selected the Merge document option, the Check Out Model window will open to allow you to review all the differences between the local and repository versions and to apply or reject them as appropriate.

Checking Out a Model or File not Present in Your Local Workspace

You can check out a model or file not present in your local workspace from the Repository tab of the Browser.

1. Select the Repository tab of the Browser to view the contents of the repository.
2. Right-click a model or file, and select Check Out to open the Check Out Document window (see [Using the Check Out Document window](#) on page 22).
3. Select the appropriate parameters (see [Check out parameters](#) on page 23), and then click OK to check out the model or file. If you have selected the Open document option, the document will be added to your workspace and opened.

Updating a Model or File Present in Your Local Workspace

You can update a model or file present in your local workspace from the Local tab of the Browser.

1. On the Local tab of the Browser, right click the model or file that you want to update, and select Update from Repository to open the Check Out Document window (see [Using the Check Out Document window](#) on page 22).
2. Select the appropriate parameters (see [Check out parameters](#) on page 23), and then click OK to check out the model or file.

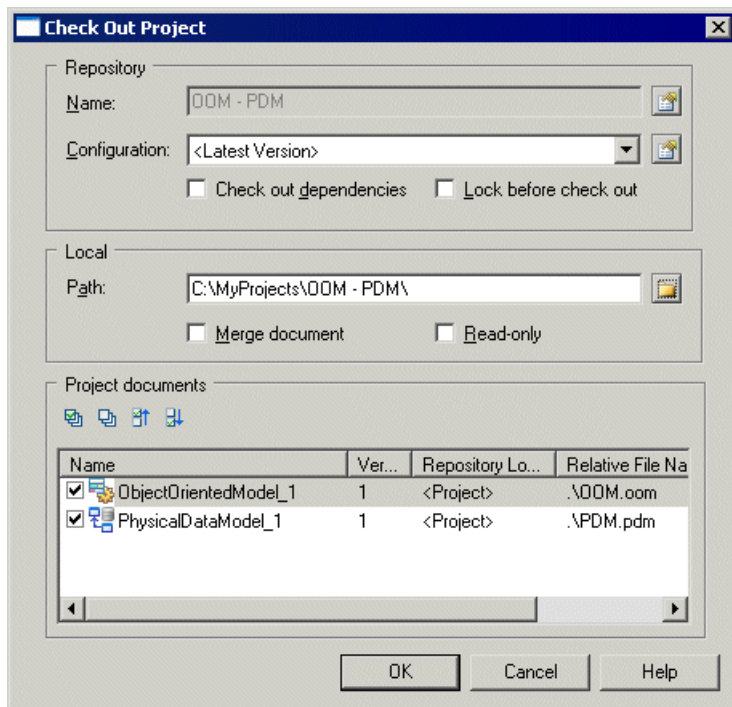
If you have selected the Merge document option, the Check Out Model window will open to allow you to review all the differences between the local and repository versions and to apply or reject them as appropriate.

Note: You can select multiple models or files to check out in either the Local or Repository tab of the Browser, in which case, the Check Out Multiple Documents window will open (see [Using the Check Out Multiple Documents window](#) on page 24).

Using the Check Out Project Window

The Check Out Project window lets you specify the project and all its associated models or files to be checked out, and set check out parameters.

For more information on check out parameters, see [Check out parameters](#) on page 23. The window is pre-populated with the project documents, and those that are not up-to-date are selected by default.

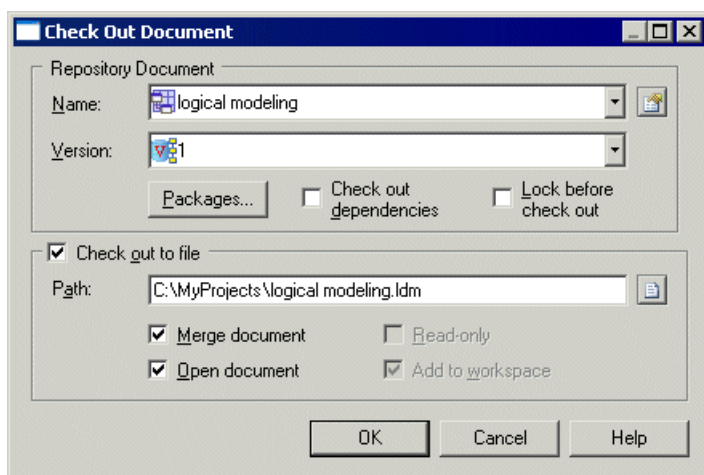


You can arrive at this window in any of the following ways:

- Right-click a project in the Repository tab of the Browser, and select Check Out. The window is pre-populated with details of the project.
- Right-click a project in the Local tab of the Browser, and select Update from Repository. The window is pre-populated with details of the project.

Using the Check Out Document Window

The Check Out Document window lets you specify the model or file to be checked out and set check out parameters.



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

- Right-click a model or file in the Repository tab of the Browser and select Check Out. The window is pre-populated with details of the document.
- Right-click a model or file in the Local tab of the Browser and select Update from Repository. The window is pre-populated with details of the document, and some of the parameters are unavailable.

- Select a model or file in the Check Out Multiple Documents window (see [Using the Check Out Multiple Documents window](#) on page 24) and click the Settings tool. The window is pre-populated with details of the document.
- Click the Add Document tool in the Check Out Multiple Documents window (see [Using the Check Out Multiple Documents window](#) on page 24). The fields are empty, allowing you to specify a model or file to check out.
- Select a resource file, and click the **Update from Repository** tool in the List of *Resource Type*. The window is pre-populated with details of the document, and many of the parameters are unavailable.

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 45.
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 33.
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.

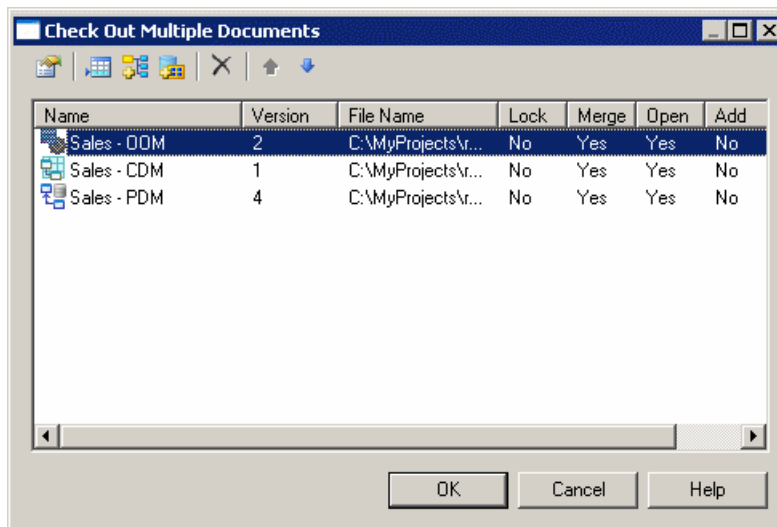
Parameter	Description
Merge document	Opens the Check Out Model window (see Using the Check Out Model window on page 25) before checking out, to allow you to compare the local and repository versions of a document, and to review and approve or reject differences for checking out. If you check-out multiple 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.

Using the Check Out Multiple Documents Window








The Check Out Multiple Documents window allows you to check multiple model or file documents out from the repository.



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

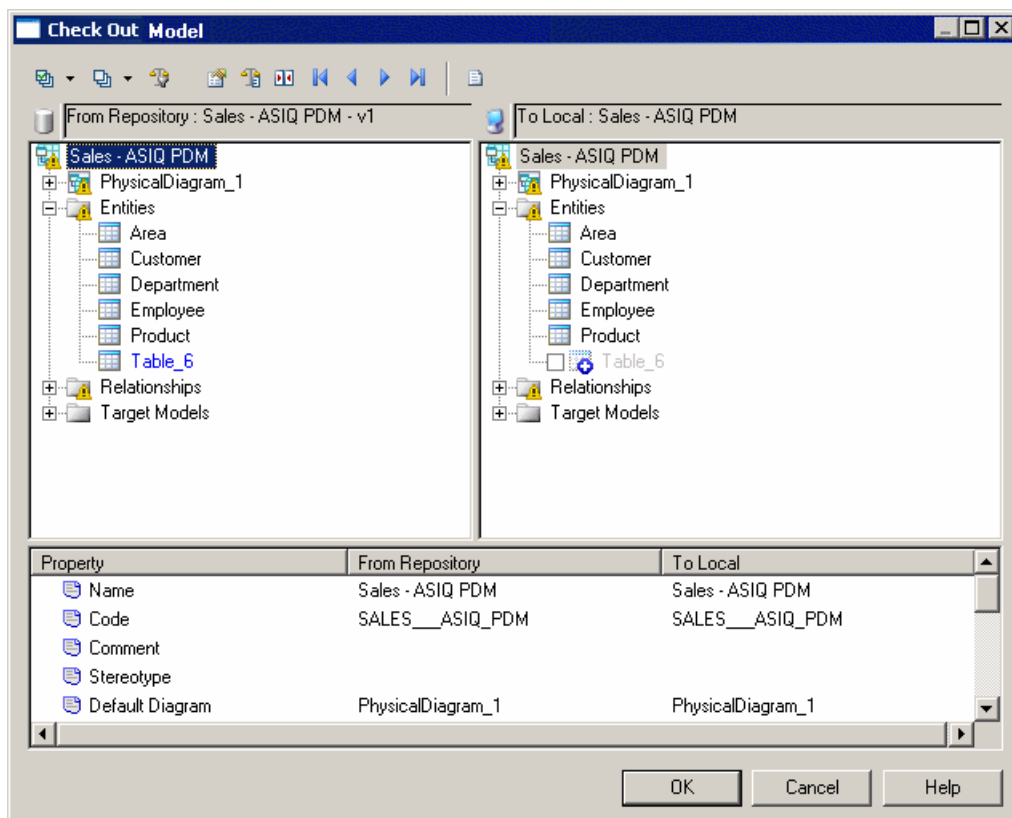
- Right-click the workspace root or a folder containing multiple models and files or Ctrl-select multiple models and files in the Repository tab of the Browser, and select Check Out.
- Right-click the workspace root or a folder containing multiple models and files or Ctrl-select multiple models and files in the Repository tab of the Browser, and select Update from Repository in the contextual menu.
- Select **Repository > Check Out**.
- Press Ctrl+Alt+E

The following tools are available in this window:

Tool	Description
	Settings – Opens the Check Out Document window (see Using the Check Out Document window on page 22) for the selected document to allow you to specify checking out parameters.
	Add Document Version - Opens the Check Out Document window (see Using the Check Out Document window on page 22) to allow you to select a document to add to the check out list.
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation links to the check out list. This tool allows you to check out the global environment of a model.
	Add Document Versions from a Configuration – Opens a selection window allowing you to specify a configuration to add to the check out list. A configuration is a set of documents defined in the repository under a label. When you select a configuration, you add all the document versions included in it to the check out list.
	Delete - Removes the selected document from the check out list.
	Move Up - Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

Using the Check Out Model Window

When you check out PowerDesigner models, the Check Out Model window allows you to review all the differences between the local and repository versions and to apply or reject them as appropriate.



You arrive at this window if you are checking out a PowerDesigner model to an existing file on your computer and you have selected Merge model in the Check Out Document and Check Out Project windows.

In the Check Out Model window, the repository version of the model is displayed on the left, and the local version on the right. You can select or reject each of the differences for check in.

When you select the Merge Document feature and depending on the state of the Open Document check box, PowerDesigner performs the following actions:

Local model	Open Document check box	Merge document impact
Opened	Selected	Merge models, and save local model
Opened	Deselected	Merge models, and save and close local model
Closed	Selected	Open local model, merge models, and save local model
Closed	Deselected	Open local model, merge models, and save and close local model




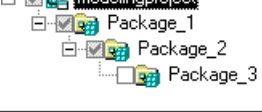
For more information on merging models, see the Comparing and Merging Models chapter in the *Core Features Guide*.

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.

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 47), 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 70.
- 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 79.
- Version Info – specifies when the repository was created and last modified, see [Repository Document and Object Properties](#) on page 54.

Repository Folders

Folders can be used to organize the contents of the repository. For example, you could create different folders for each of the subject areas, or teams working on a project. Folders can be useful for organizing permissions, as any documents checked in in a folder inherit its permissions.

You must have write permission on the location where you want to create the folder.

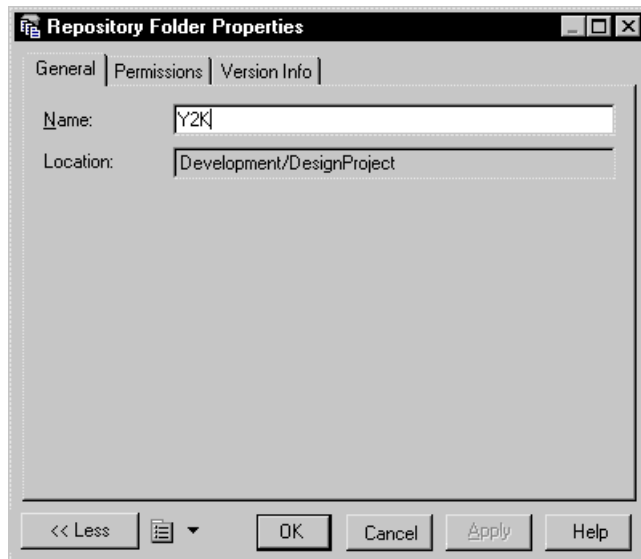
Note: A project behaves in a similar fashion to a folder. You can rename a project, delete it, freeze it, or use it to grant access permissions, etc. as for a repository folder. For information about creating a project, see the Projects and Framework Matrices chapter in the *Core Features Guide*.

Creating a Folder

You can create a folder from the Repository tab of the Browser.

1. In the Repository tab of the Browser, right-click the root node, a project or a folder in which you want to create the folder, and select **New > Folder**.

The property sheet for the new folder opens:



2. Enter a name for the folder. The name must be unique at this level in the repository.
3. Click the Permissions tab, and add permissions for all appropriate users and groups. All documents checked in in this folder will inherit these permissions. For more information about permissions, see [Controlling Permissions for Repository Items](#) on page 79.
4. Click OK to complete the creation of the folder.

Renaming a Folder

You can easily rename a folder in the repository. When you rename a folder, you must keep in mind that PowerDesigner applies checks on name uniqueness for parent folders directly linked with the root node.

1. Select the folder node in the browser and press the f2 key.
2. Type a new name and click out of the edit zone.

Moving a Folder

You can move folders and documents in the repository. When you move a document, you move it with its entire version tree. You cannot move a single document version nor can you move objects from one model to another.

Note that you cannot move a project under another project.

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.

Deleting a Folder

You can only delete empty folders. You must delete or move any contents before deleting the folder.

1. Right-click the folder in the Repository tab of the Browser, and select Delete.
A confirmation box is displayed.
2. Click OK to confirm the deletion.

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
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
	Check Out - Checks out the selected version. See Checking Documents Out From the Repository on page 20.
	Compare - Compares two versions. See Comparing Models in the Repository on page 32.

Tool	Description
	Freeze - Freezes the selected version. See Document Versions on page 36.
	Unfreeze - Unfreezes the selected version. See Document Versions on page 36.
	Lock - Locks the selected version. See Document Locks on page 38.
	Unlock - Unlocks the selected locked version. See Document Locks on page 38.
	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 sub-object collections, where these are supported by the item. For example, in a CDM, the Collections tab in the property sheet of an entity would display the business rules attached to the entity.
- 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 79.

Finding Objects in PowerDesigner Models

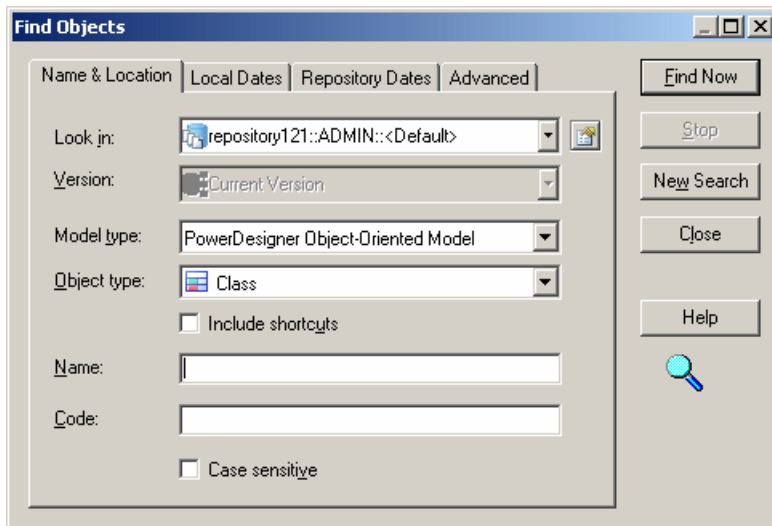
You can search for PowerDesigner model objects in the repository. The Find Objects feature is not available for external application documents.

You do not need special user rights and the List permission is enough to find objects in PowerDesigner models.

Different users can use the Find Objects feature to perform different tasks:

- A team member – may use the Find Objects feature to search the repository for objects to reuse. When he finds such objects, he can check out the model in order to copy objects and create shortcuts in other models.
 - An administrator – may search for objects that are different but have the same name in order to check out the models and normalize them, or enter User and Date filters to check who performed changes on selected objects and when.
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.

The Find Objects dialog box opens to the Name & Location tab:

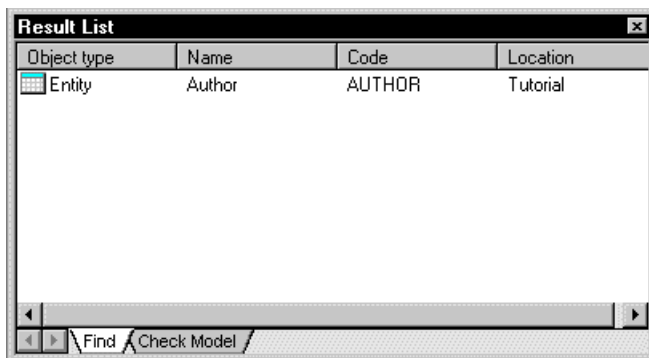


2. Define the appropriate parameters in the different tabs of the dialog box, and then click the Find Now button.

The Find process messages are displayed in the Output window in the lower part of the PowerDesigner window.

Note: Once you have started the Find Objects process, you can stop it at any time by clicking the Stop button in the Find Object dialog box.

The Find Objects dialog box remains open until the end of the process, then it is closed 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.

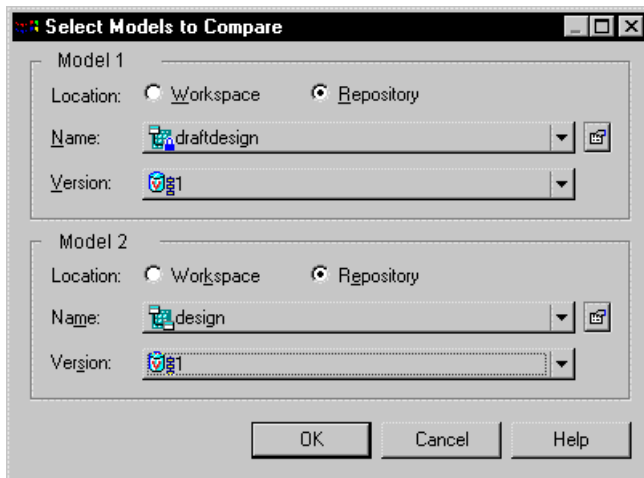
You can compare:

- 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.

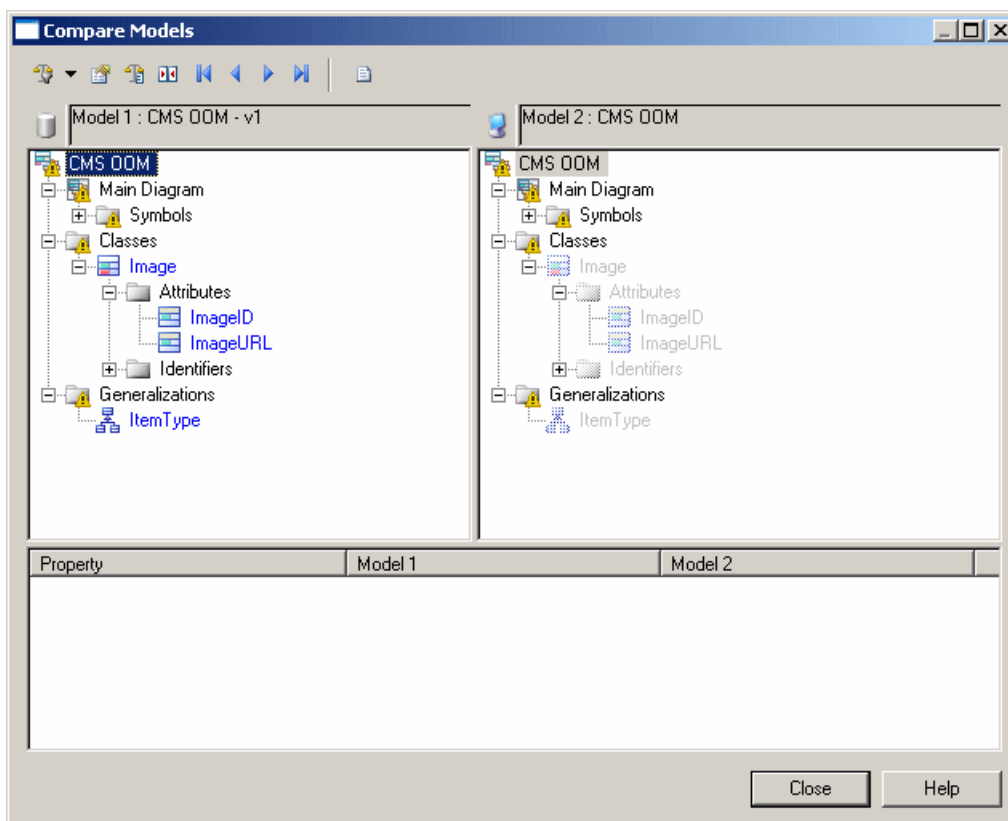
The Select Models to Compare window opens:



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.

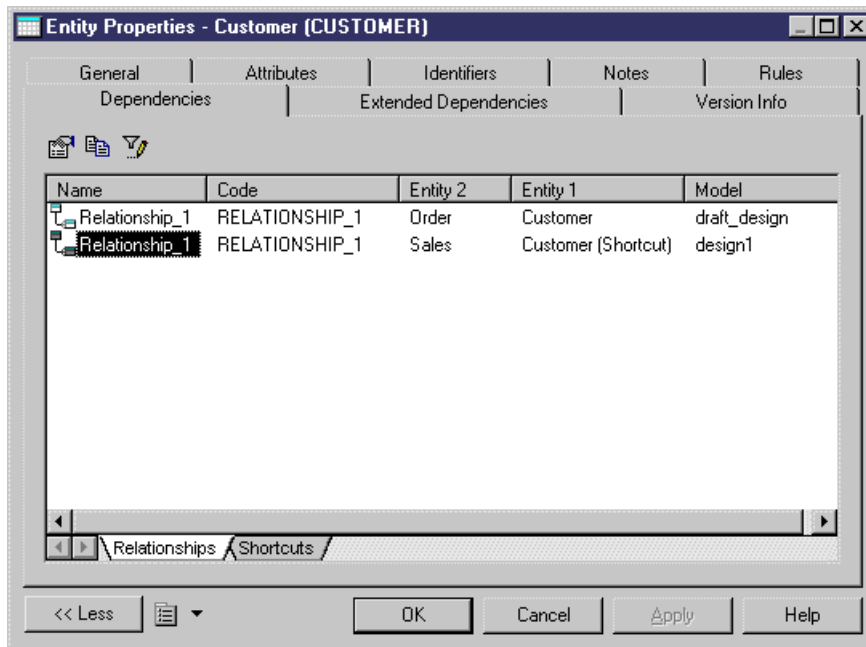
The external dependencies retrieved from the repository reflect the state of a model as it is stored in the repository. They are likely to be out of date if the local model has been modified, or not yet checked in.

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:



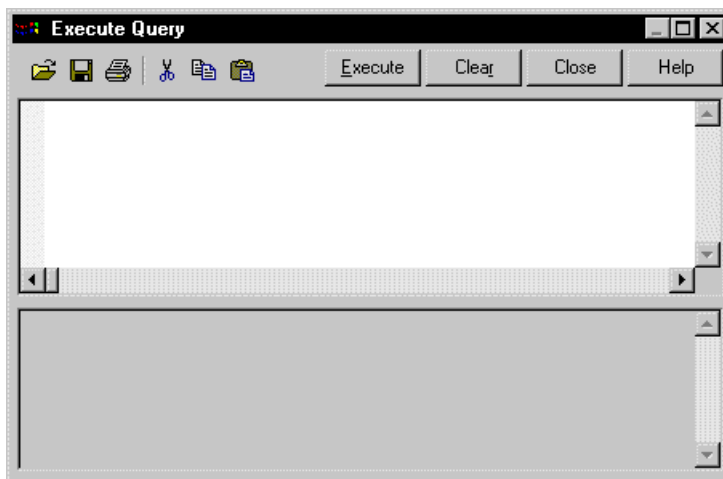
Querying the Repository Using SQL

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 > 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.

Repository 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.

Browser

The following options are available in the Browser group box:

Option	Description
Auto-refresh	Refreshes the Repository tab of the Browser at the interval specified in the Refresh every box.
Display	Specifies whether the name or code of items is displayed in the Repository tab of the Browser.
Show objects	Specifies that PowerDesigner diagrams and objects are displayed in the Repository tab of the Browser.
Display version number	Specifies that the version number of documents and objects is displayed in the Repository tab of the Browser.
Display status icons in workspace	Specifies that the state of a local document versions (against those stored in the repository) is displayed in the Repository tab of the Browser.

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 47).

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.
Store prepared diagrams	Specifies that the Store prepared diagrams check box in the Check In window is selected by default.

Document Versions

A *version* is a snapshot of a document or model object at a given moment in time. Versions are created to keep track of changes during the lifetime of a document or object, and appear as a tree on the Versions tab of repository object property sheets. Details of the current version are displayed on the Version Info tab.

By default, and unlike many repository systems, the PowerDesigner repository does not create a new version of a document each time you check it in. To create a new version of a document you must specify that you want to *freeze* it. You can freeze documents during check in or check out, or directly in the repository.

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.

Note: 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.

Freezing Document Versions

There are various ways to freeze a document (and any objects it contains) in the current branch.

- Right-click a document, folder, or root node and select the Freeze command. This command is available only if you have the Freeze versions right and the appropriate permissions on the documents.
- Use the Freeze tool on the property sheet of a:
 - document (Versions tab)
 - branch (Members tab – multiselection available)
 - configuration (Members tab – multiselection available)
- Before or after check in using the options in the Check In Document and Check In Project windows (see [Checking in Changes to Documents Stored in the Repository](#) on page 11)

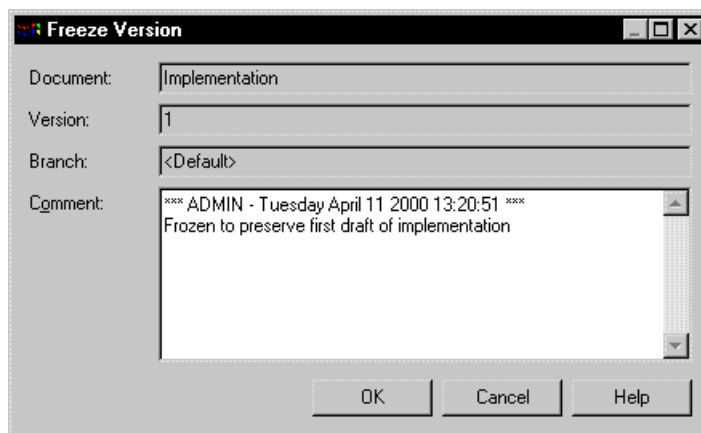
You must have the Write or Full permission to freeze a document version.

Note: This option specifies that the Freeze after check in option in the Check In window is selected by default.

Freezing a Document Version from the Repository Browser

To freeze a document version from the repository browser:

1. Right-click a document node in the repository browser and select Freeze to open the Freeze Version dialog box:



2. Enter an appropriate comment in the Comment box.
3. Click OK to freeze the version. Further changes to the document will be made to a new version.

Unfreezing Document Versions

There are various ways to unfreeze the most recent version of a document in the current branch.

- Right-click a frozen document, (or a folder or the root node) and select the Unfreeze command.

- Use the Unfreeze tool on the property sheet of a:
 - document (Versions tab)
 - branch (Members tab – multiselection available)
 - configuration (Members tab – multiselection available)

You must have the Write or Full permission to unfreeze a document version. You cannot unfreeze any but the most recent version.

Deleting Document Versions

There are various ways to delete the most recent version of a document in the current branch.

- Right-click a frozen document and select the Delete Version command.
- Use the Delete tool on the property sheet of a:
 - document (Versions tab)
 - branch (Members tab – multiselection available)
 - configuration (Members tab – multiselection available)

Warning! Deleting a version cannot be undone.

You must have the Full permission to delete a document version. You cannot delete any but the most recent version. When you delete a version, you delete the objects it contains.

Note: When you delete a version, you may 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.

Delete All Versions of a Document

To delete all versions of a document in all repository branches, right-click the document and select Delete Document. This action cannot be undone.



Document Locks

A *lock* is a temporary protection you set on a document version when you check it out, to prevent other users from updating it before you check it in. You must have at least the Lock Versions right to lock a document.

Only you or a user with the Full permission on the document can check in the document. If another user tries to check it in, a message warns him that the version is locked. Generally, you will release the lock when you check in the document.

You can also lock documents in the repository without checking them out.

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

Locking Documents

There are various ways to lock a document in the current branch.

- Right-click a document, folder, or root node and select the Lock command. This command is available only if you have the Lock versions right and the appropriate permissions on the documents.

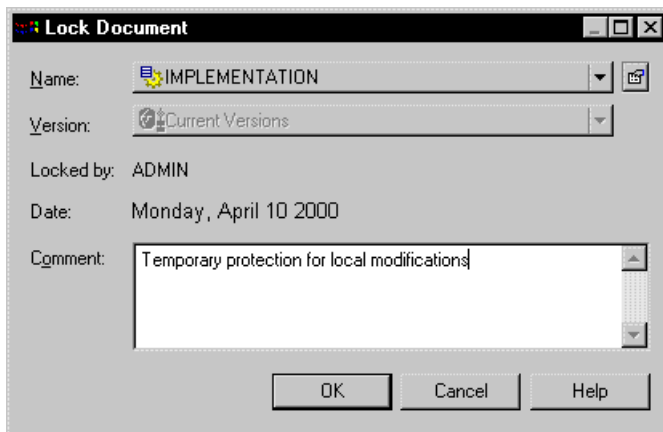
- Use the Lock tool on the property sheet of a:
 - document (Versions tab)
 - branch (Members tab – multiselection available)
 - configuration (Members tab – multiselection available)
- Using the List of Locks (see [Managing document locks in the List of Locks](#) on page 39)
- Before check out using the option in the Check Out Document and Check Out Project windows (see [Checking Documents Out From the Repository](#) on page 20)

You must have the Lock permission to freeze a document version.

Locking a Document Version from the Repository Browser

To lock a document version from the repository browser:

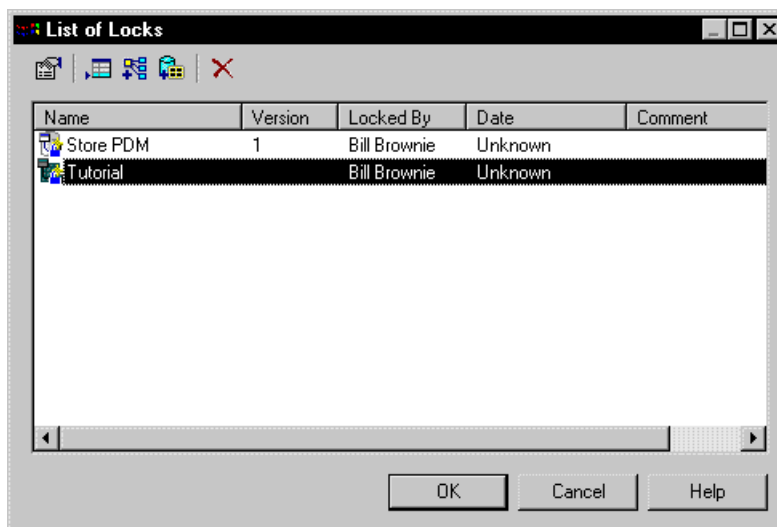
1. Right-click a document node in the repository browser and select Lock to open the Lock Document dialog box.
2. Enter an appropriate comment in the Comment box.








3. Click OK to lock the version. Other users will not be able to check in changes to the version until you unlock it.

Managing Document Locks in the List of Locks

You can review the documents that are currently locked, and lock additional documents by selecting **Repository > Locks** to access the List of 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.

Unlocking Documents

There are various ways to unlock a locked document.

- Right-click a locked document, (or a folder or the root node) and select the Unlock command.
- Use the Delete tool in the List of Locks (see [Managing document locks in the List of Locks](#) on page 39):
- Use the Unfreeze tool on the property sheet of a:
 - document (Versions tab)
 - branch (Members tab – multiselection available)
 - configuration (Members tab – multiselection available)
- After check in using the option in the Check In Document and Check In Project windows (see [Checking in Changes to Documents Stored in the Repository](#) on page 11)

You must have locked the document or have the Full permission to unlock a document.

Repository Branches

A *branch* is a view of the version tree, which shows all the document versions present in its parent branch at the time the new branch is created.

The branch from which the branch is created is called its *base branch*. By default, the repository contains a single branch called <default>. There is no limit to the number or level of branches.

Each time you connect to the repository, you have a *current branch*, and the repository browser displays only the document versions that exist in that branch, AND those in its base branch.

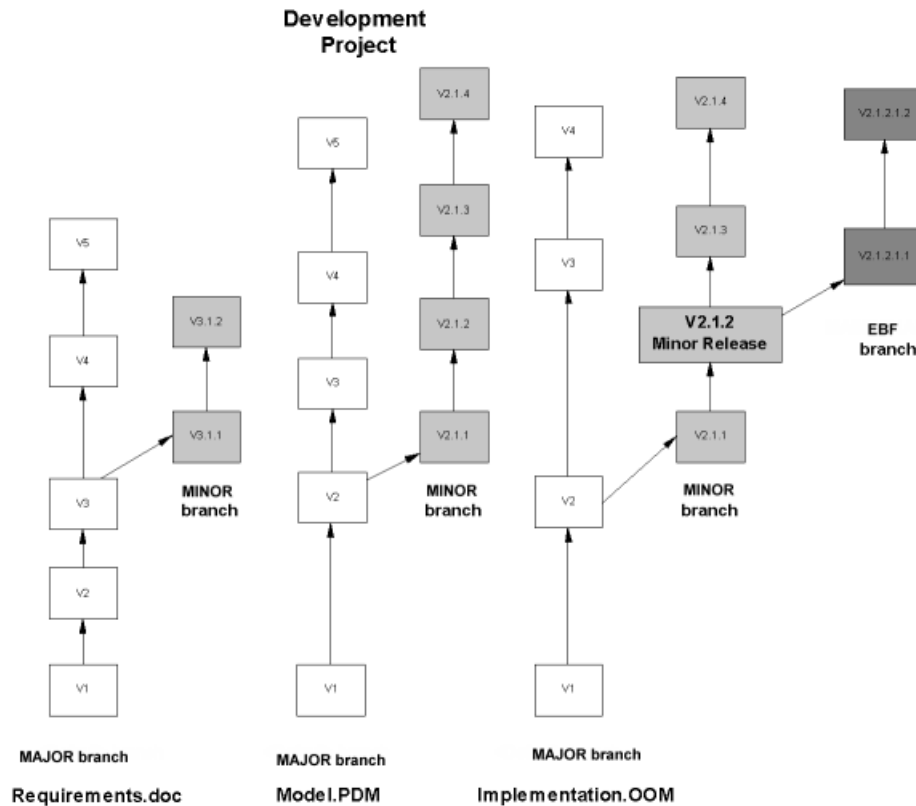
Branches are used to allow parallel updates of a document, and can be used to organize and make accessible multiple versions of a document.

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.

The project includes three branches used to perform different tasks:

- The MAJOR branch is used to work on the major release of the software: by specifying new requirements, designing and implementing major features.
- The MINOR branch is used to work on a minor release to correct bugs and add small features.
- The EBF (Emergency Bug Fix) branch is created after the minor release to let engineers correct bugs on a released software version.



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:

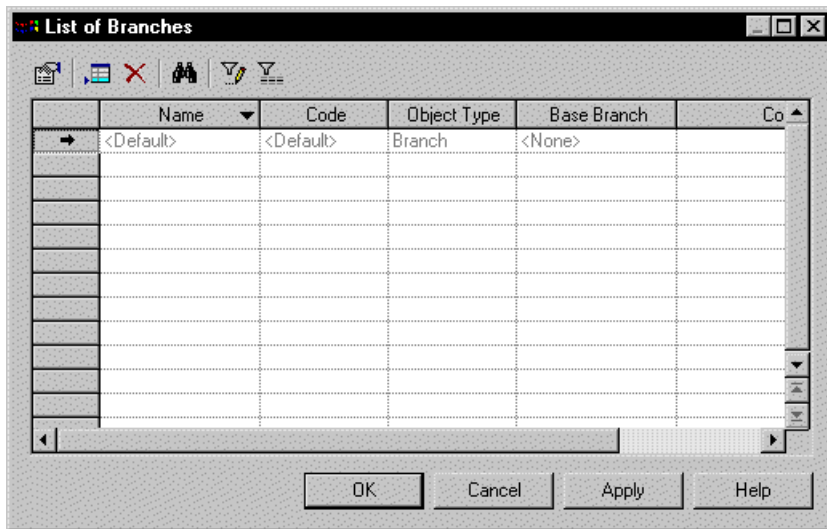
Branch	Documents versions
MAJOR	Requirements.doc V5 Model.PDM V5 Implementation.OOM V4
MINOR(based on MAJOR)	Requirements.doc V3.1.2 Model.PDM V2.1.4 Implementation.OOM V2.1.4
EBF (based on MINOR)	Requirements.doc V3.1.2 Model.PDM V2.1.4 Implementation.OOM V2.1.2.1.2

Creating a Branch

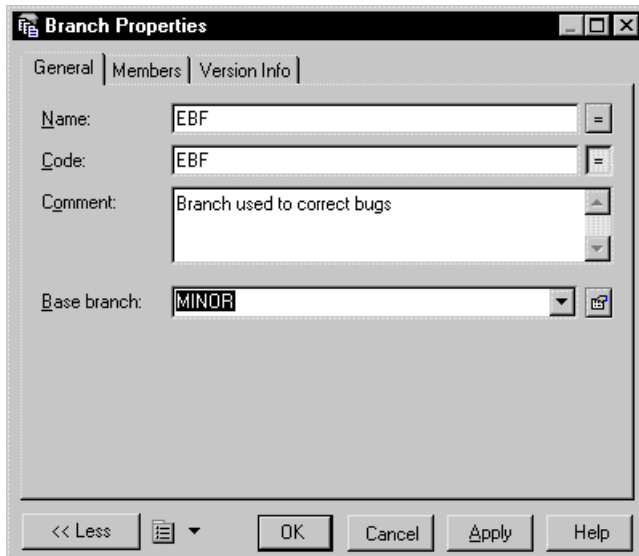
When you connect to the repository for the first time, the <Default> branch is automatically created. This branch is the trunk of the version tree. All future branches that you create will have <Default> as either their base branch or a more distant ancestor.

If you have the Manage Branches right, then you can create branches. Usually, the team leader has this right.

- 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 a base branch from the list. If you select <None>, the new branch will not have a base branch, and will be empty upon creation.
5. Click OK to return to the List of Branches.

Changing Branch

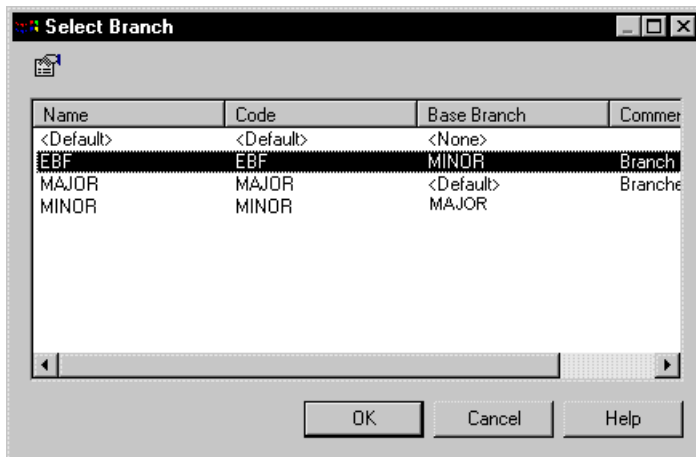
Each time you connect to the repository, you are connected to a specific branch. If you have not created any branches, then you are connected to the <Default> branch.

Your current branch is displayed on the root node in the repository browser as follows:

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

You can change your branch in order to access alternate document versions.

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 new branch.

Checking a Document into a Branch

When you check in a document, it is always checked into the branch to which you are currently connected. If you want to check your document into another branch, you must change branch before beginning your check in.

For more information about changing branch, see [Changing branch](#) on page 42.

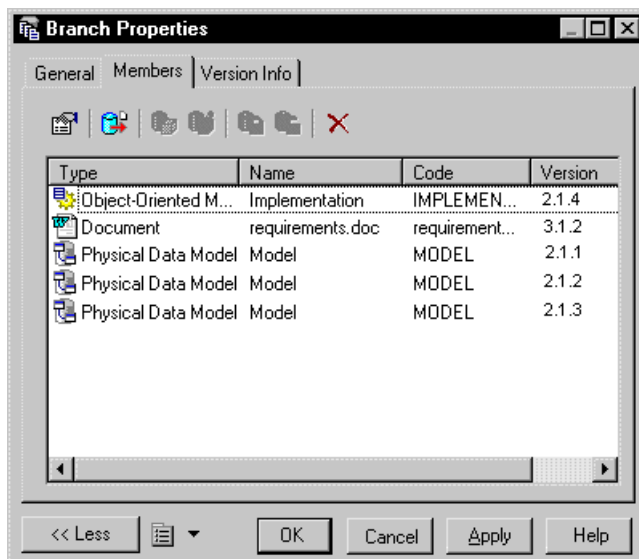
When you have completed your check in, you can confirm that your document version has been added to the correct branch by opening the property sheet of the branch and clicking its Members tab (see [Managing branch members](#) on page 43).

For more information about checking documents into branches, including the various modes available, see [Check in modes](#) on page 18.








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 From the Repository on page 20
	Freeze – Freezes the selected repository document version. See Freezing document versions on page 37
	Unfreeze – Unfreezes the selected repository document version. See Unfreezing document versions on page 37.
	Lock – Locks the selected repository document version. See Locking documents on page 38
	Unlock – Unlocks the selected repository document version. See Unlocking documents on page 40.
	Delete – Deletes the selected document version.

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.

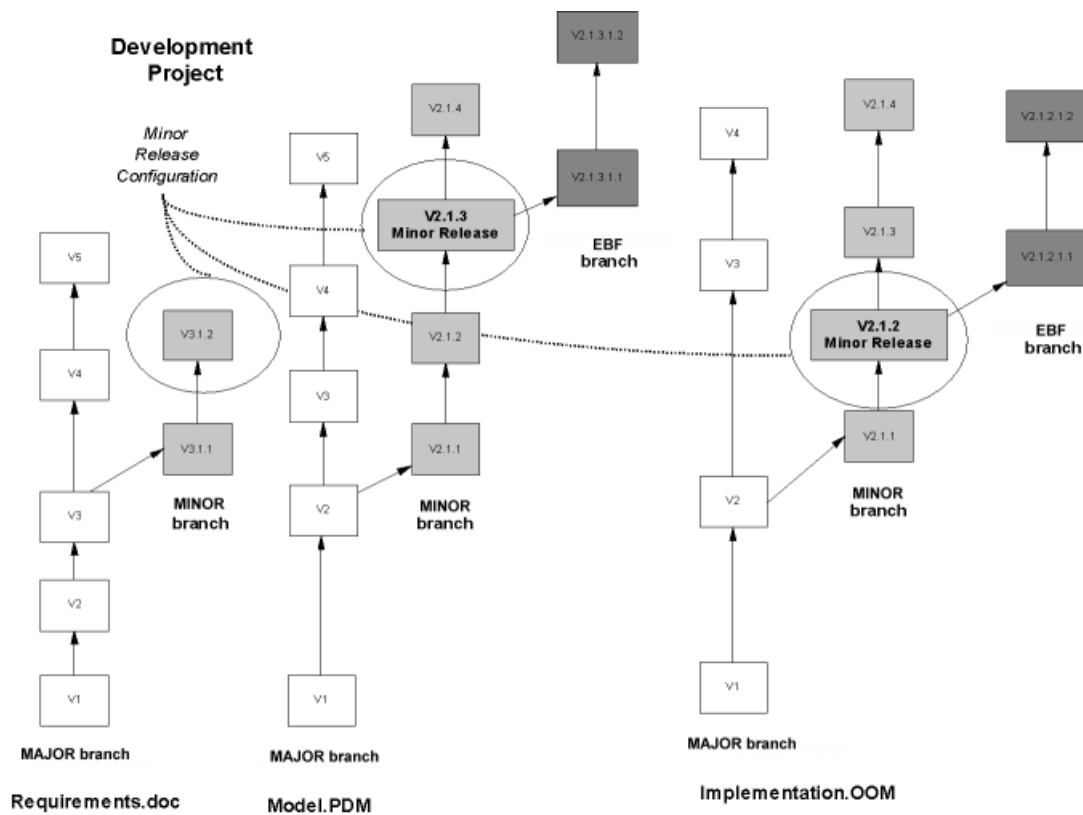
Warning! Deleting a branch cannot be undone.

1. 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.

Repository Configurations

A *configuration* is a set of document versions, which are grouped together in order to make them easier to recover. Configurations can be useful for grouping model versions that have dependencies between them (for example, through generation links or shortcuts) or documents that represent a particular point in your development, such as a release. You can easily check out all the documents included in the configuration.

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



Creating a Configuration

If you have the Manage Configurations right, you can create configurations.

1. Select **Repository > Configurations** to open the List of Configurations (or right-click a folder or a project and select Create Configuration to open the configuration property sheet).
2. [list of configurations only] Click the Add a Row tool and then click the Properties tool to open its property sheet.
3. Enter an appropriate Name, Code, and Comment to describe the configuration.

The screenshot shows the Configuration Properties dialog box with the General tab selected. The fields are as follows:

- Name:** Minor Release
- Code:** Minor Release
- Comment:** This configuration flags all versions used for the minor release of the software

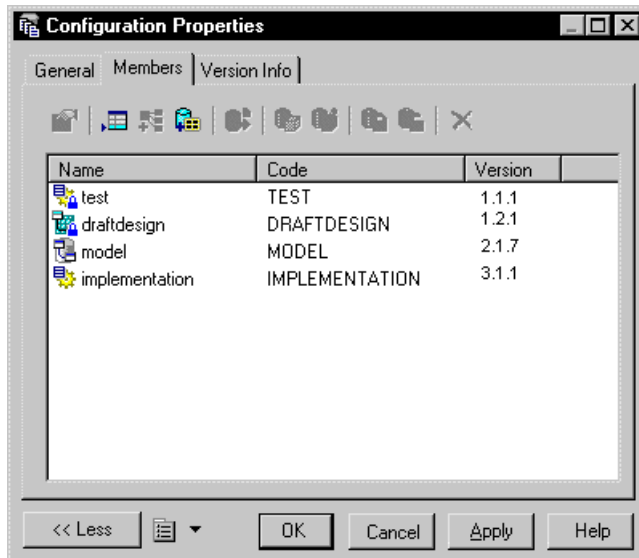
At the bottom, there are buttons for << Less, OK, Cancel, Apply, and Help.

4. Click the Members tab to add document versions to the configuration (see [Managing configuration members](#) on page 46).

- Click OK to return to the List of Configurations or to close the configuration property sheet.

Managing Configuration Members

You add and control the document versions belonging to a configuration on the Members tab of its property sheet.



You can add versions in a configuration using the following methods:

- Add an individual document version
- Add the document versions of another configuration
- Add document versions related to an existing member to the configuration

The following tools are available on the Members tab:

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 From the Repository on page 20
	Freeze – Freezes the selected repository document version. See Freezing document versions on page 37
	Unfreeze – Unfreezes the selected repository document version. See Unfreezing document versions on page 37.
	Lock – Locks the selected repository document version. See Locking documents on page 38
	Unlock – Unlocks the selected repository document version. See Unlocking documents on page 40.
	Delete – Deletes the selected document version from the configuration.

Checking Out a Configuration

You can check out all the document versions contained within a configuration in a single check out.

1. Select **Repository > Check Out** to open the Check Out Multiple Documents dialog box (see [Using the Check Out Multiple Documents window](#) on page 24).
2. Click the Add Document Versions from a Configuration tool, select the configuration to check out, and click out.

The document versions of the configuration are added 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.

Checking Out a Project Configuration

When you check out a project, you can check out all the project document versions contained within a project configuration in a single check out.

You must create the configuration from the project contextual menu (see [Creating a configuration](#) on page 45).

1. Open the Check Out Project window.
2. Select the appropriate parameters (see [Check out parameters](#) on page 23) including the configuration to check out, and then click OK to check out the document versions of the configuration, and add them to your workspace. If you have selected the Merge document option, the Check Out Model window will open to allow you to review all the differences between the local and repository versions and to apply or reject them as appropriate.

Sharing Resources in the Repository

You can store PowerDesigner resource files (DBMSs, target languages, and extended model definitions) 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 extended model definition.

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 extended model definitions 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.

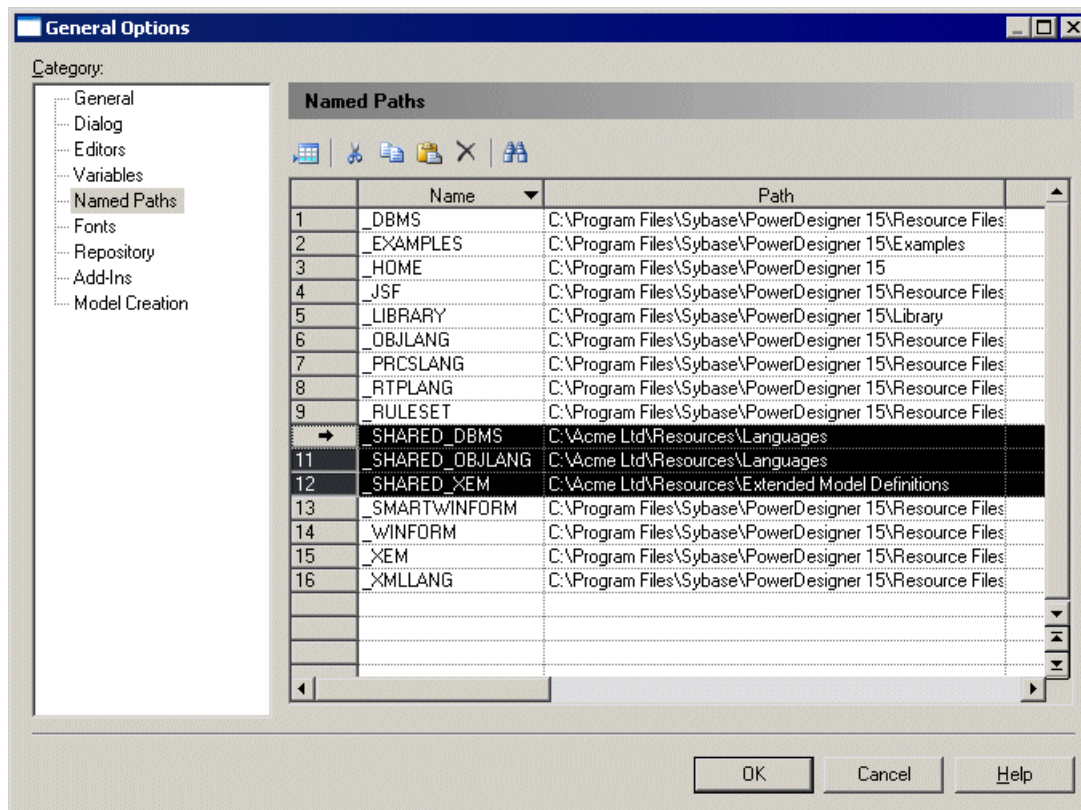
The named paths you create will be deployed to users via a user profile and will enable all users to point to the appropriate resources.

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 and Object, business process, and XML languages
 - Extended model definitions
 - Model category sets - to enforce the use of shared resources at model creation time (see [Making Shared Resources Available in the New Model Dialog](#) on page 49)
 - Model templates - to be referenced by your model category sets
 - User profiles - to configure each of these elements
3. Click **OK** to close the dialog box.

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



For more information about named paths, see "Defining Named Paths" in the Customizing your Modeling Environment chapter of the *Core Features Guide*.

Selecting and Preparing Resources to Share

The resource manager selects the resource files to share and places them in the folders for which she has created named paths 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 "Working with PowerDesigner Resource Files" in the Resource Files and the Public Metamodel chapter of the *Customizing and Extending PowerDesigner* book).

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 intend 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.

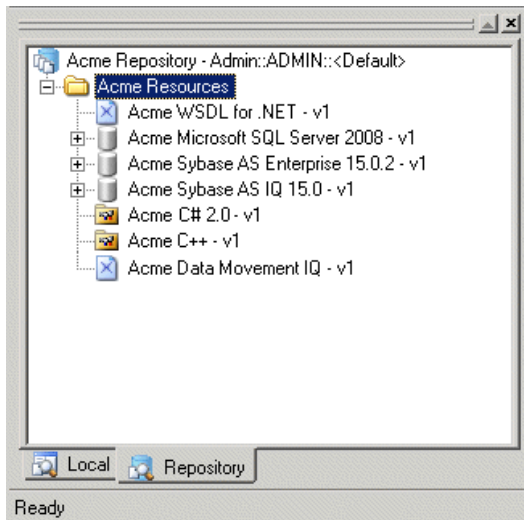
Checking Resources to Share into the Repository

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 [Creating a Folder](#) on page 27). 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 extended model definitions 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 PowerDesigner General Options.

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 49).

Creating a User Profile to Enable Resource Sharing

The resource manager creates a user profile to deploy the settings required to share 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.

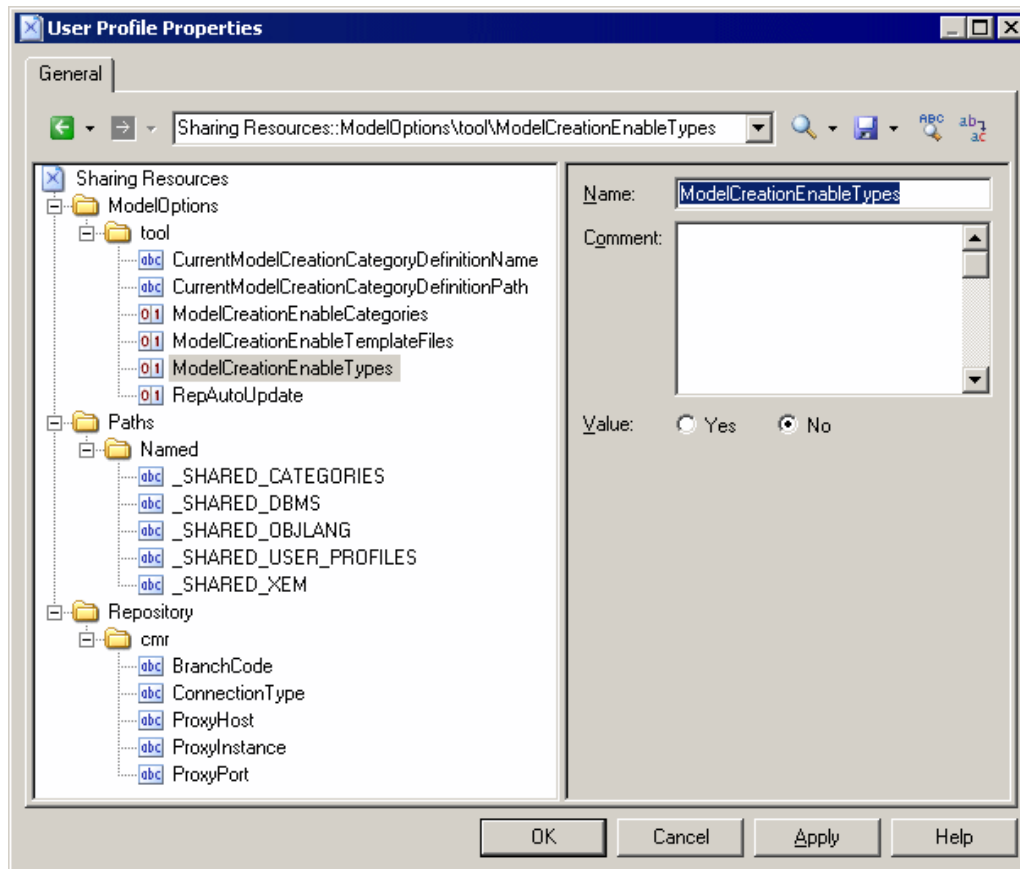
Since user profiles are themselves PowerDesigner resource files, and you will need to deploy the user profile to your users, you will need to store them with your other resources to share and specify a named path to point to them.

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
  \Extended Model Definitions
  \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 box, 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 49).

For more information about user profiles, see "User Profiles" in the Customizing your Modeling Environment chapter 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. The Update mode is selected automatically (see [Using the Check In Document window](#) on page 12).
3. Select the appropriate parameters (see [Check in parameters](#) on page 13), and click **OK** to check in your changes with the version of the resource held in the repository.

To check in changes to multiple resource files, use the Check In Multiple Documents window (see [Using the Check In Multiple Documents Window](#) on page 15).

Comparing Shared Resources Files

Anyone with at least the Read right on one or more resource files can perform a comparison on them 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 have to select resources of the same type: DBMS, object languages, XEM, etc. You cannot compare resources of different types.

For more information about the comparison process in the Repository, see [Comparing Models in the Repository](#) on page 32.

Browsing the Repository Via the Web

The PowerDesigner Repository Web Browser 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 Repository Web Browser server before you can access the repository via the web. For installation details, see "Installing the Repository Web Browser" in the Installing the Repository chapter of the *Installation Guide*.

Logging into the Repository Web Browser

In order to log into the Repository Web Browser, 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`):

SYBASE PowerDesigner Repository Web Browser

Sign in to Sybase PowerDesigner Repository Web Browser.
To sign in, select a connection profile, enter user name and password.

Connection profile: LOCAL-SSL
[Show profile information](#) [Advanced options](#)

User name:

Password: [Ligon](#) [Help](#)

©Copyright 2007-2009, Sybase Inc. All rights reserved. PowerDesigner Repository Web Browser Version 15.1

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 Ligon to go to the Repository Web Browser home page.

Each Repository Web Browser 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

Repository Explorer

The Repository Explorer page is the main page of the Repository Web Browser. It contains on the left, a tree view of the contents of the repository, and on the right, the properties of the selected item:

Branch: <Default>

- <Default>
 - Manufacturing
 - WebLibrary
 - Analysis and design
 - WebLibrary Conceptual Data - v1
 - WebLibrary Conceptual Data2 - v1
 - WebLibrary Data Flow - v1
 - Child Organization Units
 - Conditional Links
 - Data
 - Diagrams
 - Processes
 - Web Library Context Diagram - v1
 - Conditional Links
 - Data
 - Diagrams
 - Processes
 - Create Account - v1

Repository Object Properties: Create Account - v1

General | **Attributes** | Relationships | Versions

Attributes:

Name	Value
Action Type	<Undefined>
Creation Date	Feb 15, 2008
Creation User	lpommier
Duration	
Implementation Type	None
Loop Expression	
Loop Type	While
Modification Date	Feb 15, 2008
Modification User	lpommier
Number ID	1
Reusable Process	true
Timeout	
Use Parent Namespace	true

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

You can click any document or model object in the repository to see its property sheet. The following tabs are available:

- General - displays basic, read-only, information about the object.
- Attributes - [models, packages, and model objects only] displays the properties of the object.
- Relationships - [objects only] displays lists of the other objects with which the object is related
- Versions - displays the various versions of the document stored in the repository.
- Version Info - displays information about the current version of the document, along with information about its creation and last modification.

- Permissions - [models, packages and other documents only] lists those users and groups with permissions on the document. For more information, see [Setting permissions](#) on page 55.
- Compare - [models, and objects only] Allows you to compare different versions of models and objects. For more information, see [Comparing model and object versions](#) on page 55.
- Subscriptions - [projects, folders, and models only] Allows you to subscribe to email notifications of changes to the object. For more information, see [Subscribing to change notifications](#) on page 55.

Setting Permissions

You can change access permissions on an item from the Repository Web Browser.

Adding User or Group Permissions for a Repository Item

To add user or group permissions for a repository item:

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.

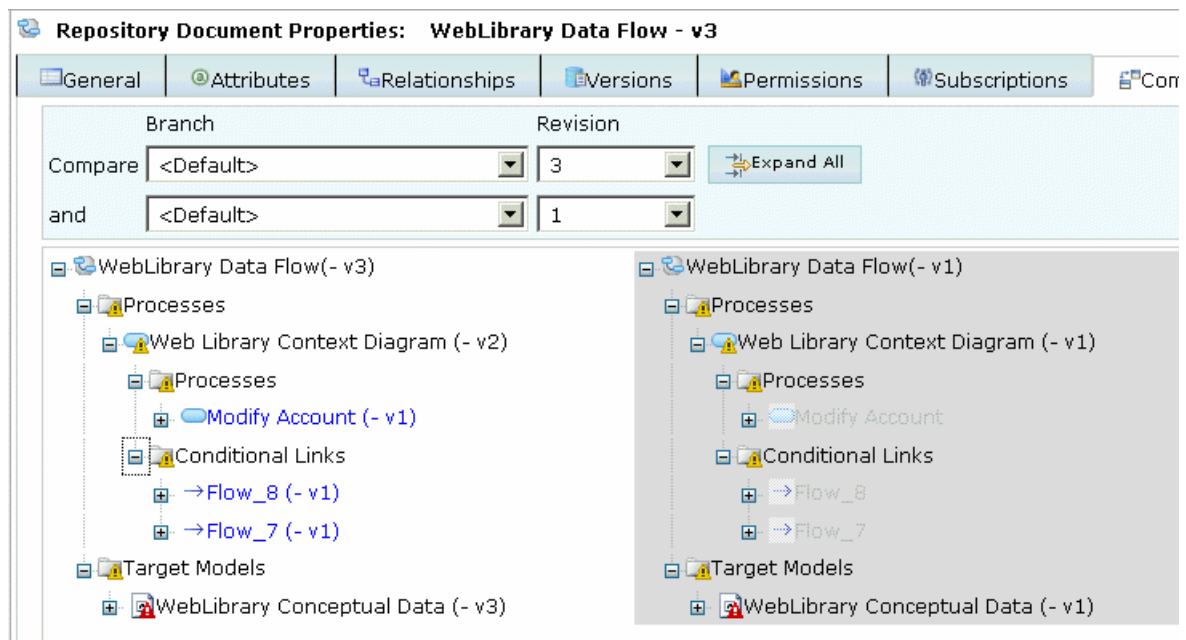
1. Navigate to the project, model, or folder for which you want to subscribe in the Repository Explorer, and then click on the Subscriptions tab.
2. Select the Email checkbox against your name in the Notification Method column, and then click Save.

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 Repository Web Browser 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, and/or select the Show only changes checkbox to show only nodes that have been changed.
4. [optional] Click any of the child nodes to view changes to their properties.

Repository Web Browser Search

You can search for PowerDesigner model objects in the Repository Web Browser, by clicking the Search tab. The Read permission suffices to find objects. Note that you cannot search external application documents.

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

2. [optional] Select a branch, model, and model version on the left of the Search 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.

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

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
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)

Note that the case sensitivity of your searches depends on your DBMS.

6. Click the Add button to add the complete search clause to the Condition table.
7. [optional] Create additional search clauses and add them to the Condition table, choosing one of the following conjunctions:
 - AND – results must satisfy all clauses
 - OR – results must satisfy one of the clauses
8. [optional] Click the Others tab and specify additional search clauses based on creation and modification dates and users:

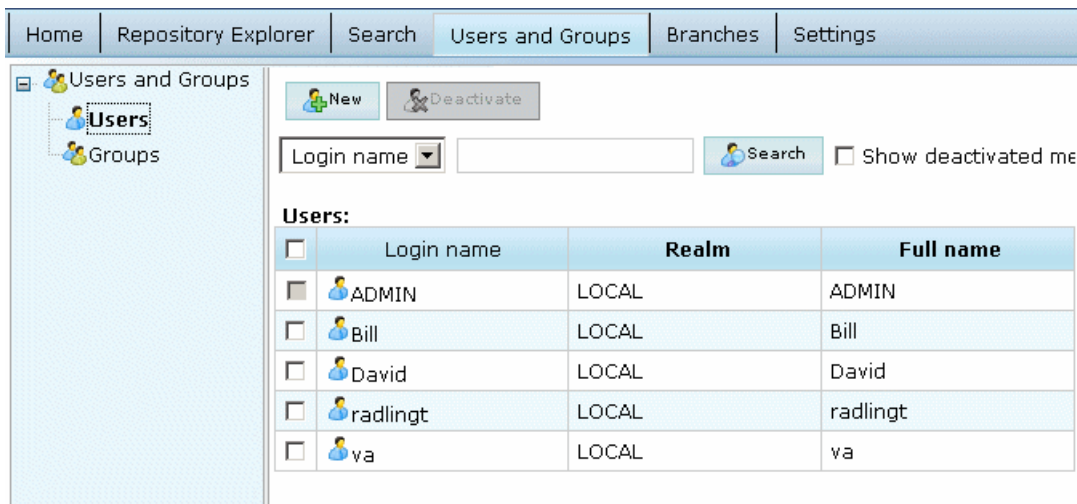
- Click the Search button to start your search.

Your search results are displayed. Click on an object to display all its available properties. Depending on the object type, you may also be able to browse and change permissions and subscriptions, compare different versions and/or view the diagram. You can search within your results by entering a value in the search box at the top of the screen.

- [optional] Click the Report button to generate a PDF version of the results for printing. Click the Back button to modify your search string.

Repository Web Browser Users and Groups

Repository users with the Manage users right can view, create, modify, and delete users and groups, and manipulate their assignment to groups via the Repository Web Browser.



Creating a Repository User

You can create users from the Users and Groups screen.

- Click the Users and Groups tab and click on Users on the left of the screen to display the list of users.
- 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.

- [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.
- [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 59)
- [optional] Click the Rights tab and assign the user any appropriate rights (see [Granting Rights to Users and Groups](#) on page 59)
- 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 60)
5. [optional] Click the Rights tab and assign the group any appropriate rights (see [Granting Rights to Users and Groups](#) on page 59)
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:

Home | Repository Explorer | Search | Users and Groups | Branches | Settings

Users and Groups

Users

Groups

Save Reset

User properties : Bill

General Parent Groups Rights Subscriptions Permissions

Available groups:
Select groups to which to add the user.

Name Search

<input type="checkbox"/>	Group name	Realm
<input type="checkbox"/>	ADMN	LOCAL
<input type="checkbox"/>	Design	LOCAL
<input type="checkbox"/>	EXTERNAL	LOCAL
<input type="checkbox"/>	Implementation	LOCAL

Parent groups:
The user belongs to th

Name

PUBLIC

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:

Home | Repository Explorer | Search | Users and Groups | Branches | Settings

Users and Groups

Users

Groups

Save Reset

Group properties : Design

General Members Parent Groups Rights Subscriptions Permissions

Available members:
Select members to add to the group.

Name Search

<input type="checkbox"/>	Name	Realm
<input type="checkbox"/>	ADMIN	LOCAL
<input type="checkbox"/>	Bill	LOCAL
<input type="checkbox"/>	ADMN	LOCAL
<input type="checkbox"/>	EXTERNAL	LOCAL
<input type="checkbox"/>	Implementation	LOCAL

Child members:
The following member(s)

Name

David
radlingt
va

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.

Deleting a User

The repository administrator or a user with the Manage Users right can delete users from the repository. A user cannot delete himself.

Warning! Deleting a user cannot be undone.

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

The user remains in the List of Users (to retain, if appropriate, authorship traces) 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 Users or Groups on the left of the screen to display the appropriate list.
 2. Select a group in the list, click the Delete button, and then confirm the deletion by clicking the OK button.

The group is removed from the list.

Repository Web Browser Branches

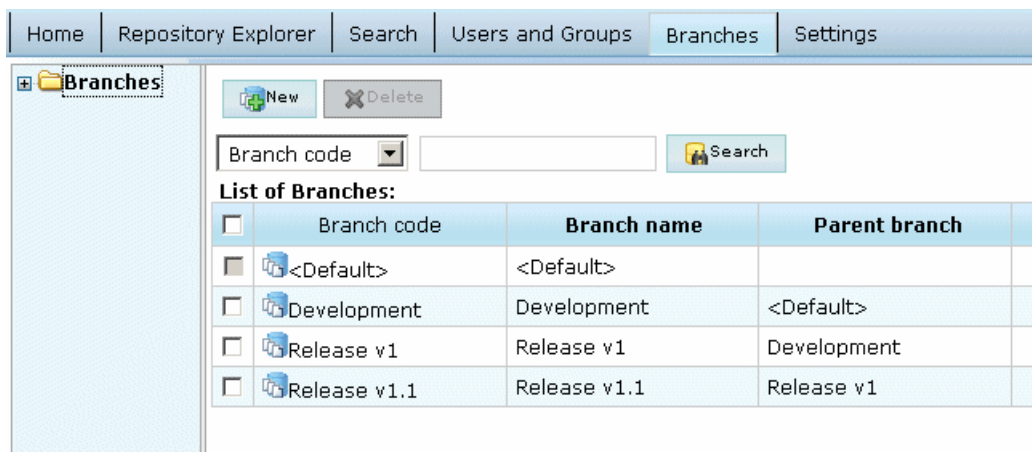
Branches allow you to split the version tree in order to support parallel development on documents.

You can easily switch between branches in the repository web browser, 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.

Repository Web Browser Settings

The Configuration tab lists various parameters that you can view and set to control your client and the repository web server.

Client Settings

The following client settings are available:

Logging

The following logging settings are available:

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.
Log file pattern	Specifies how the log file should be rolled over. You can choose between: <ul style="list-style-type: none"> 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.
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 SSL connection settings are available:

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.

Setting	Description
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.

Server Settings

The following server settings are available:

Database

The following database settings are available:

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.
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.

General

The following general settings are available:

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.

Setting	Description
Max users	Specifies the maximum number of users that may connect to the repository web browser 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:

Setting	Description
Provider URL	Specifies the URL for the LDAP provider.
Security protocol	Specifies the protocol to be used when connecting to the LDAP server. If you are using SSL, you should set this parameter to "ssl".
Default search base	Specifies the default LDAP search base to be used for authentication, roles, attribution, and self-registration.
Anonymous bind	Specifies that binding is anonymous. If this parameter is not selected, you must specify a bind DN and password.
Trusted server	Specifies that the LDAP server can be trusted and automatically creates a login in the repository for the remote LDAP user from the LDAP server.
Server type	Specifies the type of the LDAP server. You can choose between. <ul style="list-style-type: none"> • 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
Bind DN	Specifies the DN to bind against when building the initial LDAP connection. This user must have read rights on all user records that will be authenticated via the login module.
Bind password	Specifies the password to bind with when building the initial LDAP connection.
Filter	Specifies the filter to use when authenticating users.
Scope	Specifies the authentication scope. You can choose between: <ul style="list-style-type: none"> • onelevel [default] • subtree
Method	Specifies the method to use for authentication requests. You can choose between: <ul style="list-style-type: none"> • simple • DIGEST-MD5
Digest MD5 format	Specifies the DIGEST-MD5 bind authentication identity format. The default is DN.
Search base	Specifies the default authentication search base.
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.

Setting	Description
Scope	Specifies the role search scope. You can choose between: <ul style="list-style-type: none"> • onelevel [default] • subtree
Referral	Specifies the treatment of referrals. You can choose between: <ul style="list-style-type: none"> • 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 63.

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.

Server Status Settings

The following server status settings are available:

Setting	Description
Server Information	Displays the host name, port and status of the server. Click the Restart button the restart the browser service.

Setting	Description
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 Repository Web Browser

The following items may help if you experience problems using the repository web browser:

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.

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 72
- Document and folder permissions – provide access to individual documents, or to the contents of folders. See [Controlling Permissions for Repository Items](#) on page 79

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.

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

You can drop the tables of a repository database. If you are not connected to the repository, the Connect dialog box is displayed to let you identify the repository where you will drop the tables.

The Drop Repository Tables command removes all repository tables and their content from the database and interrupts the connection with the current database. You should be very careful when using this feature since you may lose all checked in documents in the database.

Warning! Drop Repository Database cannot be undone. In addition, you cannot drop the repository tables through a proxy connection. You must perform a direct ODBC connection instead.

Drop Tables Script

Drop statements appear first in the Repository Table Drop Script window. You can use this window to browse the statements.

If you need to modify the statements you have to copy the text, and paste it in a script interpreter. The Separator list lets you select the separator that corresponds to your interpreter. When you change separator using the Separator list, the script automatically displays this change.

You have to click Cancel in the Repository Table Drop Script window, then modify and execute the statements with your script interpreter.

Warning! The Repository Table Drop Script window is read-only. You cannot edit any statement.

Dropping Repository Tables

To drop repository tables :

1. Select **Repository > Administration > Drop Repository Tables** to display the Repository Table Drop Script window.
2. Click OK to confirm deletion.

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.

Out of the box, members of the External users group have only the Connect right granted, while members of Public have read access to everything in the repository, but the administrator can grant them 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 71).

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).

Parameter	Description
Provider URL	Specifies the URL for the LDAP provider.
Security protocol	Specifies the protocol to be used when connecting to the LDAP server. If you are using SSL, you should set this parameter to "ssl".
Default search base	Specifies the default LDAP search base to be used for authentication, roles, attribution, and self-registration.
Trusted server	Specifies that the LDAP server can be trusted. This parameter must be selected in order to enable access to the repository for LDAP-authenticated users.

Parameter	Description
Server type	Specifies the type of the LDAP server. You can choose between. <ul style="list-style-type: none"> • 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
Anonymous bind	Specifies that binding is anonymous. If this parameter is not selected, you must specify a bind DN and password.
Bind DN	Specifies the DN to bind against when building the initial LDAP connection. This user must have read rights on all user records that will be authenticated via the login module.
Bind password	Specifies the password to bind with when building the initial LDAP connection.
Filter	Specifies the filter to use when authenticating users.
Scope	Specifies the authentication scope. You can choose between: <ul style="list-style-type: none"> • onelevel [default] • subtree
Method	Specifies the method to use for authentication requests. You can choose between: <ul style="list-style-type: none"> • simple • DIGEST-MD5
Digest MD5 format	Specifies the DIGEST-MD5 bind authentication identity format. The default is DN.
Search base	Specifies the default authentication search base.
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: <ul style="list-style-type: none"> • onelevel [default] • subtree
Referral	Specifies the treatment of referrals. You can choose between: <ul style="list-style-type: none"> • 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 76).

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. Add to him to any appropriate groups (see *Inserting a User into a Group* on page 76).

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.

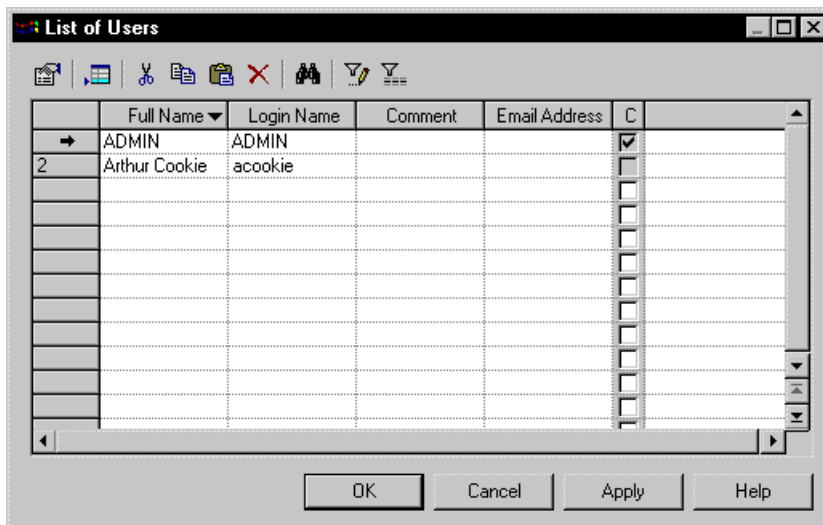
When you install the PowerDesigner repository, the following groups are created by default:

- 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 the different users of the repository.

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.

- Click the Properties button to open the property sheet of the user:

- 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.
- Click OK.

Each repository user has the following properties:

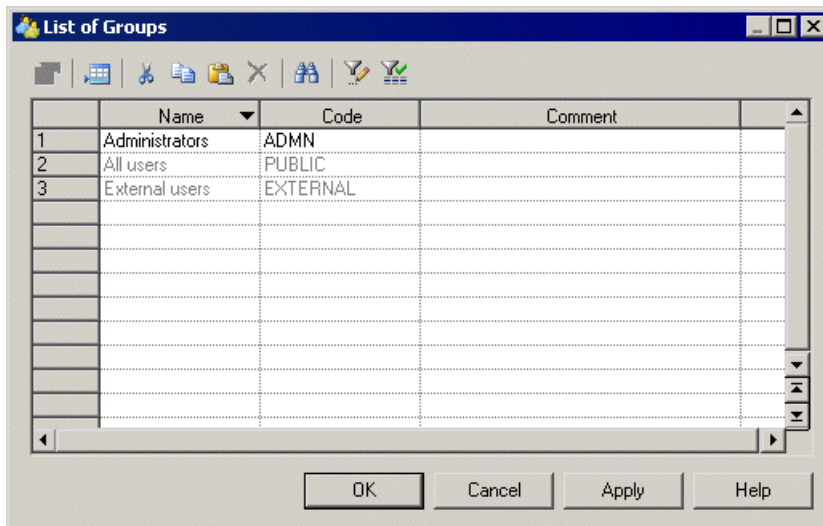
Property	Description
Full Name	Specifies the real name of the user.
Login Name	Specifies the name used for logging in to the repository.
Comment	Specifies any additional information about the user
Email	Specifies the email address of the user
Password	Specifies the password of the user (the next field lets you confirm the password)

Note: When you create a user, make sure you grant him/her access rights on the repository document either by inserting this user into a group (see [Inserting a user into a group](#) on page 76), or directly (see [Specifying a user's rights](#) on page 74). 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 of the repository.

- Select **Repository > Administration > Groups** to open the List of Groups, which lists, by default, the groups that are available upon repository initialization:



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 a group's rights](#) on page 75). If a user connects to the repository without access rights, he cannot see any document in the browser.

Specifying a User's Rights

A new user has no rights assigned by default. He belongs to the group PUBLIC that has no rights either. The repository administrator has to define the rights of a new 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
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 79.

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 36.
 - Lock Versions - User can lock document versions. See [Document Locks](#) on page 38.
 - Manage Branches - User can create, modify, and delete branches. See [Repository Branches](#) on page 40.
 - Manage Configurations - User can create, modify, and delete configurations. See [Repository Configurations](#) 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.

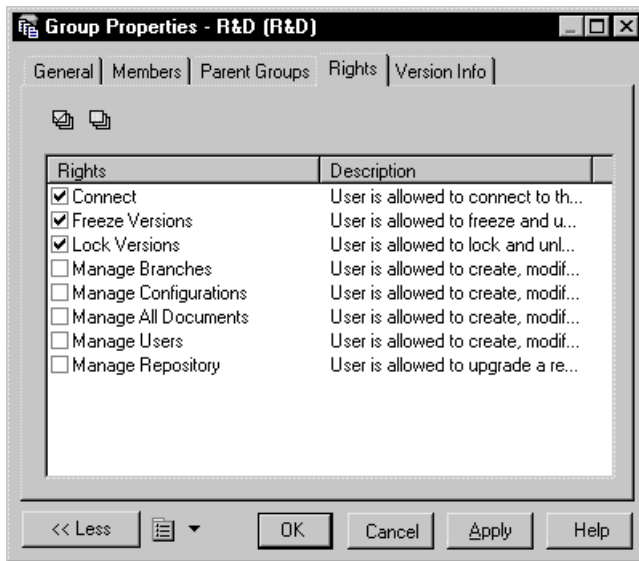
3. Click OK to save your changes and close the use property sheet.

Specifying a Group's 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: Group rights are always added to user rights, and never reduce rights granted individually to a user. Similarly, when you insert a group into another group, the sub-group retains all of its rights in addition to inheriting those of the parent group. For example, the group Designers has the right to freeze and lock versions in the repository. If you add a user who is also allowed to manage branches into this group, he retains this right and gains those of the group.

1. Select **Repository > Administration > Groups** to display the List of Groups dialog box.
2. Select a group and click the Properties tool to display the group 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 a user's rights](#) on page 74).



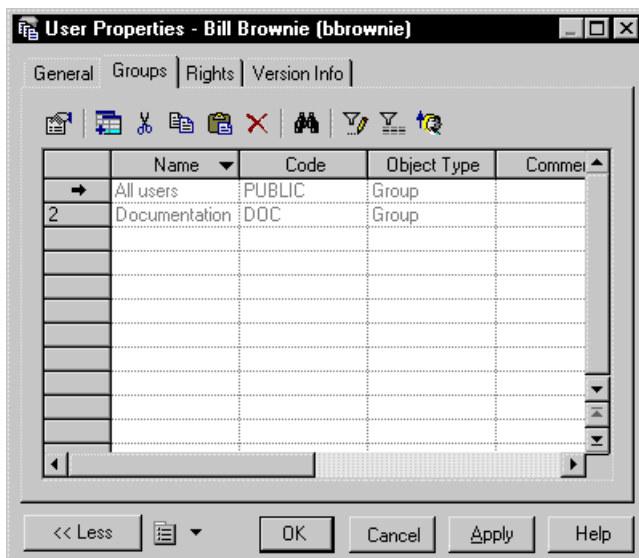
5. Click OK to close the property sheet.

Inserting a User into a Group

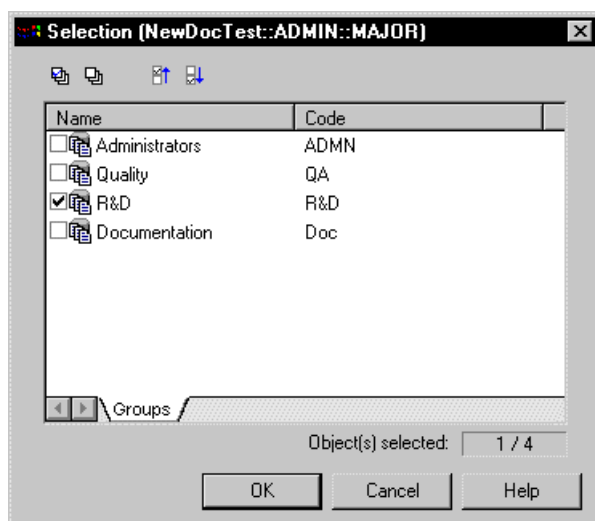
You can define groups of users in order to grant them rights and permissions on documents more easily. Each new user is automatically inserted into the PUBLIC group. You can insert a user into other groups.

For more information about creating a group, see [Creating a group](#) on page 73.

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.

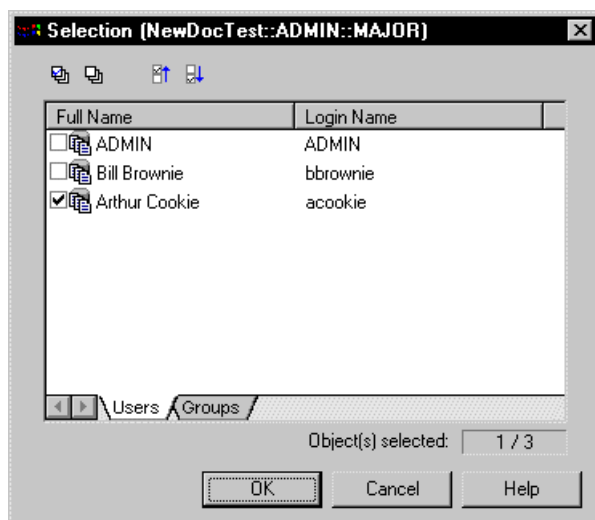
The groups appear in the list of groups.

Adding a User or a Group to a Group

You can add a user or a group to the current group. The user or group will then be granted the rights of the group in addition to his own.

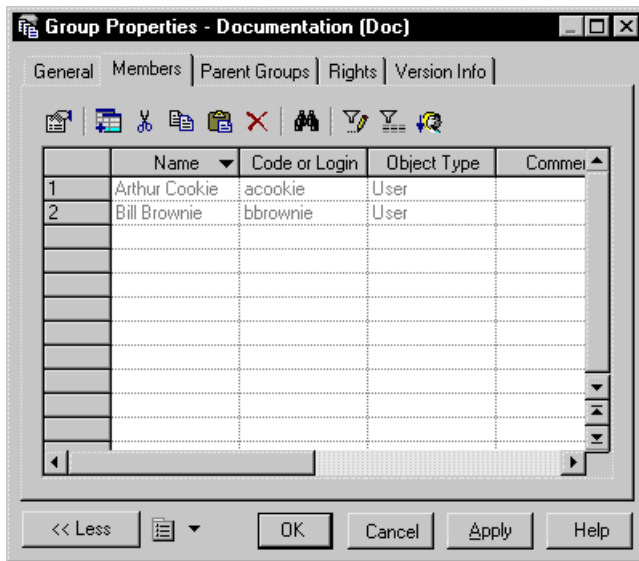
For information about how to insert a user into a group from the user's property sheet, see [Inserting a user into a group](#) on page 76.

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 users and groups appear in the list of group members.



- 6.** Click OK to close the group property sheet.

Inserting the Current Group into a Group

You can insert the current group into a larger group. For example, in a development team you have created several groups corresponding to the different teams: 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 click OK to add the group to them.

Deleting a Repository User

The repository administrator or a user with the Manage Users right can delete users. A user cannot delete himself, even if he has the Manage Users right.

Warning! Deleting a user cannot be undone.

1. Select **Repository > Administration > Users** to display the List of Users dialog box.
2. Select a user from the list and click the Delete tool:
 - If the user has created or modified documents – he remains in the List of Users (to retain the appropriate authorship traces) but his rights and permissions are canceled.
 - If the user has not created or modified documents - he is removed from the List of Users and completely removed from the repository.

Deleting a Group

You can delete a group 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 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 72).

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)

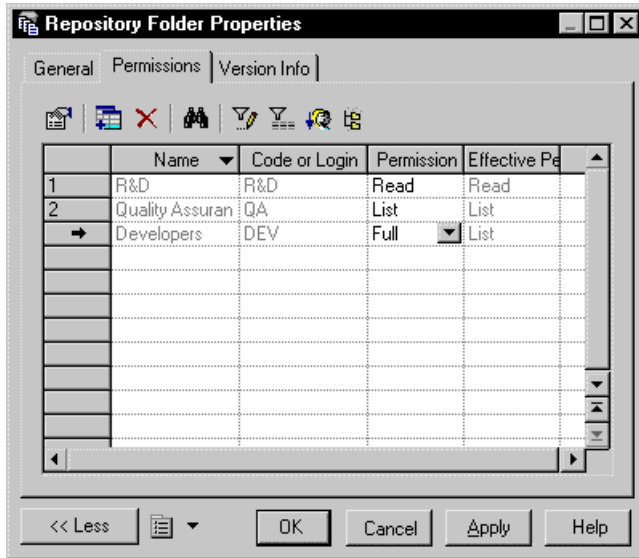
Defining Access Permissions on a Repository Item

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
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:

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

The node property sheet opens to the Permissions tab.



- Click the Copy Permissions to All Children tool.
- Click Apply.

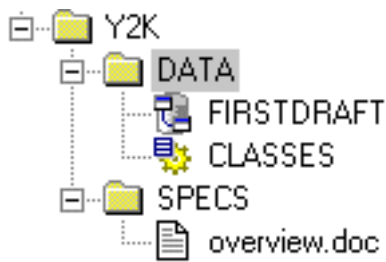
The permissions are copied to all child members, and the tool Copy Permissions to All Children tool is no longer pressed in the toolbar.

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
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:

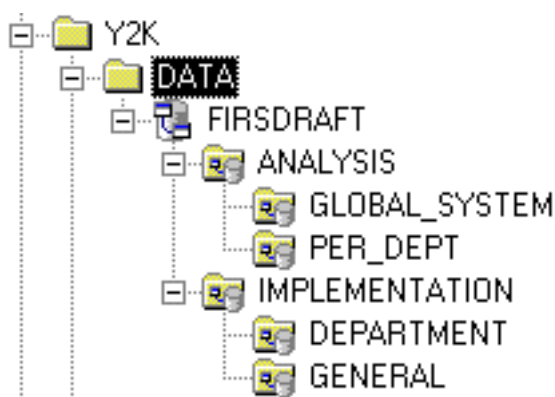
Groups or users	Right
Repository and Data Administrator	Connect, Manage All Documents, Manage Users, and Manage Repository
Project leader	Connect, Freeze Versions, Lock Versions, Manage Branches, and Manage Configurations
Developer team leader	Connect, Freeze Versions, and Lock Versions
Developers	Connect,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

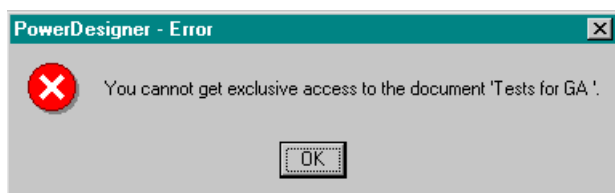


The following permissions are assigned to the developers:

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

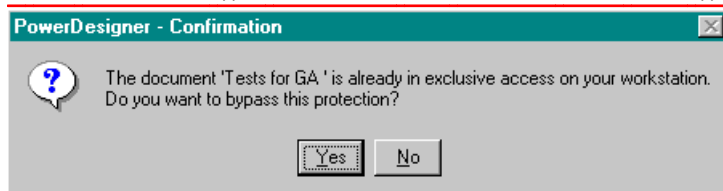
When you check in, freeze, lock or delete a document in the repository, a temporary protection is applied to all the versions of the document to ensure the update is completed correctly, and an error message is given if other updates are attempted.



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.

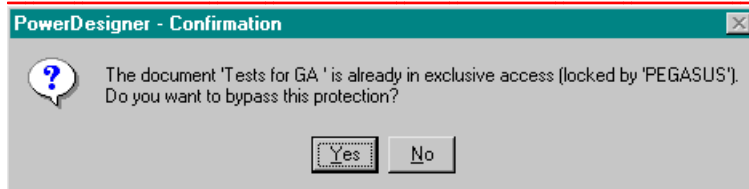
If the Document Was Protected and Blocked from Your Workstation



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

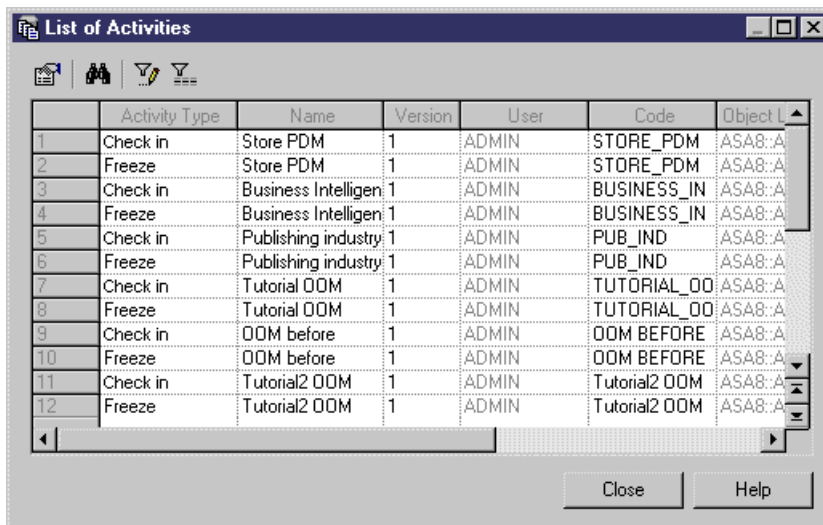
Users with the Manage All Documents right can audit repository use through the List of Activities. Activities are actions that modify repository documents, such as check in, freezing, and deleting.

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

- access
 - permission 79
 - protection 83
- activity (repository)
 - audit 84
 - consult 84
 - list 84
 - manage all documents 84
 - open list 84
- ADMIN 2, 4, 72
- audit repository activity 84

B

- backup repository database 69
- base branch 40
- batch update 16
- branch
 - base 40, 41
 - change 42
 - check in 9, 18
 - create 41
 - current 40
 - default 40
 - delete 44
 - delete version 38
 - manage 40
 - manage members 43
 - none 40
- browser
 - diagram 26
 - document 26
 - find objects 30
 - folder 26, 27
 - node 26
 - object 26
 - project 26
 - repository 26
 - root 26, 27

C

- change
 - branch 42
 - repository password 7
- change notifications 55
- check in 11
 - batch update 16
 - branch 18
 - conflict 17
 - current branch 9
 - document 10, 12
 - external application document 20
 - file 10
 - freeze current version 20
 - freeze version 9

- integrate 18, 19
- interactive update 16
- keep repository changes 16
- large numbers of locks in New mode 18
- large transactions 18
- lock 20
- Manage All Documents right 9
- merge conflict 17
- mode 18
- model 10
- multiple documents 15
- multiple files 11
- multiple models 11
- new 18
- package 20
- parameters 13
- permission 9
- PowerDesigner models 17
- project 11
- project file 10
- project model 10
- resolve manually 16
- resource 49, 52
- temporary protection 83
- undo 20
- update 17, 18
- update dependencies 34
- update options 16
- update version 9
- use my changes 16
- work organization 9
- Write permission 9
- check out
 - configuration 21
 - dependencies 21
 - document 20, 22
 - file 21
 - find objects 30
 - lock 21
 - merge 21
 - merge models 25
 - model 21
 - multiple documents 24
 - name 21
 - options 21
 - package 26
 - parameters 23
 - path 21
 - permissions 26
 - project 21
 - read-only 21
 - repository 21
 - update dependencies 34
 - version from configuration 47
 - version from project configuration 47
- Check Out Document 22
- client/server environment 1
- close repository connection 7

- compare
 - models 32
 - resources 52
- configuration
 - check out project 47
 - check out version 47
 - create 45
 - delete version 38
 - manage 44
 - manage versions 46
- conflict during check in 17
- connect
 - automatic 6
 - repository 35
- copy permissions to all children 81
- copy script 69
- current branch 40

D

- database
 - drop repository 69
 - size 37
 - upgrade 69
- delete
 - branch 44
 - document 38
 - empty folder 28
 - temporary protection 83
 - user 78
 - version 38
- dependency
 - external 33
 - external shortcut 33
 - intermodel generation 33
 - internal 33
 - update from repository 33, 34
- diagram 26
- direct connection 2, 4
- disconnect repository 7, 69
- display
 - data 35
- document 26
 - check in 10
 - check in multiple 15
 - check out 20, 22
 - check out multiple 24
 - copy permissions 81
 - delete 38
 - external application 20
 - GUID 18
 - lock 38, 39
 - properties 28
 - status 9
 - temporary protection 83
 - update 20
 - version 37

E

- execute
 - SQL 35
- execute script 69

- external application 20

F

- find objects
 - check out 30
 - display properties 30
 - model 30
 - Result list 30
- folder 26
 - copy permissions 81
 - create 27
 - delete 28
 - move 28
 - node 27
 - rename 28
- freeze
 - all document versions 37
 - temporary protection 83
 - undo check in 20
 - version 9, 37

G

- group
 - additive rights 75
 - code 73
 - create 73
 - group members 73
 - insert into group 77, 78
 - insert user 77
 - member of groups 78
 - name 73
 - permissions 73
 - repository 72
 - rights 75
 - user 76
 - user members 73, 77
- GUID 18

I

- impact analysis 33
- integrate mode for check in 19
- interactive update 16

L

- LDAP
 - controlling repository access 70, 71
- Local Document 13, 23
- lock 21
 - add configuration 39
 - add related models 39
 - all document versions 40
 - check in 40
 - check out 40
 - document 38
 - multiple documents 39
 - temporary protection 83
 - undo check in 20

- version 38
- locks in the server during check in 18
- log of repository activities 84

M

- manage
 - branches 40
 - lock 38
- manage all documents 84
- member of a branch 43
- merge 21
 - check out model 25
 - conflict 17
 - general option 25
- mode
 - branch 18
 - integrate 18, 19
 - new 18
 - update 18
- model
 - check in 17
 - check out a package 26
 - check out with merge 25
 - copy permissions 81
 - find objects 30
- models
 - compare 32
- modify script 69

N

- named path
 - creating for resource 47
- new mode for check in 18

O

- object 26
 - category 26
 - properties 30
- option
 - auto-connect 6
 - browser 35
 - check in 16, 35
 - check out 35
 - connection 35
 - merge models 25
 - update 16

P

- package 21
 - check in 20
 - check out 26
 - copy permissions 81
 - permissions 20
 - shortcuts 26
- password
 - change for a repository 7
 - user 72
- path 21

- permission
 - check in 9
 - copy on document 81
 - copy on package 81
 - copy on sub-folder 81
 - group 73
 - inherit 80
 - list 79, 80
 - no access 79
 - on documents 79
 - on folder 79
 - package 20, 26
 - read 79, 80
 - repository 79
 - Show All Granted Users 80
 - user 72, 80
 - write 79, 80
- project 26
- Project Documents 13, 23
- proxy connection 2, 4, 69

Q

- query
 - execute 35
 - SELECT 35

R

- rebuilding clustered index 69
- rename folder 28
- repository
 - access permission 79
 - activity 84
 - add document 10
 - admin 72
 - auto-connect 6
 - backup database 69
 - browser 26
 - change password 7
 - check in document 10
 - check out 21
 - check out document 20
 - connect 5
 - create user 72
 - data consistency 1
 - data integrity 1
 - database size 37
 - database upgrade 69
 - definition 2, 4
 - delete user 78
 - direct connection 2, 4
 - disconnect 7
 - drop script 69
 - drop tables 69
 - find objects 30
 - folder 27
 - general options 6
 - group 72, 73
 - LDAP parameters 70, 71
 - list of activities 84
 - proxy 1

- proxy connection 2, 4
 - purpose 1
 - rebuilding clustered index 69
 - shared data 1
 - sharing resources 47
 - teamwork 1
 - temporary protection 83
 - test definition 2, 4
 - update dependencies 33
 - update dependencies from repository 34
 - update document 20
 - upgrade 69
 - upgrade failure 69
 - user 72
 - user rights 74
 - web access 2
 - web browser 53
 - repository definition
 - ADMIN 2, 4
 - Repository Document 13, 23
 - repository web browser 53
 - branches 61
 - changing branch 62
 - client settings 63
 - comparing models 55
 - creating a branch 61
 - creating a group 58
 - creating a user 58
 - deleting a branch 62
 - deleting a group 61
 - deleting a user 61
 - document properties 54
 - groups 58–60
 - logging in 53
 - permissions 55
 - repository explorer 54
 - rights 59
 - server settings 64
 - settings 62
 - subscriptions 55
 - users 58–60
 - resource
 - sharing 48
 - automatic update 47
 - checking in 49, 51
 - checking in changes 52
 - comparing 51, 52
 - creating named path 47
 - creating template 49
 - creating user profile 50
 - enabling sharing 47, 49, 50
 - preparing sharing 47
 - sharing policies 51
 - sharing via the repository 47
 - Result list
 - find tab 30
 - right
 - freeze versions 37
 - group 75
 - user 74
 - root 26
 - browser 27
 - copy permissions 81
 - freeze all documents 37
 - lock all documents 40
- ## S
- script
 - copy 69
 - drop tables 69
 - execute 69
 - modify 69
 - SELECT query 35
 - shortcut of package 26
 - show
 - all parent groups 78
 - size
 - database 37
 - SQL query 35
 - subscriptions 55
- ## T
- table
 - drop script 69
 - template
 - creating for resource 49
 - temporary protection 83
 - bypass 83
 - document 83
- ## U
- undo check in 20
 - unfreeze
 - version 37
 - update
 - check in 18
 - dependencies from repository 34
 - document 20
 - file document 21
 - model 17
 - model document 21
 - options 16
 - project 21
 - version 9
 - upgrade 69
 - user
 - group 76
 - insert into group 76, 77
 - password 72
 - permissions 72
 - repository 72, 78
 - rights 74
 - show all parent groups 76
 - user profile
 - creating for resources 50
- ## V
- version
 - check out from configuration 47

- check out from project configuration 47
- configuration 46
 - delete 38
 - freeze 37
 - locks 38
 - manage 37
 - unfreeze 37
 - updatable 37

- user right 37

W

- web access 2
- web browser 53

