USER GUIDE | PUBLIC
SAP PowerDesigner
Document Version: 16.6.10 – 2019-12-17

Core Features Guide

SAP PowerDesigner Documentation Collection

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN SAP

Content

1 Modeling with PowerDesigner. 6

1.1 Getting Started with PowerDesigner. 6

The Welcome Page. 7

PowerDesigner Models. 9

Creating a Model 12

Accessing the Repository. 15

Connecting to a Database. 22

Managing Your PowerDesigner License. 30

PowerDesigner Documentation. 33

SAP Platform Support. 34

Next Steps. 36

1.2 The Browser. 37

The Glossary. 38

The Library. 42

The Repository Browser. 44

Workspaces 44

Folders 45

Models. 45

1.3 Projects and Frameworks. 46

Creating a Project. 48

Adding Model and File Documents to a Project. 49

Completing Framework Diagrams and Matrices. 57

Designing Framework Diagrams and Matrices. 65

Project Templates. 81

Project and Framework Properties. 85

1.4 Objects 92

Creating Objects from the Toolbox. 92

Creating Objects from the Browser. 94

Creating Objects from an Object List. 95

Importing Objects from Excel Files. 95

Importing Objects from XML Files. 102

Moving and Copying Objects. 103

Deleting Objects. 106

Object Properties. 107

Object Lists. 117

Finding Objects. 133

Business Rules. 137
File Objects. 141
Packages 143
Areas. 145
Traceability Links. 146
Extended Objects, Sub-Objects, and Links. 147
Checking Models. 148
Naming Conventions. 155
Spell Checking. 167
Extending Objects. 167
Applying Transformations to Objects. 183
Exporting Objects to Troux Semantics. 185
1.5 Diagrams, Matrices, and Symbols. 185
Diagrams 185
Dependency Matrices. 199
Symbols. 203
Display Preferences. 228
Importing and Exporting Model Graphics. 240
1.6 The Free Model (FEM). 241
Sketching Objects with the Free Model. 241
Creating New Types of Model with the Free Model. 242
1.7 The Repository. 243
Browsing the Repository. 244
Checking Documents into the Repository. 247
Change Lists. 259
Checking Documents Out of the Repository. 266
Document Versions. 273
Grouping Document Versions in a Configuration. 278
Branching Version Trees. 281
Impact and Lineage Analysis in the Repository. 290
Finding Objects in the Repository. 292
Comparing Models in the Repository. 293
Deleting Documents, Document Versions, and Folders. 295
Repository Web Client. 295
1.8 Reports. 295
Creating a Model Report. 296
Creating a List Report. 303
Creating a MultiModel Report. 311
The Report Editor. 312
The Legacy Report Editor. 334
Generating a Model Report. 356

Report Templates. 357
Report Properties. 358
1.9 Comparing and Merging Models. 366
Comparing Models. 366
Merging Models. 369
Excluding Objects and Properties from a Compare or Merge. 373
Filtering Objects and Changes During a Compare or Merge. 375
Repairing or Breaking Links Between Objects. 376
Previewing, Printing, and Saving Comparison Results. 377
1.10 Customizing Your Modeling Environment. 380
User Profiles. 381
General Options. 388
Organizing Views 402
Organizing Toolbars. 404
Customizing Menus, Toolbars, and Toolboxes. 405
PowerDesigner Resource Files. 408
Shortcut Keys. 416

2 Linking and Synchronizing Models. 420
2.1 Getting Started with Linking and Syncing. 420
Creating Links Between Models. 422
Viewing Links Between Models. 422
2.2 Generating Models and Model Objects. 423
Generating Models. 424
Generating Model Objects. 435
The Generation Links Viewer. 439
Using Impact Analysis with Generated Models. 441
2.3 Shortcuts and Replicas. 443
Creating Shortcuts. 446
Creating Replicas. 451
Synchronizing Shortcuts and Replicas. 458
Viewing Shortcut and Replica Dependencies. 459
Working with Target Models. 460
Generating Shortcuts and Replicas. 461
2.4 Object Mappings. 463
Creating a Data Source. 466
The Mapping Editor. 468
Creating Mappings from an Object's Property Sheet. 482
Object and Sub-Object Mapping Properties. 485
Metamodel Mapping Properties. 487
Metamodel Object Properties. 488
2.5 Impact and Lineage Analysis. 489

Reviewing and Modifying an Analysis. . . . 492
Reviewing and Modifying an IAM Model. . . . 497
Printing an Analysis. . . . 501
IAM Custom Checks. . . . 501
Obtaining Cross-Model Dependencies Without a Repository Connection. . . . 501
Analysis Rule Sets. . . . 503
Repository Rule Set Limitations. . . . 508

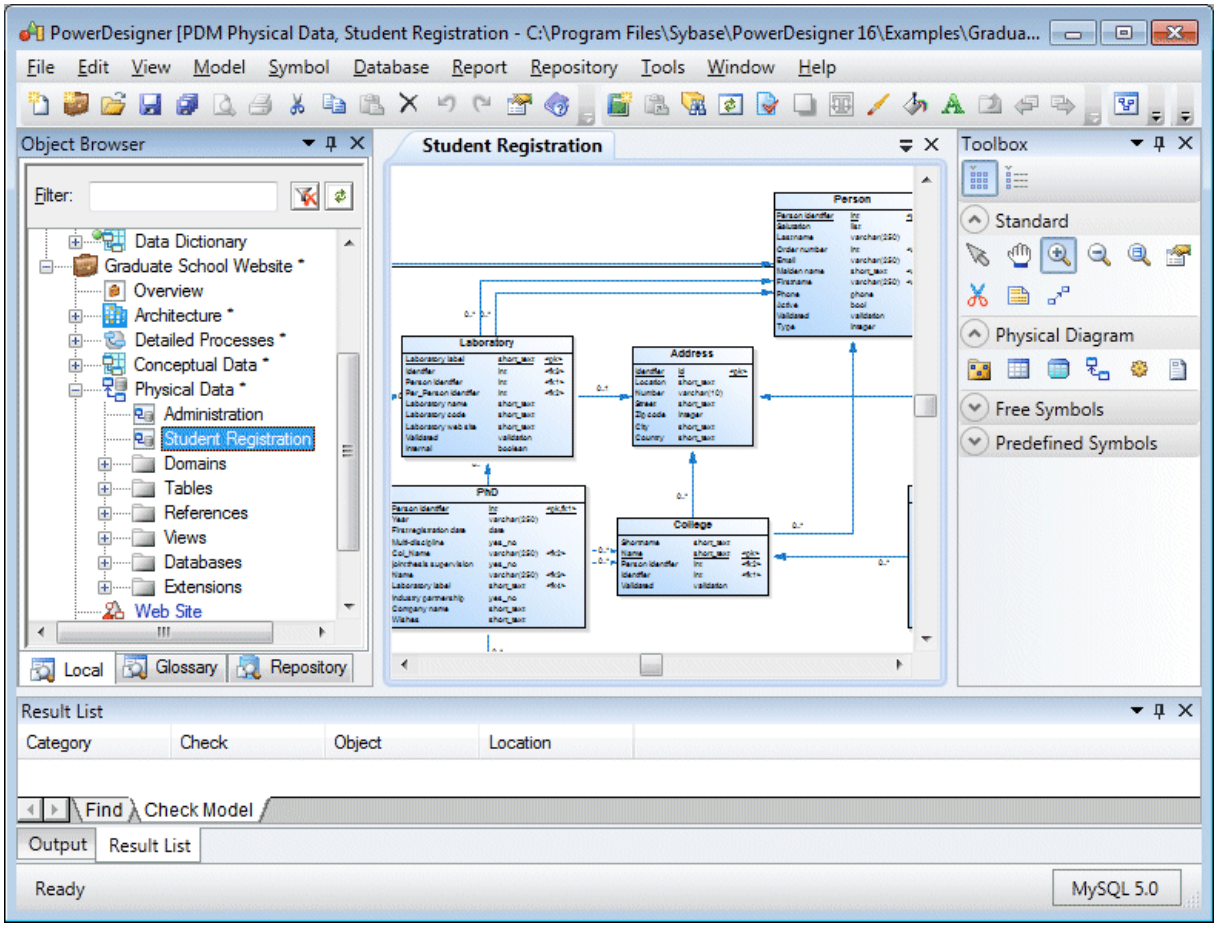
3 Administering PowerDesigner. . . . 510
3.1 Controlling Repository Access. . . . 510
Connecting to an LDAP Server for User Authentication. . . . 513
Connecting to an SMTP Server for Notifications. . . . 517
Defining a Password Policy. . . . 518
Enabling Secure Connections (SSL/TLS) for PowerDesigner Web. . . . 520
Enabling Single Sign-On for PowerDesigner Web. . . . 527
Creating Repository Users. . . . 530
Creating Repository Groups. . . . 531
Granting Privileges to Users and Groups. . . . 533
Granting Access Permissions on Repository Items. . . . 534
Unblocking Blocked Users 536
Deactivating Users. . . . 537
Monitoring Repository Activities. . . . 537
Querying the Repository Using SQL. . . . 539
Obtaining Emergency Access to the Repository. . . . 539
3.2 Deploying an Enterprise Glossary and Library. . . . 540
Deploying an Enterprise Glossary. . . . 540
Deploying an Enterprise Library. . . . 549
Controlling Library and Glossary Write Permissions. . . . 554
3.3 Customizing the PowerDesigner Interface. . . . 554
Customizing the New Model Dialog. . . . 554
Using Profiles to Control the PowerDesigner Interface. . . . 561
3.4 PowerDesigner Security Guide. . . . 569
Auditing and Logging. . . . 570
Data Protection and Privacy. . . . 570
Identity and Access Managment. . . . 571
Network and Communication Security. . . . 573
Cookies. . . . 575
Secure Delivery, Configuration, and Change Management. . . . 575
Operational Security. . . . 576

1 Modeling with PowerDesigner

The chapters in this part introduce you to the SAP® PowerDesigner® interface and the core concepts available for all kinds of models.

1.1 Getting Started with PowerDesigner

SAP® PowerDesigner® is a graphical enterprise architecture and design solution supporting standard methodologies and notations and providing automated code reverse engineering and generation through customizable templates. PowerDesigner provides powerful reporting capabilities, is highly extensible, and offers a scalable enterprise repository solution with strong security and versioning capabilities to aid multi-user development.



The default PowerDesigner window displays the following views:

- The Browser - displays your models and the objects belonging to them (see The Browser [page 37]). The Browser also allows you to view a PowerDesigner repository (see The Repository [page 243]) and may display an enterprise library of shared models (see The Library [page 42]) and an enterprise glossary (see The Glossary [page 38]).
- The Canvas - is a tabbed view that displays your open diagrams (see Diagrams [page 185]) and report outlines (see Reports [page 295]).
- The Toolbox - displays graphical tools to help you quickly build model diagrams. The available tools depend on the type of the currently selected diagram (see Creating Objects from the Toolbox [page 92]).
- The Result List - displays the results of a search or a model check.
- The Output view - (grouped, above, with the Result List) displays the progress of any PowerDesigner process, such as checking a model or generating or reverse engineering a database.

For information about moving, docking, grouping, splitting, and hiding these views, see Organizing Views [page 402].

In addition, you will work extensively with:

- Object property sheets - which permit you to view and edit the object's properties. You can access an object's property sheet by double-clicking its symbol or browser entry, or right-clicking it and selecting *Properties* (see Object Properties [page 107]).
- Object lists - provide a spreadsheet-like presentation of, and allow for the easy creation and modification of objects in your model. Lists are available under the *Model* menu and on the property sheets of composite objects such as entities, tables, and classes, which contain sub-objects (see Object Lists [page 117]).

1.1.1 The Welcome Page

The Welcome page gives you one-click access to all your recent projects, workspaces, and models, as well as providing a direct link to the New Model and New Project dialogs and a range of help materials. To suppress this page, select the *Do not show this page again* check box. You can redisplay it at any time by selecting *View* > *Welcome Page*.

By default, the following items are available:

- *Getting Started*
  - *Create Model* - Opens the New Model dialog to let you create a model (see Creating a Model [page 12]).
  - *Create Project* - Opens the New Project dialog to let you create a project (see Creating a Project [page 48]).
  - *Open Model or Project* - Opens a standard Open dialog to let you browse for a model or project to open.
  - *Examples* - Opens the PowerDesigner Examples folder to let you select an example model to open.
  - *Help* - Opens the PowerDesigner online help.
  - *What's New in PowerDesigner* - Opens the PowerDesigner New Features Summary.
  - *Documentation and Videos* - Opens a Web page listing the available PowerDesigner documentation and videos.
  - *PowerDesigner Web Site* - Opens the PowerDesigner page on the SAP® website.
  - *PowerDesigner Newsgroup* - Opens the PowerDesigner newsgroup page.
- *Recent projects and workspaces* - Lists your most recent projects and workspaces. Click a project or workspace to open it.

- *Recent models* - Lists your most recent models. Click a model to open it.

1.1.2 PowerDesigner Models

PowerDesigner enables you to closely integrate the design and maintenance of your application's core data layers with your project requirements, business processes, OO code, XML vocabularies, and database replication information and to align these with your overall enterprise architecture and corporate standards. By providing you with a comprehensive set of models at all levels of abstraction, PowerDesigner helps you broaden the reach of your iterative design process to all aspects of your system architecture, from conception to deployment, and beyond.

The following types of PowerDesigner models are available:

Icon	Description
	A requirements model (RQM) helps you analyze any kind of written requirements and link them with users and groups who will implement them and with design objects in other models. You can use an RQM to represent any structured document (e.g. functional specification, test plan, business goals, etc.) and import and export hierarchies of requirements as MS Word documents. File extension: **`.rqm`** Backup: **`.rqb`**
	An enterprise architecture model (EAM) helps you analyze and document your organization, its capabilities and processes, the applications and systems that support them, and the physical architecture on which they are implemented. File extension: **`.eam`** Backup: **`.eab`**
	A business process model (BPM) helps you identify, describe, and decompose business processes. You can analyze your system at various levels of detail, and focus alternatively on control flow (the sequence of execution) or data flow (the exchange of data). SAP® PowerDesigner® supports Analysis, SOA, DFD, SAP® Solution Manager, BPMN (including for SAP BPM), and BPEL process languages, and process simulation through SIMUL8. File extension: **`.bpm`** Backup: **`.bpb`**
	A conceptual data model (CDM) helps you analyze the conceptual structure of an information system, to identify the principal entities to be represented, their attributes, and the relationships between them. A CDM is more abstract than a logical (LDM) or physical (PDM) data model. File extension: **`.cdm`** Backup: **`.cdb`**
	A logical data model (LDM) helps you analyze the structure of an information system, independent of any specific physical database implementation. An LDM has migrated entity identifiers and is less abstract than a conceptual data model (CDM), but does not allow you to model views, indexes and other elements that are available in the more concrete physical data model (PDM). File extension: **`.ldm`** Backup: **`.ldb`**

Icon	Description
	A physical data model (PDM) helps you to analyze the tables, views, and other objects in a database, including multidimensional objects necessary for data warehousing. A PDM is more concrete than a conceptual (CDM) or logical (LDM) data model. You can model, reverse-engineer, and generate for all the most popular DBMSs. File extension: **.pdm** Backup: **.pdb**
	A data movement model (DMM) provides a global view of the movement of information in your organization. You can analyze and document where your data originates, where it moves to, and how it is transformed on the way, including replications and ETL. File extension: **.dmm** Backup: **.dmb**
	An object-oriented model (OOM) helps you analyze an information system through use cases, structural and behavioral analyses, and in terms of deployment, using the Unified Modeling Language (UML). You can model, reverse-engineer, and generate for Java, .NET and other languages. File extension: **.oom** Backup: **.oob**
	An XML model (XSM) helps you analyze an XML Schema Definition (.XSD), Document Type Definition (.DTD) or XML-Data Reduced (.XDR) file. You can model, reverse-engineer, and generate each of these file formats. File extension: **.xsm** Backup: **.xsb**
	A free model (FEM) provides a context-free environment for modeling any kind of objects or systems. You can use it to sketch simple visualizations or create entirely new forms of model by defining your own concepts and graphical symbols through an extension file. See The Free Model (FEM) [page 241]. File extension: **.fem** Backup: **.feb**
	The glossary model (GLM) helps you capture and organize the terminology to be used for naming your model objects. An administrator deploys the glossary, and users enable it in their models to provide autocompletion for object names and model checks to ensure compliance. The glossary appears in the Browser *Glossary* tab and updates are pushed to users each time they connect to the repository, or on demand. Only one glossary is permitted per repository. See The Glossary [page 38] and Deploying an Enterprise Glossary [page 540]. File extension: **.glm** Backup: **.geb**
	A multimodel report (MMR) is a PowerDesigner report that can document any number of models together and show the links between them. To create such a report, you must have at least one model open in the workspace, and you can add additional models at any time. See Reports [page 295]. File extension: **.mmr** Backup: **.bmr**

PowerDesigner does not impose any particular software engineering methodology or process. Each company can implement its own workflow, defining responsibilities and roles, describing what tools to use, what validations are required, and what documents to produce at each step in the process.

A development team will comprise multiple user roles, each of whom will use different PowerDesigner models:

- Business Analysts – define the architecture of the organization, the business requirements and high-level business flows, and may use:

    - EAM - for providing a big picture of the organization, defining its structure, and analyzing high-level functions, processes, and flows. These architectural objects can be attached to implementation objects in any of the other models.
    - RQM - for defining business requirements to be refined into technical requirements by Analysts and Designers.
    - BPM - to define the high-level business process flows in existing and new systems, and to simulate business processes to improve productivity.
- Data Analysts and Designers - map technical requirements to business requirements, and may use:
    - OOM - to define use cases and map them to requirements.
    - BPM - to write functional specifications and define more precisely the nature and details of each process, the application and its data structure.
    - CDM - to provide a clear view of business data across all systems.
- Database Administrators use the well-defined data structure to optimize, denormalize, and create the database, and may use:
    - PDM - to represent a database running on a server with complete information on the structure of the physical objects, such as tables, columns, references, triggers, stored procedures, views, and indexes, and to generate to and reverse-engineer from scripts or a live database connection. By maintaining a PDM and a CDM, you can ensure that your analysis and design efforts are reflected exactly in your actual systems.
    - LDM - which can act as a bridge between a CDM and a PDM, allowing you to resolve many-to-many and super/sub-type relationships, de-normalize your data structures, and define indexes, without specifying a particular RDBMS.
    - DMM - to represent the replication of information from a source database to one or several remote databases.
- Developers model the application, defining object structures and behaviors, and Object/Relational mappings, and may use:
    - RQM - to write technical specifications.
    - OOM - to represent objects and their interactions in UML, and to generate and reverse engineer OO code. Close integration with your BPM, CDM, and PDM can greatly simplify the maintenance and development of your system.
    - XSM - to graphically model the complex structure of an XML file and to generate and reverse-engineer DTDs, and XSDs and link them to your PDM and OOM.
- Testers will use the RQM, CDM, and other models, together with the design documents to understand how the application should work and how it is developed.

1.1.3 Creating a Model

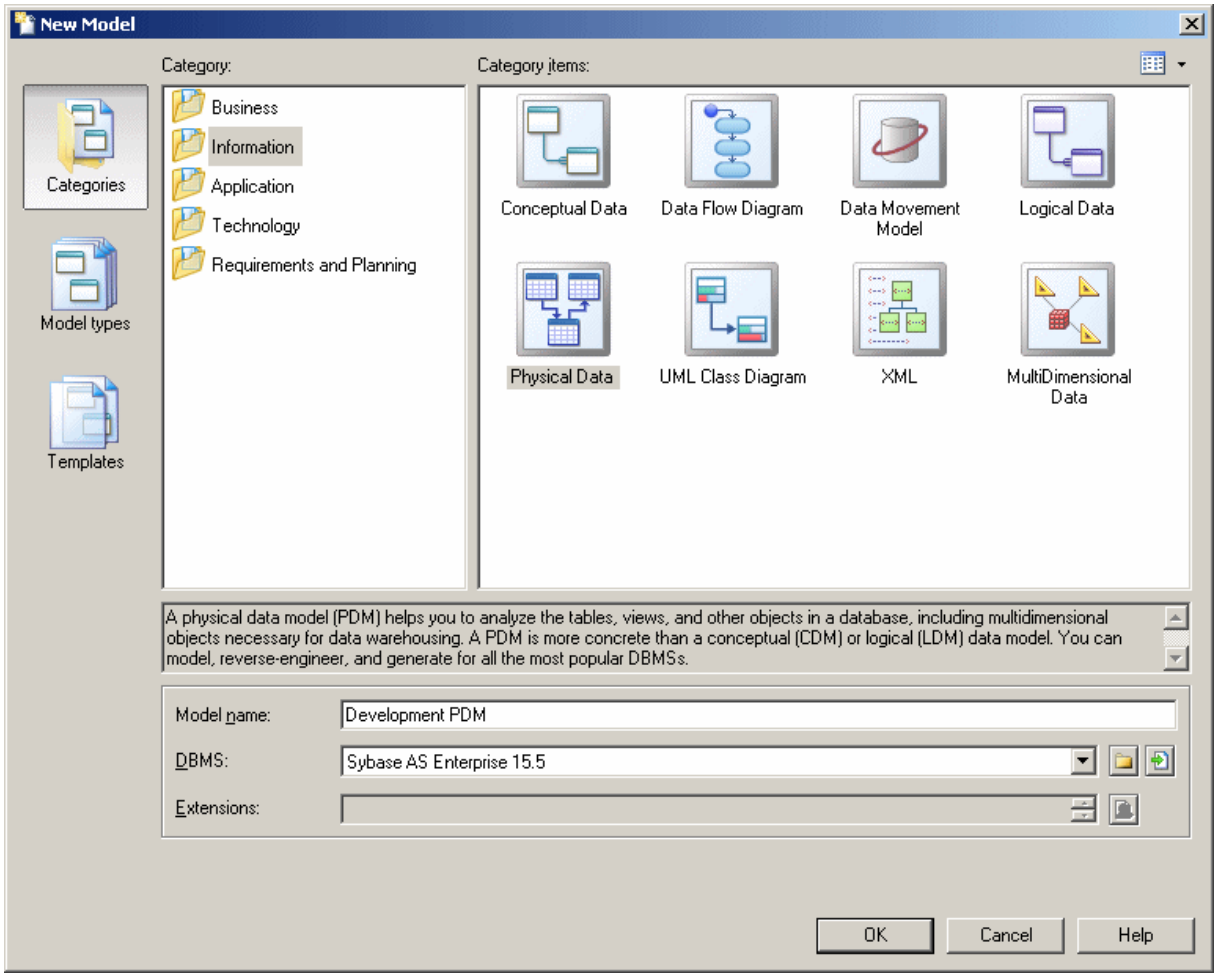
You create a new model by selecting *File* > *New Model*.

Context

> **i Note**
>
> A project can provide a convenient environment for working with multiple interconnected models and other files (see Projects and Frameworks [page 46]).

The New Model dialog is highly configurable, and your administrator may hide options that are not relevant for your work or provide templates or predefined models to guide you through model creation. When you open the dialog, one or more of the following buttons will be available on the left hand side:

- *Categories* - which provides a set of predefined models and diagrams sorted in a configurable category structure.
- *Model types* - which provides the classic list of PowerDesigner model types and diagrams.
- *Template files* - which provides a set of model templates sorted by model type.



Procedure

1. Select *File* > *New Model* to open the New Model dialog.
2. Click a button, and then select a category or model type (as appropriate) in the left-hand pane.

   For a list of PowerDesigner model types, see PowerDesigner Models [page 9].
3. Select an item in the right-hand pane. Depending on how your New Model dialog is configured, these items may be first diagrams or templates on which to base the creation of your model.

   Use the *Views* tool on the upper right hand side of the dialog to control the display of the items.
4. Enter a model name. The code of the model, which is used for script or code generation, is derived from this name using the model naming conventions.
5. [BPM, PDM, OOM, and XSM] Select a target *resource file* , which customizes PowerDesigner's default modifying environment with target-specific properties, objects, and generation templates.

   By default, PowerDesigner creates a link in the model to the specified file. To copy the contents of the resource and save it in your model file, click the *Embed Resource in Model* button to the right of this field. Embedding a file in this way enables you to make changes specific to your model without affecting any other models that reference the shared resource.
6. [optional] Click the *Select Extensions* button and attach one or more extensions to your model.
7. Click *OK* to create and open the model .

> **i** Note
>
> Sample models are available in the Example Directory.

1.1.3.1 Attaching Extensions At Model Creation Time

The Select Extensions dialog allows you to attach extensions to your model at creation time. Extensions may contain additional types of objects or extensions to standard objects, along with additional generation targets and other extensions to PowerDesigner's standard capabilities.

Procedure

1. Click the *Select Extensions* tool in the New Model dialog.
2. Review the different sorts of extensions available by clicking the sub-tabs and select one or more to attach to your model.

   By default, PowerDesigner creates a link in the model to the specified file. To copy the contents of the extension file and save it in your model file, click the *Embed Resource in Model* button in the toolbar. Embedding a file in this way enables you to make changes specific to your model without affecting any other models that reference the shared resource.

3. Click *OK* to close the dialog and return to the New Model dialog.

> **i Note**
>
> You can attach extensions to your model after creation by selecting *Model* > *Extensions* and clicking the *Attach an Extension* tool.

1.1.3.2 Model Templates

A model template is a shell model that can set model options and display preferences and contain extensions and objects to initialize new models. Model templates are stored in the model templates directory, which is, by default, in your Documents folder.

Context

You can make any model available as a model template by saving it to the model templates directory. Alternatively, you can manage your model templates from the New Model dialog by selecting *File* > *New Model* and clicking the *Templates* button).

> **i Note**
>
> The New Model dialog is highly customizable. If the *Enable model template files* option is deselected or if no valid template directory is selected then the *Templates* button will not be available (see Model Creation Options [page 401]).

The New Model dialog lists the model templates that are available, sorted by model type, and provides the following tools:

Tool	Description
	Copy Model as Model Template – allows you to select one or more models as model templates, and copies them to the model templates folder.
	Change Model Templates Folder – allows you to select a folder for use as the model templates folder.

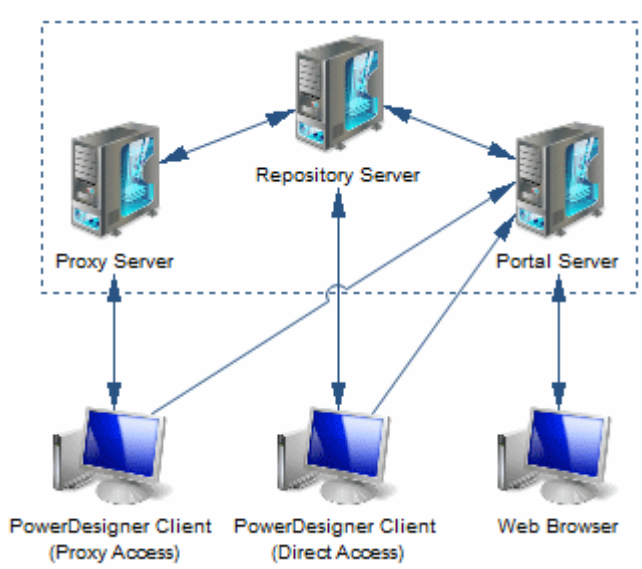
> **i Note**
>
> You can make model templates available to other users by checking them into the repository `Library/Resources/Model Templates` folder (see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

1.1.4 Accessing the Repository

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

The following diagram shows the various ways that users can connect to the repository:

- Proxy Access [if the repository proxy is installed] - see Setting up PowerDesigner for Access Via the Proxy [page 18]
- Direct Access – see Setting up PowerDesigner for Direct Access [page 15]
- Web Browser [if PowerDesigner Web is installed] – see (see *PowerDesigner Web*



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

1.1.4.1 Setting up PowerDesigner for Direct Access

To create a direct repository definition, you must have a login name and password for the repository AND a user name and password for the database where it is stored. You may also need to install a database client or

driver in order to connect to the repository database. Your repository administrator will supply you with this information.

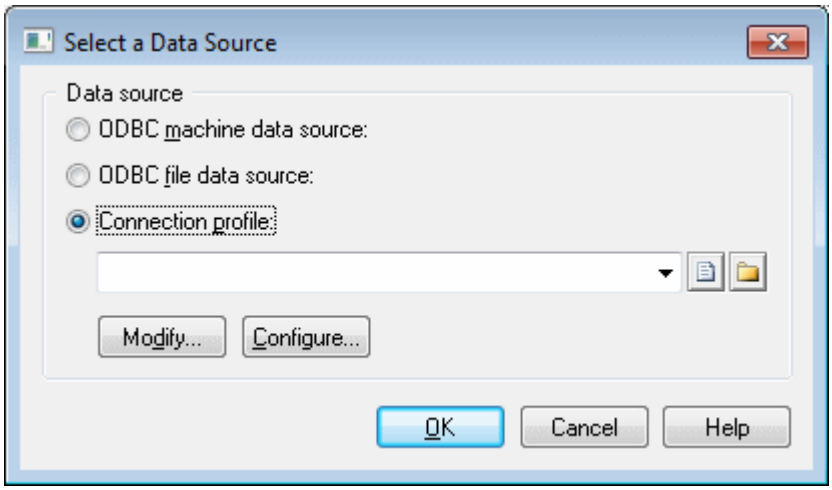
Context

If your administrator has installed the repository proxy, see Setting up PowerDesigner for Access Via the Proxy [page 18].

> **i Note**
>
> For information about installing the repository and making the first connection to it as an administrator, see the *Installation Guide* .

Procedure

1. Select *Repository* > *Repository Definitions* to open the List of Repository Definitions.
2. Click the *Add a Row* tool and type a name in the *Repository Name* column to identify the definition for future use.
3. Type your user name in the *Repository User* column. If you are an administrator and are creating the first repository definition that will be used to initialize the repository (see Controlling Repository Access [page 510]), you must use the `ADMIN` repository user name.
4. Press F4 or click the Ellipsis button in the *Data Source Name* column to open the Select a Data Source dialog:



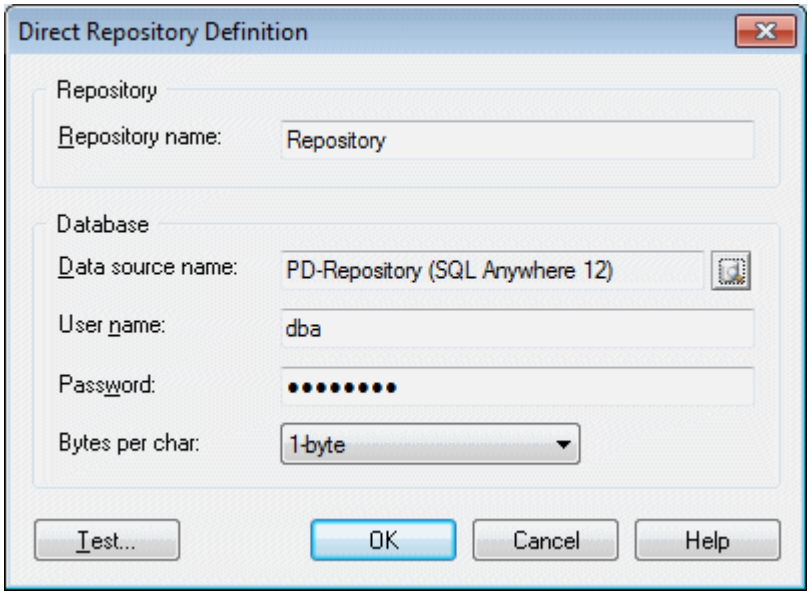
5. Select one of the following options:
   - ODBC machine data source
   - ODBC file data source

- Connection profile (for JDBC connections)

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

For detailed information about creating, configuring, and using connection profiles, see Configuring Connection Profiles [page 27].

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



This window has two group boxes:

- *Repository* – specifies the name of the repository definition.
- *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. The user name and password are optional, but if you do not specify them here you will need to enter them each time you connect to the repository. For non-Oracle unicode or multi-byte character set databases, you should also specify the bytes per character used by the database:
  - 1-byte - For SBCS (Single-Byte Character Set)
  - 2-byte - [default] For DBCS (Double-Byte Character Set)
  - 3-byte - For Unicode or MBCS (Multi-Byte Character Set)

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

1.1.4.2 Setting up PowerDesigner for Access Via the Proxy

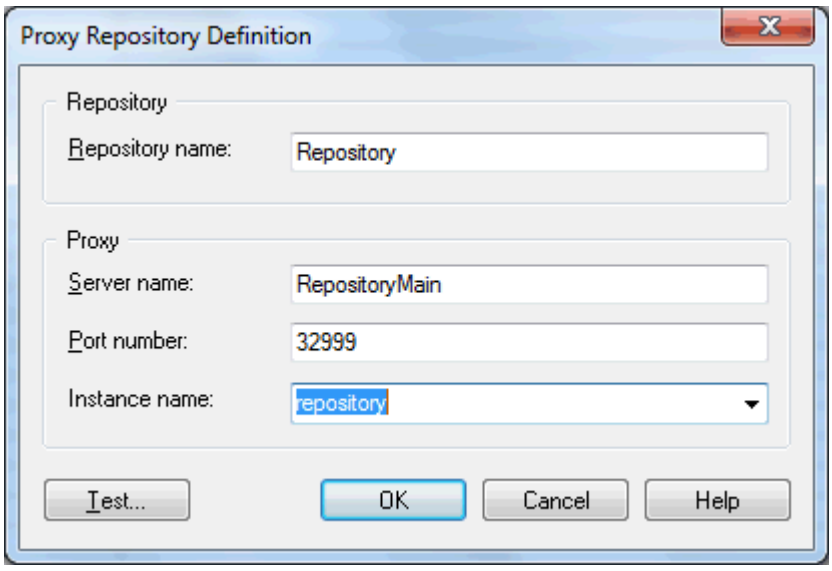
To create a proxy repository definition, you must have a login name and password for the repository and know the server on which the proxy is installed. Your repository administrator will supply you with this information.

Context

If you are connecting directly, see Setting up PowerDesigner for Direct Access [page 15].

Procedure

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



This window has two group boxes:

- *Repository* – specifies the name of the repository definition.
- *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.

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

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

1.1.4.3 Connecting to a Repository

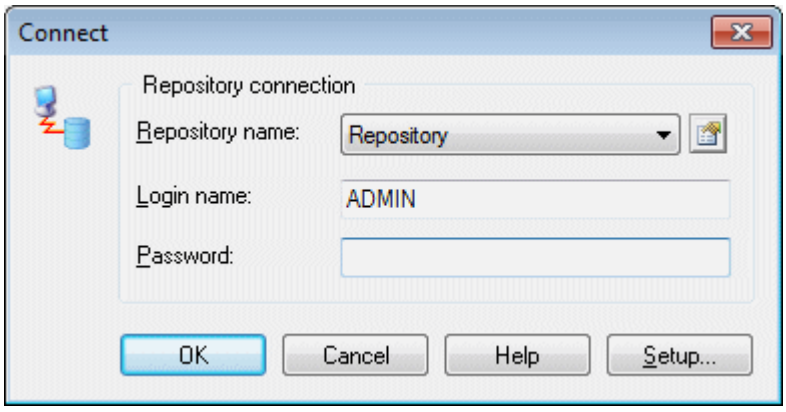
Before you can connect to a repository, an administrator must install and configure it and provide you with a user name and password, and you must have created a repository definition.

Context

For information about creating a repository definition, see Accessing the Repository [page 15].

Procedure

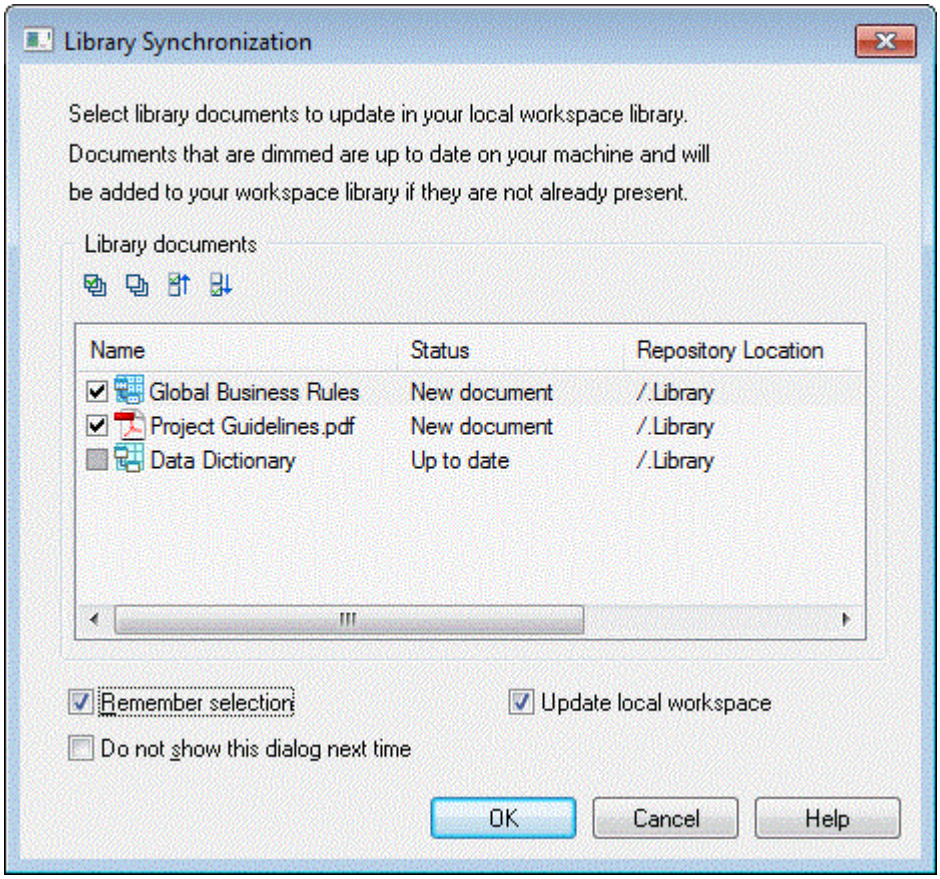
1. Select *Repository* > *Connect* (or double-click the root in the Browser *Repository* tab) to open the Connect dialog.



2. Select a repository to connect to and enter your login name and password. To review the repository definition, click the *Properties* tool.
3. Click *OK* to connect.

   The contents of the repository are displayed in the *Repository* tab of the Browser. The root node displays the name of the repository and your login.

   If any of the local copies of Library files (see The Library [page 42]) are not up to date, the Library Synchronization dialog will open to invite you to update them:



4. Select documents to update in your local library folder, and set options as appropriate:

Option	Description
Remember selection	Specifies to retain the same selection for updating next time you connect to the repository.
Update local workspace	Specifies to update the files displayed in the **`Library`** folder in the local workspace. Deselecting this option allows you to display only the subset of library files that you commonly use in the **`Library`** folder (or to hide the folder altogether), while keeping a wider selection of library files on your client machine up-to-date.
Do not show this dialog next time	Specifies to have the library update silently from now on without displaying this dialog.

> **i Note**
>
> You can display the Library Synchronization dialog and change your settings at any time by selecting *Repository* > *Synchronize Library*.

5. Click *OK* to perform any requested updates and begin modeling.

   To review the information stored for your repository user account, select *Repository* > *User Account*.

To disconnect from the repository, select *Repository* > *Disconnect* (or right-click the root in the Browser *Repository* tab and select *Disconnect*).

1.1.4.3.1 Connecting Automatically to the Default Repository

You can specify that you want to connect automatically to a repository each time you open PowerDesigner. On opening PowerDesigner will prompt for your password to the repository to which you were last connected.

Procedure

1. Select *Tools* > *General Options* to open the General Options dialog and click the Repository category.
2. Select *Auto-connect* in the Connection group box, and then click *OK*.
   For information about the other options on this tab, see Repository Options [page 396].

1.1.4.3.2 Changing Your Repository Password

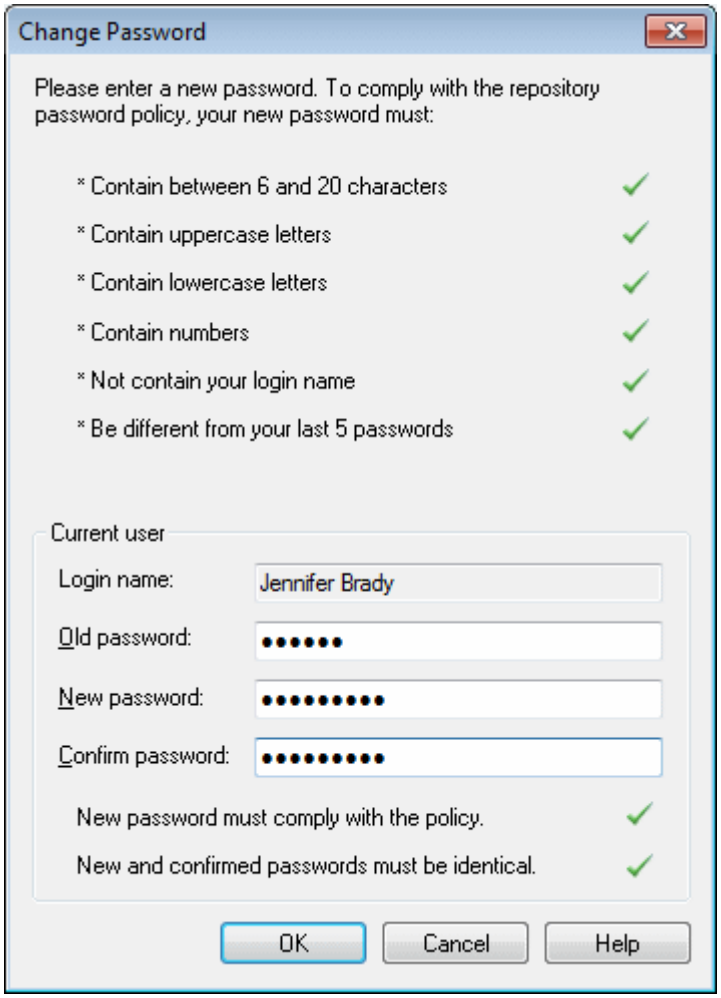
You can change the password that you use to connect to the repository at any time. You may be required to change your password if it has expired or if the password policy has become more restrictive.

Context

> **i Note**
>
> In the event that you have forgotten your password or have been blocked for multiple failed logins or inactivity, you must ask your administrator to reset your password (see Unblocking Blocked Users [page 536]).

Procedure

1. Select *Repository* > *Change Password* to open the Change Password dialog.
   The dialog lists the rules of the repository password policy that all passwords must respect (see Defining a Password Policy [page 518]).
2. Enter your old password, and then enter your new password twice.



When your new password conforms to all the policy rules and the values entered in the *New password* and *Confirm password* fields are identical, the *OK* button becomes available.

3. Click *OK* to complete the change and return to the repository.

1.1.5 Connecting to a Database

PowerDesigner allows you to define data connections to access information in different database management systems (DBMSs) using Structured Query Language (SQL).

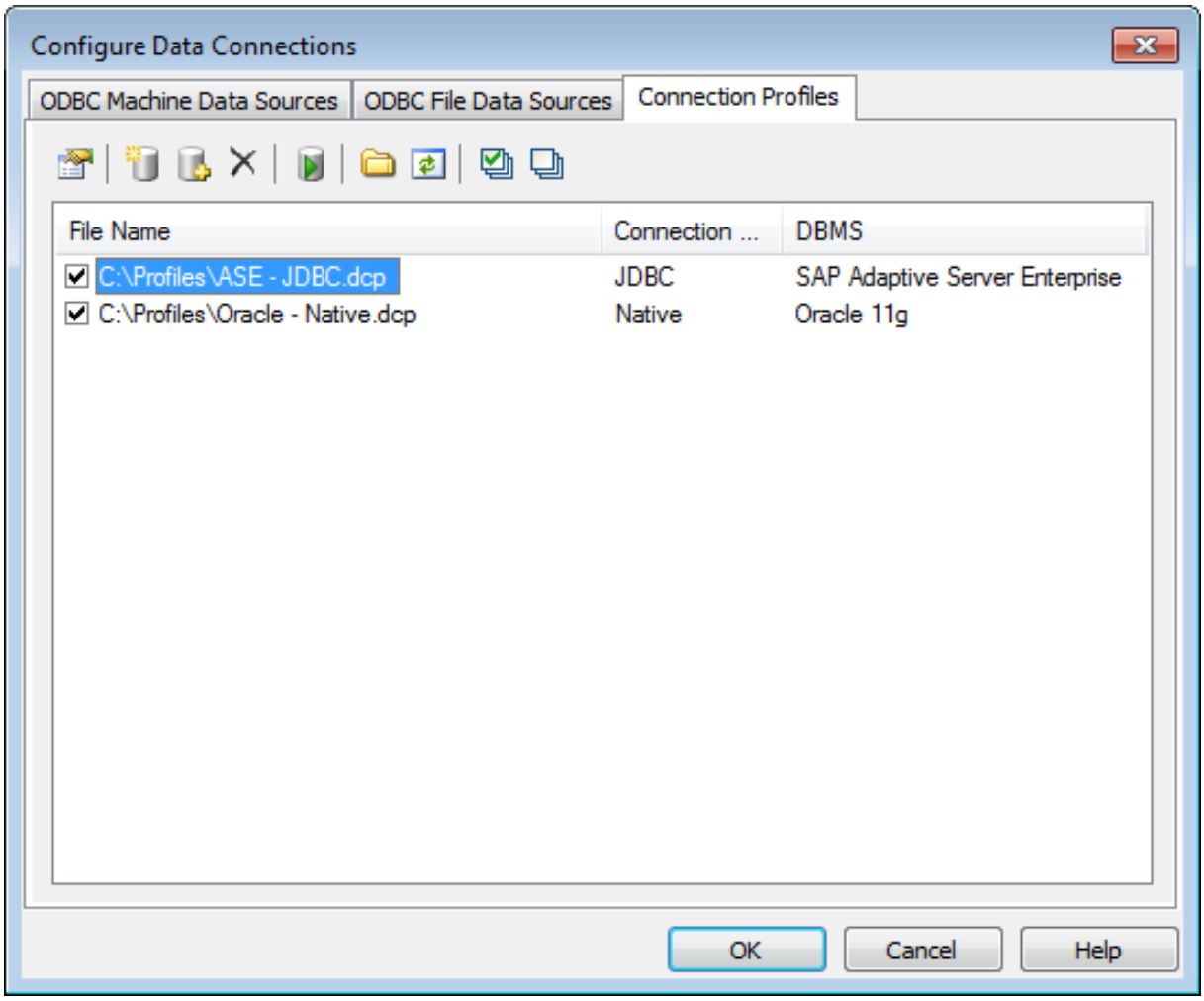
A data connection processes function calls and SQL requests coming from PowerDesigner and sends them to a data source, and returns results to PowerDesigner.

The PowerDesigner physical data model (see *Data Architecture*) and data movement model (see *Data Movement Architecture*) allow you to define data connections for target databases in order to generate your models and reverse engineer your data sources, while the repository (see The Repository [page 243]) requires a database to store your models and other design documents.

PowerDesigner supports the following forms of connection:

You have...	Configure a connection of type...
ODBC driver	ODBC machine or file data source (see Configuring ODBC Machine and File Data Sources [page 24]).
JDBC driver	JDBC connection profile (see Configuring Connection Profiles [page 27]).

To configure your database connections, select *Database* > *Configure Connections*:



> **Note**
>
> In order to create a data source on a Windows 7 machine, you may need to launch PowerDesigner or the ODBC Data Source Administrator by right-clicking its icon or menu item and selecting *Run as administrator*.

The following tools are available on each of the tabs:

Tool	Description
	Properties – Opens the ODBC Setup or Connection Profile Definition dialog for the selected profile.
	Add Data Source – Creates a new connection.
	Browse Data Source File [ODBC file data sources and connection profiles] – Opens a file browser to select a .dcp profile file.
	Delete – Deletes the selected connection.
	Test Connection – Tests the selected connection.
	ODBC Administrator [ODBC machine and file data sources] – Opens the ODBC Data Sources Administrator window.
	Change Connection Profiles Directory [Connection profiles] – Opens a file browser in which to search for profiles. The default directory is `<install_dir>/Connection Profiles`.
	Refresh – Refreshes the list of connections.
	Select All – Selects all the connections in the list. Connections that are selected will be displayed in lists in the PowerDesigner interface.
	Unselect All - Unselects all the connections in the list. Connections that are not selected will not be displayed in lists in the PowerDesigner interface.

1.1.5.1 Configuring ODBC Machine and File Data Sources

Before you can connect to a database through ODBC, you must create an appropriate ODBC data source. You define data sources using the Windows ODBC Administrator, which you can access directly from PowerDesigner.

Procedure

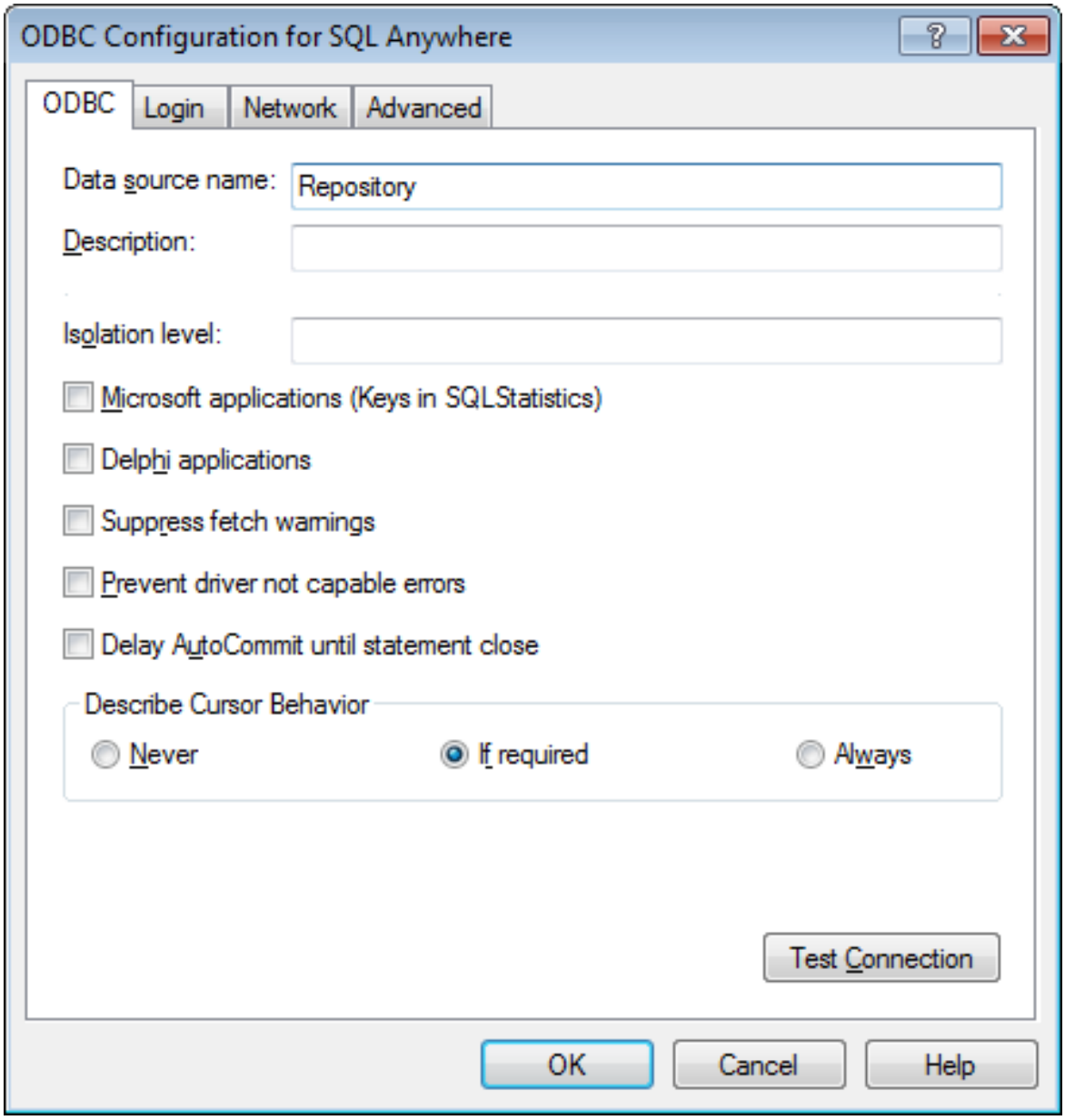
1. Select *Database* > *Configure Connections* to open the Configure Data Connections window, and click the ODBC Machine Data Source or ODBC File Data Sources tab:

Data source	Description
Machine	The data source is created on the client machine (stored in the current user settings in the registry), and is only available to the user currently logged onto the system.

Data source	Description
System	The data source is created on the client machine (stored in the machine settings in the registry), and is available to all users of the machine.
File	The data source is stored as a file with the extension .dsn, which can be used by users if it is placed in the default location for file data sources. File data sources are usually managed by database administrators.

> **i Note**
>
> In order to create a data source on a Windows 7 machine, you may need to launch PowerDesigner or the ODBC Data Source Administrator by right-clicking its icon or menu item and selecting *Run as administrator*.

2. Click the *Add Data Source* tool to open the *Create New Data Source* dialog, select a File, User, or Machine data source as appropriate and click *Next*.
3. Select the appropriate driver for your DBMS and click *Next* and then *Finish* to access a driver-specific configuration dialog:



4. Enter your DBMS parameters, then click *OK* to close the profile definition, and return to the Configure Data Connections window, and click *OK* again to return to your model.

1.1.5.2 Configuring Connection Profiles

Before you can connect to a database through a JDBC driver, you must create an appropriate connection profile.

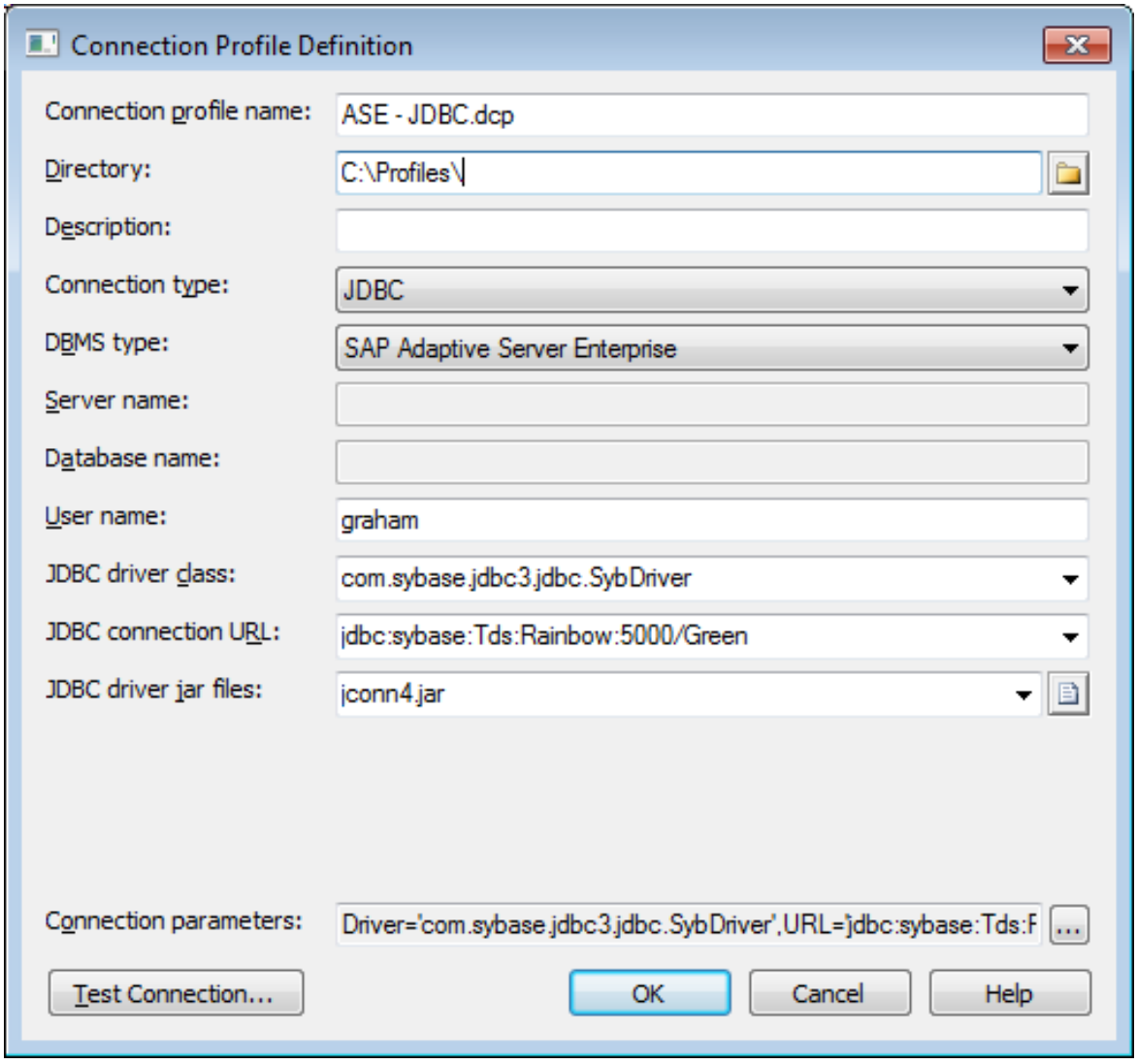
Procedure

1. Select *Database* > *Configure Connections* to open the Configure Data Connections window, and click the *Connection Profiles* tab.
2. Click the *Add Data Source* tool to open the *Connection Profile Definition* dialog, and enter the following properties:

Property	Description
Connection profile name	Specifies the name of the connection profile.
Directory	Specifies the directory in which the .dcp connection file will be created. By default .dcp files are stored in the Connection Profiles directory directly beneath the PowerDesigner installation directory.
Description	Optional additional description of the connection profile.
Connection type	Specifies the type of connection profile. JDBC is supported.
DBMS type	Specifies the DBMS to which the connection profile will connect. The list includes only those DBMSs supported for JDBC.
Server name	[not used]
Database name	[not used]
Service	[not used]
User name	Specifies the username to use when connecting.
Password	Specifies the password to use when connecting.
JDBC driver class	Specifies the driver class to use for the connection.
JDBC connection URL	Specifies the connection URL to use for the connection.
JDBC driver jar files	Specifies the driver jar file to use for the connection.

Property	Description
Connection parameters	[read-only] Lists advanced connection parameters. Click the parenthesis tool to the right of this field to add parameters. In the Connection Parameters dialog, click the *Add a Row* tool to create a new parameter and enter a name and value pair in the two columns. To connect to a HANA server over SSL, you must use the `encrypt=true` connection option and may need also to use the `validateCertificate=true` and `hostNameInCertificate=<fullyQualifiedHostName>` options. For more information about these and other HANA connection properties, see JDBC Connection Properties in the *SAP HANA Client Interface Programming Reference*.

In this example, I have installed the SAP® Adaptive Server® Enterprise JDBC driver:



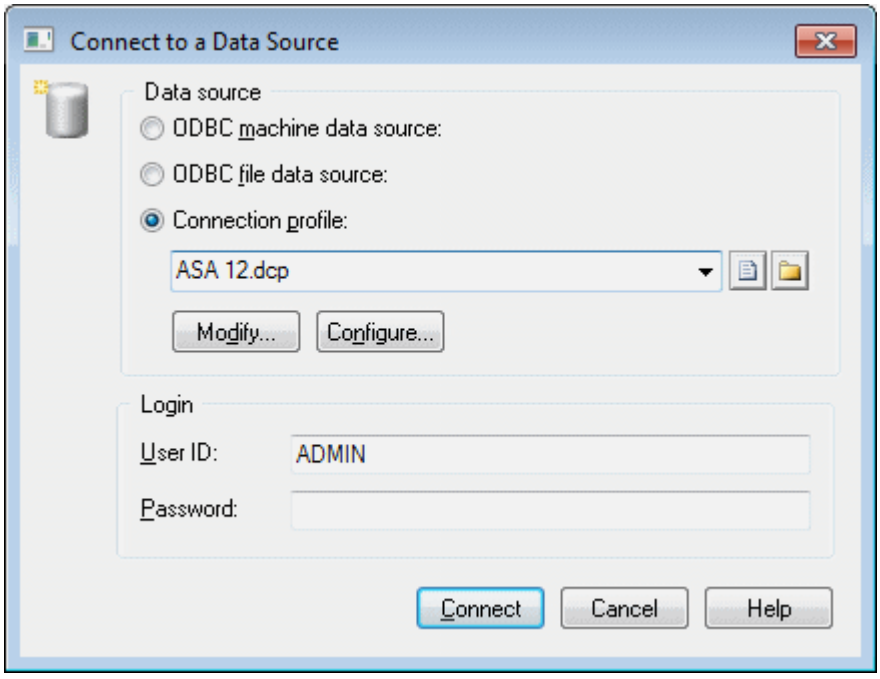
3. Click the *Test Connection* button to open the Test Connection window, enter the necessary password, and click *OK* to test your connection.
4. Click *OK* to close the profile definition, and return to the Configure Data Connections window, and click *OK* again to return to your model.

1.1.5.3 Connecting to a Data Source

When you connect to your database, PowerDesigner can communicate with it for reverse-engineering, generation or any other form of request.

Procedure

1. Select *Database* > *Connect* to open the Connect to a Data Source window:



2. Select one of the following radio buttons, depending on your chosen method for connecting to your data source:
   - ODBC machine data source (see Configuring ODBC Machine and File Data Sources [page 24])
   - ODBC file data source (see Configuring ODBC Machine and File Data Sources [page 24]) - use the tool to the right of the data source field to browse to a new file
   - Connection profile (see Configuring Connection Profiles [page 27]) - use the tools to the right of the data source field to browse to a new directory or file

   You can use the *Modify* and *Configure* buttons to modify or configure the selected connection.

3. Enter your user ID and password, and then click *Connect*. If prompted by your data source, you may need to enter additional connection parameters.

> **i Note**
>
> You stay connected until you disconnect or terminate the shell session.

1.1.5.4 Running SQL Queries Against Your Database

You can query your PDM database by selecting *Database* > *Execute SQL*. Administrators can query the repository database by selecting *Repository* > *Administration* > *Execute SQL*.

In addition to the standard tools (see Text Editor Tools [page 226]), the following tools are available in this dialog:

Tool	Description
	Run (F5). Executes the current script
[n/a]	Insert Bookmark (Ctrl + F2) – inserts a blue bookmark box at the cursor position. Press *Ctrl+F2* to delete the bookmark.
[n/a]	Go to Next Bookmark (F2)
[n/a]	Go to Previous Bookmark (Shift + F2)

1.1.6 Managing Your PowerDesigner License

You must select a license when installing PowerDesigner. If you subsequently need to change your type of license or the package of modules you use, then you should use the License Management wizard. If you use a served license and will be out of contact with the license server, you should request a temporary mobile license.

Context

You may need to access the License Management wizard when:

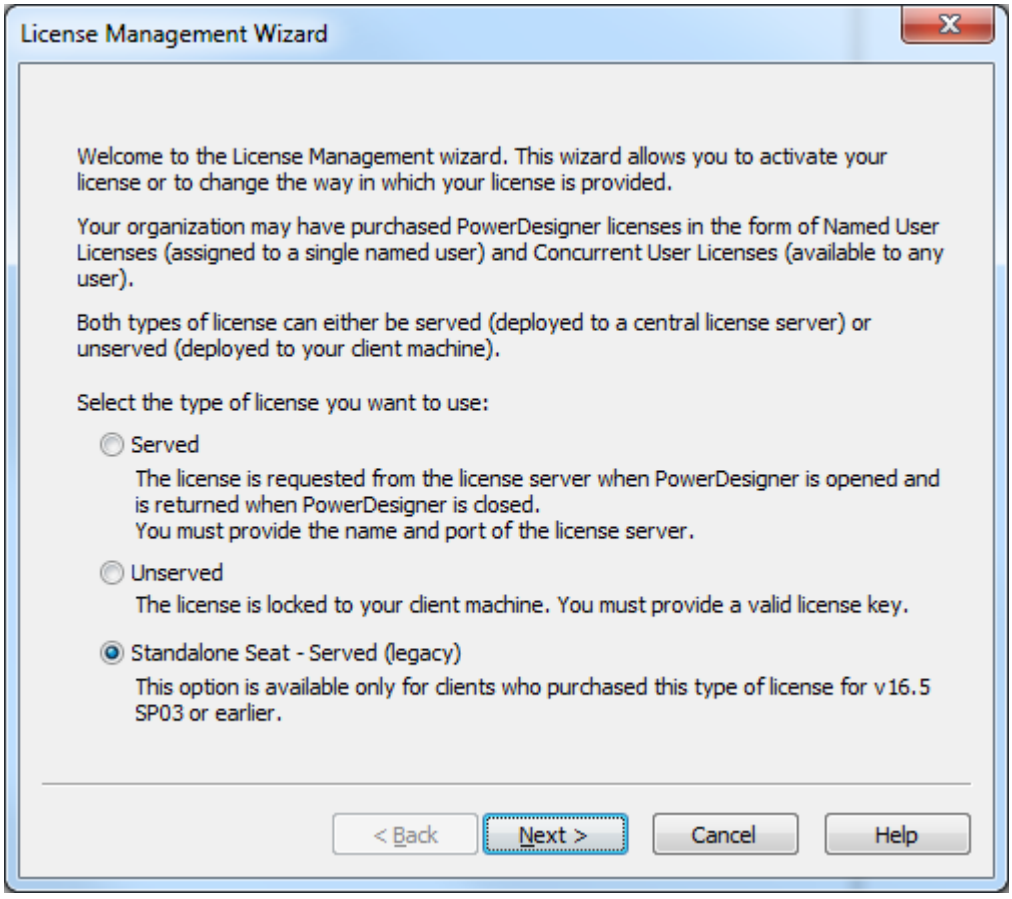
- Moving from a trial license to a regular license - You may need to rerun the setup program before launching the wizard if you need to install additional modules.
- Extending a trial license – After having requested your extension (which will be in the form of a time-limited unserved license) from SAP.
- Obtaining a trial of a new package – You must rerun the setup to install the additional modules before launching the wizard.

- Moving from an unserved to a served license – You must be able to connect to the license server to make this change.

Procedure

1. Select *Tools* > *License Parameters* to open the wizard.

   > **i Note**
   >
   > If you have taken a mobile license, you will be prompted to return it before accessing the wizard. If you are using a served license and the server cannot be contacted, an error message will be displayed and you will not be able to access the wizard.

2. Select a type of license and then click *Next*. The following types of license are available:
   - Served - Specify the name of the license server (and, if your administrator has specified one, a port number), and click *Next*.
     The license is requested from the SySAM license server when PowerDesigner is opened, and is returned when PowerDesigner is closed, or after 3 hours of inactivity. You must be connected to the license server in order to obtain a license, and must remain in contact (brief losses of contact only are permitted) in order to continue using it. When a user begins to work after a period of inactivity, PowerDesigner will automatically try to obtain a license, and if it is unable to do so, will display a message suggesting that you save your work.

> **Note**
>
> For periods when you must be out of contact with the server, you can request a mobile license for up to 30 days.

- Unserved - Follow the instructions to load a valid license key and then click *Finish* to validate your license and exit the wizard.
  The license is locked to your client machine. PowerDesigner reviews the local license file and authorizes only the use of modules for which a valid license is present. A license server is not needed.
- Standalone Seat - Served (legacy) - This option is available only for clients who purchased this type of license for v16.5 SP03 or earlier.

3. Specify the PowerDesigner package that you want to use, and then click *Finish* to validate your license and exit the wizard.

> **Note**
>
> The list displays all the packages that are available on the server but does not necessarily mean that a license for a particular package is available. If no license is available for the selected package, you must select an alternate package or click *Cancel* to restore your previous license configuration.

- If you are in trial mode, a license agreement dialog box is displayed.
- If you have selected an unserved license, the license file information is added to the license folder as a new license file.
- If you have selected a served license, a new license file is created to store the server name and port number.
- If you have selected a different package, the new modules will be available the next time you start PowerDesigner.

1.1.6.1 Obtaining a Mobile License

When you are using a served license and anticipate being out of contact with the license server, you can borrow a mobile license to allow you to continue using PowerDesigner for up to 30 days.

Procedure

1. Select *Tools* > *Take Mobile License*.

   A confirmation dialog box is displayed.
2. Click *Yes* to obtain a mobile license and click *OK* to confirm.

   Once you have obtained a mobile license, the 30 day limit is renewed automatically each time you connect to the license server. If you do not contact the server for 30 days, the mobile license is returned to the pool and you will not be able to use PowerDesigner until you reconnect.

> **Note**
>
> When you no longer need the mobile license you should return it to the server to make your license seat available to the general license pool. To do so, select *Tools* > *Return Mobile License*.

1.1.6.2 Troubleshooting Licensing Problems

In rare cases, PowerDesigner may repeatedly fail to obtain authorization from an unserved license file or the license server. If this happens, you can troubleshoot as follows:

Procedure

1. Exit PowerDesigner and navigate to the following directory:

   C:\Documents and Settings\All Users\Application Data\PowerDesigner <x>
2. Move any file with the .lic extension this folder to a safe place.
3. Make a backup copy of the file sysam.properties, and open it in a text editor. Remove the lines starting "Pd.LicenseMode" and "Pd.Package" at the end of the file and save the file.
4. Restart PowerDesigner.

   You will be prompted to open the License Management Wizard (see Managing Your PowerDesigner License [page 30]). Re-load your unserved license key or select the appropriate served license type.

1.1.7 PowerDesigner Documentation

Full documentation is provided as context-sensitive help by pressing *F1* and as PDFs by selecting *Help* > *Online Documentation*.

The following manuals are provided with your PowerDesigner installation:

- *New Features Summary*
- *Quick Reference*
- *Installation Guide*
- *Core Features Guide*
- *Requirements Management*
- *Enterprise Architecture*
- *Exploring the EA Example Model*
- *Business Process Architecture*
- *Data Architecture*
- *Data Architecture for SAP HANA*
- *Data Movement Architecture*

- *Object-Oriented Architecture*
- *XML Architecture*
- *Customizing and Extending PowerDesigner*
- *PowerDesigner Web*

This documentation is also available for consultation or download from https://help.sap.com/viewer/p/SAP_POWERDESIGNER.

Demonstration videos are also available. They can be installed with PowerDesigner (select *Help* > *Tutorial Videos*) or viewed from the installation CD).

For documentation of the PowerDesigner metamodel, click *Help* > *Metamodel Objects Help* (or click the *Find in MetaModel Objects Help* button at the bottom-right of the *Version Info* tab of the property sheet of any object).

1.1.8 SAP Platform Support

PowerDesigner provides support for various components in the SAP platform.

SAP HANA

PowerDesigner supports round-trip reverse-engineering and generation of HANA tables and other catalog objects directly or via CDS entities for SAP HANA® 1.0 and 2.0 databases. In addition, for SAP HANA® 1.0 environments, PowerDesigner supports initializing and reverse-engineering graphical calculation views (including the XML definition and projection list) for inclusion in impact analysis.

For detailed documentation, see *Data Architecture* > *DBMS Definition Reference* > *SAP HANA* .

SAP HANA Core Data Services (CDS)

PowerDesigner supports modeling and generation of CDS artifacts to SAP HANA® 1.0 Classic Repository and SAP HANA® 1.0 and 2.0 HDI environments.

For detailed documentation, see *Data Architecture* > *DBMS Definition Reference* > *SAP HANA* .

SAP Business Suite

An SAP Business Suite installation is built on a complex database structure, which comprises many thousand tables with often cryptic names, and may include large numbers of customizations. In certain environments, there may be multiple servers, each with different customizations. PowerDesigner allows you to browse the application component and package hierarchy and to import them and their supporting logical objects for

analysis, comparison, and merging of data dictionaries, and can also help you prepare a HANA table structure to allow consolidated reporting on one or more SAP® Business Suite installations.

PowerDesigner supports importing any recent version of SAP® Business Suite.

For detailed documentation, see *Data Architecture > DBMS Definition Reference > SAP Business Suite* .

SAP BusinessObjects

PowerDesigner supports generation of a SAP® BusinessObjects™ universe from your PDM for editing in the BusinessObjects Universe Design or Information Design tools, or for direct consumption by the Web Intelligence rich client. Generating a universe from your PDM gives you access to table, view, and column names and comments and more reliable cardinality information than if you create a universe directly from your database.

To generate universes, you must have SAP® BusinessObjects™ SBOP BI Platform Clients 4.0 SP04 Patch 3 (v14.0.4.819) or higher installed on your workstation. On Windows Vista or Windows 7 machines, if PowerDesigner fails to recognize a valid BusinessObjects installation, it may be necessary separately to launch the Universe Design tool one time with administrator privileges to enable the BusinessObjects SDK.

For detailed documentation, see *Data Architecture > Building Data Models > Generating and Reverse-Engineering Databases > Generating an SAP BusinessObjects Universe* .

SAP Solution Manager

PowerDesigner supports round-trip importing and exporting of SAP® Solution Manager v7.1 SP07 to SP13 projects through the desktop client. You can import SAP® Solution Manager v7.2 process diagrams via the PowerDesigner Web client and export changes to those diagrams back to the Solution Manager v7.2 server.

For detailed documentation, see *Business Process Architecture > SAP Solution Manager* and *PowerDesigner Web > Business Processes > Importing SAP Solution Manager 7.2 Process Diagrams* .

SAP NetWeaver

PowerDesigner supports round-trip importing and exporting of SAP BPM v7.3 and higher BPMN 2.0 files. PowerDesigner Web makes your process models available for browsing and importing directly into the SAP NetWeaver Developer Studio Process Composer v7.3 EHP1 SP15 or higher.

For detailed documentation, see *Business Process Architecture > BPMN 2.0 Descriptive and Executable > Importing and Exporting BPMN2 Files* .

SAP Data Services

PowerDesigner supports round-trip importing and exporting of SAP Data Services v4.2 and higher XML files.

For detailed documentation, see *Data Movement Architecture* > *ETL Modeling* > *SAP Data Services* .

SAP Information Steward

PowerDesigner can be used as a metadata integrator with SAP Information Steward v 4.2 SP01 and higher.

For detailed documentation, see the Information Steward documentation at https://help.sap.com/viewer/p/SAP_INFORMATION_STEWARD.

1.1.9 Next Steps

Now that you are familiar with the PowerDesigner Interface, you will want to start experimenting with a particular type of model suitable to your needs.

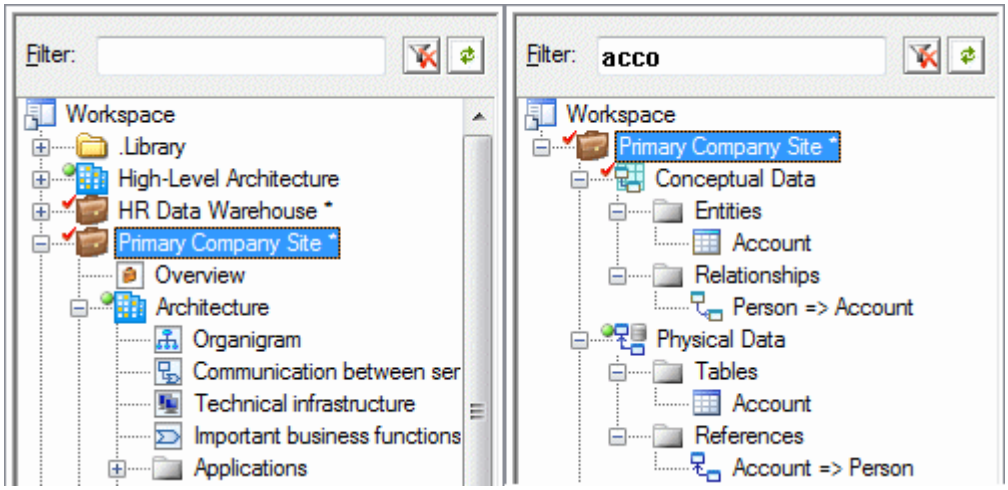
- Requirements Model (RQM) - Now turn to the *Requirements Management* guide where you will learn how to:
  - Add, modify, promote and demote requirements in the requirements hierarchy
  - Allocate requirements to particular team members and add traceability links to other model components
  - Import requirements from a structured MS Word document
  - View traceability and user allocation reports
- Enterprise Architecture Model (EAM) - Now turn to the *Enterprise Architecture* guide where you will learn how to:
  - Create diagrams in the business, application, and technology layers
  - Import Visio diagrams
  - Export and Import objects to and from other models
- Business Process Model (BPM) - Now turn to the *Business Process Architecture* guide where you will learn how to:
  - Create Process Hierarchy, Business Process, Process Service, and Composite Process diagrams
  - Manipulate Service Description Objects
  - Generate an executable BPM, and implement processes
- Conceptual Data Model (CDM), Logical Data Model (LDM), or Physical Data Model (PDM) - Now turn to the *Data Architecture* guide where you will learn how to:
  - Create CDM entities and relationships between them
  - Create LDM or PDM tables, columns, primary keys, indexes, and references and define referential integrity
  - Create PDM views, triggers, and abstract data types
  - Create business rules, domains, and data items
  - Generate a PDM from your CDM or LDM
  - Reverse engineer from and generate to database scripts
  - Work with PDM multidimensional diagrams
- Data Movement Model (DMM) - Now turn to the *Data Movement Architecture* guide where you will learn how to:

    - Define databases, replication processes, event scripts, and other replication objects
    - Reverse engineer from and generate to the SAP® Replication Server® replication engine
    - Design extract, transform, and load (ETL) operations to move data between databases
- Object-Oriented Model (OOM) - Now turn to the *Object-Oriented Architecture* guide where you will learn how to:
    - Design class, use case, and other standard UML diagrams
    - Generate a PDM with O/R mapping
    - Create an EJB
    - Deploy a component
- XML Model (XSM) - Now turn to the *XML Architecture* guide where you will learn how to:
    - Define elements, entities, and other components of an XML schema
    - Reverse engineer from and generate to DTD, XSD, and XDR files
    - Generate an XSM from a PDM or OOM

1.2 The Browser

The *Local* tab of the Browser provides a hierarchical view of all the model objects in your workspace. If the Browser is not visible, select *View* > *Browser* (or press Alt + 0) to display it.

To expand all nodes, press the numpad plus sign (+). To collapse all nodes, press the numpad minus sign (-). To filter the Browser tree, start typing the name of an object that you want to find in the *Filter* box:



PowerDesigner monitors your local documents against the versions stored in the repository, and communicates their status using icons to the left of the document icons:

Status	No Locks	Locked by You (Green Padlock)	Locked by Other (Red Padlock)
Up to Date (Green Circle)			
Local Changes (Red Checkmark)			
Changes on Server (Yellow Exclamation)			
Local and Server Changes			

> **Note**
>
> When a model file is closed (as all are when you open PowerDesigner) its icon bears a red dot in its bottom-left corner. To open a closed model, double-click it or right-click it and select *Open* or *Open as Read-only*. To close an open model to free up memory, right-click it and select *Close*.

You can control the display of status icons with the *Display status icons in Local Browser* general option (see Repository Options [page 396]).

A typical hierarchy of objects in the Browser *Local* tab may include:

- Workspace - The root of every Browser tree, which contains and organizes all your models and supporting documents (see Workspaces [page 44]).
- Library - A folder containing enterprise models that are pushed automatically to all users when they connect to the repository (see The Library [page 42]).
- Projects – Act as containers for models and other documents, allowing you to save them as a single unit in the repository (see Projects and Frameworks [page 46]).
- Folders - Organize models and other files (see Folders [page 45]).
- Models - Are the basic design unit in PowerDesigner and contain one or more diagrams and any number of model objects (see Models [page 45]).
- Packages - Divide large models into smaller "sub-models" to avoid manipulating large sets of objects, and can be used to assign different tasks or subject areas to different development teams (see Packages [page 143]).
- Diagrams - Graphical views of model objects (see Diagrams [page 185]).
- Model objects – Include all items belonging to a model (see Objects [page 92]).
- Reports – Are generated to document your models (see Reports [page 295]).

You can create objects from the Browser (see Creating Objects from the Browser [page 94]) and drag, drop, and copy objects in the Browser, or between the Browser and the Canvas (see Moving and Copying Objects [page 103]).

To find the Browser entry for a diagram symbol, right-click it and select *Edit* > *Find in Browser*. To find the diagram symbol for a Browser entry, right-click it and select *Edit* > *Find in Diagram*.

1.2.1 The Glossary

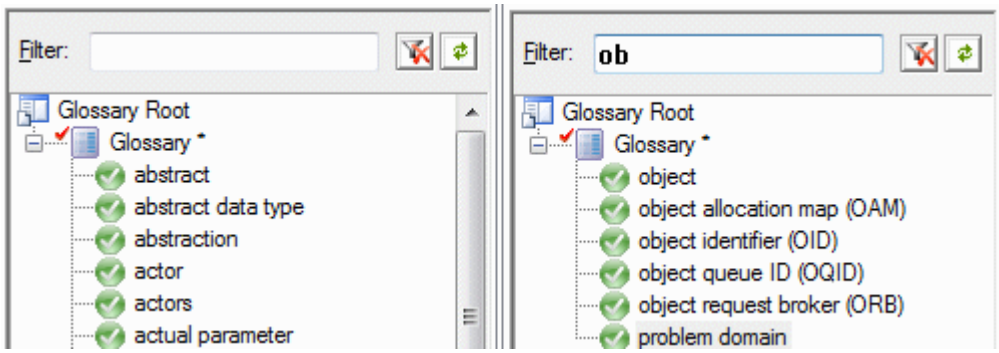
In environments with a repository, administrators can deploy an enterprise glossary to help everyone to use the same approved terms between projects for enterprise-wide consistency. Updates to the glossary are pushed to

users each time they connect to the repository or on demand. The glossary appears in the Browser *Glossary* tab.

The glossary can contain the following objects:

- Terms - are words or phrases that are approved for use in naming model objects. When you enable the glossary for your model, glossary terms are proposed as you enter object names.
- Synonyms - are words or phrases that are not approved for use in naming objects, and which are associated with the terms that should be used instead. The approved terms are proposed when you begin entering a synonym as an object name.
- Categories - to organize your terms. Each term can only appear in a single category.
- Glossary terms lists (see Glossary Terms Lists [page 546]), dependency matrices (see Dependency Matrices [page 199]), and reports (see Reports [page 295]) - to present glossary terms in various ways.
- Business Rules - see Business Rules [page 137].

Use the *Glossary* browser tab to review the approved terms, and double-click a term or synonym to open its property sheet, to view supporting information or references. Start typing the word you want to find in the Filter box at the top of the Browser to filter the glossary in real time:



> **Note**
>
> You can only edit the glossary if you have Write permission on the glossary model (see Deploying an Enterprise Glossary [page 540]).

Select *Tools* > *General Options* and select the *Glossary* category to control the appearance of the *Glossary* tab with the following options:

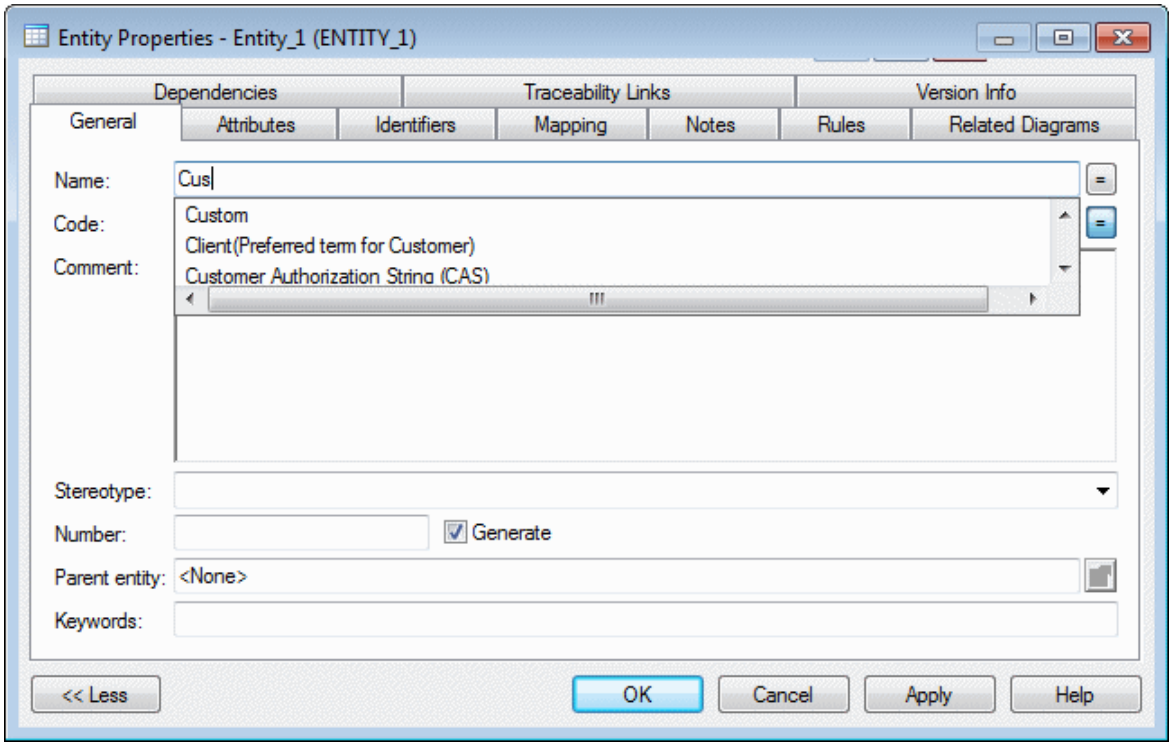
Option	Description
Show glossary tab	Shows the Browser *Glossary* tab.
Display	Controls whether names or codes appear in the Browser *Glossary* tab.

1.2.1.1 Using Glossary Terms in Your Models

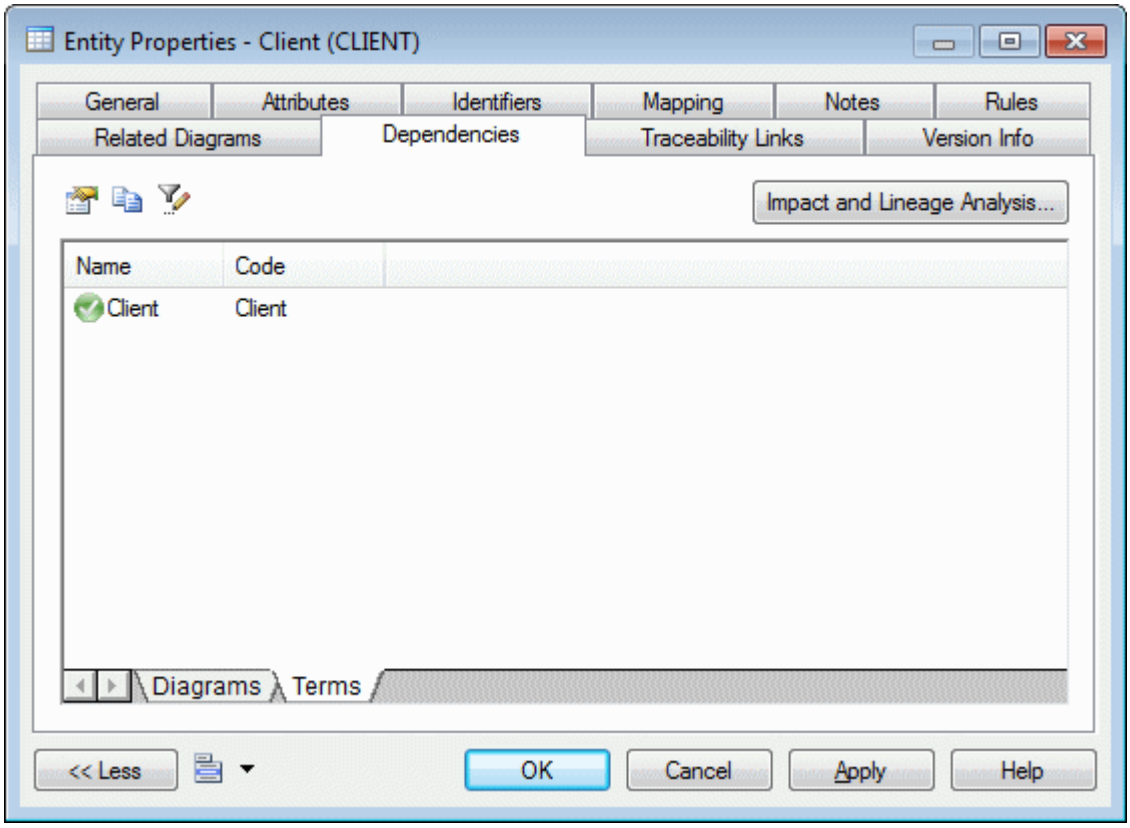
Enabling the glossary in your model provides autocompletion for object names (including guidance to discourage the use of synonyms) and enables model checks to guide your models toward glossary compliance.

Procedure

1. Select *Tools* > *Model Options*, then click *Naming Convention* in the category list and select *Use glossary for autocompletion and compliance checking* to enable the use of the glossary in your model.
2. When entering the names of objects in property sheets, diagram symbols, object lists, or in the Browser, glossary terms are suggested to autocomplete the name. Continue typing or use the down arrow to descend to a term in the list, and press *Enter* to select it.



If you click the object's *Dependencies* tab, the link to the chosen term is listed there:



3. [optional] Perform a model check to see which objects have names containing terms that are not included in the glossary (see Checking Glossary Compliance [page 154]).

1.2.1.2 Term Properties

To view a term's properties, double-click its Browser entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Description	Provides an explanation for the definition of the term. For information about using the RTF editing tools, see Text Editor Tools [page 226].

Property	Description
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Reference URL	Specifies a site, which provides more information about the term.

The following tabs are also available:

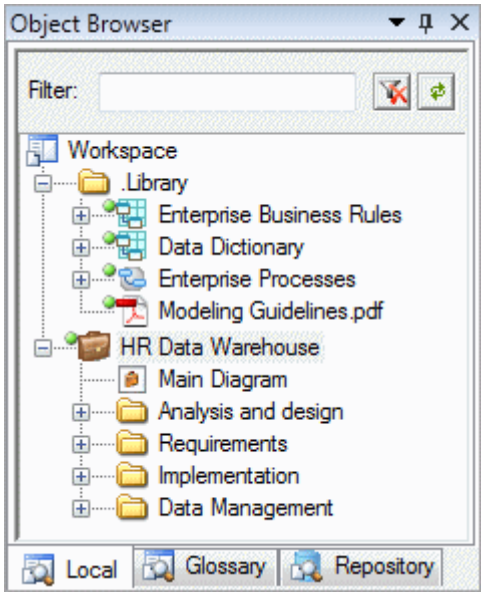
- *Synonyms* - Specifies the synonyms of the term, which are not approved for use. The *Parent* property of a synonym specifies the term with which the synonym shares a meaning, and which should be used instead of the synonym.
- *Related Terms* - Specifies other terms associated with the term.

> **i Note**
>
> Only an administrator or other user with Write permission on the glossary can edit terms (see Creating Terms, Synonyms, and Related Terms [page 541]).

1.2.2 The Library

In environments with a repository, administrators can deploy an enterprise library containing reference models, model templates, resource files, and other documents. Updates to the library are pushed to users each time they connect to the repository or on demand. Models and other documents appear in a special folder in the Browser *Local* tab.

The library appears by default at the top of your workspace as a folder named *Library* and provides access to enterprise reference models (such as a data dictionary or a PDM containing standard domains) for reuse in your projects as shortcuts or replicas, and other reference documents:



You can rename your local library folder, move it in the Browser tree or detach it from your workspace. To reinstate your Library if you have removed it, select *Repository* > *Reinitialize Local Library*.

The library documents and resource files are automatically updated whenever you connect to the repository (see Connecting to a Repository [page 19]) and you can control which ones will be pushed to your workspace. You can, at any time, confirm that you have the very latest version of library documents by selecting *Repository* > *Synchronize Library*.

Resource files, such as DBMS or language definition files and extensions, are not displayed in the Browser library folder unless you have permission to modify them, but they appear automatically in the new model dialog, resource file lists, and anywhere in the interface where a resource file can be selected.

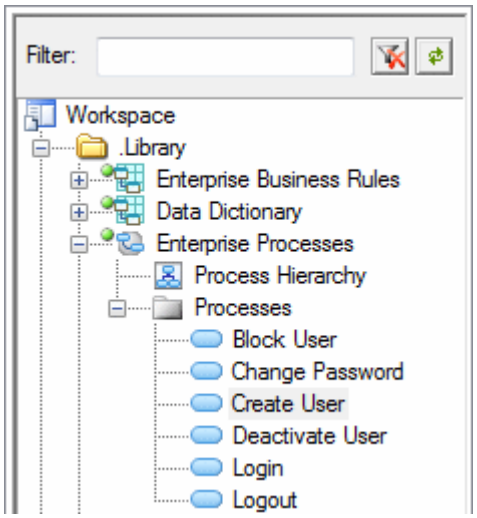
> **i Note**
>
> By default, library documents are checked out to a subfolder inside `My Documents/PowerDesigner Libraries`, controlled by the `_SHARED` named path (see Named Paths [page 393]). You can only check in changes to library documents if you have write permission on the library folder or on one or more of its children (see Deploying an Enterprise Library [page 549]).

1.2.2.1 Reusing Library Objects in Your Models

Library models can contain reference objects that are intended for reuse in your models in the form of shortcuts or replicas.

Procedure

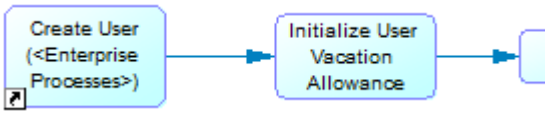
1. Expand the library model in the Browser to show the object that you want to reuse:



2. Right-click on the object in the library model and drag and drop it into your model diagram where you want to reuse it (or onto a model or package in the Browser).

3. Release the right mouse button, and select one of the following:
   - *Create Shortcut(s) Here* - when you want create a reference to the library object.
   - *Replicate Here* - when you want to create a local copy of the library object and make changes to one or more of its properties

   The shortcut or replica is created in your model, and can be connected to any appropriate locally-created objects. In this example, a shortcut to the **`Create User`** library object is connected to other locally-created objects:



   If the library object subsequently changes, any updates will cascade automatically to your model.

> **i** Note
>
> For detailed information about working with shortcuts and replicas, see Shortcuts and Replicas [page 443].

1.2.3 The Repository Browser

The *Repository* tab in the Browser displays the content of the repository to which you are currently connected.

For more information, see Browsing the Repository [page 244].

1.2.4 Workspaces

The *Local* tab of the Browser contains a workspace to organize all your modeling and supporting files. You can create projects, models, and folders in a workspace. To add external files, drag and drop them from Windows Explorer.

We recommend that you save your workspace regularly:

- To perform a standard save - select *File* > *Save Workspace* or right-click the workspace node in the Browser and select *Save*. You will be prompted to save any models that contain changes.
- To save the workspace under another name - select *File* > *Save Workspace As* to open a standard Save As dialog, or right-click the workspace node in the Browser and select *Save As*. You will be prompted to save any models that contain changes.
- To save the workspace and all the models it contains - select *File* > *Save All*, or click the *Save All* tool.

> **i** Note
>
> Workspace files (`*.sws`) contain only the structure of the workspace. Deleting the workspace file in Windows Explorer will not delete any projects, models, or other referenced files.

To create a new workspace, simply close the current workspace by selecting *File* > *Close Workspace*. You will be prompted to save the workspace and any models that contain changes, and a new, empty workspace will open.

You can create multiple workspaces, but only one can be open at a time. To change workspace, select *File* > *Open Workspace*. You will be prompted to save the current workspace and any models that contain changes, and the selected workspace will open. The models contained in the workspace are not open by default.

1.2.5 Folders

You can use folders to organize the projects, models, and other files in your workspace. The folder structure is saved in the workspace file.

To create a folder, right-click the workspace root or another folder and select *New* > *Folder*, then enter a name in the highlighted area. Folder names must be unique in the workspace.

To create a project, model, or multi-model report in a folder, select the folder and use the standard tools, or right-click the folder and select *New* > `<New item>`.

To add an existing project, model, or multi-model report to a folder from another location in the Browser, drag and drop it onto the folder. You can also drag and drop PowerDesigner and other files from Windows Explorer to the Browser.

To delete a folder, select it and click the *Delete* tool or right-click it and select *Edit* > *Delete*.

> **i Note**
>
> Deleting a folder cannot be undone. When you delete a folder, you remove the models and reports it contains from the workspace, but you do not delete them.

1.2.6 Models

Models are the basic work unit in PowerDesigner. You can create a model directly in the workspace, in a folder, or as part of a project.

To create a model, click the *New* tool or select *File* > *New Model* (see Creating a Model [page 12]). For a list of available types of models, see PowerDesigner Models [page 9]. For detailed information about a particular type of model, and for information about creating a model by importing, generating or reverse engineering, see the appropriate modeling guide.

To add an existing model to the workspace, click the *Open* tool or select *File* > *Open*, browse to the model file, and click *Open*. To open the model as read-only, select the *Open as read-only* check box.

> **i Note**
>
> When a model file is closed (as all are when you open PowerDesigner) its icon bears a red dot in its bottom-left corner. To open a closed model, double-click it or right-click it and select *Open* or *Open as Read-only*. To close an open model to free up memory, right-click it and select *Close*.

When a model contains unsaved changes, an asterisk is appended to its name in the Browser. When you save a model or a multi-model report, PowerDesigner assigns it a unique identifying number called GUID (Global Unique ID, which is used to identify documents in the Repository and during model generation), and creates a backup copy of your file with the same identifying number:

- To perform a standard save - select *File* > *Save*, click the *Save* tool, or right-click the model entry in the Browser and select *Save*.
- To save the model as a backup version with the same GUID - select *File* > *Save As*, or right-click the model node in the Browser and select *Save As*.
- To save the model as a new model with a new GUID - select *File* > *Save As New Model*, or right-click the model node in the Browser and select *Save As New Model*. If you check the new model into the repository, it must be as a new model without any revision history. Other models with shortcuts pointing to the original model will continue to be connected to that model.
- To save the workspace and all the model it contains - select *File* > *Save All*, or click the *Save All* tool.

> **i Note**
>
> The following formats are available when you save a model:
>
> - XML - [default] Larger and somewhat slower to load, but editable in a text editor. Recommended for small to medium-sized models.
> - Binary - Smaller and faster to load, but not editable outside of PowerDesigner. Recommended for large models.

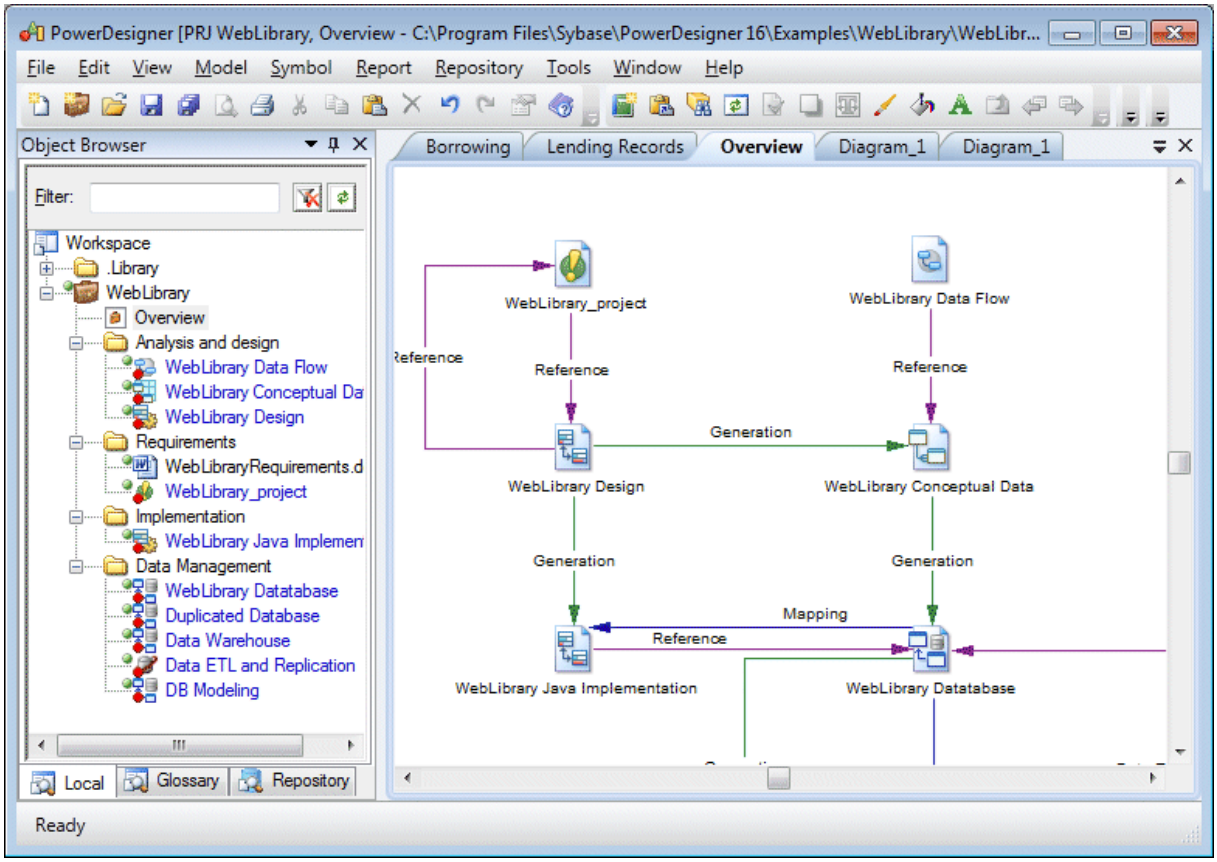
To change the display name of a model in the Browser, select it and press *F2*, or right-click it and select *Rename*.

To delete a model from the workspace, select it and click the *Delete* tool, press *Del*, or right-click it and select *Detach From Workspace*. The model is removed from the workspace, but the file is not deleted from your disk.

1.3 Projects and Frameworks

A project allows you to group together all the models and other types of documents you need for a particular modeling task, and save them as a simple entity in your repository.

A project can contain one or more project diagrams (see Project Diagrams [page 50]), which show the connections between models and other documents:



Your project can also contain one or more framework diagrams and/or a framework matrix, which direct how your project must be modeled and list the documents that are needed (see Completing Framework Diagrams and Matrices [page 57]).

You can create a project from scratch or from a template. PowerDesigner provides a set of predefined project templates, and you can also create your own (see Project Templates [page 81]).

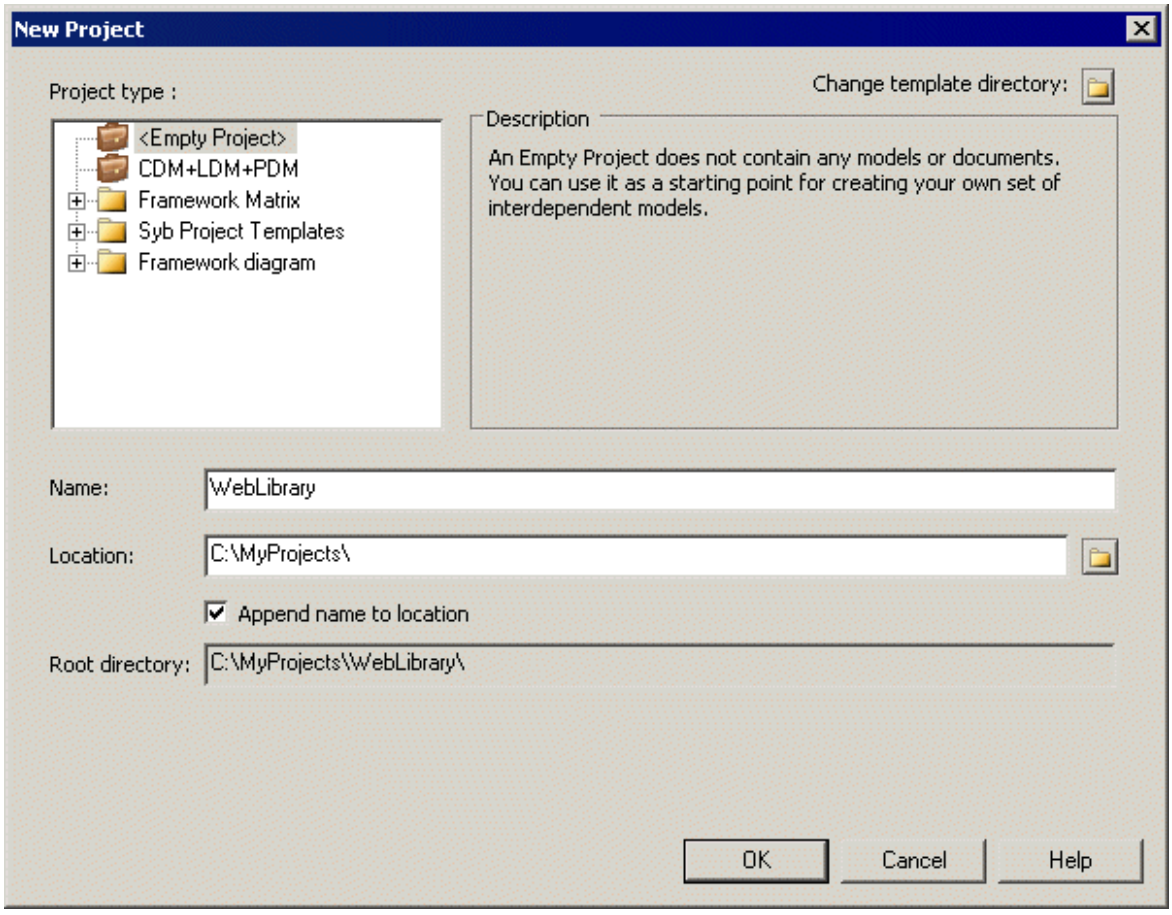
> **i Note**
>
> Projects and framework diagrams and matrices are only available with Enterprise versions of PowerDesigner.

1.3.1 Creating a Project

Every project (.prj) contains at least one diagram, and any number of model or file documents. The project can also contain one or more framework diagrams, a framework matrix, and one or more dependency matrices.

Procedure

1. Select *File* > *New Project* to open the New Project dialog box.



2. Select a type of project in the tree. You can choose to create:
   - An empty project.
   - A project based on a template (see Project Templates [page 81]). Click the *Change Template Directory* tool to search for templates in another location.
3. Enter a project name and location, and select the *Append Name To Location* check box if you want to add the project name to the root directory.
4. Click *OK* to close the dialog box, and create the project.

   The project opens. You can add models and other documents to your project (see Adding Model and File Documents to a Project [page 49] and Completing Framework Diagrams and Matrices [page 57]).

1.3.2 Adding Model and File Documents to a Project

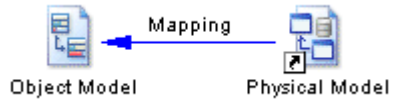
You can create models in your project and add existing models and files to it. Project documents are listed in the Browser and displayed as icons in the project diagram. You can open a model or file by double-clicking its icon in the diagram or in the Browser.

Creating Models	Adding Existing Models and Documents
• Click the *New Model* tool in the Toolbox, and click in the diagram to open the New Model dialog (see Creating a Model [page 12]). • Right-click a project or a project folder, select *New* > *Model* to open the New Model dialog.	• Click the *Add Project Document* tool in the Toolbox, and click in the diagram to open an Open dialog box. Browse to and select one or more documents and then click *Open*. • Drag and drop one or more documents from the Browser, or the Windows Explorer to the project node in the Browser or to the project diagram. **i Note** If you drag a folder with sub-folders from Windows Explorer, the hierarchy is preserved but, since projects are not synchronized with the file system, any files you subsequently add to these folders outside PowerDesigner will not be displayed in the Browser.

The models contained in your project are visible in the Browser, beneath the project node and also in the List of Models available (when the project has context) by selecting *Model* > *Models*.

When you add existing documents from other locations to your project, they are listed under the project node in the Browser, but are not moved to the project directory. Such documents display a shortcut overlay on their symbols in the Browser and project diagrams to indicate that they are located outside the project folder, and their entries in the List of Models have the read-only *[I]nside Project* check box deselected.

In the following example, Object Model is inside the project and Physical Model is outside the project:



In order to profit from the convenience of the project as a container, we recommend that you create (or place) all its models and files inside the project directory in your file system.

To move a document from outside to inside the project, right-click its Browser entry or its symbol in the project diagram, and select *Move to Project Directory*. The file is moved to inside the project directory in your file system, the shortcut overlay disappears, and the *[I]nside Project* check box is selected in the List of Models.

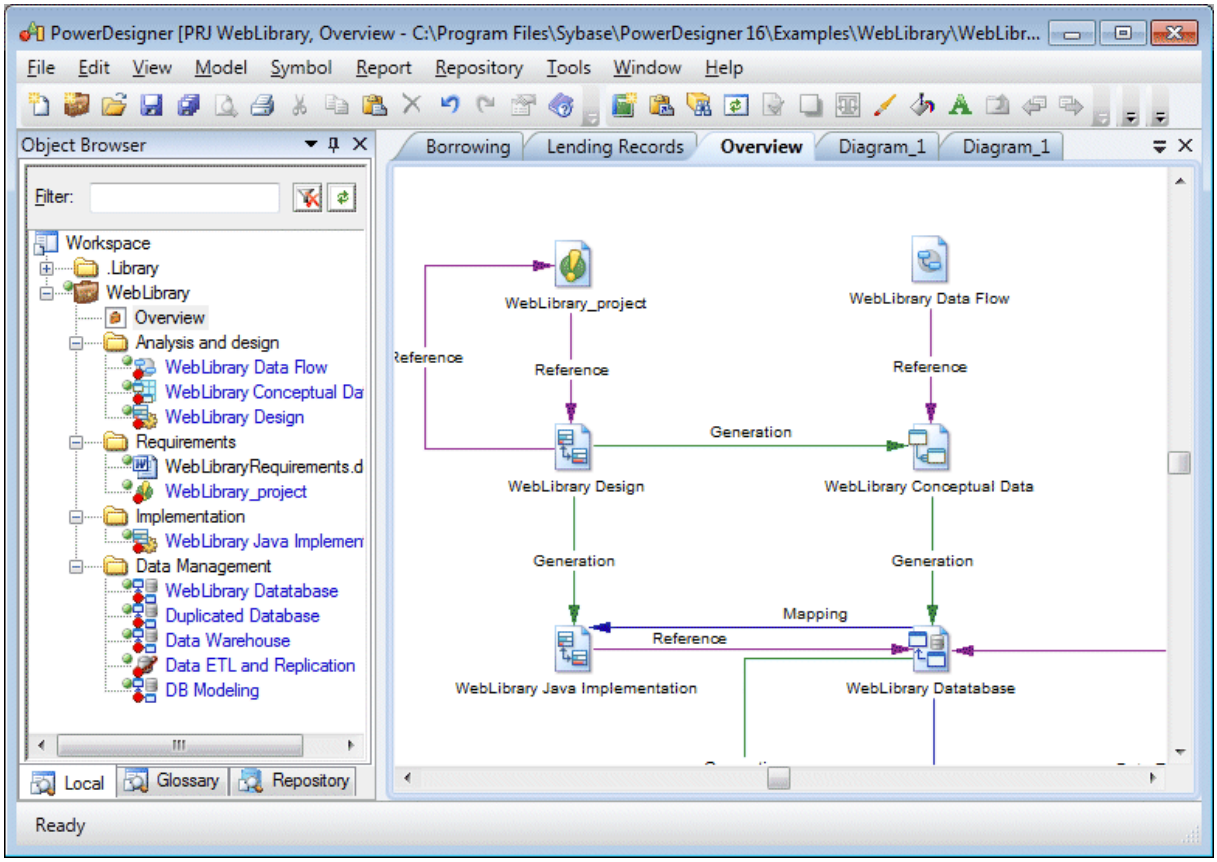
1.3.2.1 Project Diagrams

A project diagram allows you to display project documents and the relationships between them.

You can create one or more project diagrams in an existing project to display different views of your project in any of the following ways:

- Right-click a project in the Browser, and select *New* > *Project Diagram*.
- Right-click the background of any project diagram, and select *Diagram* > *New Diagram* > *Project Diagram*.
- Select *View* > *Diagram* > *New Diagram* > *Project Diagram*.

In the following example, the models and other documents for the WebLibrary project are organized into sub-folders, and the connections between them (by reference, mapping, and generation) are displayed in the Main Diagram project diagram:



The following tools are available in the Toolbox of a project diagram:

Tool	Description
	New Model – Creates a PowerDesigner model in the project.

Tool	Description
	Add Project Document – Adds an existing document to the project, which can be: • A PowerDesigner model (PDM, OOM, LDM etc.). • An external file (PDF, text file etc.).
N/A	Dependency links, such as generation, mapping, reference, and file links, are automatically created between documents. These links cannot be created, but you may need occasionally to rebuild them (see Rebuilding Dependency Links [page 53]).

1.3.2.2 Converting a Set of Repository Documents into a Project

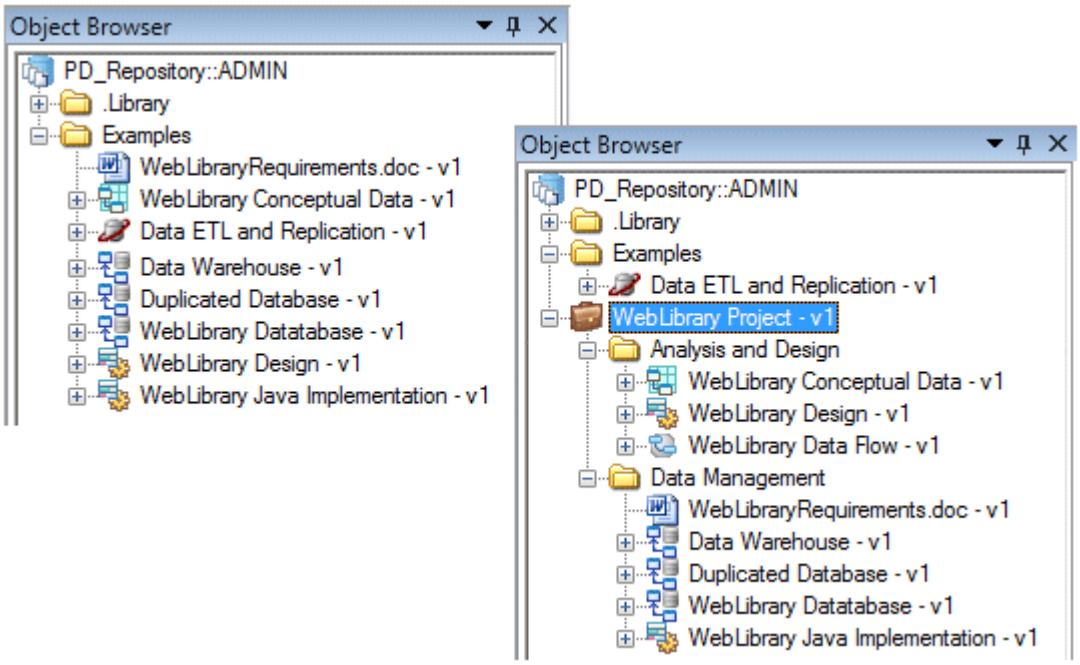
You can retrieve a set of documents checked into the repository and convert them into a project to benefit from the convenience of the project as a container.

Procedure

1. In the Repository Browser, create a project root folder to contain the project documents.
2. [optional] Create project sub-folders to organize documents according to your needs.
3. Drag and drop repository documents into these folders.
4. Right-click the project root folder, and select *Convert to Project* to convert it and all its contents into a project.

Results

In the following example, repository documents in the Examples folder are are transformed into the WebLibrary Project:



1.3.2.3 Opening Project Documents

Once you have added a PowerDesigner model to a project, you should only open the model within the context of the project, as links between it and any other models in the project are managed by the project file. If you attempt to open the model outside of the project, you will be prompted to cancel the operation, and to open the project instead.

You can open a project document in the context of its project by double-clicking it in:

- a framework matrix or diagram
- a project diagram
- the Browser under its parent project

Similarly, models that belong to a project should only be checked in and out of the repository via the project.

1.3.2.4 Viewing Dependency Links

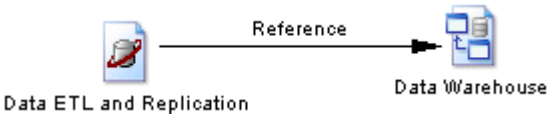
You can explore the details of any of the dependency links in your diagram by right-clicking it and selecting Show Dependencies.

Context

Each type of link has its own viewer:

- Generation – displays the generation links between models in the Generation Links Viewer (see The Generation Links Viewer [page 439]).
- Mapping – displays the mapping links between models in the Mapping Editor (see The Mapping Editor [page 468]).
- Reference – displays the shortcuts and replications between models in the Shortcuts and Replications dialog box.

In the following example, right-clicking the reference link between Data ETL and Replication and Data Warehouse models opens the Shortcuts and Replications dialog box which displays Data Warehouse object shortcuts:



1.3.2.5 Rebuilding Dependency Links

Dependency links are automatically generated in your project diagram when you add documents that are linked to each other by, for example, generation, mappings, shortcuts, etc.

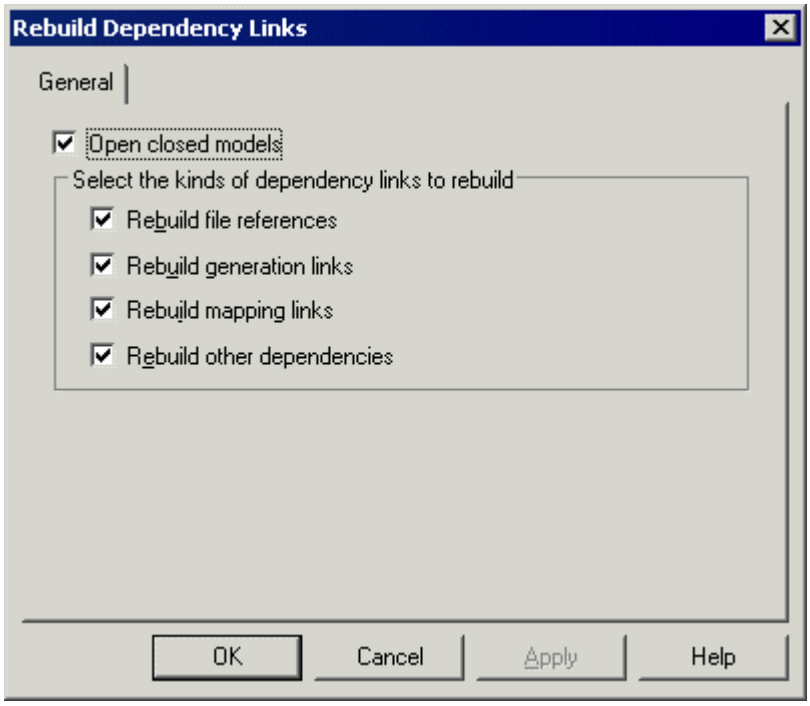
Context

Rebuilding dependency links can be useful to update:

- Links between closed models you add to your project.
- Links deleted by error in your diagram.

Procedure

1. Select *Tools* > *Rebuild Dependency Links* to open the Rebuild Dependency Links dialog box.



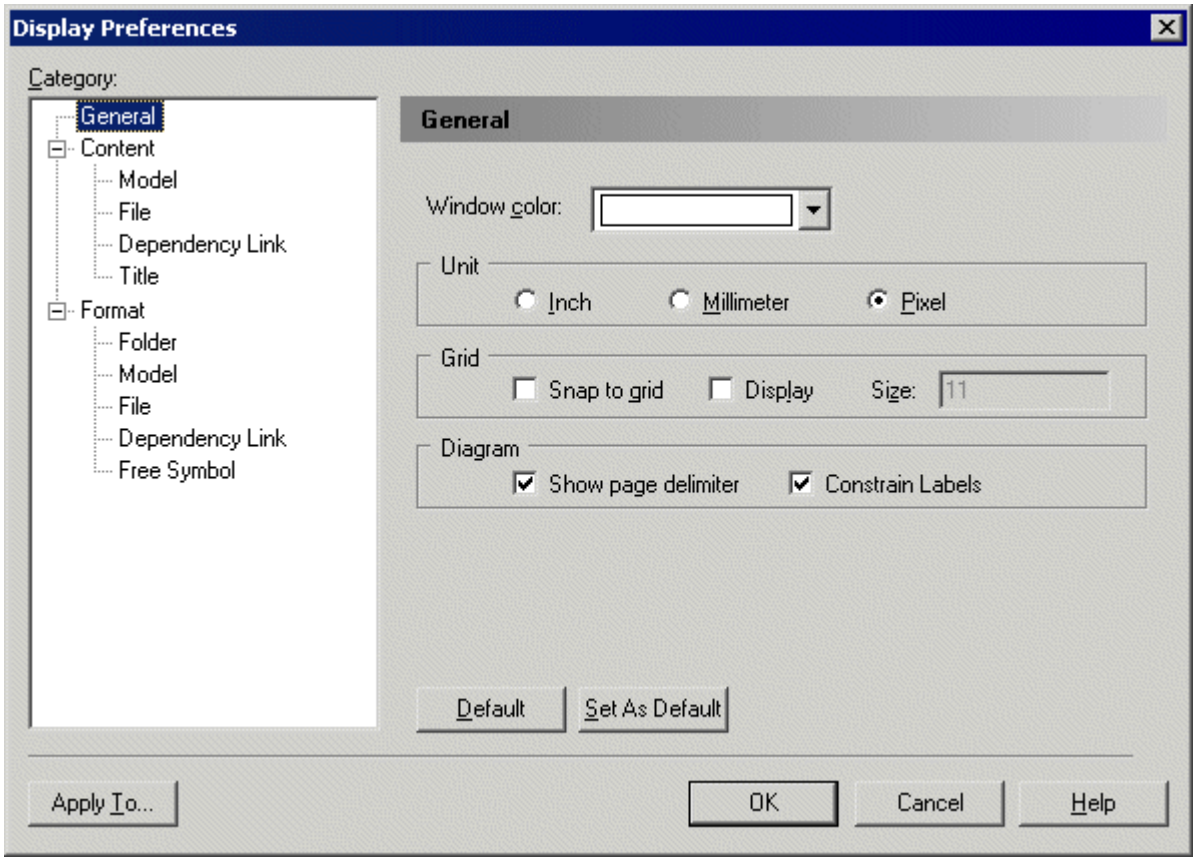
2. [optional] Clear the Open closed models check box to rebuild only those links in a limited number of models.
3. Select the check boxes that correspond to the dependency links you want to rebuild.
4. Click OK to close the dialog box and return to the diagram. Any missing links are updated in the diagram.

1.3.2.6 Project and Framework Display Preferences

PowerDesigner display preferences allow you to customize the format of object symbols, and the information that is displayed on them.

To set project display preferences, select *Tools* > *Display Preferences* (or right-click the diagram background and select Display Preferences).

> **i** Note
>
> For information about changing the format of symbols, see Format Display Preferences [page 234]. Note that only framework architects (see Designing Framework Diagrams and Matrices [page 65]), are allowed to modify the framework diagram and matrix format display preferences (see Symbol Format Properties [page 214]).



To set display preferences for a particular object, select it under the Content category. The following table lists the display preferences available. The objects available to be customized in the Display Preferences window depend upon the current diagram type.

Preference	Description
Code	Displays the code of the object.
Comment	Displays the comment of the object.
Completion status	[node and cell only] Displays a rectangle, which represents the work completion status according to the percentage entered in node and cell floating lists and property sheets.
Documents	[node and cell only] Displays the list of documents in the node or cell, instead of in floating lists.
Stereotype	Displays the stereotype of the object.
Type	[dependency link only] Displays the type of the dependency link, which can include mapping links, shortcut links, generation links, etc.

1.3.2.7 Project Checks

You can check the validity of your project at any time by pressing *F4*, selecting *Tools* > *Check Project*, or right-clicking the diagram background, and selecting *Check Project*.

Context

For information about using the Check Project Parameters dialog, see Checking Models [page 148].

1.3.2.7.1 Model and File Checks

PowerDesigner provides default model checks to verify the validity of models and files.

Check	Description and Correction
Model and file location uniqueness	Multiple models or files cannot have the same location. • Manual correction: Delete unnecessary model or file. • Automatic correction: None.
Model location empty	[model only] There must be a path in the model document Location field. • Manual correction: Save the model to specify a model location. • Automatic correction: None.
External document URL not under a named path	A model or file is defined outside the project directory. • Manual correction: Right-click the model or file, and select *Move to Project Directory*. • Automatic correction: None.
Target models not in project	[model only] One or more target models of a model contained in the project is missing. • Manual correction: Add missing target model with appropriate dependency link. • Automatic correction: Yes.

1.3.2.7.2 Framework Node and Cell Checks

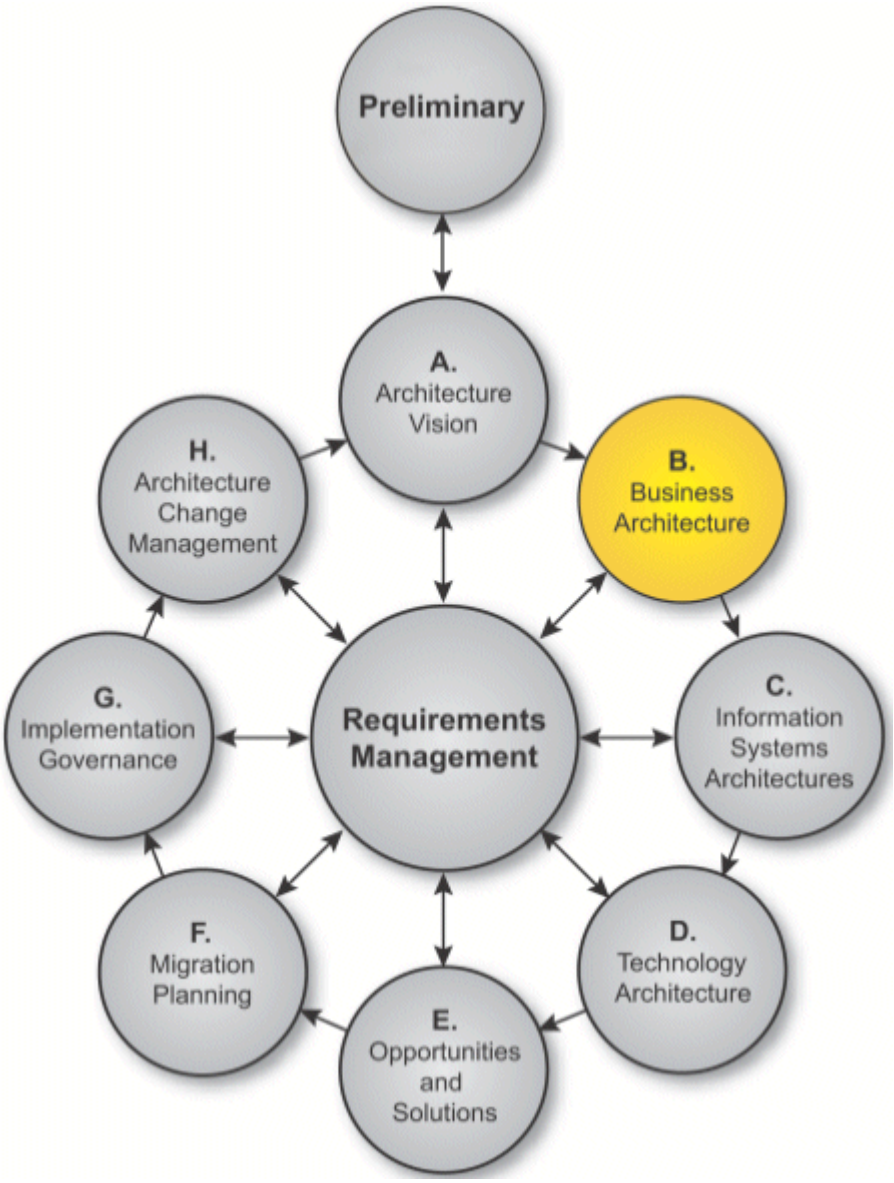
PowerDesigner provides default model checks to verify the validity of framework diagram nodes and matrix cells.

Check	Description and Correction
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Documents not consistent with node/cell actions	A document in a node and cell must be consistent with its source action. • Manual correction: Modify the node's and cell's content according to its specified actions. • Automatic correction: None
Mandatory actions not executed	A mandatory action in a node and cell must be executed. • Manual correction: Execute the mandatory action. • Automatic correction: None

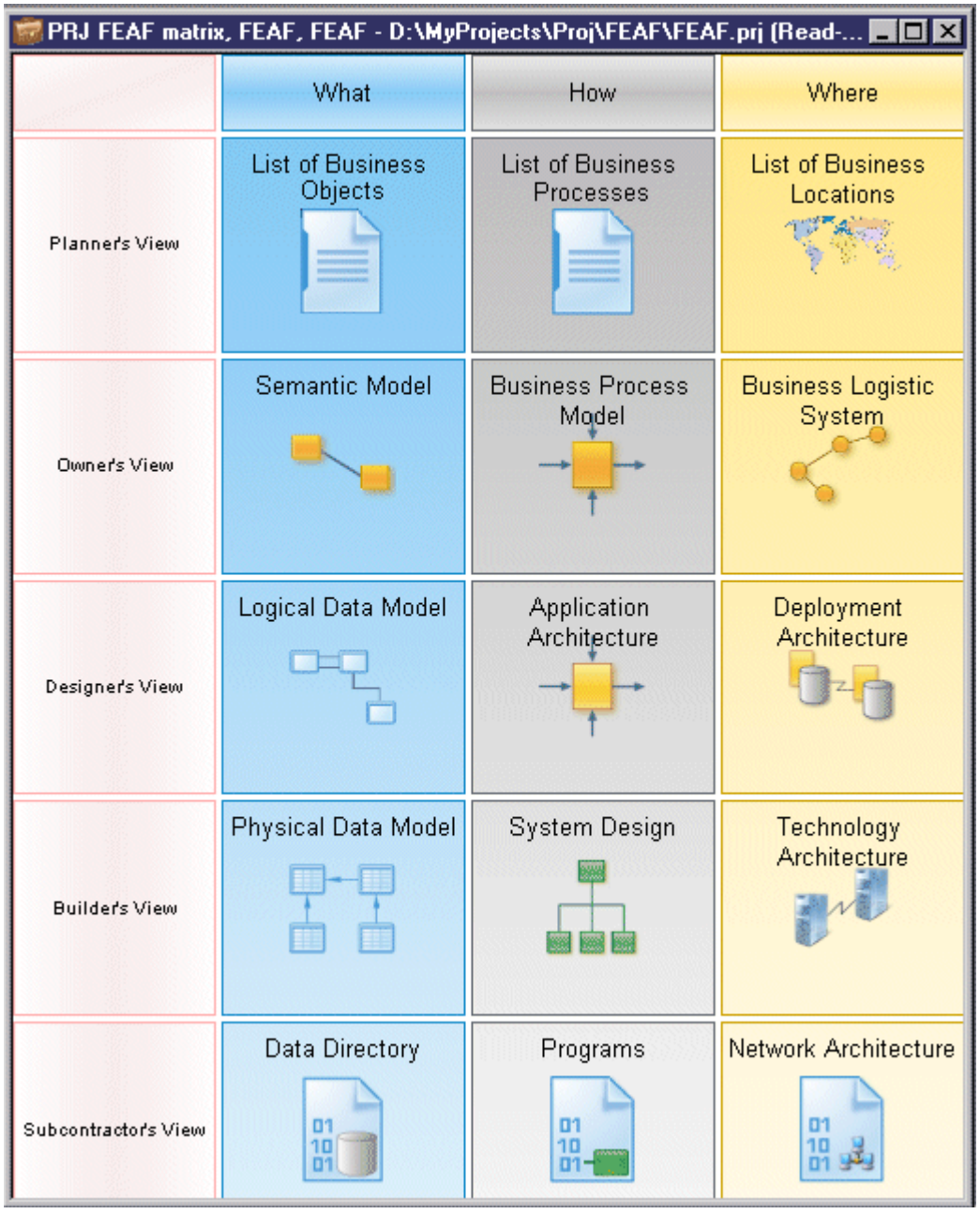
1.3.3 Completing Framework Diagrams and Matrices

Your project may contain one or more framework diagrams and/or a framework matrix, which direct how your project should be built.

- A framework diagram - lets you model a framework with nodes and links, which you can style as necessary.



- Framework matrix - lets you model a framework in a matrix, in which columns usually represent various aspects of the enterprise that can be described or modeled, and the rows represent various viewpoints from which the aspects can be described. Each cell formed by the intersection of a column and a row represents an aspect of the enterprise modeled from a particular viewpoint.



PowerDesigner provides a set of predefined framework diagrams and matrices, and users with framework design rights can also create their own (see Designing Framework Diagrams and Matrices [page 65]). You complete a framework diagram or matrix by creating (or adding existing) documents in nodes, cells, and additional areas according to predefined rules:

- Documents — can be of different types and can each be associated with multiple nodes, cells, and additional areas:
  - Model – a PowerDesigner model, such as a BPM or a PDM, created from scratch, based on a template, or generated from another model.
  - Diagram – a PowerDesigner model diagram, such as an OOM use case diagram.

- Dependency matrix – a grid that displays the links between model objects (see Dependency Matrices [page 199]).
- List – a list of PowerDesigner model objects, such as BPM processes, or PDM tables.
- File – an external file, such as a .doc or .txt file.

- Nodes and cells [when active] — must be completed by particular kinds of documents.
- Additional areas [framework matrix only] — can be defined outside of the grid (for example, the global view in the FEAF framework matrix that is created at the end of the enterprise architecture implementation).

1.3.3.1 Navigating in Framework Diagrams and Matrices

You can interact with framework diagram and matrix elements using various mouse actions.

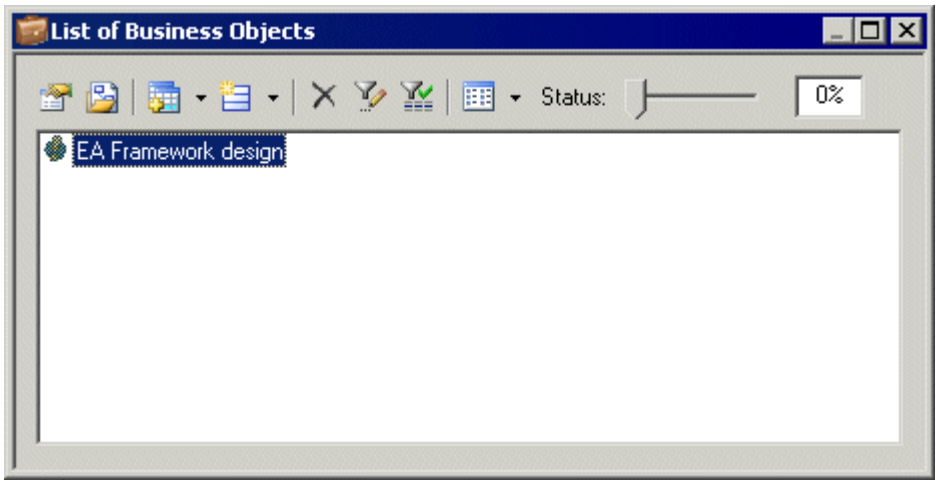
- Hover over a framework element – to display its name and any comment.
- Click a node or cell – to open a floating list, which lets you manage the documents attached to the node or cell (see Working with Floating Lists [page 63]).
- Right-click a node or cell – to opens contextual menu (see Working with the Framework Node and Cell Contextual Menu [page 64]).
- Double-click a framework matrix element [framework architects only] – to open the property sheet of the element (see Framework Matrix Properties [page 87]).

1.3.3.2 Attaching a Document to a Framework Node or Cell

You can attach new (or existing) documents to each node of a framework diagram and cell of a framework matrix.

Procedure

1. Open the framework diagram or matrix by double-clicking its Browser entry.
2. Click the node or cell to which you want to add the document to open its floating list, which displays the documents attached to it.



3. Click one of the following tools:
    - *New Document* – to create a document of the specified kind:
        - Model [from scratch or based on a template] – creates the model and opens it in the diagram window.
        - Model [generated from another PowerDesigner model] – opens the *Model* Generation Options dialog box to let you create a model (see Generating Models [page 424]).
        - Diagram – opens a model selection dialog box to let you select a new or existing model to create the diagram.
        - Dependency matrix – opens a model selection dialog box to let you select a new or existing model to create the dependency matrix (see Dependency Matrices [page 199]).
        - List – opens the List of *Objects* dialog box to let you create objects in a new or existing model.
        - File – opens a standard Save As dialog box to let you enter a file name. Click Save to open the file in the application corresponding to the file extension.
    - *Add Document*, and select:
        - *Attach Project Document* – to add documents already in the project. Select the type of document you want to attach.
        - *Add Document* – to add documents from outside the project. Select the type of document you want to add.

    > **i Note**
    >
    > You can also drag a project document from the Browser and drop it onto a node, a cell or a floating list, or right-click a node or a cell, and select the New Document, Attach Project Document, and Add Document commands. Documents can also be moved from one node or cell to another.

4. [optional – New Document only] Complete the creation of the new document by, for example, adding objects to the model or diagram, or entering text in the file, etc.

    The document is displayed in the floating list.

1.3.3.2.1 Example: Creating a PDM from a Cell's Floating List

In this example, we create a PDM, which has been specified as a requirement to complete a framework matrix cell.

Procedure

1. Click a cell to open its floating list, which displays the documents attached to the cell.
2. Click the *New Document* tool, and then select *New PDM* to create the required PDM.

    The new PDM document is displayed in the cell, and in the Browser in the folder specified by the framework architect.

For information about specifying a model as a requirement to complete a framework node or cell, see Example: Specifying a Model Action [page 68].

1.3.3.2.2 Example: Creating a List of Processes from a Node's Contextual Menu

In this example, we create a list of BPM processes, which has been specified as a requirement to complete a framework diagram node.

Procedure

1. Right-click a node, and select *New* > *Document* > *List of processes* to open the List of Processes dialog.
2. Create an appropriate number of processes in the list, entering a name for each, and then click *OK* to close the dialog box.

   The list of processes document is displayed in the node, and in the Browser under a BPM node in the folder specified by the framework architect.

   For information about specifying a list as a requirement to complete a framework node or cell, see Example: Specifying a List Action [page 71].

1.3.3.3 Deleting Documents from Framework Nodes and Cells

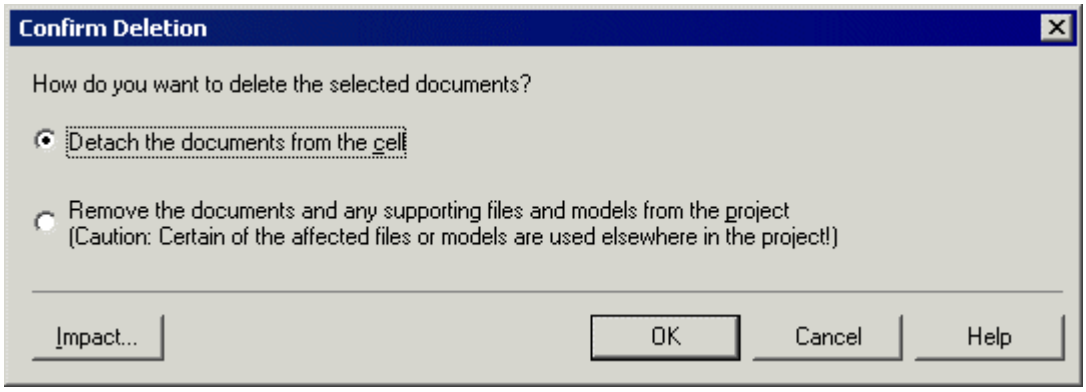
You can delete documents from framework nodes and cells. You can choose to simply detach a document from the node or cell or completely remove it from the project.

Procedure

1. Click a node or cell to open its floating list, select one or more documents in the list, and then click the *Delete* tool.

   Alternatively, select one or more documents in a node or cell and press the *Del* key or right-click a document and select *Edit* > *Delete*.

   The Confirm Deletion dialog box is displayed.



2. Choose one of the following delete options:
   - Detach the document from the node or cell - the document is detached from the node or cell but remains available in the project to be used elsewhere.
   - Remove the document and any supporting file or model from the project - their deletion from the project may affect files or models used elsewhere in the project.
3. Click OK to close the dialog box and confirm the deletion.

1.3.3.4 Working with Floating Lists

You open a floating list by clicking a framework diagram node or matrix cell.

> **i** Note
>
> Floating lists are not available when you select the Documents display preference (see Project and Framework Display Preferences [page 54]).

Floating lists contain the following tools to let you manage project documents in the framework diagram and matrix:

Tool	Description
	Properties – opens the property sheet of the selected document.
	Open Document – opens the selected document.
	Add Document – contains the following tools: • Attach Project Document – lets you select a document of the specified kind from inside the project. • Add Document – lets you select a document of the specified kind from outside the project.
	New Document – creates a document of the specified kind.

Tool	Description
	Delete – deletes the selected document from the node and cell. You can choose to: • Detach the document from the node and cell. • Remove the document and any supporting file or model from the project. For information see Deleting Documents from Framework Nodes and Cells [page 62].
	Customize Columns and Filter – allows the definition of filter expression on the columns of the floating list (see Customizing Object List Columns and Filtering Lists [page 123]).
	Enable/Disable Filter – enables or disables the filter.
	Views – controls the display of documents in the floating list. You can choose between: • Large Icons – displays documents as large icons. • List – displays documents in multiple columns to optimize space. • Details – displays documents in a grid, which can be filtered.
Status	Status – indicates the percentage of work completed in the node and cell.

1.3.3.5 Working with the Framework Node and Cell Contextual Menu

The contextual menu available on framework nodes and cells allows you to perform various actions.

The contextual menu of a framework node and cell contains the following commands. For information about the additional commands available only in Framework Design Mode, see Styling Framework Nodes and Cells [page 81]):

Command	Description
List of Documents	Opens the node's or cell's floating list.
New Document	Creates a document of the specified kind in the node or cell. An action must be specified in the node or cell in order for this command to be available.
Attach Project Document	Attaches a document from inside the project to the node or cell.
Add Document	Adds a document from outside the project to the node or cell.

1.3.4 Designing Framework Diagrams and Matrices

Framework architects (users who have selected the Authorize Framework Design user profile when installing PowerDesigner) can create, design, and delete framework diagrams and matrices in projects, by enabling the Framework Design Mode.

Procedure

Select *Tools* > *Framework Design Mode*.

In the framework matrix, you can alternately right-click the framework matrix top-left corner, and select *Framework Design Mode*. The design icon displays in the matrix top-left corner.

In the framework diagram, the Toolbox displays in the diagram.

1.3.4.1 Creating a Framework Diagram

Framework architects can create one or more framework diagrams to guide their modeling teams through building a project. You can specify actions for each node in the framework diagram that control what kind of document (model, diagram, model objects list etc.) must be attached to them.

Procedure

1. Right-click a project in the Browser, and select *New* > *Framework Diagram* to open the framework diagram property sheet.
2. Enter a name for the framework diagram on the *General* tab, and click *OK* to close the property sheet.
3. Click the *Node* tool in the Toolbox, click in the diagram, and create an appropriate number of framework diagram nodes.
4. Click the *Link* tool in the Toolbox, click in the diagram, and create an appropriate number of framework diagram links between the nodes to express connections between them.
5. [optional] Decompose one or more nodes to analyze them in more detail (see Decomposing a Framework Node [page 66]).
6. Create actions for each node to specify the types of documents that your modelers can attach to the node (see Specifying Framework Node and Cell Actions [page 67]).
7. [optional] Select *File* > *Save As Template* to open the Project Template Wizard, and convert the project containing your framework diagrams into a template to allow it to be reused as the basis for future projects (seeProject Templates [page 81]).

1.3.4.1.1 Decomposing a Framework Node

You can decompose one or more nodes in the framework diagram to analyze them in more detail. The decomposed node has its own sub-diagram, which models the links between its sub-nodes. Sub-nodes can, themselves, be decomposed into further sub-nodes, and so on until you obtain a sufficient level of detail.

Context

You can decompose a node in a framework diagram in any of the following ways:

- Press *ctrl* and double-click the node symbol (this will open the sub-node directly)
- Open the property sheet of the node and, on the *General* tab, select the *Composite* check box.

Decomposed node symbols in a framework diagram carry a plus-sign symbol to indicate that they contain further detail

The sub-diagram of the decomposed node is empty at first. You have to create the appropriate number of nodes and links between them to show their connections (see Creating a Framework Diagram [page 65]).

1.3.4.2 Creating a Framework Matrix

Framework architects can create a framework matrix to guide their modeling teams through building a project. You can specify actions for each cell in the framework matrix that control what kind of document (model, diagram, model objects list etc.) must be attached to them.

Procedure

1. Right-click a project in the Browser, and select *New* > *Framework Matrix Diagram* to open the framework matrix property sheet, or right-click the background of any project diagram, and select *Diagram* > *New Diagram* > *Framework Matrix*, or select *View* > *Diagram* > *New Diagram* > *Framework Matrix*.

   > **i** Note
   >
   > You can create only one framework matrix per project.

2. Enter a name for the framework matrix on the *General* tab (see Framework Matrix Properties [page 87]).
3. Click the *Columns* tab, and create an appropriate number of framework matrix columns, entering a name for each.
4. Click the *Rows* tab, and create an appropriate number of framework matrix rows, entering a name for each.
5. [optional] Click the *Additional Areas* tab, and create an appropriate number of additional areas, which will be displayed outside the main matrix grid, entering a name for each.

6. Click *OK* to close the property sheet and create your framework matrix.
7. Create actions for each cell to specify the types of documents that your modelers can attach to the cell (see Specifying Framework Node and Cell Actions [page 67]).
8. [optional] Select *File* > *Save As Template* to open the New Template Wizard, and convert the project containing your framework matrix into a template to allow it to be reused as the basis for future projects (see Project Templates [page 81]).

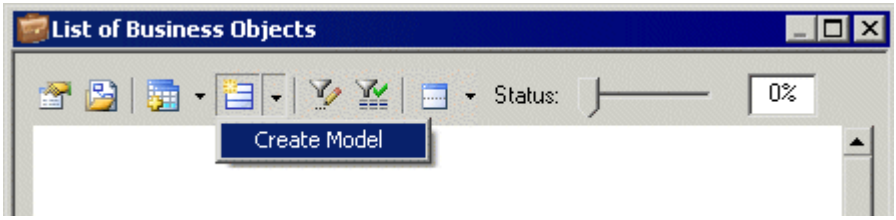
1.3.4.3 Specifying Framework Node and Cell Actions

Actions control the types of documents that can be attached to nodes and cells when completing framework diagrams and matrices. You can specify one or more action types per node and cell.

Procedure

1. Double-click a node or a cell to open its property sheet.
2. On the *General* tab, enter a node or a cell name, and any other appropriate properties (see Framework Node, Cell and Additional Area Properties [page 88]).
3. Click the *Actions* tab, and click the *Add a Row* tool to create a new action.
4. Double-click the action to open its property sheet and, on the *General* tab, enter an action name, and any other appropriate properties (see Framework Action Properties [page 89]).
5. Click the *Detail* tab, and select one of the following types of actions as a means to populate the node or cell:
   - *Model* – a PowerDesigner model (see Example: Specifying a Model Action [page 68]).
   - *Diagram* – a PowerDesigner model diagram (see Example: Specifying a Diagram Action [page 69]).
   - *List* – a list of PowerDesigner model objects (see Example: Specifying a List Action [page 71]).
   - *Model Generation* –a PowerDesigner model generated from another model (see Example: Specifying a Model Generation Action [page 72]).
   - *Object Generation* –a PowerDesigner model populated with one type of objects generated from another model (see Example: Specifying an Object Generation Action [page 74]).
   - *File* – an external file (see Example: Specifying a File Action [page 75]).
   - *Matrix* – a PowerDesigner dependency matrix showing links between model objects (see Example: Specifying a Matrix Action [page 77]).
   - *Script* - a VB script that can define any scriptable action (see Example: Specifying a Script Action [page 78]).
6. Specify any other appropriate properties to refine the action.
7. Click *OK* to close the action and node or cell property sheets and return to the framework diagram or matrix.

   The action is now available for use by framework users (see Attaching a Document to a Framework Node or Cell [page 60]). Note that only fully-defined actions are available for selection.



1.3.4.3.1 Example: Specifying a Model Action

The Model action lets a framework user (add or) create one or more models of the specified kind in the node or cell. The model will be displayed in the Browser under the project node and as a document in the node or cell to which it has been attached.

Context

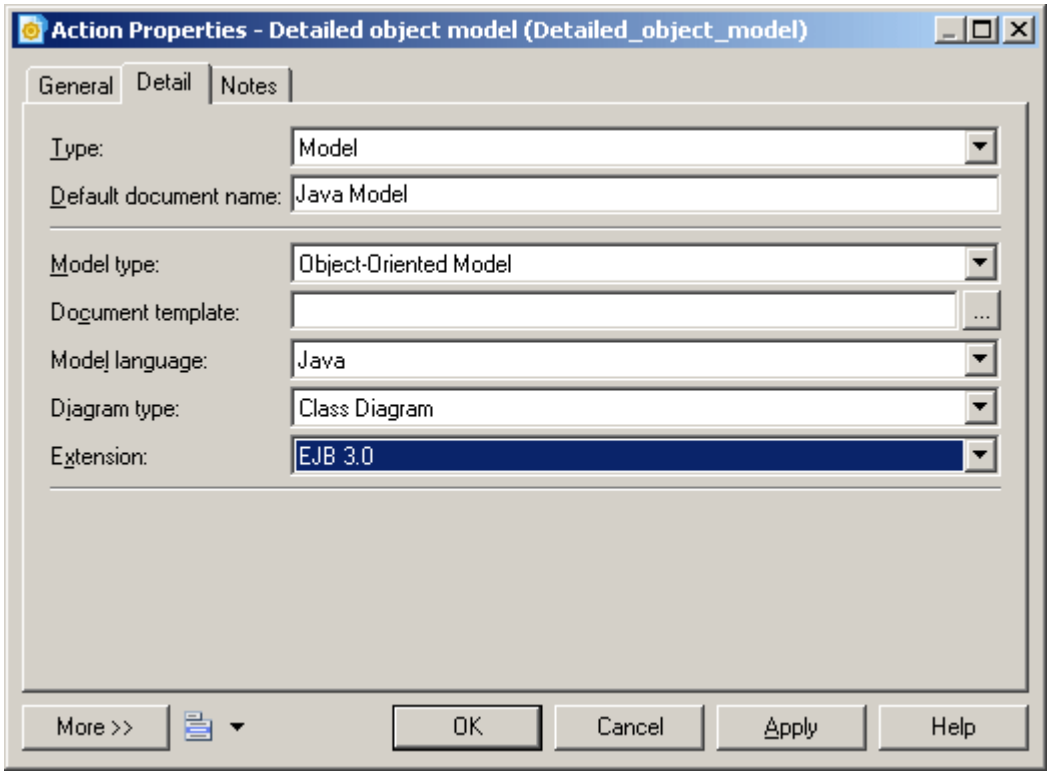
In the following example, we will specify an action in a framework diagram node that enables the user to create one or more Java OOMs with the EJB 3.0 xem attached.

Procedure

1. Double-click a framework node to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter "*Detailed object model*" in the Name field.
4. Select the *Multiple* check box to specify that more than one OOM can be attached to the node.
5. Click the *Detail* tab, and specify the following properties:

Property	Select or enter
Type	*Model*
Default document name	*Java Model*
Model type	*Object-Oriented Model*
Document template	Leave blank
Model language	*Java*
Diagram type	*Class Diagram*

Property	Select or enter
Extension	*EJB3.0*



6. Click OK to close the action and node property sheets and return to the framework diagram.

   The Model action is now available for use by framework diagram users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.2 Example: Specifying a Diagram Action

The Diagram action lets a framework user (add or) create one or more diagrams of the specified kind in the node or cell. The diagram will be displayed in the Browser under an existing or new model node and as a diagram in the node or cell to which it has been attached.

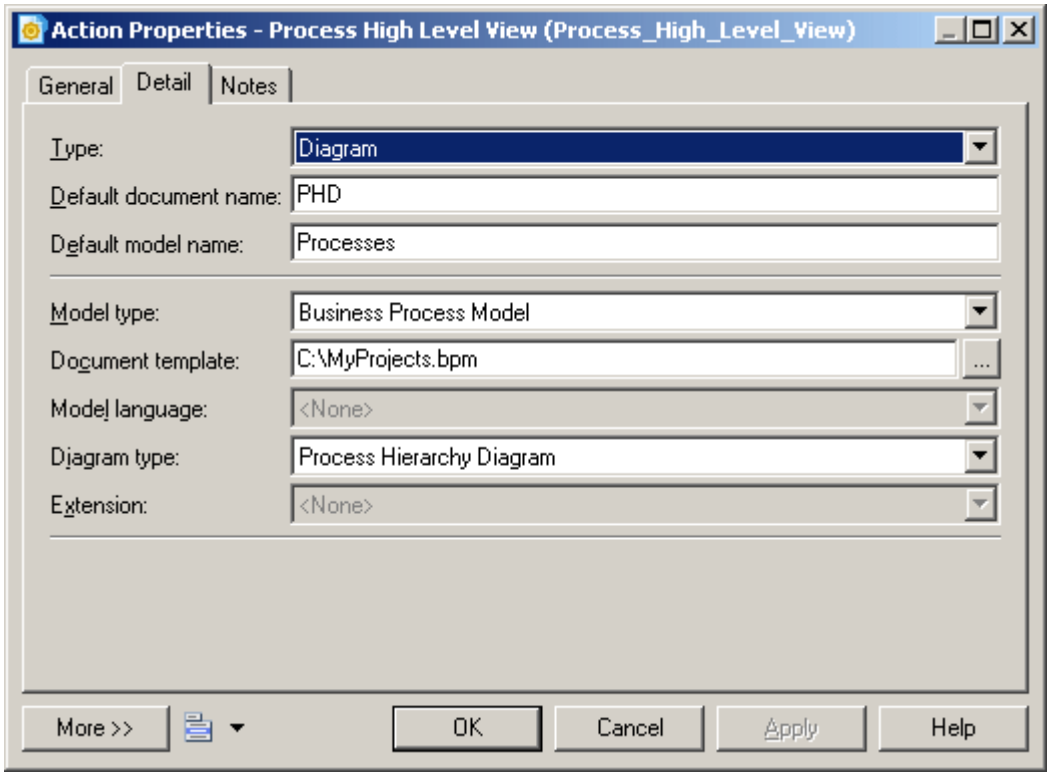
Context

In the following example, we will specify an action in a framework matrix cell that enables the user to create exactly one BPM process hierarchy diagram from a previously defined template.

Procedure

1. Double-click a framework cell to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter "*Process high level view*" in the Name field.
4. Clear the *Multiple* check box to specify that exactly one BPM process hierarchy diagram can be attached to the cell.
5. Click the *Detail* tab, and specify the following properties:

Property	Select or enter
Type	*Diagram*
Default document name	*PHD*
Default model name	*Processes*
Model type	*Business Process Model*
Document template	Click the Ellipsis button to select the *Orders.bpm* template. For information about creating model templates see Model Templates [page 14].
Diagram type	*Business Process Diagram*



6. Click *OK* to close the action and cell property sheets and return to the framework matrix.

   The Diagram action is now available for use by framework matrix users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.3 Example: Specifying a List Action

The List action lets a framework user (add or) create one or more model object lists of the specified kind in the node or cell. The list will be displayed in the Browser under the appropriate model node and as a document in the node or cell to which it has been attached.

Context

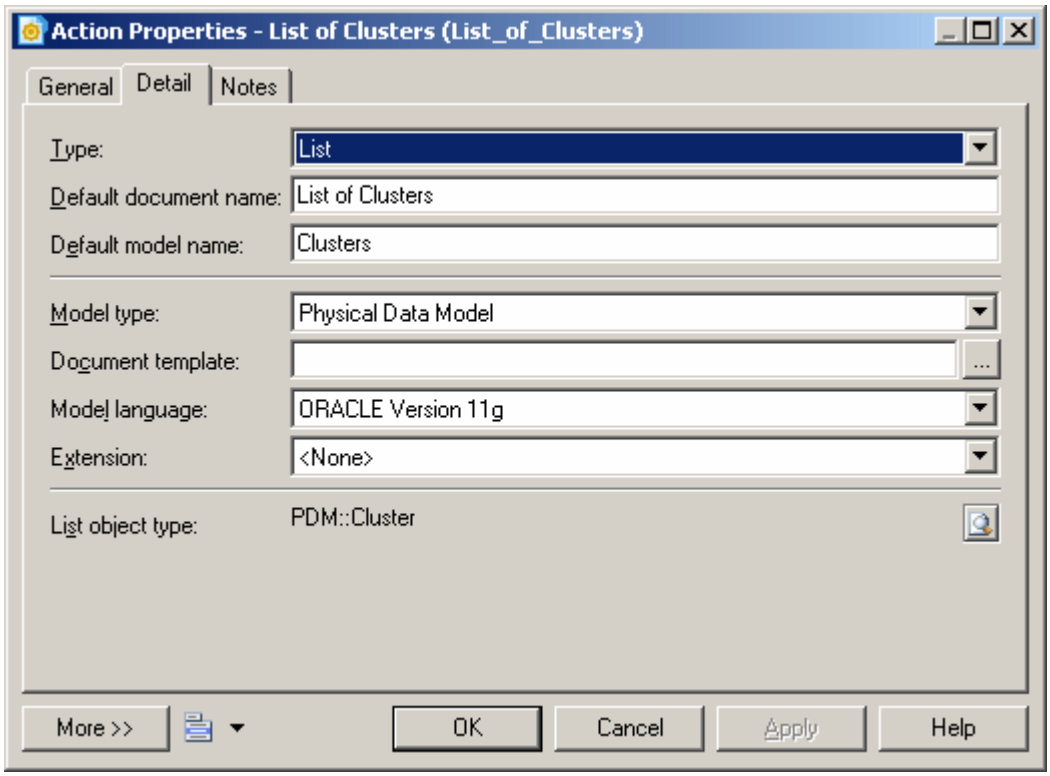
In the following example, we will specify an action in a framework diagram node that requires the user to create at least one list of PDM clusters.

Procedure

1. Double-click a framework node to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter "*List of PDM clusters*" in the Name Field.
4. Select the *Mandatory* check box to specify that at least one list of PDM clusters must be attached to the node.
5. Click the *Detail* tab, and specify the following properties:

Property	Select or enter
Type	*List*
Default document name	*List of clusters*
Default model name	*Clusters*
Model type	*Physical Data Model*
Document template	Leave blank
Model language	*ORACLE Version 11g*
Extension	*<none>*

Property	Select or enter
List object type	*PDM::Cluster* (click the *Select Object* tool, navigate to Oracle Version 11g, and select the Cluster node)



6. Click *OK* to close the action and node property sheets and return to the framework diagram.

   The List action is now available for use by framework diagram users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.4 Example: Specifying a Model Generation Action

The Model Generation action lets a framework user generate one or more models from another model. The generated model will be displayed in the Browser under the project node and as a document in the node or cell to which it has been attached.

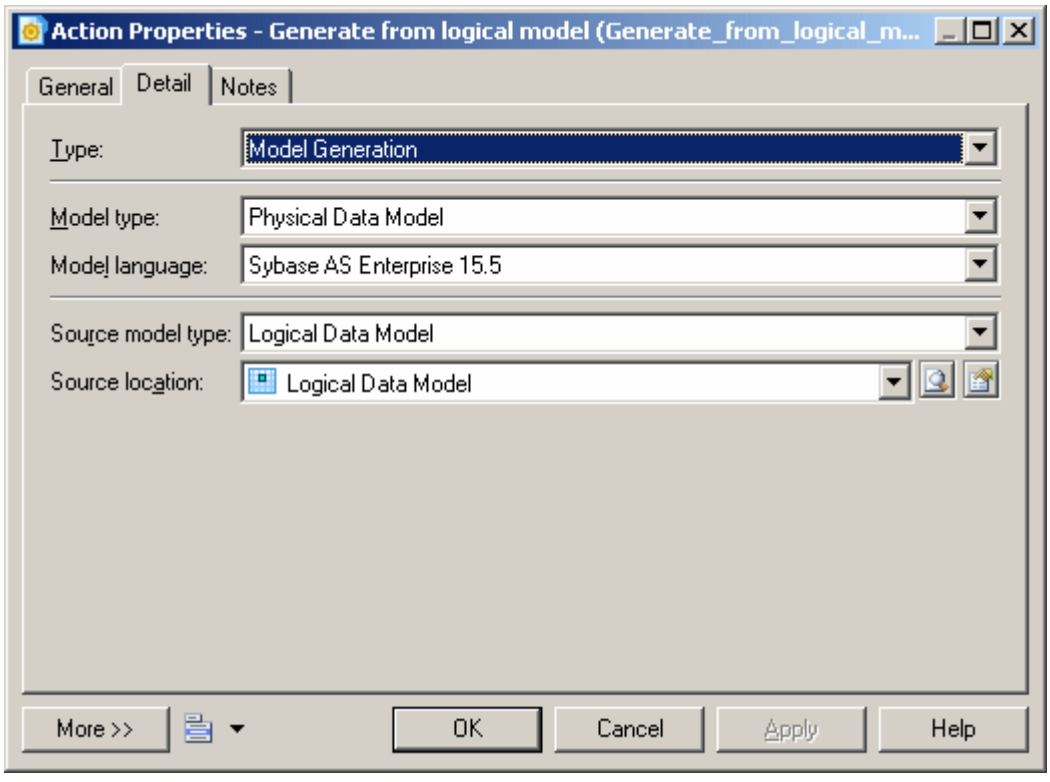
Context

In the following example, we will specify an action in a framework matrix cell that enables the user to generate exactly one Sybase AS Enterprise 15.0.2 PDM from a Logical Data Model.

Procedure

1. Double-click a framework cell to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter **`Generate from Logical Model`** in the *Name* field.
4. Clear the *Multiple* check box to specify that exactly one generated PDM can be attached to the cell.
5. Click the *Detail* tab, and specify the following properties:

Property	Select or enter
Type	*Model Generation*
Model type	*Physical Data Model*
Model language	*Sybase AS Enterprise 15.5*
Source model type	*Logical Data Model*
Source location	*Logical data model*



6. Click *OK* to close the action and cell property sheets and return to the framework matrix.

   The Model Generation action is now available for use by framework matrix users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.5 Example: Specifying an Object Generation Action

The Object Generation action lets a framework user generate objects from one model to another. The generated model will be displayed in the Browser under the project node and as a document in the node or cell to which it has been attached.

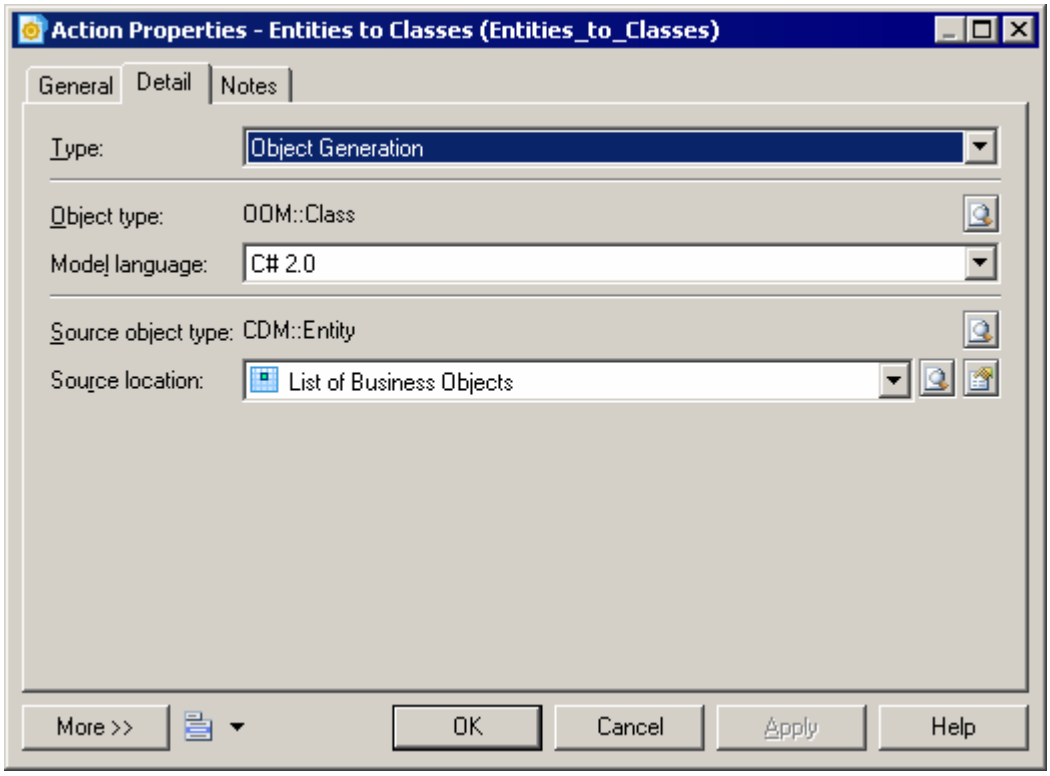
Context

In the following example, we will specify an action in a framework matrix cell that enables the user to generate exactly one Java OOM containing classes from a list of entities.

Procedure

1. Double-click a framework cell to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter `Classes from Entities` in the *Name* field.
4. Clear the *Multiple* check box to specify that exactly one generated OOM can be attached to the cell.
5. Click the *Detail* tab, and specify the following properties in the fields:

Property	Select or enter
Type	`Object Generation`
Object type	Click the *Select Object* tool, navigate to *PdOOM* > *Class* in the dialog, and click *OK* to select it.
Model language	*C# 2.0*
Source object type	Click the *Select Object* tool, navigate to *PdCDM* > *Entity* in the dialog, and click *OK* to select it.
Source location	`List of Business Objects`



6. Click *OK* to close the action and cell property sheets and return to the framework matrix.

   The Object Generation action is now available for use by framework matrix users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.6 Example: Specifying a File Action

The File action lets a framework user (add or) create one or more files of the specified kind in the node or cell. The file will be displayed in the Browser under the project node and as a document in the node or cell to which it has been attached.

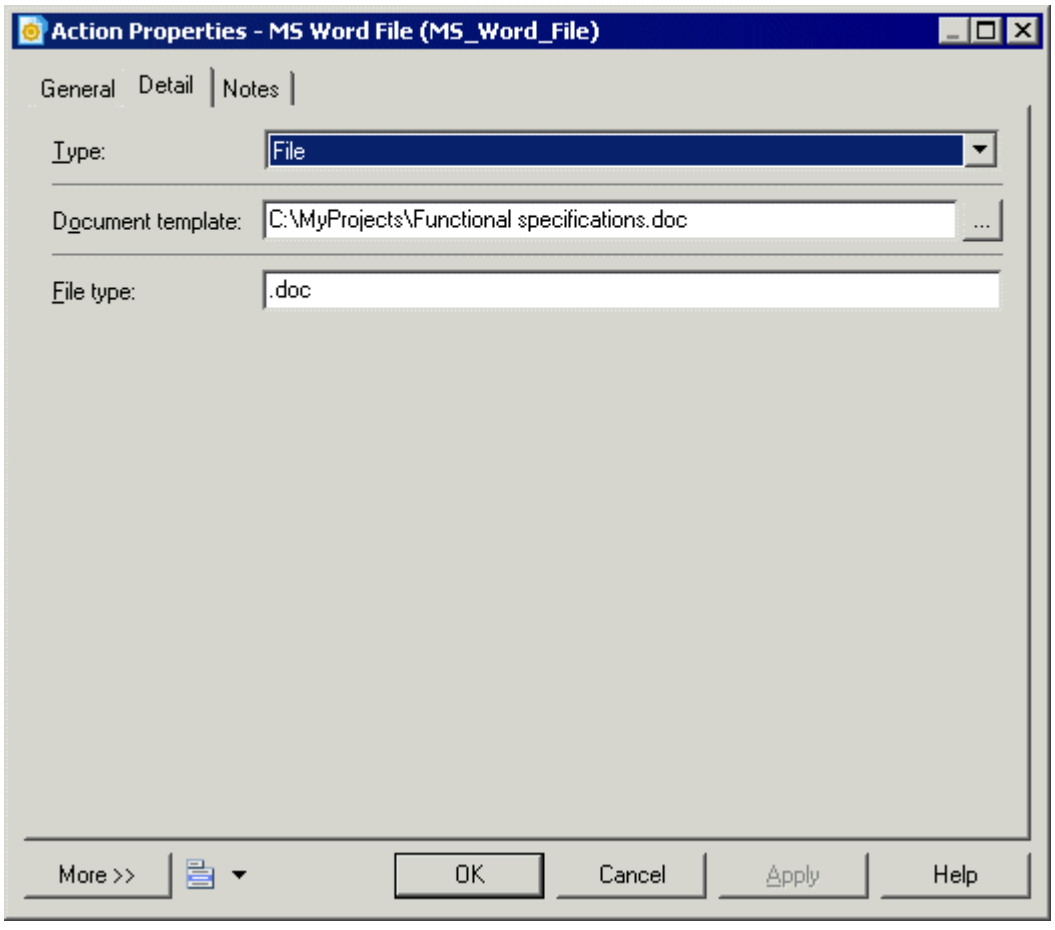
Context

In the following example, we will specify an action in a framework diagram node that requires the user to create at least one MS Word file from a predefined template.

Procedure

1. Double-click a framework node to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter "*MS Word File*" in the Name field.
4. Select the *Mandatory* check box to specify that at least one MS Word file must be attached to the node.
5. Click the *Detail* tab, and specify the following properties in the fields:

Property	Select or enter
Type	*File*
Document template	Click the *Ellipsis* button to select the *Functional specifications.doc* template.
File type	*.doc*



6. Click *OK* to close the action and node property sheets and return to the framework diagram.

   The File action is now available for use by framework diagram users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.7 Example: Specifying a Matrix Action

The Matrix action lets a framework user (add or) create one or more dependency matrices of the specified kind in the node or cell. The dependency matrix will be displayed in the Browser under the project node and as a document in the node or cell to which it has been attached.

Context

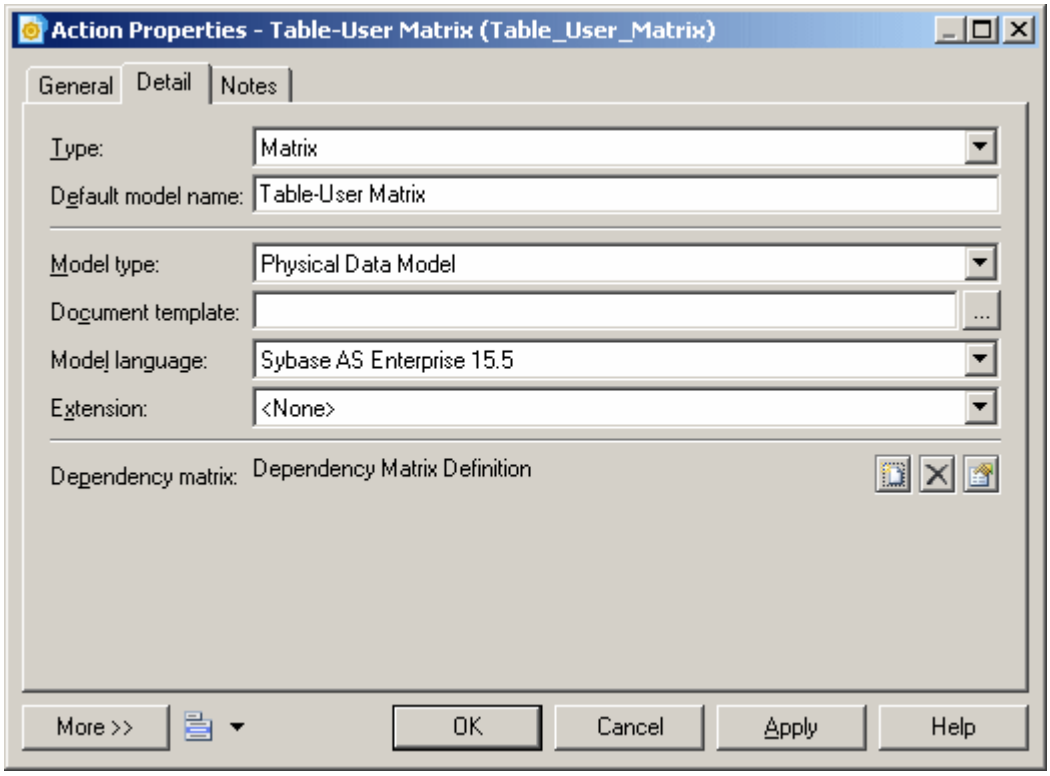
In the following example, we will specify an action in a framework matrix cell that enables the user to create exactly one dependency matrix between PDM tables and users connected by the Owner dependency.

Procedure

1. Double-click a framework cell to open its property sheet, and click the *Actions* tab.
2. Click the *Add a Row* tool to create a new action, and double-click it to open its property sheet.
3. On the *General* tab, enter **`Table-User Matrix`** in the *Name* field.
4. Clear the *Multiple* check box to specify that only one dependency matrix can be attached to the cell.
5. Click the *Detail* tab, and specify the following properties:

Property	Select or enter
Type	**`Matrix`**
Default model name	**`Table-User Matrix`**
Model type	**`Physical Data Model`**
Document template	Leave blank
Model Language	**`Sybase AS Enterprise 15.5`**
Extension	**`<none>`**
Dependency matrix	Click the *Create* tool to open the dependency matrix property sheet and specify the following values: ○ Matrix rows object type – **`Table`** ○ Matrix columns object type – **`User`** ○ Matrix cells dependency – **`Owner`**

6. Click *OK* to close the dependency matrix property sheet, and return to the action property sheet.



7. Click *OK* to close the action and cell property sheets and return to the framework matrix.

   The Matrix action is now available for use by framework matrix users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.3.8 Example: Specifying a Script Action

The Script action lets a framework user execute a script in the node or cell. You can use the script to create a project document and attach it to the node or cell.

Context

In the following example, we will specify an action in a framework diagram node that enables the user to execute a script that generates a model from another model. The result of this example is the same as that of a Generation action, but the script is intended to demonstrate the possibilities that you can script in your own projects. You could, for example, write a script to connect to a live database and reverse engineer it into a cell. For information about writing scripts for use in your models and projects, see *Customizing and Extending PowerDesigner > Extension Files > Methods (Profile)* .

Before you can specify a script action, you have to create a method script on the FrameworkAction metaclass in an extension, and then attach it to your project. In this case, the following script is called **GenerateModel**, and will generate a PDM:

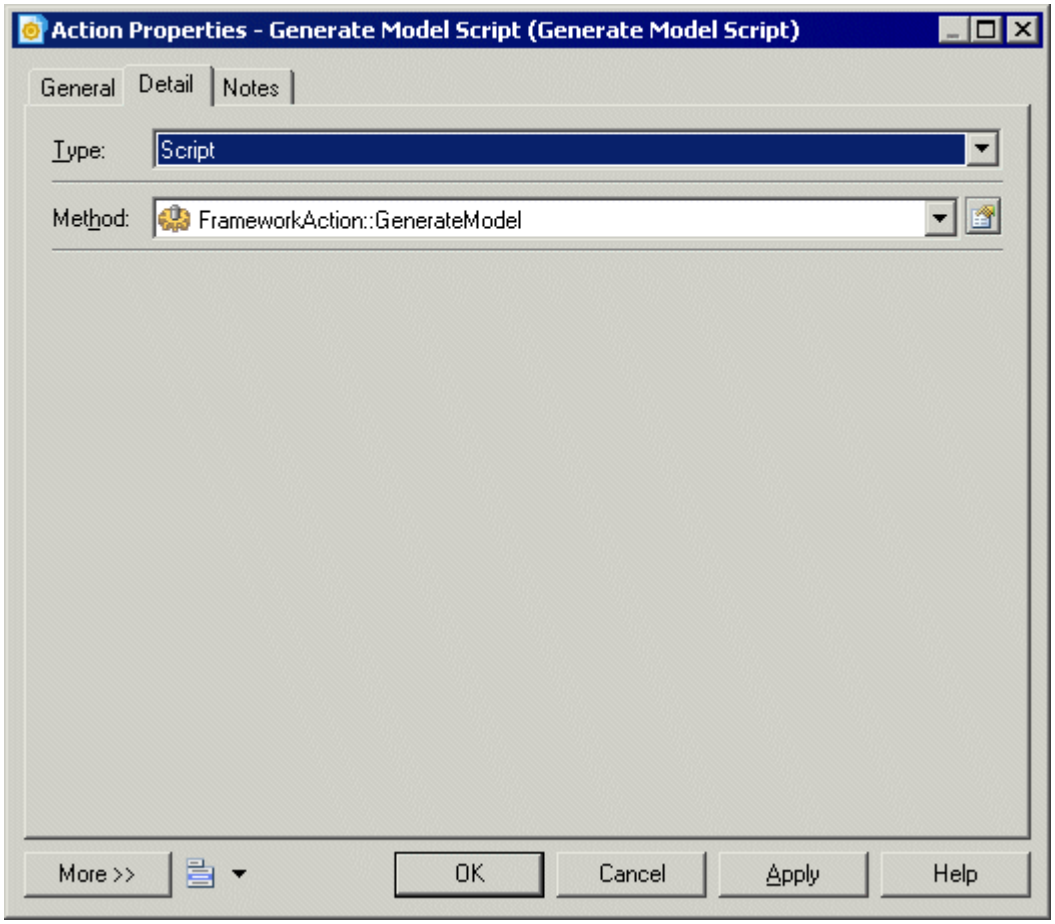
```
Sub %Method%(obj)
   ' This is a sample method that can be used in a script action
   ' It generates a PDM from a source cell CDM and attaches it to the current
cell
   ' Steps:
   ' 1- Get the source cell: found by its code "mySourceCell"
   ' 2- Get the source model: first document in the source cell
   ' 3- Generate a PDM from the source model
   ' 4- Attach the generated model to the current cell
   ' 5- Set current action as source action for the new document
   ' => That will prevent generating twice if the action is not multiple

   If obj Is Nothing Then Exit Sub
   Dim sourceCell, targetCell, modelDoc, sourceModel, targetModel
   ' The script is defined on the action so the current cell is simply the
action parent
   Set targetCell = obj.Parent
   ' First check if we can execute the action on the current cell
   ' for non multiple actions, the CanExecute should return true only the first
time
   If obj.CanExecute(targetCell) Then
      ' Get Source cell
      Set sourceCell = FindCellByCode(targetCell.Parent, "mySourceCell") ' See
function code below End Sub statement
      ' Get source model (supposed to be the first in its artifact document list)
      Set modelDoc = sourceCell.ArtifactDocuments.Item(0)
      Set sourceModel = modelDoc.TargetModelObject
      ' Generate PDM Model
      Set targetModel = sourceModel.GenerateModel (Nothing, PdPDM.cls_Model)
      ' Attach generated model to current cell
      Set modelDoc = targetModel.SourceModelDocument
      targetCell.AttachDocument(modelDoc)
      ' Set current action as source for the new model document
      obj.SetAsSource(modelDoc)
   Else
      ' In this sample, the action is supposed to be non-multiple
      ' Therefore, CanExecute fails if it's already a source action for an
existing document
      output "The action has already been executed"
   End If
End Sub
' FindCellByCode function (Global Script function):
Function FindCellByCode (fmx, Code)
   Set FindCellByCode = Nothing
   Dim Cell
   For Each Cell In fmx.cells
      If Cell.Code = Code Then
         Set FindCellByCode = Cell
         Exit For
      End If
   Next
End Function
>>
```

Procedure

1. Double-click a framework node to open its property sheet, and click the click the *Actions* tab.

2. On the *General* tab, enter **`Generate Model Script`** in the *Name* field.
3. Clear the *Multiple* check box to specify that exactly one generated model can be attached to the node.
4. Click the *Detail* tab, and specify the following properties in the fields:

Property	Select
Type	Script
Method	GenerateModel



5. Click *OK* to close the action and node property sheets and return to the framework diagram.

   The Script action is now available for use by framework diagram users (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.4.4 Styling Framework Nodes and Cells

Framework architects have access to additional commands that allow them to change the format of framework nodes and cells.

Context

Command	Description
Format	Opens the Symbol Format dialog box to let you specify line and fill colours, fonts and formatting for each textual element, and an image for the node or cell (see Symbol Format Properties [page 214]).
Change Image	Opens the Select Image dialog box to let you insert an image in the node or cell (see Browsing for Images [page 222]).
Get Format	Copies the format of the selected node or cell for pasting in other nodes or cells.
Apply Format	Pastes a previously copied node or cell format into the selected node or cell.
Properties	Opens the property sheet of the selected node or cell.

For information about the standard commands available outside design mode, see Working with the Framework Node and Cell Contextual Menu [page 64].

1.3.5 Project Templates

A project template can provide predefined content, rules, and formatting for your project.

Context

For example, you could create:

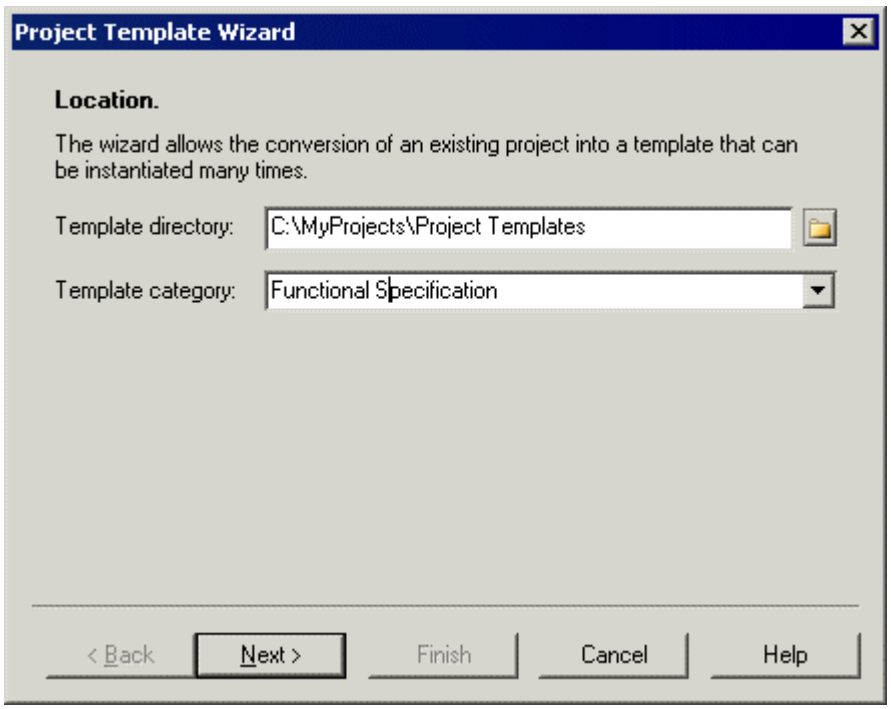
- A data project template - containing a CDM, LDM, and PDM targeting your preferred DBMS.
- An application project template - containing an OOM with the UML diagrams required for your modeling practices, along with a PDM and XSM to handle data persistance and format.
- A framework project template - containing a FEAF or other framework matrix.

Procedure

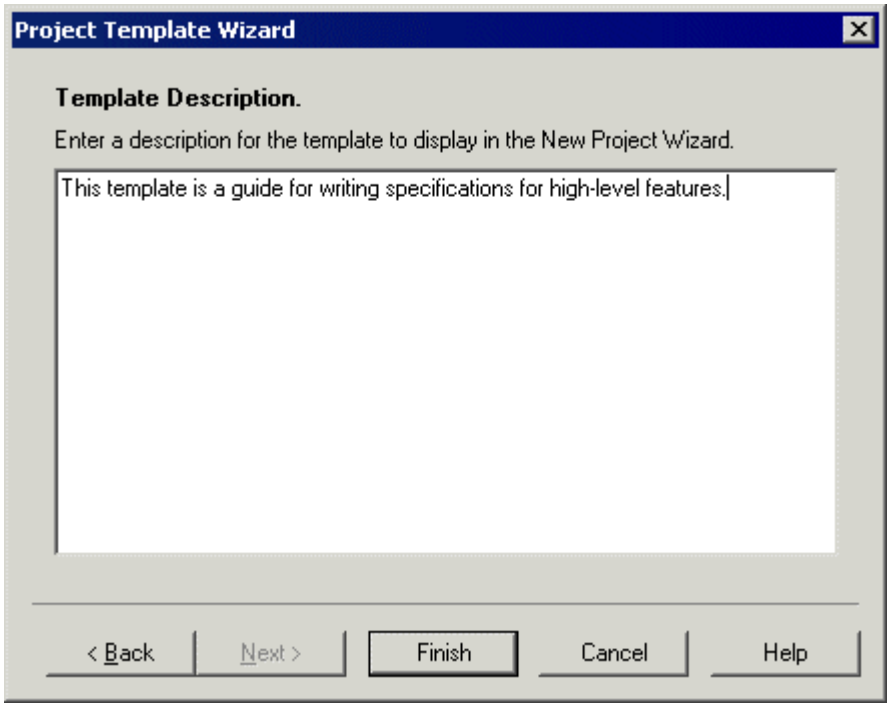
1. Create a project (see Creating a Project [page 48]).
2. Create the content that you want to make available in your template:
   - [optional] Add any appropriate models or files (see Adding Model and File Documents to a Project [page 49]).
   - [optional] Create one or more framework diagrams (see Creating a Framework Diagram [page 65]).
   - [optional] Create a framework matrix (see Creating a Framework Matrix [page 66]).
   - [optional] Specify any appropriate display preferences (see Project and Framework Display Preferences [page 54]).
3. Select *File* > *Save As Template* to open the Template Wizard and click *Next*.

> **i Note**
>
> If you have unsaved changes to the contents of your project, you will be prompted to save them before beginning the wizard.

4. Specify a directory in which to save the project template. By default, the project template directory (defined via *Tools* > *General Options* > *Project Templates*) is selected. Click the *Change Template Directory* tool to select another location.
5. Select a template category from the list or enter a template category name that will appear in the New Project dialog. You can use the \ character to create sub-categories.



6. Click *Next*, and enter a description for the new template that will be displayed in the New Project dialog.



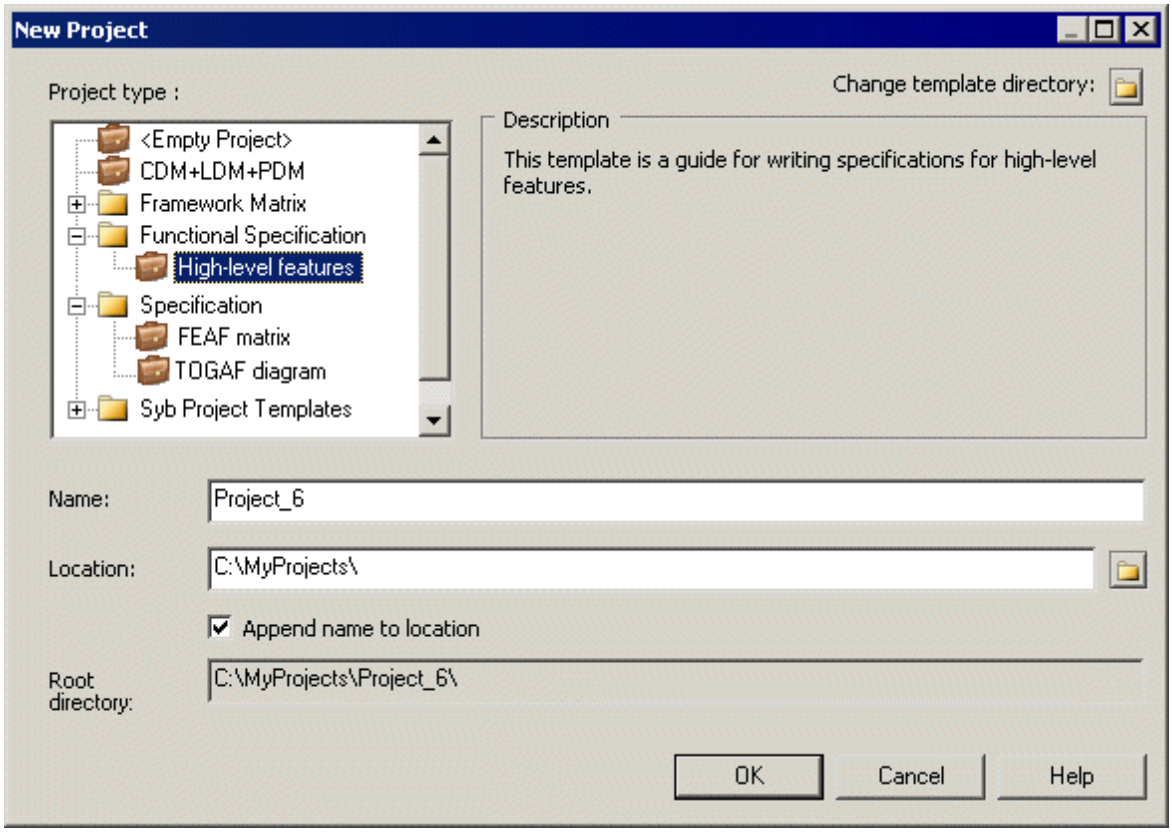
> **i** Note
>
> You can modify a template description by right-clicking a project template in the Browser, and selecting *Edit Template*.

7. Click *Finish* to close the wizard and create the template.

   The project (.prj) and the models and files it contains are copied to the template directory.

   > **i** Note
   >
   > If any documents are outside the project and only attached to it they will not be copied to the template directory and thus may not be available to a user when she creates a project from the template.

   When you select *New* > *Project*, the template is available for selection in the Project type tree:



8. [optional] Check the project template into the repository `Library/Resources/Project Templates` folder to make it available to other users (see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

1.3.6 Project and Framework Properties

Projects, model and file documents, and framework diagrams and matrices have a variety of properties that you can configure to control their behavior.

1.3.6.1 Project Properties

To view or edit a project's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
File name	Specifies the location of the project file. This field is empty if the project has never been saved.
Author	Specifies the author of the project. If you enter nothing, the Author field in diagram title boxes displays the user name from the project property sheet Version Info tab. If you enter a space, the Author field displays nothing.
Version	Specifies the version of the project. You can use this box to display the repository version or a user defined version of the project. This parameter is defined in the Title page of the project display preferences.
Default diagram	Specifies the diagram displayed by default when opening the project.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.3.6.2 Model and File Document Properties

To view or edit a document's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name and code are read-only for model documents. You can optionally add a comment to provide more detailed information about the object.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Location	Specifies the path or URL to the document. Read-only for model documents.
[read-only]	The property sheet provides read-only information on the extension, size, and creation and modification dates of the document, as well as noting whether it is inside the project or simply referenced by it.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.3.6.3 Dependency Link Properties

To view or edit a dependency link's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Type	[read-only] Specifies the type of the dependency link, which can include generation links, reference links, mapping links, and file links.
Influent object	[read-only] Specifies the source object of the dependency link. Click the Properties tool beside the list to open its property sheet.
Dependent object	[read-only] Specifies the target object of the dependency link. Click the Properties tool beside the list to open its property sheet.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.3.6.4 Framework Matrix Properties

Framework architects can modify a framework matrix properties from its property sheet by enabling the Framework Design Mode, and double-clicking the framework matrix top-left corner.

For more information on enabling the Framework Design Mode, see Designing Framework Diagrams and Matrices [page 65].

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

In addition to the *General* tab, the framework matrix property sheet contains the following tabs:

- Rows tab – displays a sortable list of rows, in which you can create, edit or delete rows. See Framework Column and Row Properties [page 87].
- Columns tab – displays a sortable list of columns, in which you can create, edit or delete columns. See Framework Column and Row Properties [page 87].
- Cells tab – displays a list of cells, which is computed from the intersection of columns and rows or from the creation of additional areas. You can edit cells but you cannot delete them unless you delete the columns or rows to which they belong. See Framework Node, Cell and Additional Area Properties [page 88].
- Additional Areas tab – displays the four additional areas available to the framework matrix. You can create, edit or delete additional areas. See Framework Node, Cell and Additional Area Properties [page 88].

1.3.6.5 Framework Column and Row Properties

[Framework architects] To view or edit a column or row's properties, enable the Framework Design Mode, and double-click a column or row header in the framework matrix.

For more information about enabling the Framework Design Mode, see Designing Framework Diagrams and Matrices [page 65]).

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Help file	Specifies a location for the help file associated with the row or column.
Default folder	Specifies a default folder for the creation of the cell's documents. If no default folder is specified in the cell's property sheet, the row's default folder is used first, then the column's default folder.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.3.6.6 Framework Node, Cell and Additional Area Properties

[Framework architects] To view or edit a node, cell or additional area's properties, enable the Framework Design Mode, and double-click a node in the framework diagram or a cell or an additional area in the framework matrix.

For more information about enabling the Framework Design Mode, see Designing Framework Diagrams and Matrices [page 65].

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.

Property	Description
Help file	Specifies a location for the help file associated with the node and cell.
Default folder	Specifies a default folder for the creation of the node's and cell's documents.
Cell location	[framework matrix only - read-only] [cell] Specifies the name of the column and row for which the cell is the intersection. Click the Properties tool beside Row and Column fields to open their property sheet. [additional area] Specifies the location of the additional area: Left, Top, Right, or Bottom area.
Completion status	Controls the percentage of work completed in the node and cell. Enter a value between 1 and 100 in the box. This value is also displayed in the node's and cell's floating list and a rectangle on the node and cell shows the work progression if you have selected the Completion status display preference.
Inactive	[cell only] Specifies a cell which is not available.
Constrain content	Specifies a node and cell, whose content can only be completed by its defined actions.
Composite	[node only] Specifies whether the node is decomposed into sub-nodes displayed in sub-diagrams. When selected a Sub-Nodes tab is displayed in the property sheet to list these sub-nodes. If you deselect this property, then any sub-nodes that you have created will be deleted.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

In addition to the *General* tab, the framework node and cell property sheet contains the following tabs:

- *Actions* tab – displays a list of actions, in which you can create, edit or delete actions (see Framework Action Properties [page 89]).
- *Documents* tab – displays a list of documents, in which you can create or add existing documents, and also edit or delete documents (see Attaching a Document to a Framework Node or Cell [page 60]).

1.3.6.7 Framework Action Properties

[Framework architects] To view or edit an action's properties, enable the Framework Design Mode, and double-click an action on the *Actions* tab of a node or cell property sheet.

> **i Note**
>
> For information about enabling the Framework Design Mode, see Designing Framework Diagrams and Matrices [page 65].

The *General* tab contains the following properties:

Property	Description
Parent	[read-only] Specifies the name of the parent node and cell to which the action belongs.
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Multiple	When selected, specifies that one or more documents of the selected type can be created.
Mandatory	When selected, specifies that at least one document of the selected type must be created.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

Detail Tab

The *Detail* tab specifies the nature of the action.

Property	Description
Type	Specifies the nature of the document to attach to the framework nodes and cells (see Specifying Framework Node and Cell Actions [page 67]).
Default document name	[Model, Diagram, and File only] Specifies the model, diagram or file name created by default in the Browser.
Model type	[not for File and Dependency Matrix] Specifies a model type to create, for example Enterprise Architecture Model.
Document template	[not for Generation and Dependency Matrix] Specifies the model or file template on which the model or file to create is based, for example a PDM, an LDM, or a .doc, a .xls etc. Click the Ellipsis button to browse for a file on your system.

Property	Description
Model language	[not for File and Dependency Matrix] Specifies a target language associated with the model to create, for example Analysis for a BPM or Java for an OOM. Select a model language in the list. If you do not specify a language, your system default model language is used.
Diagram type	[Model and Diagram only] Specifies a diagram type for the model, for example Class Diagram for an OOM, or Business Process Diagram for a BPM.
Extension	[not for File and Dependency Matrix] Specifies an extension file attached to the model, for example SIMUL8 for a BPM.
Object type	[List only] Specifies the model object type in the list, for example entity, process, table, class etc.
Object stereotype	[List only] Specifies a stereotype to filter the metaclass, for example a <<cluster>> stereotype on an extended object in a PDM targeted for the Oracle Version 9i DBMS.
Dependency matrix	[Dependency Matrix only] Specifies a dependency matrix definition between model objects. Click the Create tool to create a new dependency matrix.
File type	[File only] Specifies the extension of the file to create, for example, .doc, .xsl, .txt, etc.
Source model type	[Generation only] Specifies the model type from which to generate the model, for example Business Process Model, Conceptual Data Model etc.
Source cell	[Generation only] Specifies the node and cell containing the model type from which to generate the new model.

1.3.6.8 Framework Link Properties

[Framework architects] To view or edit a framework link's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

For more information about enabling the Framework Design Mode, see Designing Framework Diagrams and Matrices [page 65].

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
First object	Specifies the node that the link leads from. You can click the Properties tool to the right of the list to view the properties of the selected object.
Second object	Specifies the node that the link leads to. You can click the Properties tool to the right of the list to view the properties of the selected object.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.4 Objects

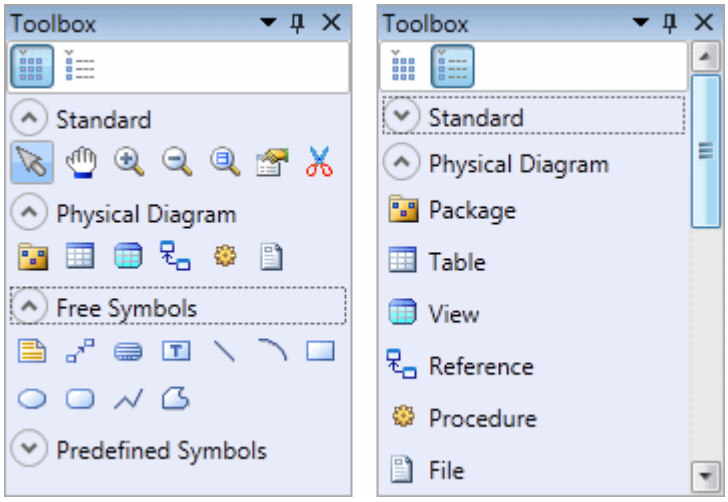
Modeling objects, such as tables and entities, are the building blocks of your models. All the objects in a model are listed as items in the Browser and they may also appear as symbols in your diagrams.

Each type of model diagram (see Diagrams [page 185]) supports certain specific types of objects, and each object created in a diagram is represented by a symbol (see Symbols [page 203]). A single object can have multiple symbols associated with it to allow it to appear in multiple diagrams or multiple times in the same diagram (see Creating Graphical Synonyms for Object Symbols [page 220]).

> **i** Note
>
> For detailed information about a specific type of model or object, see the appropriate modeling guide. The following sections describe generic features available for all objects.

1.4.1 Creating Objects from the Toolbox

The toolbox is a toolbar that lets you quickly create objects in diagrams. To create an object, click the appropriate tool and then click in the diagram. When you release the mouse button, the object is created, and you can click again elsewhere to create a second object and so on.



To release the tool, simply select another tool or right-click in free space. To revert to a previously selected tool, hold down `Ctrl` and perform a double right-click.

To create a link, click the appropriate link tool, click in the object from which you want the link to begin and hold the mouse button as you drag the pointer to the object in which you want the link to end. When you release the mouse button, the link is created.

The toolbox contains four categories that can be collapsed or expanded by clicking on the up and down arrows:

- *Standard* - containing the basic tools available in all diagrams:

Icon	Action
	Pointer [default] – Selects, moves, and resizes individual symbols. Double-click to select all the objects in the diagram. To switch back to the Pointer from another tool, right-click anywhere in the diagram.
	Grabber – Single-click to select the entire diagram. Double-click to display the entire diagram, centered.
	Zoom In/Zoom Out.
	Open Diagram - Click a composite object or package to open its default diagram.
	Delete - Click a symbol to delete it and, optionally, its associated object (see Deleting Objects [page 106]).
	Properties - Click a symbol to open the property sheet of its object (see Object Properties [page 107]).

- `<Type>` *Diagram* - containing tools specific to the current diagram (see the appropriate modeling guide) and the following tools for objects that are available in all diagrams:

Icon	Action
	Package – see Packages [page 143].
	File - see File Objects [page 141].

- *Free Symbols* - containing tools for drawing free shapes and links to decorate your diagrams (see Decorative Symbols [page 225]).
- *Predefined Symbols* - containing tools for drawing standard flowchart shapes.

> **i** Note
>
> To rename an object from its symbol, select it and then press `F2`. If the symbol carries additional editable properties (but not lists of sub-objects), you can navigate to the next with `Tab` (or the previous with `Shift` + `Tab`). Select the *Edit in place after creation* general option to have the name of each object that you create immediately selected for editing (see General Options [page 388]).

1.4.2 Creating Objects from the Browser

You can create any object, including objects that do not have symbols, from the Browser. If the Browser is not visible, select *View* > *Browser* (or press *Alt+0*) to display it.

Procedure

1. Right-click the model, package, object, or category where you want to create the object, and select *New* > `<object type>`.
2. Type an object name and code in the *General* tab of the property sheet, and then add any other relevant properties (see Object Properties [page 107]).
3. Click *OK* to confirm the creation of the object.

   The object is created in the appropriate category in the Browser. If the object has a symbol, it will also be added (where possible) to the current diagram.

1.4.3 Creating Objects from an Object List

You can create most types of objects in object lists, available under the *Model* menu. Some property sheet tabs, such as a table's *Columns* tab or a class's *Attributes* or *Operations* tabs also contain object lists. Lists provide a method to quickly create multiple objects and change their properties simultaneously.

Procedure

1. Select *Model* > `<object type>` (or open an appropriate property sheet tab).
2. Click the *Add a Row* tool or, if the list is ordered, the *Insert a row* tool.

   A new item is added at the end, or before the selected row, of the list.
3. Enter an object name and code and, for link objects, a source and destination object.
4. Repeat as necessary to create additional objects of the same type, and then click *OK*.

Results

For detailed information about object lists, see Object Lists [page 117].

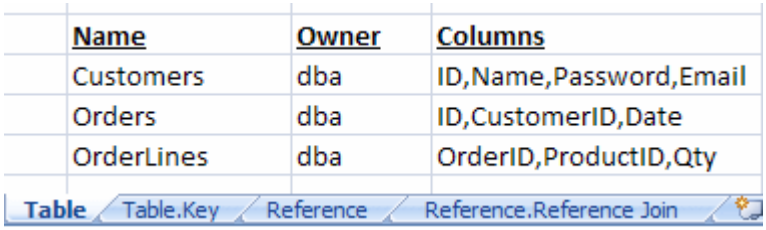
1.4.4 Importing Objects from Excel Files

The Excel Import extension provides a wizard to guide you through mapping tables of objects in an Excel file (v2003 and higher) to PowerDesigner objects and properties, to import them to your model. For example, you could create a list of database tables in an Excel file, specifying any appropriate properties, and then import them into a PDM.

Context

You can import any number of different types of objects from a single Excel file, so long as they can all be imported to a single type of model. Each type of object should be listed in its own table on a separate worksheet of the Excel file. Each row in a table represents one object to import, and each column represents one property (an attribute or list of associated objects) of the object.

In the following example, the Excel file contains worksheets listing tables, keys, references, and reference joins. The **`Table`** sheet, contains a list of three table to import. The columns **`Name`**, **`Owner`**, and **`Columns`** will be imported as the corresponding table properties:



	Name	Owner	Columns
	Customers	dba	ID,Name,Password,Email
	Orders	dba	ID,CustomerID,Date
	OrderLines	dba	OrderID,ProductID,Qty

Table | Table.Key | Reference | Reference.Reference Join

For detailed information about how to organise the file, see Preparing Your Excel File for Import [page 98].

> **i Note**
>
> An example Excel file for importing into a PDM is provided at `<install_dir>\Examples\excel_import.xls` and a set of nine files for importing into an EAM is at `<install_dir>\Examples\EAExample`.

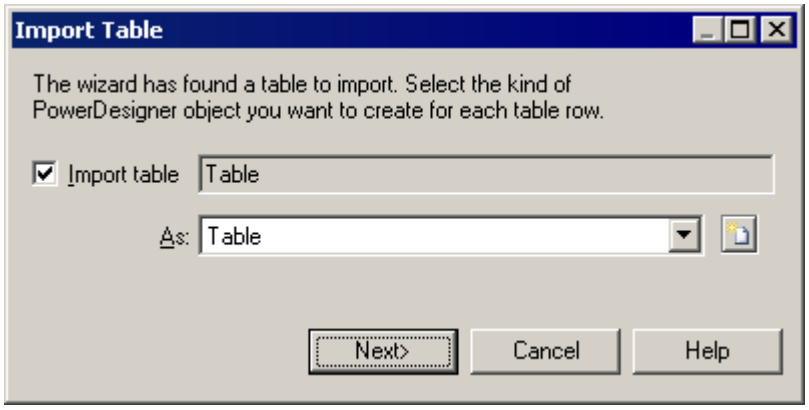
Procedure

1. [optional] Create extensions to allow you to easily import objects or properties not defined in the standard PowerDesigner metamodel (see Extension Files [page 411]).

   If you do not predefine extensions, the wizard will allow you to specify them during the definition of the mappings prior to beginning the import, but predefined extensions can be more sophisticated and allow the import to proceed more quickly.
2. Open the Excel Import Wizard:
   - To create a new model, select *File* > *Import* > *Excel File*. Specify the kind of model you want to create in the New Model dialog (see Creating a Model [page 12]), and then click *OK*.
   - To import objects into an existing model, attach the Excel Import extension (available for all model types on the *Import* subtab of the Select Extensions dialog) to your model (see Attaching Extensions to a Model [page 413]), then right-click the model in the Browser, and select *Import Excel File*.

3. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

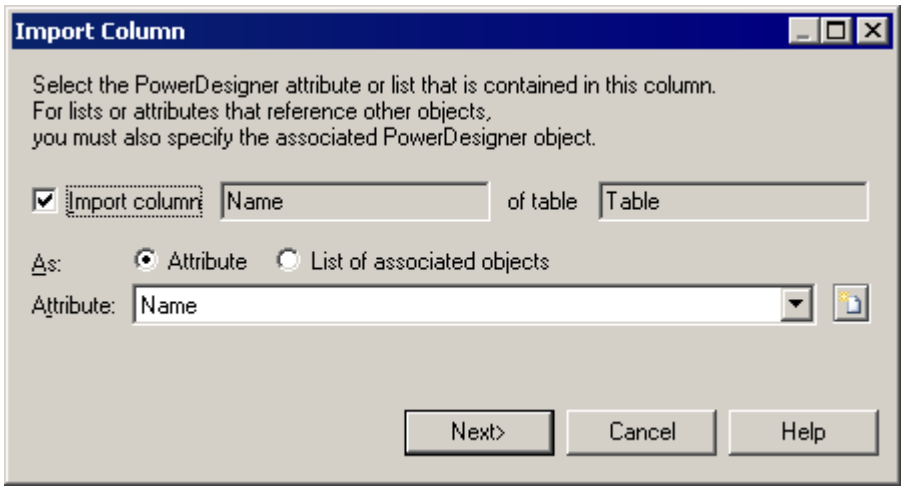
4. [optional] Click the *Options* button to open the Import Options dialog (see Excel Import Options [page 100]), specify any appropriate options and then click *Close* to return to the wizard.
5. Click *Next* to go to the *Import Table* page:



On this page, you must either:

- Select a PowerDesigner object type to import the table lines *As* from the list.
- Select to import the table lines as a new object type by clicking the *New* tool to the right of the list to open the New Object Type dialog.
- Deselect the check box to not import this table. When you click *Next*, PowerDesigner will search the Excel file for another table of objects to import.

6. Click *Next* to go to the *Import Column* page:



On this page, you must either:

- Select a PowerDesigner attribute to import the table column as from the *Attribute* list. To create a new attribute, click the *New* tool to the right of the list.
- Select the *List of associated objects* radio button to import the column as a list, selecting the *List* to use and (if the list can contain multiple types of objects), the appropriate *Object*. To create a new list, click the *New* tool to the right of the list.
- Deselect the check box to not import this table column.

7. Click *Next*. The wizard will search for the next column in the table and if there is none (or if you have selected to skip importing the table), will search for the next sheet containing a table of objects.

   When all the worksheets and columns have been processed, the import will begin. You can stop the import at any time by clicking the *Cancel* button in the bottom-right corner. When the import is complete, a dialog will appear showing how many objects have been created.
8. Click *OK* to return to your model and review the imported objects.

   If you have instructed PowerDesigner to create new object types or new properties, a new embedded extension called **`Import Extensions`** is created under the **`Extensions`** category in the Browser to hold the definitions of these extensions.

Results

During the import, in addition to the objects that are created, PowerDesigner creates an Excel Import object that contains your import parameters, along with a Table Mapping object for each of the imported tables, which lists the column mappings used. Right-click the Excel Import Object to access the following commands:

- *Import* - to re-import your file based on the options and mappings you have already defined. All the objects will be reimported, overwriting any existing objects based on their names. If you have added new objects to your worksheet tables they will be added to the model, but any objects that you have deleted from your worksheet will not be deleted from the model.
- *Change Mappings* - to re-import your file via the Excel Import Wizard in order to modify your table or column mappings.
- *Change Options* - to change the import options preparatory to re-importing with the command *Import*.
- *Properties* - to open the Excel Import properties sheet.

> **i** Note
>
> Advanced users may want to modify the Excel Import XEM to enable the import of data from other external file types. For detailed documentation about its implementation, open the XEM (which is located at `<install_dir>\Resource Files\Extended Model Definitions\ExcelImport.xem` in the Resource Editor and read the detailed commentary on the root element.

1.4.4.1 Preparing Your Excel File for Import

By following certain simple rules, you can greatly help PowerDesigner to recognize the information in your Excel files, and thus improve the accuracy and speed of the import.

Follow these rules to ensure that your import goes as smoothly and quickly as possible:

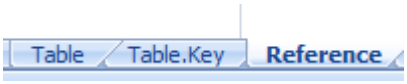
Determine your extensions strategy	The wizard can create simple extensions (new object types or new properties not included in the PowerDesigner metamodel) for you during the import. However, to define more complex extensions or to speed up the import process, you may prefer to predefine your extensions beforehand (see Extension Files [page 411]) and have the import automatically map them.

Insert only one table of objects per worksheet

If you have more than one table on a worksheet, only the first will be recognized and imported.

Name your worksheet after the object to import

If the name of your worksheet matches the name of a PowerDesigner object type, then the wizard will suggest mapping the table to that object type. The wizard does not recognize plural forms so, if your sheet contains tables, you should call it **`Table`**, and if it contains classes, call it **`Class`**.



For sub-objects (objects linked to their parent by composition), you must prefix the name of the object with that of its parent object, (eg **`Table.Key`**) or it will not be recognized.

Name table columns after the property they contain

If you name each of your columns after the attribute or collection that it contains, then the wizard will suggest mapping the column to that attribute or collection. As a general rule, attributes are singular and collections plural so that, for example, you should name the column that contains your tables' names **`Name`** and the column containing its columns, **`Columns`**.

Name	Owner	Columns
Customers	dba	ID,Name,Password,Email

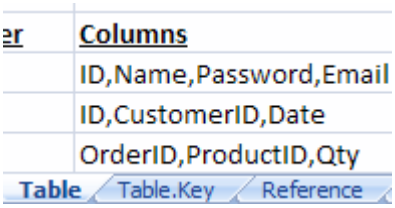
Determine how you will reference other objects

Properties can contain references to other objects. For example, a table Owner property contains a reference to a user object and a business process references the organization unit that performs it. The import tries to resolve references first by searching for an appropriate object in the Excel file, and then in any of the models open in the Workspace (creating a shortcut to the referenced object). Follow these suggestions to optimize your references:

- Control whether references will be made via object names or codes using the *Reference associated objects by* option.
- To maximize the clarity of dependencies between objects, place objects that are to be referenced on a worksheet before the objects that will reference them.
- Ensure that the necessary target models are open in the Workspace to create shortcuts, if appropriate, and select the *Search for referenced objects in other models* option.
- Use the *Create associated objects if not found* option to instruct PowerDesigner to create the necessary objects if it cannot resolved references either in the Excel file or in models open in the Workspace.

Quickly create collections by listing objects in a single cell

You can create a collection simply by listing its items in a single cell, separated by commas:



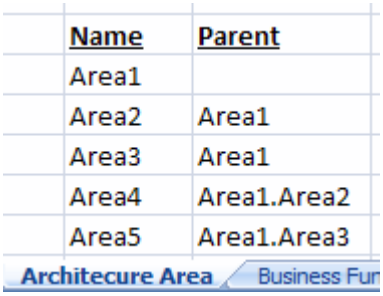
er	Columns
	ID,Name,Password,Email
	ID,CustomerID,Date
	OrderID,ProductID,Qty

Table | Table.Key | Reference

To change the separator, use the *List value separator* option.

Represent relationships of composition using a `Parent` column

To specify composition relationships (such as columns in a table, classes in a package, or attributes in a class) between objects on different worksheets, specify the parent or location for the sub-object in a **`Parent`** column. You can specify a multi-level hierarchy of parents separated by a qualified name separator.



	Name	Parent
	Area1	
	Area2	Area1
	Area3	Area1
	Area4	Area1.Area2
	Area5	Area1.Area3

Architecure Area | Business Fun

> **i Note**
>
> Whereas specifying a collection of sub-objects in a single cell on the parent worksheet allows you to quickly populate the collection, specifying sub-objects on a separate worksheet allows you to detail all their properties.

1.4.4.2 Excel Import Options

The Excel Import Wizard provides a number of options to allow some flexibility in the format of the data it can import.

Option	Description
Auto-map columns to properties	Instructs the wizard to automatically map Excel tables and columns to objects and properties in the PowerDesigner metamodel. If all your Excel worksheets and table column headings are named from the PowerDesigner metamodel (or your predefined extensions), this option can make the import run without requesting further input (see Preparing Your Excel File for Import [page 98]). Default: Unselected

Option	Description
Check data integrity	Enables validation of objects and property values (such as the uniqueness of names and codes) during import. Disabling this option can provide significant performance improvements when importing larger files. Data integrity can be confirmed after the import by selecting *Tools* > *Check Model* (or pressing *F4*) in the model. Default: Selected
Create symbols in active diagram	Instructs the wizard to create symbols, where appropriate, for the imported objects in the currently selected diagram. Default: Unselected
Create associated objects if not found	Instructs the wizard to create objects referenced by the imported objects, if they cannot be found in the Excel file (or in an open model, if the *Search for referenced objects in other models* option is selected). For example, in a list of PDM tables, the **`Owner`**, property references a user object. If you do not supply a list of users on another worksheet or in an open model (or if your list is incomplete), PowerDesigner will create user objects for any references that cannot be resolved. Default: Selected
Search for referenced objects in other models	Instructs the wizard to search any models open in the workspace to resolve references as shortcuts if the referenced object cannot be found in the Excel file. Default: Unselected
Reference associated objects by	Specifies whether the object's name or code (the latter of which does not usually permit spaces or special characters) is used in columns that reference the object. In the example above, you could use the user's name or code to reference it in the **`Owner`** column of the list of tables. Default: Name
Qualified name separator	Specifies the character used to separate namespaces in a qualified name, which is used to specify the location or parent of an object. For example, to specify that area **`Area3`** is inside **`Area2`**, which is in turn inside **`Area1`**, you would enter **`Area1.Area2`** in the **`Parent`** column for **`Area3`**. Default: Dot
List value separator	Specifies the character used to separate items in a list in an Excel cell. For example, you could list all the columns of a table in a cell as follows: **`ID,Name,Email,Password`**. Default: Comma
String literal for 'True' value	Specifies the string used to signify 'True' for boolean attributes. False is signified by an empty cell. Default: X ('Y', 'Yes', 'True', and '1' also always signify 'True'.

1.4.5 Importing Objects from XML Files

PowerDesigner allows you to import objects from an XML file by defining mappings between the elements in the XML schema and objects in the PowerDesigner metamodel (and any extensions). The import mappings are defined in an extension file, can be shared with all users through the repository library, and are accessed by selecting *File* > *Import* > `<Import Name>`.

Context

Once an import is defined for a particular XML schema, any XML data files conforming to the schema can be imported. If you do not have access to the schema, PowerDesigner can infer a schema from a data file.

> **i Note**
>
> The following procedure gives an overview of the process for defining an XML import. If an administrator has already defined an import and provided the resource file to you, you can perform an import directly by selecting *File* > *Import* > `<Import Name>`.

Procedure

1. Select *Tools* > *Resources* > *Extensions* > `<Model Type>` to open the appropriate extension file list.

   > **i Note**
   >
   > As XML imports are mapped to a specific set of objects in the PowerDesigner metamodel, you must choose the correct type of extension file. For example, if you are importing business processes, you should create the import definition in a business process model extension file. If the objects defined in your schema do not relate to any of the standard PowerDesigner model types, you should choose a free model (see The Free Model (FEM) [page 241]) extension file. If appropriate you can, alternately, define the import in a DBMS or language definition file (see Creating and Copying Resource Files [page 410]).

2. Create a new extension file in the list to contain your XML import definition (see Creating an Extension File [page 412]).
3. Create an XML import definition in the extension file (see *Customizing and Extending PowerDesigner > Extension Files > XML Imports (Profile)*).
4. Save the extension file and test the import by selecting *File* > *Import* > `<Import Name>`.
5. [optional] Check the extension file into the repository library to have it automatically deployed to all users (see Sharing Resource Files via the Library [page 551]).

1.4.6 Moving and Copying Objects

Objects in the PowerDesigner working environment are easy to manipulate and to reuse from one model or package to another.

Drag and Drop

You can drag objects from one package or model to another to copy, or move them, or to create a shortcut or replica. You can drag objects from the Browser, a diagram, an object list, or the Result List to the Browser or a diagram. By default, you perform a move when the start or drop point is the Browser or create a shortcut when dragging between diagrams. You can change this behavior temporarily, or set a different default.

To temporarily change the behavior; press and hold the following keys while dragging:

- `Shift` - Move
- `Ctrl` - Copy
- `Shift` + `Ctrl` - Create shortcut (see Creating Shortcuts [page 446])
- `Shift` + `Alt` - Create replica (see Creating Replicas [page 451])

To obtain any behavior without modifier keys, right-click and drag and release to open a contextual menu listing the available drop actions.

To modify the default Browser drag and drop behavior, select *Tools* > *General Options* and select the appropriate *Default action* in the *Browser drag & drop* group box (see General Options [page 388]).

Copy and Paste

You can copy objects from the Browser, a diagram, an object list, or the Result List and paste them into the Browser, a diagram, or an object list. Symbols can be pasted as images into MS Word or image applications, while lines from object lists or the Result List can be pasted in CSV format into Excel, Word, or other text editors.

> **i Note**
>
> To create a graphical synonym of a symbol, which is merely a second instance of the object in the diagram to improve readability or reduce link lengths, see Creating Graphical Synonyms for Object Symbols [page 220]

To copy one or more objects, select them, and then:

- Select *Edit* > *Copy* from the menu bar.
- Press `Ctrl` + `C`.
- Right-click and select *Edit* > *Copy*.

Click in the place where you want to paste the object and then:

- Select *Edit* > *Paste* from the menu bar.

- Press Ctrl + V.
- Right-click and select *Edit* > *Paste*.
- [to paste as a shortcut] Select *Edit* > *Paste As Shortcut* from the PowerDesigner menu bar, or right-click and select *Edit* > *Paste As Shortcut*. For more information, including the rules governing the creation of shortcuts, see Shortcuts and Replicas [page 443].

When you copy an object, you also copy its child objects, and any references to other objects. For example, if you copy a CDM entity, you also copy its attributes and any references to business rules attached to that entity. When you paste an object, you transfer all of its properties from the Clipboard and create a new object, and not a graphical synonym or a new instance of the copied object.

When you paste in PowerDesigner, the creation of the object must not violate name uniqueness rules (see Object Namespaces [page 116]). If an object with the same name and code already exists in the model or package, PowerDesigner will append a number to the name and code, and display a message in the Output pane to warn you that the object was renamed.

> **Note**
>
> When a paste conflict occurs on a CDM entity, the copied entity is given a new name and code and, if the *Allow Reuse* model option is not selected, new attributes will be created, with the same names as those being copied. If the *Unique Code* model option is not selected, the codes because they are in the context of the new entity) a new identifier is created (which will have the same name according to the data item options set in the model.

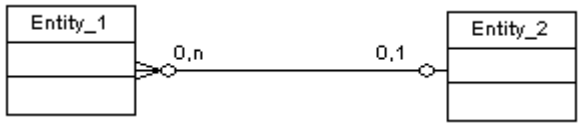
1.4.6.1 Moving Objects from One Package to Another

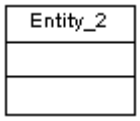
You can move objects from one package to another by drag and drop or cut and paste. Moving an object does not involve copying it, but a shortcut will be created in the original model or package, which points to the object's new location. You can only move an object to a model that supports this kind of object, and global objects (such as business rules, domains, storages etc.) cannot be moved into packages.

Context

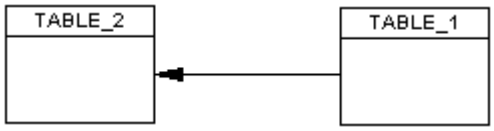
When you move an object from a package to another, linking objects that you move keep their links in the target package and a shortcut is usually created in the source package. The general rule being that conceptual modeling must be preserved.

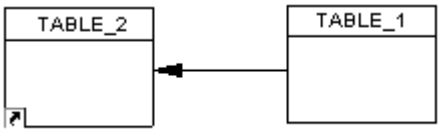
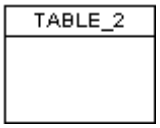
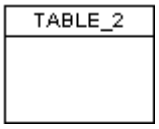
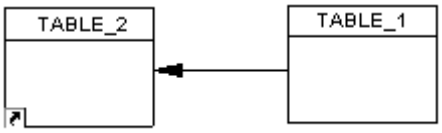
When you move an object with a non-oriented link:



Source package	Target Package
A shortcut of the moved entity is created: 	The moved entity is displayed: 

When you move an object with an oriented link:



Move Table	Source Package	Target Package
Parent TABLE_2	A shortcut of the parent TABLE_2 is created: 	The parent TABLE_2 is displayed: 
Child TABLE_1	Only the parent TABLE_2 is displayed: 	The child TABLE_1 is displayed with the reference link and a shortcut is created for the parent TABLE_2. 

When moving CDM entities containing data items, the data items will be moved to the new package or model with the entity unless they are reused by other entities, in which case copies will be made.

For more information about shortcut and copy rules, see Creating Shortcuts [page 446].

1.4.7 Deleting Objects

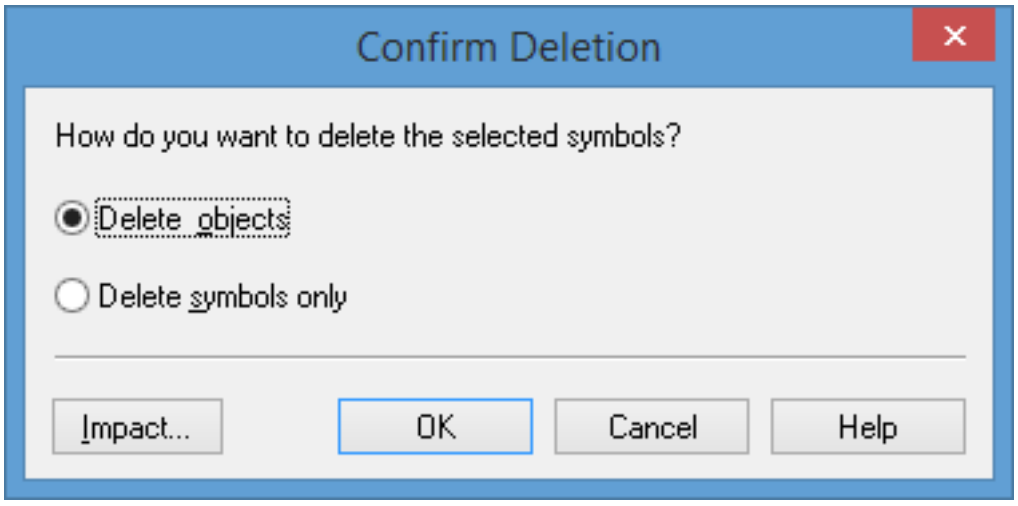
You can delete an object from a diagram, the Browser, or an object list.

Since PowerDesigner gives you the freedom to create multiple symbols in multiple diagrams to represent the same object, when deleting a symbol in a diagram, you can choose to delete just the symbol or the entire object.

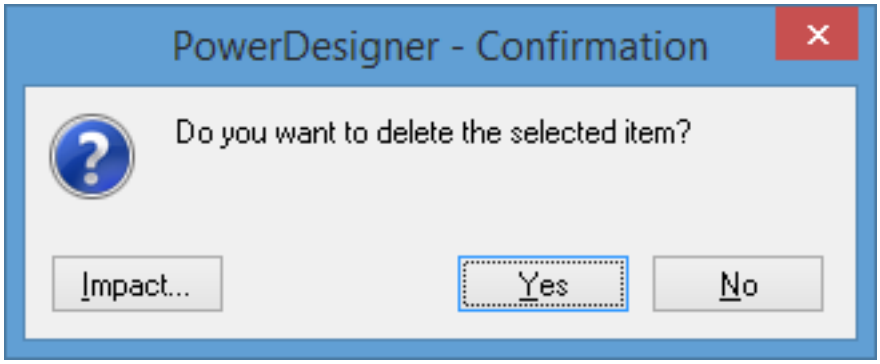
When you delete an object, you also delete any sub-objects it contains (for example, when you delete a table from a PDM, you delete its columns, keys, triggers and indexes), along with all its diagram symbols. If you delete an object that is connected to another object via a link, the link is also deleted.

To delete an object, do one of the following:

- Select its symbol in a diagram and press the `Del` key or right click it and select *Edit* > *Delete*. A dialog will open asking you whether you want to delete the object itself (including any sub-objects it contains) or just the symbol:



  You can review the impact that deleting the object would have on other objects in your environment by clicking the *Impact* button on the confirmation dialogs (see Impact and Lineage Analysis [page 489]). You can undo a deletion by clicking the *Undo* tool.
  Make your choice and click *OK*. If you delete only the symbol, you can restore it to the diagram, by selecting *Symbol* > *Show Symbols* and reselecting the object in the Show Symbols dialog.
- Select its symbol in a diagram and press `Shift` + `Del`. The object, any sub-objects it contains, and any associated diagram symbols will be deleted immediately without the need for confirmation.
- Select it in the Browser and press the `Del` key or right click it and select *Edit* > *Delete*. A dialog will open asking you to confirm the deletion:



Click *OK* to delete the object, any sub-objects it contains, and any associated diagram symbols.

- Select it in an object list and click the *Delete* tool or press the Del key. The object, any sub-objects it contains, and any associated diagram symbols will be deleted immediately without the need for confirmation.

> **i** Note
>
> You can suppress the display of the Confirmation dialogs by deselecting the *Confirm Object Deletion* general option (see General Options [page 388]).

Deleting Domains and Data Items

If you have specified that domains and data items can be reused by multiple objects in a CDM or PDM and you delete a parent object to which they belong, these sub-objects will not be deleted with their parent. For more information, see *Data Architecture* .

1.4.8 Object Properties

All model objects created in PowerDesigner have property sheets, which organize object properties on tabs.

You can open an object property sheet in any of the following ways:

- In the Browser, double-click the object symbol or its entry.
- Right-click the object symbol or its Browser entry, and select *Properties*.
- Select the object from an object list or in the property sheet of its parent object, and click the *Properties* tool.
- Select an object symbol and press Alt + Enter.

Property sheets let you assign extensive and sophisticated properties to objects. Most PowerDesigner objects have the following tabs and properties, but many others, specific to the particular object type, may be available:

- *General* – provides basic information about the object. Almost all objects have these properties:
  - *Name* - clearly identifies the object. By default, names can have up to 254 characters, and can include uppercase, lowercase, and mixed-case strings. Names are controlled by naming conventions (see

Naming Conventions [page 155]) and must be unique within a namespace (see Object Namespaces [page 116]). If the enterprise glossary is enabled in the model, then autocompletion and glossary compliance checking is available (see The Glossary [page 38]).

- *Code* – is used in scripts generated from the model. Codes are controlled by naming conventions and must be unique within a namespace. By default, codes are synchronized with names, but you can decouple them individually by clicking to deselect the equal sign button to the right of this field or in the model as a whole by unselecting the *Synchronize code* model option (see Naming Conventions [page 155]).
- *Comment* – provides a more detailed description of the object.
- *Stereotype* - extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
- *Keywords* - Provide a way of loosely grouping objects through tagging. Separate multiple keywords with commas. You can use keywords as criteria for finding objects (see Finding Objects [page 133]), and in impact and lineage analysis (see Impact and Lineage Analysis [page 489]).

Property sheets of link objects display a picture of the link together with its extremities. Additional information, such as the cardinality value and the role, may also appear.

- *Definition* – Provides fuller information about the object. See Definition Tab [page 110].
- *Rules* – lists the business rules with which the object must comply. A business rule may be a government-imposed law, a customer requirement, or an internal guideline. See Business Rules [page 137].
- *Version Info* - provides details about the object owner, creation, and modification date and allows you to access help for the PowerDesigner metamodel metaclass on which the object is based. See Version Info Tab [page 115]
- *Dependencies* - lists all the objects that depend on the object. See Dependencies Tab [page 115]
- *Traceability Links* - lists all the objects on which the object depends. See Traceability Links [page 146]
- *Requirements* – lists the project requirements that the object is intended to satisfy. By default, this tab does not appear. See Requirements Tab [page 114]

> **i Note**
>
> Use `Ctrl` + `Page Down` or `Ctrl` + `Page Up` to move to the next or to the previous tab and display the corresponding object type tab.
>
> By default, property sheets open to the *General* tab. You can choose to open property sheets at the last accessed tab by selecting *Tools* > *General Options* > *Dialog*, and selecting *Keep Last Tab* in the *Property Sheets* groupbox.

You can customize, create, and hide properties and property sheet tabs using extension files (see *Customizing and Extending PowerDesigner* > *Extension Files*). Administrators can prepare customized property sheets through permission profiles (see Customizing the PowerDesigner Interface [page 554]).

You can control the form of the symbols, the background color, and the text format with display preferences (see Display Preferences [page 228]). For display preferences to be applied to the mini-diagrams in link object property sheets, click *Set As Default* in the Display Preferences dialog for each modification.

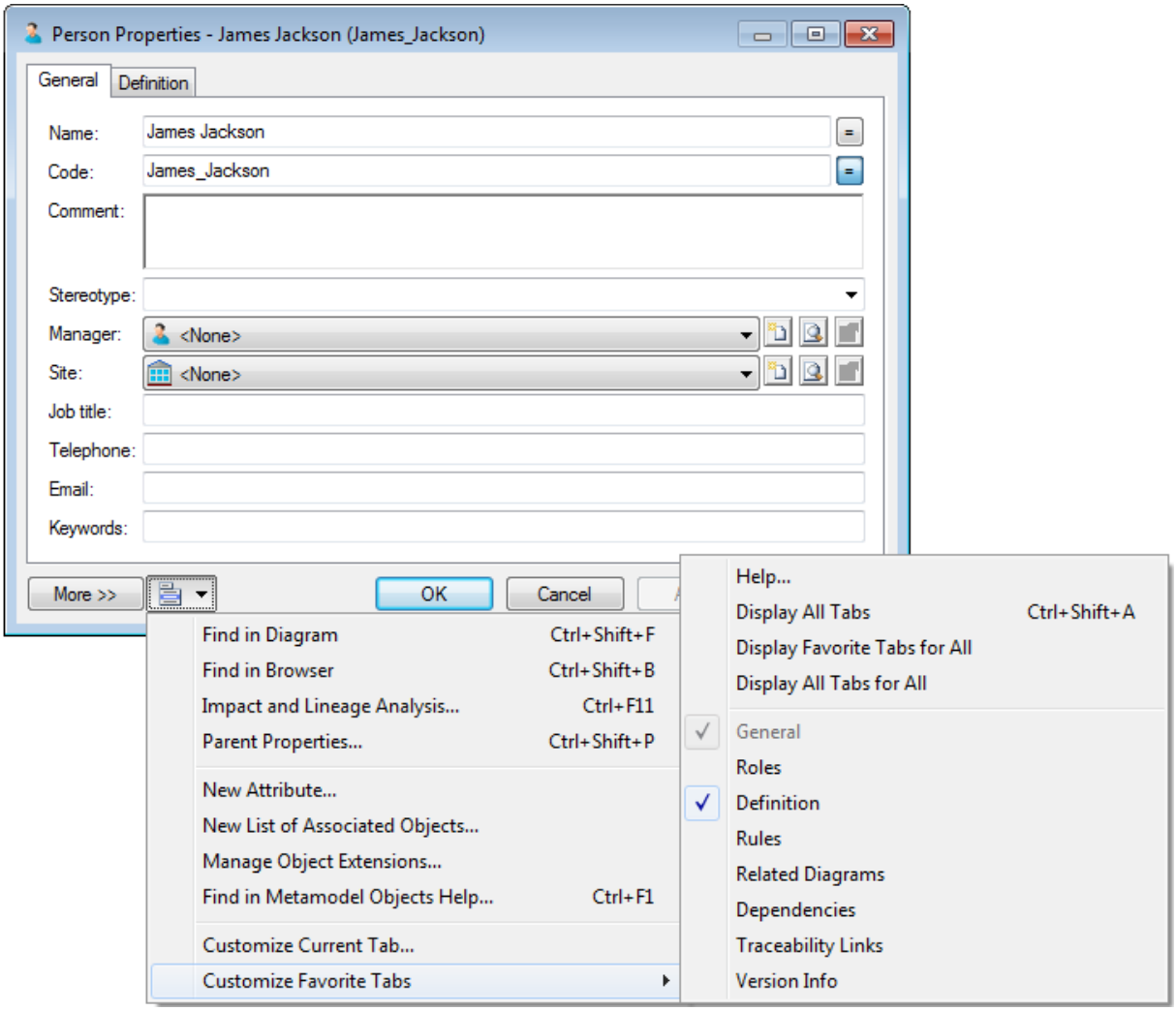
1.4.8.1 Customizing a Property Sheet

Since PowerDesigner can hold a rich variety of information about your model objects, property sheets can become overloaded. The *More* or *Less* button at the bottom left corner of the property sheet allows you to toggle between displaying all the available property sheet tabs, and a subset of "favorite" tabs.

You can customize your list of favorite tabs, and access various other property sheet features, from the property sheet menu, which is accessible from the bottom-left corner of all property sheets.

Command	Description
Find in Diagram	Finds the object in the diagram, selects it, and centers the diagram on it.
Find in Browser	Finds the object in the Browser and highlights it.
Impact and Lineage Analysis	Opens the Impact and Lineage Analysis dialog (see Impact and Lineage Analysis [page 489]).
Parent Properties	Opens the property sheet of the parent object.
New Attribute	Opens the *New Attribute* dialog, which lets you add new properties to your object property sheets (see Adding a New Attribute to an Object [page 168]).
New List of Associated Objects	Opens the *New List* dialog, which lets you add new lists of associated objects to your object property sheets (see Adding a New Collection to an Object [page 171]).
Manage Object Extensions	Opens the *Manage Object Extensions* dialog, which lets you access the metaclass on which the object is based in the Resource Editor (see Managing Extensions [page 174]).
Find in Metamodel Objects Help	Opens the MetaModel Objects Help for the metaclass on which the object is based.
Customize Current Tab	Opens a dialog to change the properties displayed in the current tab (see Customizing an Object Property Sheet Tab [page 173]).
Customize Favorite Tabs	Opens a sub-menu, which allows you to define favorite tabs: • Help – opens this help topic. • Display All Tabs - Displays all available property tabs for the current property sheet. • Display Favorite Tabs for All - Displays favorite tabs for all property sheets throughout all models. • Display All Tabs for All - Displays all tabs for all property sheets in all models. Beneath the sub-menu are listed all the tabs available for the present object. Favorite tabs have a check against them. Click a tab in the list to select or remove it from the list of favorite tabs. Note that the *General* tab cannot be unchecked. When you check or uncheck a tab such as *Definition*, *Rules*, and *Dependencies*, which are common to many objects, you are asked to confirm the change for all other objects. If you click *No*, only the current property sheet is modified.

In the following example, only the *General* and *Definition* tabs are selected and are displayed:



Any changes in the display of tabs are immediately applied to the current property sheet but not to other property sheets that are currently open. The changes become the default setting for any property sheet of the same type.

> **i** Note
>
> You can control whether tabs display on one or several rows in property sheets by selecting *Tools* > *General Options* > *Dialog* and using the options available in the **`Property sheets`** groupbox (see Dialog Box Options [page 390]).

1.4.8.2 Definition Tab

The *Definition* tab in an object property sheet can be used to provide detailed documentation about the object.

- *Description* - Provides detailed information about the object. For example, a description of the Employee entity might read: *This entity has one occurrence for each employee in our worldwide operations. This base should grow by 20 percent by 2020.*

- *Annotation* - Provides information about the implementation of a model or the objects it contains. For example, an annotation of the Employee entity might read: *Verify list of attributes with Director of Human Resources.*

Both fields can be edited directly in the tab with the internal PowerDesigner RTF editor and you can insert existing text and RTF files into them (see Text Editor Tools [page 226]).

> **i Note**
>
> To verify that all objects have a description, open the appropriate object list (Object Lists [page 117]) and review the *Description* column. For information about adding columns to a list, see Customizing Object List Columns and Filtering Lists [page 123].

1.4.8.3 Preview Tab

Click the *Preview* tab in the property sheet of the model, packages, and various other model objects in order to view the code that will be generated for it.

The following tools are available on the *Preview* tab toolbar:

Tools	Description
	Editor Menu (Shift + F11) - Contains the following commands: • *New* (Ctrl + N) - Reinitializes the field by removing all the existing content. • *Open...* (Ctrl + O) - Replaces the content of the field with the content of the selected file. • *Insert...* (Ctrl + I) - Inserts the content of the selected file at the cursor. • *Save* (Ctrl + S) - Saves the content of the field to the specified file. • *Save As...* - Saves the content of the field to a new file. • *Select All* (Ctrl + A) - Selects all the content of the field. • *Find...* (Ctrl + F) - Opens a dialog to search for text in the field. • *Find Next...* (F3) - Finds the next occurence of the searched for text. • *Find Previous...* (Shift + F3) - Finds the previous occurence of the searched for text. • *Replace...* (Ctrl + H) - Opens a dialog to replace text in the field. • *Go To Line...* (Ctrl + G) - Opens a dialog to go to the specified line. • *Toggle Bookmark* (Ctrl + F2) - Inserts or removes a bookmark (a blue box) at the cursor position. Note that bookmarks are not printable and are lost if you refresh the tab, or use the *Show Generation Options* tool • *Next Bookmark* (F2) - Jumps to the next bookmark. • *Previous Bookmark* (Shift + F2) - Jumps to the previous bookmark.
	Edit With (Ctrl + E) - [OOM only] Opens the previewed code in an external editor. Click the down arrow to select a particular editor or *Choose Program* to specify a new editor. Editors specified here are added to the list of editors available at *Tools* > *General Options* > *Editors*.
	Save (Ctrl + S) - Saves the content of the field to the specified file.

Tools	Description
	Print (Ctrl + P) - Prints the content of the field.
	Find (Ctrl + F) - Opens a dialog to search for text.
	Cut (Ctrl + X), *Copy* (Ctrl + C), and *Paste* (Ctrl + V) - Perform the standard clipboard actions.
	Undo (Ctrl + Z) and *Redo* (Ctrl + Y) - Move backward or forward through edits.
	Refresh (F5) - Refreshes the Preview tab. You can debug the GTL templates that generate the code shown in the Preview tab. To do so, open the target or extension resource file, select the *Enable Trace Mode* option, and click *OK* to return to your model. You may need to click the *Refresh* tool to display the templates.
	Select Generation Targets (Ctrl + F6) - Lets you select additional generation targets (defined in extensions), and adds a sub-tab for each selected target. For information about generation targets, see *Customizing and Extending PowerDesigner > Extension Files > Generated Files (Profile) > Generating Your Files in a Standard or Extended Generation* .
	Show Generation Options (Ctrl + W) - Opens the Generation Options dialog, allowing you to modify the generation options and to see the impact on the code.
	Ignore Generation Options (Ctrl + D) - [PDM only] Ignores changes to the generation options made with the *Show Generation Options* tool.

1.4.8.3.1 Finding Text Using Regular Expressions

You can use regular expressions to find script text displayed in the Preview or Script tab of a table property sheet, as well as in the Edit/Run Script editor, the Resource Editor, and various other windows.

In order to be able to search for text using regular expressions, you must select the Regular Expression check box in the dialog box and enter a regular expression in the Find What box.

Regular expressions can contain ordinary characters and the following *metacharacters*:

Character	Description
\	Matches a special character Examples: "n" matches "n". "\n" matches a newline character. "\\" matches "\" and "\(" matches "("

Character	Description
^	Matches the position at the beginning of the input string Examples: "^Win" matches strings beginning with "Win
$	Matches the position at the end of the input string Examples: "then$" matches strings ending with "then"
*	Matches the preceding character zero or more times Examples: "zo*" matches "z" and "zoo"
+	Matches the preceding character one or more times Examples: "zo+" matches "zo" and "zoo", but not "z"
?	Matches the preceding character zero or one time Examples: "to?" matches either "t" or "to"
.	Matches any single character except the newline \n Examples: ".ork" matches "Work", "Fork" etc
[ ]	Matches any one of the enclosed character Examples: "[abc]" matches "a", "b", or "c" A range of character can be indicated with a dash "[a-z]"

For a complete list of metacharacters and their behavior in regular expressions, see the Visual Basic Documentation.

1.4.8.4 Requirements Tab

You can attach one or more requirements to an object using the Requirements tab in the object property sheet. This tab is not displayed by default, and can only be used if one or more requirements models is open in the workspace.

Context

> **i Note**
>
> To display the Requirements tab, select *Tools* > *Model Options* and select the *Enable links to requirements* option.

Procedure

1. Open the object's property sheet and click the *Requirements* tab.
2. Click the *Add Objects* tool to open a selection dialog.
3. Select a Requirements model and, optionally, a package to display a list of the requirements contained therein.
4. Select the requirements that you want to attach to the object and click *OK*.

   The selected requirements appear in the Requirements tab.
5. [optional] Select the requirement in the list and click the *Properties* tool to open its shortcut property sheet. To open the property sheet of the requirement itself, click the *Target Object Properties* button to the right of the *Name* field.

   > **i Note**
   >
   > To access the property sheet of the requirement directly instead of passing by the shortcut, select *Tools* > *General Options* > *Dialog*, and select the *Target Object* radio button beside the *External Shortcut* option.
6. Click *OK* to close the property sheet and return to the model.

1.4.8.5 Version Info Tab

PowerDesigner automatically manages version information about model objects on the read-only *Version Info* tab of the object's property sheet.

Property	Description
Creation User	Specifies the name of the User who created the object.
Creation Date	Specifies the date of creation of the object
Last Modification User	Specifies the Name of the user who made the last modification to the object.
Last Modification Date	Specifies the date of the last modification of the object
Replicated From	[replicated objects only]Specifies the name of the origin object from which the object was replicated. Click the button to the right of this field to open the origin object property sheet
Generated From	[generated objects only]Specifies the name of the origin object from which the object was generated. Click the button to the right of this field to open the origin object property sheet

> **i** Note
>
> In addition to the standard help accessible via *F1* or by clicking the *Help* button, you can obtain information about the PowerDesigner metamodel metaclass on which this object is based by clicking the small question mark button at the bottom right of the tab.

1.4.8.6 Dependencies Tab

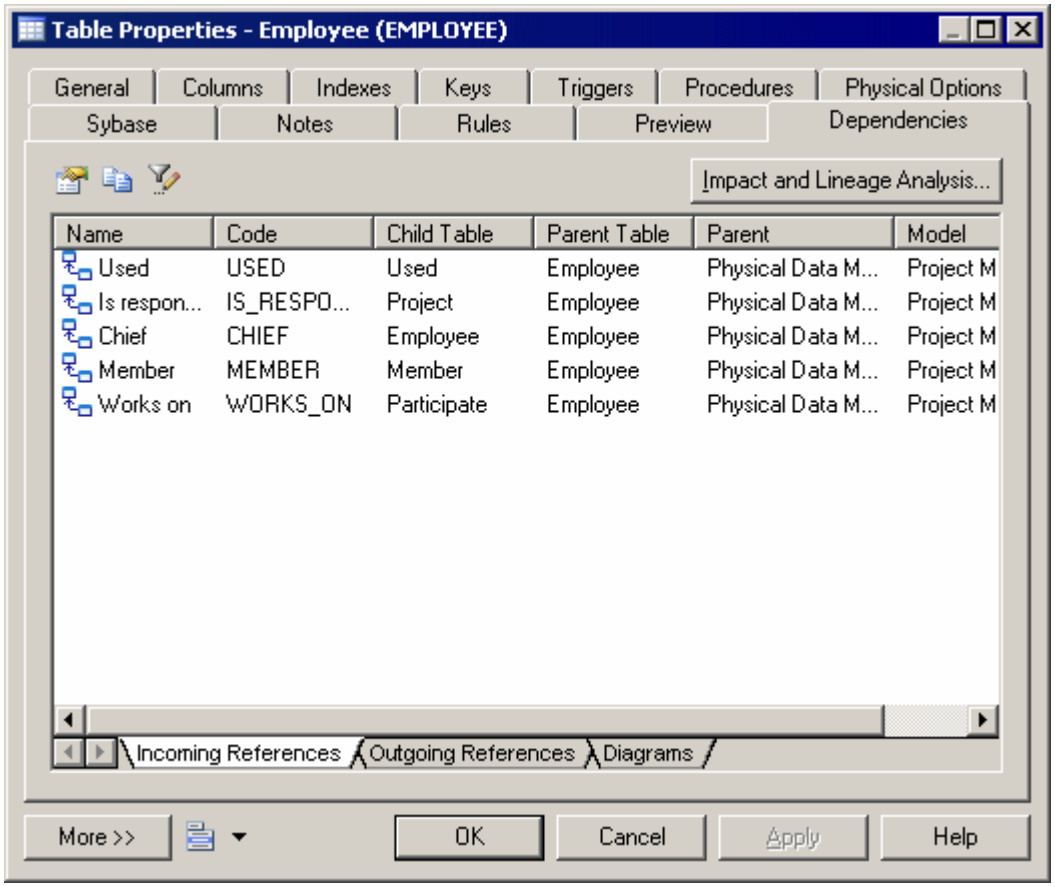
The PowerDesigner metamodel provides the capability to link objects to other objects in a variety of semantic ways. For example, when you create an external shortcut, or when you attach a business rule to an object, a dependency link is created between models or objects. The *Dependencies* tab of the object's property sheet displays these links.

Context

Dependencies can be:

- Internal - to other objects within the model. These dependencies are saved in the model and are always available on the *Dependencies* tab of an object property sheet
- External - to objects in other models via shortcuts or generation links (see Creating Links Between Models [page 422]). These dependencies are available on this tab only if the linked model is open in your

workspace, or if you have checked both models into the repository and have checked your model out with the *Check out dependencies* option (see Checkout Parameters [page 270]).



1.4.8.7 Object Namespaces

In PowerDesigner, each package can be a namespace. It is however possible to expand the namespace to the parent of a given package. You can cascade the expansion until you reach the level of the model itself.

Objects that appear in the Browser obey the general rules that follow:

In Browser	Namespace	Uniqueness rule
Objects directly under a package or a model (table, process, class)	Model	Unique name and code in model
	Package	Unique name and code in package
Objects under parent object (column, attribute)	Not applicable	Unique name and code in parent

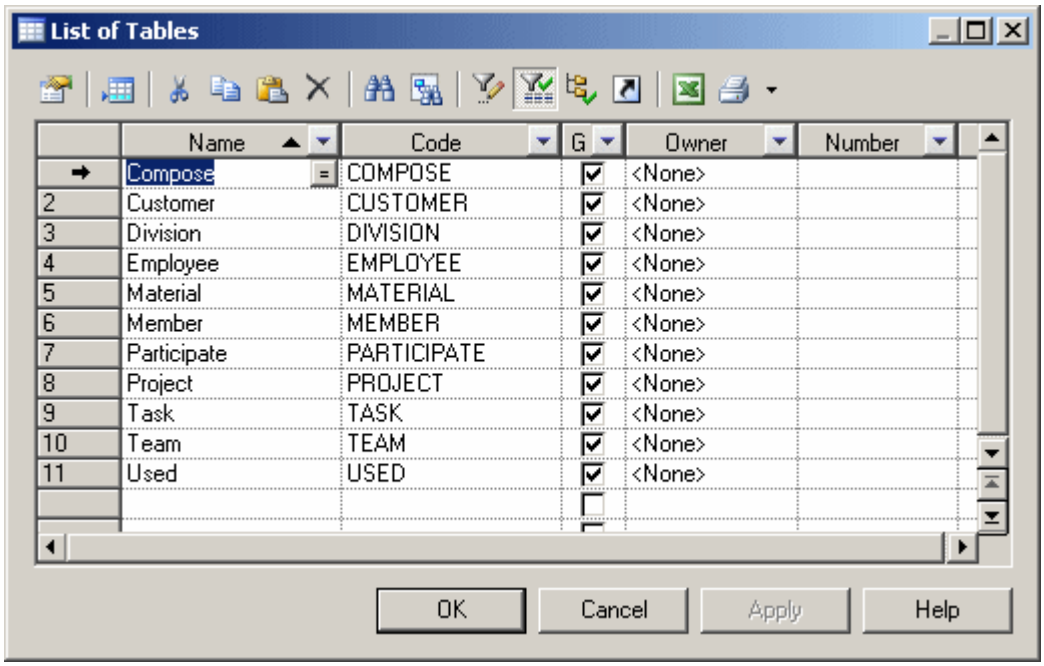
In Browser	Namespace	Uniqueness rule
Linking objects (reference, relationship)	Not applicable	Unique name and code between same end objects (i.e., parallel links with same name and code are not allowed between same end objects)

However, you may encounter some exceptions to those rules. For example only the code of a reference allows to identify the object in the entire model, when the "Unique code" option is selected in the Model Options dialog box. PowerDesigner warns you when a general rule is not respected.

1.4.9 Object Lists

PowerDesigner object lists provide a spreadsheet-like interface for manipulating large quantities of objects. They offer you an overview of the objects in your model and can save you time and effort in managing your metadata. While property sheets provide depth of detail for individual objects, lists allow you to select and manipulate multiple objects simultaneously to streamline creation and improve consistency.

Lists of all the major objects in your model are available under the *Model* menu or by right-clicking your model in the Browser and selecting *List of* `<objects>`. Each list shows all the objects of that type in the currently selected package or model, including those that do not have symbols in the current diagram:



The properties of the listed objects are organized in columns. You can order the list by a particular columns values by clicking on its column header. To filter the values in a column, click the down-arrow on the right of the column header and enter your filter (see Filtering a List Using In-Column Filters [page 120]).

You can control which property columns are displayed, and also filter the list based on the values in any of the columns (see Customizing Object List Columns and Filtering Lists [page 123]).

You can select multiple items in a list by `Ctrl`-clicking them in the far-left, numbered column. To select all the items in a list, click the top-left corner box. When multiple items are selected, any edits you make to properties are applied to all the selected items.

> **i Note**
>
> By default, you must click the *Apply* button to commit changes or the *OK* button to commit and close the list. To have changes commited immediately when you enter them in a field, enable the *Auto commit* general option (see Dialog Box Options [page 390]). You can always use the *Undo* tool to cancel the change.

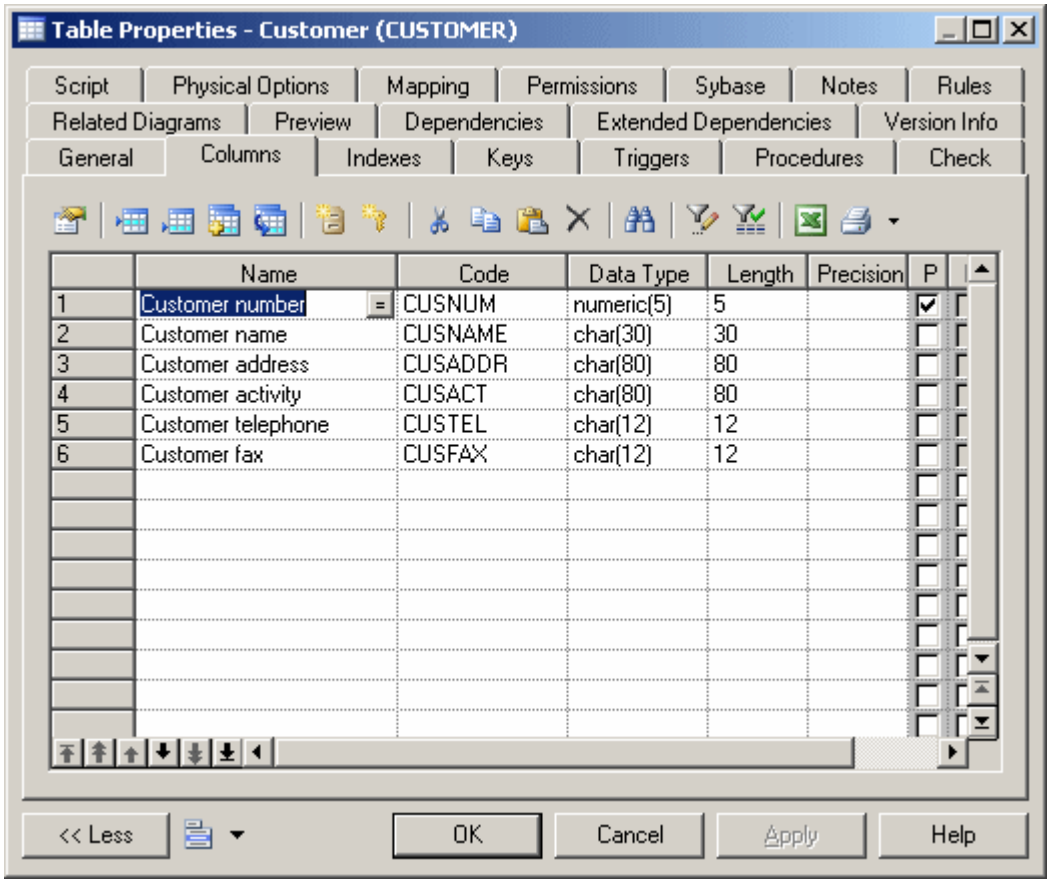
All or some of the following tools are available on object lists:

Tool	Description
	Properties - Opens the property sheet of the selected item (see Object Properties [page 107]).
	Insert a Row - [ordered lists only] Inserts a row before the selected row in the list.
	Add a Row - Adds a row at the end of the list.
	Add Objects - Opens an object selection dialog box to select objects and copy them to the list (see Adding an Item from a Selection List [page 129]).
	Create Object - Creates a new object and opens its property sheet to allow you to complete its definition.
	Reuse/Replicate Objects - Opens an object selection dialog box to reuse objects. When you select an item you create a link to the original instead of copying it to the list.
	Cut - Cuts the row and stores it in the Clipboard.
	Copy - Copies the selected row to the Clipboard.
	Paste - Pastes the contents of the Clipboard.
	Delete - Deletes the row.
	Find a Row - Opens the Find dialog to search for an item in the list.
	Find Symbol in Diagram - Finds the object's symbol in the diagram.
	Customize Columns and Filter - Opens a dialog to change the columns displayed in the list or define a filter (see Customizing Object List Columns and Filtering Lists [page 123]).
	Enable/Disable Filter - Toggles the filter specified in the Customize Columns and Filter dialog.
	Include Sub-Packages - Includes objects in sub-packages in the list.

Tool	Description
	Include Composite Objects - Includes composite objects (such as sub-processes, sub-activities, sub states, etc.) in the list.
	Include Shortcuts - Includes object shortcuts (see Creating Shortcuts [page 446]) in the list. Shortcuts are grayed as they cannot be modified. When you include shortcuts whose target model is closed, some information may be unavailable.
	Open Diagrams - [*Related Diagrams* tab] Opens a target model or a diagram (see Specifying Diagrams as Related Diagrams [page 189]).
	Shortcut - Opens an object selection list to change the target object of a shortcut
	Export to Excel - Saves the list to a *.xls, *.xlsx, or *.csv format (specify the format in the *Save as type* field in the Save as dialog).
	Print - Prints the list. Click the arrow to the right of the button to view a print preview or to access the Page Setup dialog.

Sub-Object Lists

Many objects have property sheets that contain sub-object lists, which list all of the child objects of that type belonging to the object. For example, the *Columns* tab in a table property sheet displays the list of columns in the selected table:



These lists provide the same features as the lists of objects available from the *Model* menu. Depending on the properties of the listed objects, the list may have a natural order that you can control. For example, a list of columns has such an order, and you can move objects in the list by selecting them and then clicking on one of the arrows at the bottom left corner of the list.

1.4.9.1 Filtering a List Using In-Column Filters

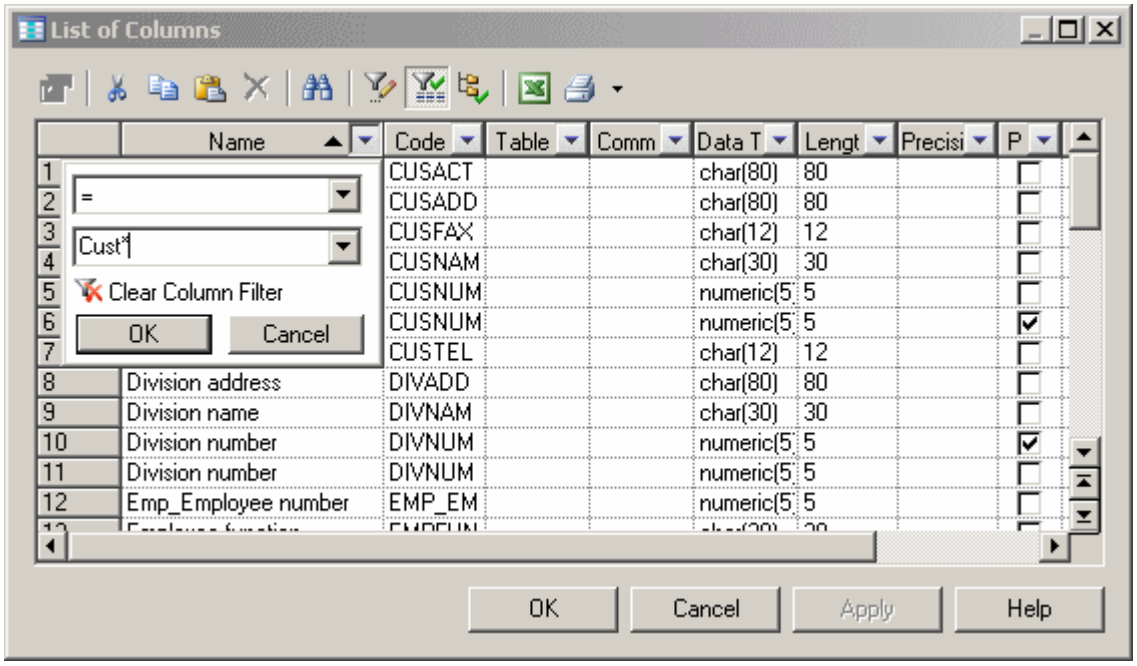
To filter a list by the values in a column, click the down arrow on the right of the column header to display the in-column filter dialog.

Context

> **i** Note
>
> If the in-column filter arrows are not displayed, click the *Customize Columns and Filter* tool and select the *Show column filter buttons* option.

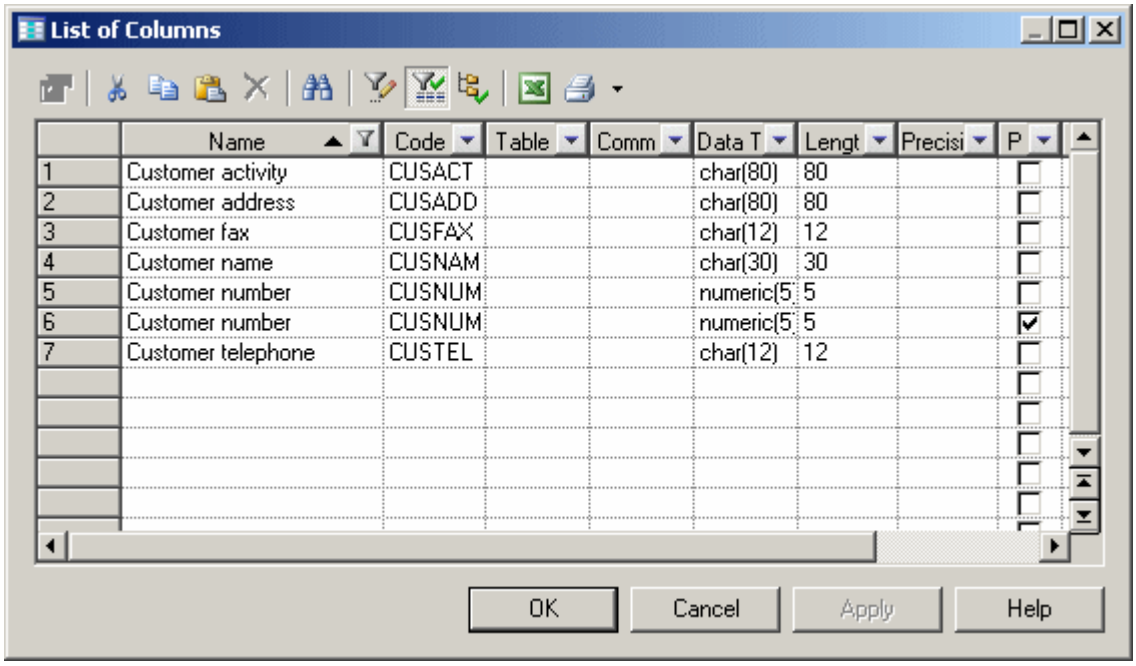
Procedure

1. Select an operator from the first list and, if appropriate, select or enter an expression value to test against it. In this example, the list will be filtered to show only rows where the *Name* column contains values beginning with **`Cust*`**.

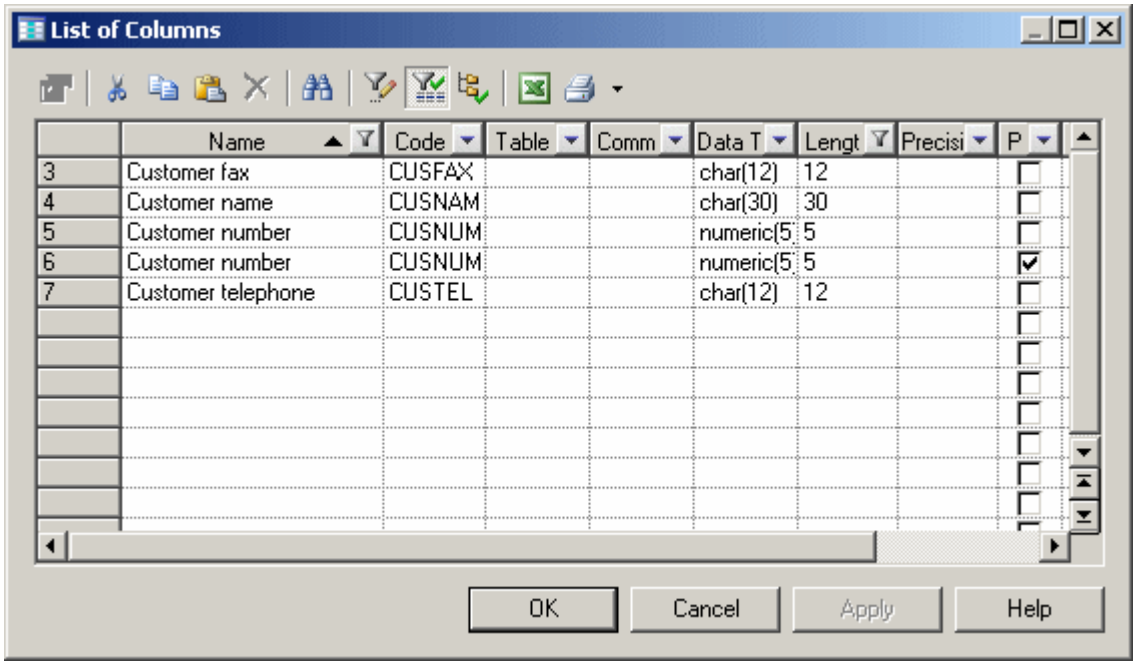


   For an explanation of the available operators and the syntax for the expressions, see Defining a Filter Expression [page 127].

2. When you are satisfied with your criteria, click *OK* to apply the filter. The down arrow is overlaid with a filter symbol to show that a filter is applied, and any rows not meeting the criteria are hidden. In addition, to *Enable/Disable Filter* button is pressed. In this example, the **`Name=Cust*`** filter is applied:



3. [optional] Further restrict the rows displayed by defining a filter on any number of other columns. In this example, both the **`Name=Cust*`** filter and a **`Length<80`** filter are applied:



> ### i Note
>
> The filter criteria are stored in the Customize Columns and Filter dialog and remain applied until you remove them. To remove all filters, click to release the *Enable/Disable Filter*. To remove a filter from a

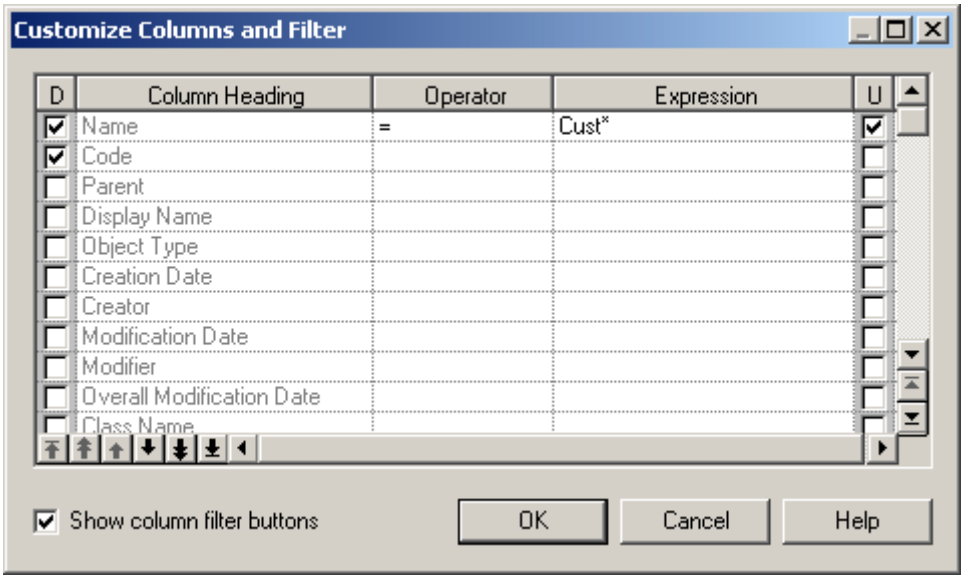
single column, click the filter button in the column header, and then click the *Clear Column Filter* button.

1.4.9.2 Customizing Object List Columns and Filtering Lists

You can choose which property columns to display in object lists, reorder them, and filter the rows to be displayed from the Customize Columns and Filter dialog.

Procedure

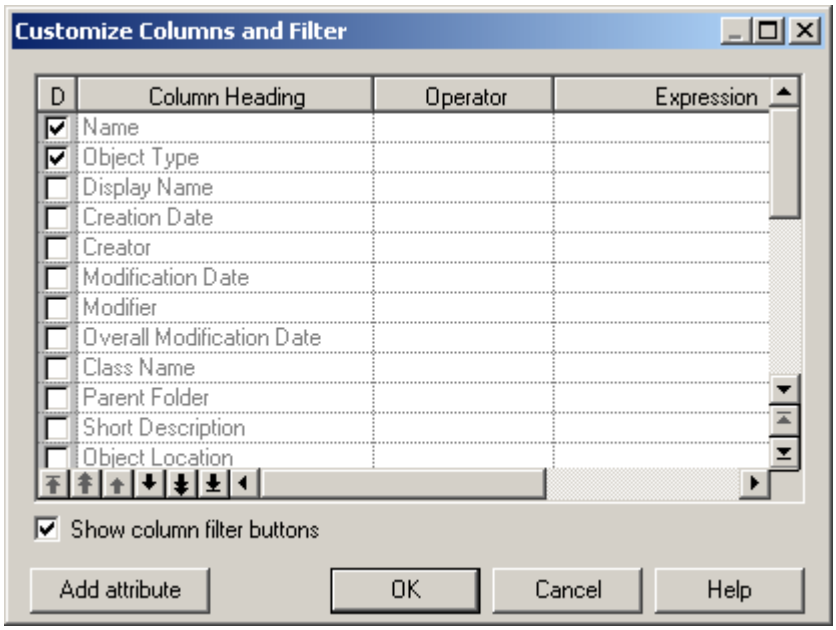
1. Click the *Customize Columns and Filter* tool on an object list toolbar to open the Customize Columns and Filter dialog.
2. Perform any of the following functions to filter the list:
   - Select columns to display - by checking the [*D*]isplay column checkbox.
   - Reorder the columns in the list - by using the arrows at the bottom left of the dialog. To move all the selected rows to the top of the list, press `Shift` + `Enter`.
   - Define one or more filter expressions - by selecting an operator in the *Operator* column and entering an expression to filter by in the *Expression* column.



   The [*U*]sed checkbox is automatically selected when you enter a filter expression. To disable the expression but keep it in memory, clear the checkbox.
   - Enable or disable in-column filtering directly in the list (see Filtering a List Using In-Column Filters [page 120]) by selecting the *Show column filter buttons* option.
3. Click *OK* to return to the list. The filter is applied by default, and the *Enable/Disable filter* tool is depressed. Click this tool to toggle between enabling and disabling the filter.

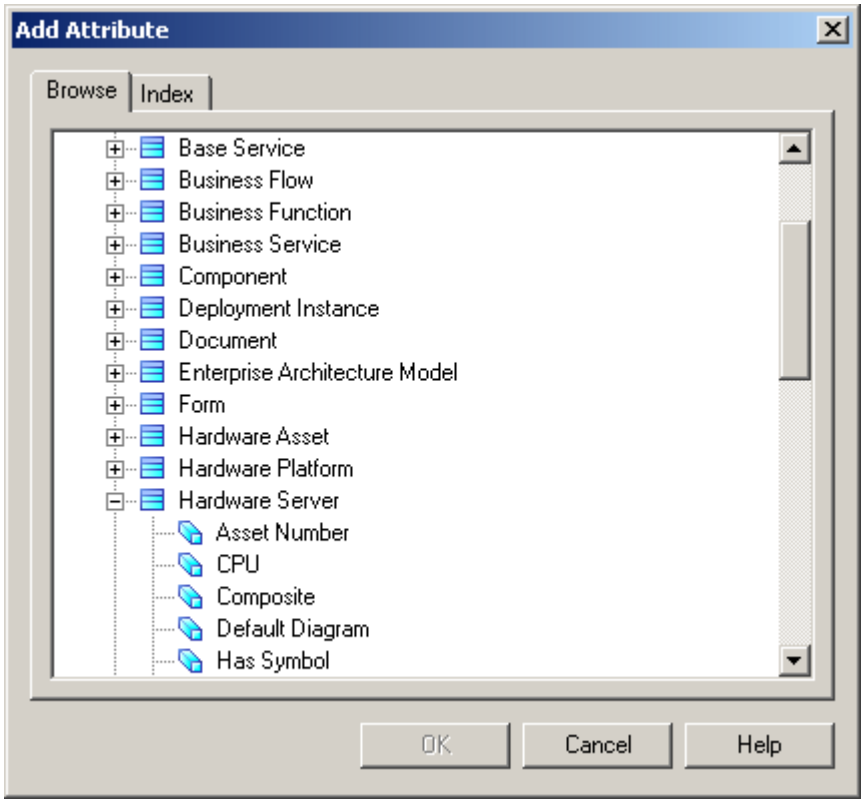
1.4.9.2.1 Customizing Columns in Lists Containing Multiple Types of Objects

PowerDesigner supports lists containing multiple types of objects. As each type of object has different sets of properties, by default only properties common to all the object types are available for selection to display in the list. However, you can customize display properties specific to one or more object types by using the *Add Attribute* button at the bottom of the Customize Columns and Filter dialog.



> **Note**
>
> The *Add Attribute* button is only available when the list can contain multiple types of objects.

The button opens a dialog that lets you select any attribute for any object that is available in this context:

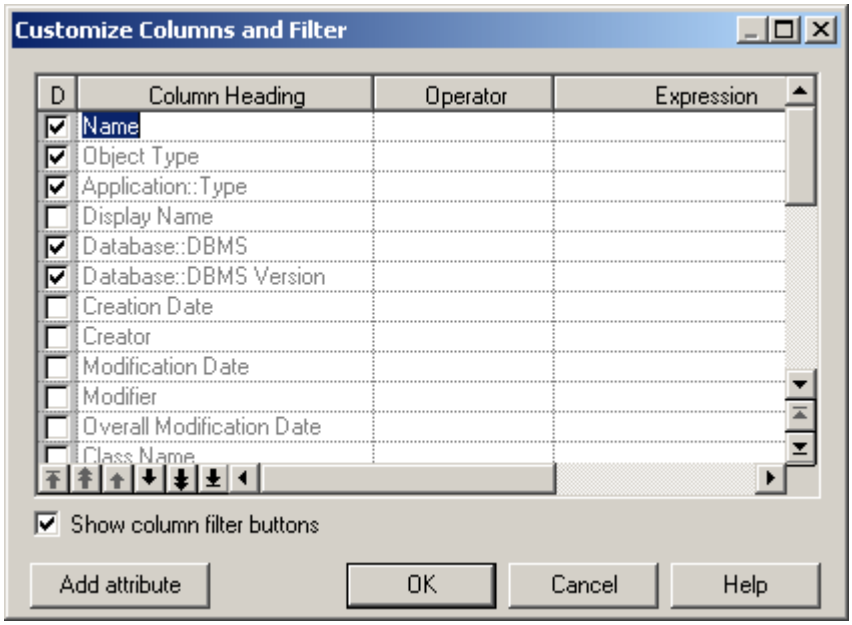


When you select an attribute and click *OK*, it is added to the list in the Customize Columns and Filter dialog and all instances of the attribute (if it is available for multiple objects) are removed from the Add Attribute dialog. You can add as many attributes as necessary and apply filters on them as appropriate.

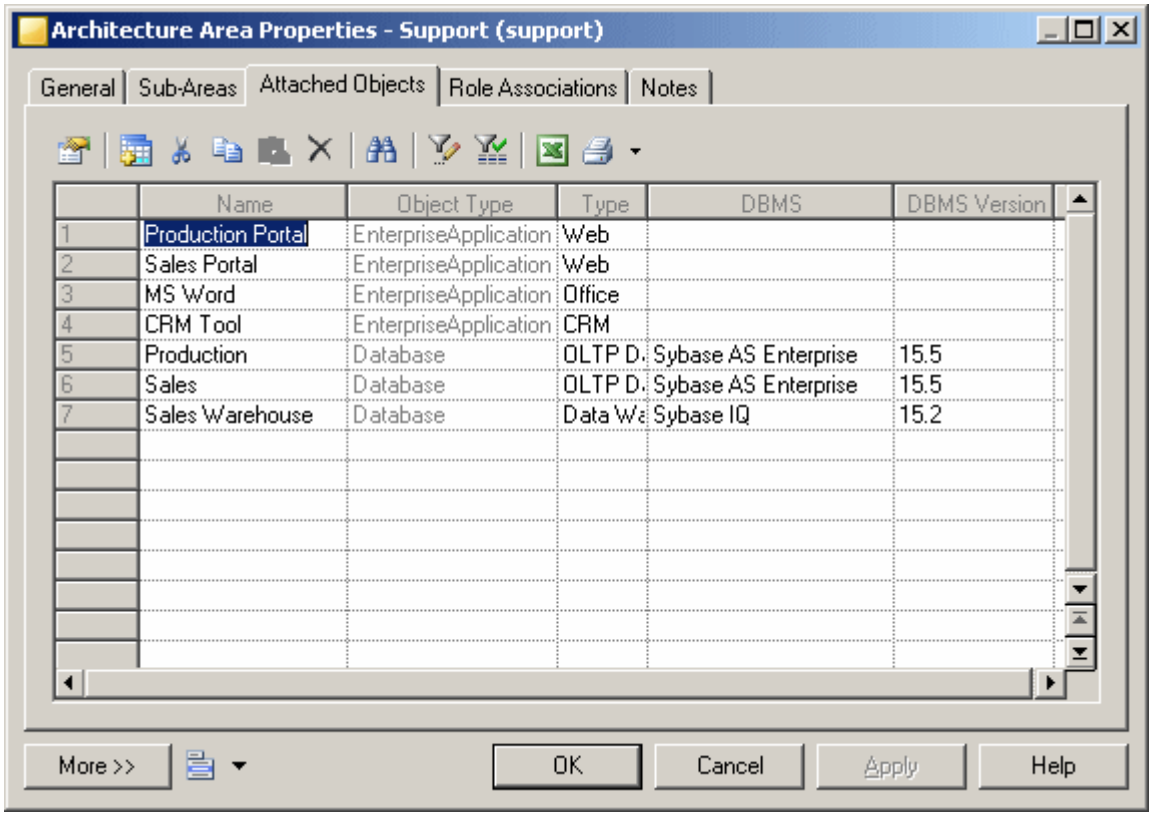
> **i** Note
>
> If you deselect an attribute in the Customize Columns and Filter dialog and click *OK*, it will be removed from the list of properties available for display. If you change your mind and want to display it once again, you will need to re-add it with the *Add Attributes* button.

For example, an area can contain many different types of objects, and these are listed on the *Attached Objects* tab of its property sheet. Here, I've added the following attributes:

- **`Application::Type`** - though I selected the Type attribute for the Application metaclass, the Type attribute present in any other metaclasses available in this context will be displayed too.
- **`Database::DBMS`**
- **`Database::DBMS Version`**



When I return to the list tab, my new attributes are displayed as columns with values where appropriate. Both my applications and databases have *Type* properties, but only my databases display values for the *DBMS* and *DBMS Version* columns:



1.4.9.3 Defining a Filter Expression

You can define expressions to filter PowerDesigner lists directly in the list with the in-column filters or through the *Customize Columns and Filter* dialog.

The following operators are available

Operator	Finds items that are...
=	[default] Equal to the expression.
>	Greater than the expression.
>=	Greater than or equal to the expression.
<	Less than the expression.
<=	Less than or equal to the expression.
Not Equal	Different from the expression.
In List	Within the comma-separated list of values given in the expression (or the items selected in the list in the in-column filter). For example: `"global", "Architecture", "proc*"`
Not In List	Outside the comma-separated list of values given in the expression (or the items selected in the list in the in-column filter).
Between	Located in the range between defined by the two values given in the expression and separated by a comma. For example, to find values between A and E, enter:`A,E`
Not Between	Located outside the range defined by the two values given in the expression and separated by a comma. For example, to find values outside of the range between A and E, enter:`A,E`
Empty	Null. No expression is necessary with this operator.
Not Empty	Not null. No expression is necessary with this operator.

You can use the following wildcards when entering a filter expression:

Wildcard	Description
*	Any string (from none to any number of characters). For example `P*` finds "protocol" and "Paris".
?	Any character. For example `?????` finds "Table" and "inner" but not "Seller".
\	Escapes the special characters `*`, `?`, and `\`. For example, `\?\\` finds "?\".

Examples

The following examples show some possible combinations of operators and expressions:

Operator	Expression	Find
=	`W*`	Work, Washington
=	`*CODE`	AREA CODE, COUNTRY CODE, CITY CODE
>	`1??`	200, 405, 609
Between	`0,8`	0,1,2,3,4,5,6,7,8
In List	`*_emp_???, *_grp_???`	div_emp_fun, _emp_idn, div_grp_fun, _grp_idn
=	`*\?`	Is this book ready for production?

1.4.9.4 Adding Items to and Reordering Items in a List

If a list is ordered you can choose to add an item at the end or at a certain point in the list, and you can reorder the items in the list. When you add a new item to a list, it is created with a default name and code that you can edit. Objects are not actually created until you click Apply or OK.

> **i** Note
>
> In general, you are not required to provide any specific properties when you create an object. However, when you create a link object, such as a reference, association link, or inheritance link, you must specify the source and the destination.

Adding Items at the End or at a Particular Point in a List

The following tools are available for adding items to a list:

Tool	Description
	Add a Row - Adds an item to the end of the list. You can also add an item to the end of a list by clicking in any empty row.
	Insert a Row [ordered lists only] - Adds an item in the row above the selected row. You can always reorder items in an ordered list by selecting and dragging and dropping them or by using the arrow tools at the bottom left of the list.

Arranging Items in an Ordered List

In ordered lists, you can drag and drop objects to reorder them, or use the following buttons at the bottom of the list:

Tool	Description
	Move the selected items to the top of the list.
	Move the selected items up one page
	Move the selected items up one line
	Move the selected items down one line
	Move the selected items down one page
	Move the selected items to the bottom of the list

1.4.9.5 Adding an Item from a Selection List

Selection lists allow you to select items from a list in order to associate them with another object.

Most selection lists display objects contained in the current model or in individual packages contained in that model. Other selection lists, such as the following, allow you to display both objects contained in the current model and other models:

- Select Diagrams. From this dialog box you can select the diagram to which you want to apply pre-defined display preferences.
- Add Shortcuts. From this dialog box you can select objects to include them as shortcut in your model or package.
- Add Objects. From this dialog box you can select objects to which you want to attach traceability links.

The following tools are available in selection lists:

Tool	Description
-	Model list - Lets you specify a model as the basis for the list.
-	Package list - Lets you specify a package as the basis for the list.
-	Owner List - [PDMs only] Lets you specify a user as the basis for the list. The list will contain only objects owned by the specified user or by no user. If you select User <NONE>, then all the objects are displayed.

Tool	Description
	Include Sub-Packages - Includes objects contained in sub-packages (Include Sub-Packages) in the list. As this tool allows you to display all objects, regardless of their package, some objects in the list may have the same name and be difficult to identify. In this situation, you can use the Customize Columns and Filter tool, to display the Object Location column to identify where the objects are defined.
	Include Composite-Objects - Includes composite objects, such as sub-process, sub-activity, sub state, in the list
	Include External Shortcuts - Includes shortcuts to objects in other models in the list. The model containing the original objects must be open for external shortcuts to be available for selection. When generating, external shortcuts are generated as ordinary objects.
	Select All - Selects all the objects in the current object type tab. To select all check boxes in all object type tabs, click the arrow and select All Lists or you can press the *Ctrl* key and click the Select All tool.
	Deselect All - Deselects all the objects in the current object type tab. To clear all check boxes in all object type tabs, click the arrow and select All Lists or press the *Ctrl* key and click the Deselect All tool.
	Use Graphical Selection - Selects only the objects currently selected in the diagram. To apply the graphical selection in all object type tabs, press the *Ctrl* key and click the Use Graphical Selection tool.
	Move Selected Items to Top - Moves all selected objects to the top of the list.
	Move Selected Items to Bottom - Moves all selected objects to the bottom of the list.
	Customize Columns and Filter - Opens the Customize Columns and Filter dialog (see Customizing Object List Columns and Filtering Lists [page 123]), which allows you to define a filter expression to apply to the selection list.
	Enable/Disable Filter - Applies the filter defined in the Customize Columns and Filter dialog to restrict the list of objects available for selection to those meeting its criteria. The currently defined filter is displayed in the Filter box underneath the object list.
	Use Filter for Selection - Applies the filter defined in the Customize Columns and Filter dialog to select all objects meeting its criteria from the list. This selection by criteria is persistent for as long as the tool is applied.
Ctrl + selection tool	Applies the action of the selection tool to all object types in the different tabs.

Saving Object Selections

In some selection lists, you can save sets of object selections in your model, so that you can reuse them easily. To save a selection, enter a name in the Selection list at the bottom of the *Selection* tab then click the *Save* tool beside the list. The selection is saved as part of the model file.

> **i** Note
>
> When reverse engineering from a live data source, object selections are saved into separate files, as you need not have any model open in the Workspace (see *Data Architecture > Building Data Models > Generating and Reverse-Engineering Databases > Reverse Engineering a Database into a PDM > Database Reverse Engineering Selection Window*). In this case, you must select a folder before being able to save the object selection.

Confirming Object Selections

In a selection list, the display of your object selection may be modified whenever you perform one of the following actions:

- Change the folder selection using the *Model* or *Package* list
- Deselect the *Include Sub-Packages/Sub-Objects* tool
- Deselect the *Include Shortcuts* tool
- Apply a filter using the *Enable Filter* tool
- Change the database or owner in the Reverse Engineering from a data source dialog box

In this case, some objects that have already been selected will no longer be displayed, and a dialog opens prompting you to:

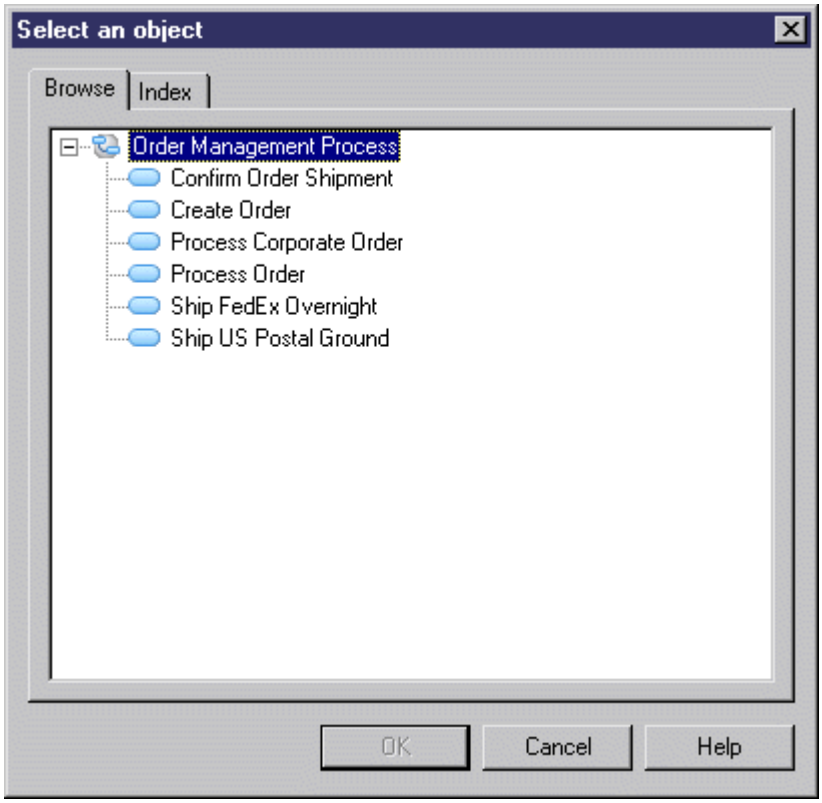
- Select only the objects displayed - Other objects that are no longer displayed are deselected.
- Keep the hidden objects in the selection – All the previously selected objects are retained, including those that are no longer shown. This allows you to take into account object selections you have made in several packages for example.
- Cancel - The commit of the selection list is canceled and the selection page now displays all objects and sub-objects in the model to let you modify your selection if necessary.

If you confirm your selection by clicking *OK*, the confirmation dialog will not be displayed again, even if you again modify your parameters.

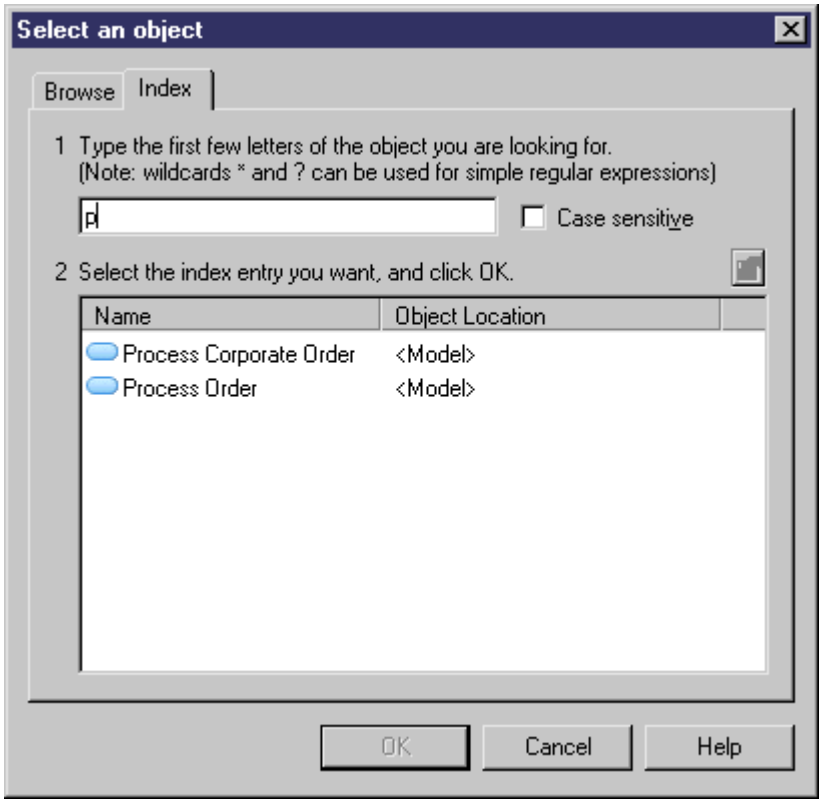
1.4.9.6 Selecting an Object from a Selection Tree

Certain selection dialogs allow you to choose objects from a tree view or by searching by name.

The *Browse* tab allows you to choose an object from a tree view:



The *Index* tab lets you search for an object by entering all or part of its name (or code depending on the *Display Name/Code* model option). Objects are sorted alphabetically in the list and dynamically update as you type. Select an object and click the *Properties* button to view its property sheet:



You can select an object on either tab and then click the other tab to retain the choice and view it in that context. Double-click an object, or select it and click *OK*, to commit the selection and close the selection dialog.

1.4.10 Finding Objects

PowerDesigner lets you search for objects within all the models currently open in your workspace.

Context

> **i** Note
>
> You can quickly find objects by name in the workspace by entering a string in the *Filter* field at the top of the Browser *Local* tab (see The Browser [page 37]).

Procedure

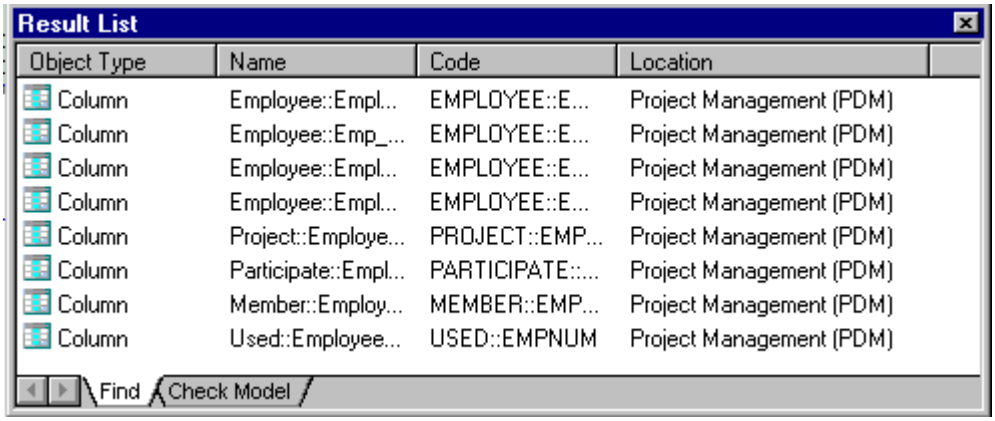
1. Select *Edit* > *Find Objects* (or press Ctrl + F) to open the Find Objects dialog.

2. Specify the appropriate constraints on these tabs:
   - *Name and Location* - to search on the name and location of PowerDesigner objects.
   - *User and Date* - to search on the user and date of creation/modification of PowerDesigner objects.
   - *Advanced*- to search on other criteria.
   - *Glossary* - to search on keywords, terms, related terms, and synonyms.

   For detailed information about these tabs, see Find Parameters [page 135].
3. Click *Find Now*.

> **i Note**
>
> Once you have started the Find Objects process, you can stop it at any time by clicking the *Stop* button.

The Find Object dialog remains open, displaying messages in the Output pane, until the end of the process, when the Result List displays the result:



Right-click an object in the list to find it in a diagram or the Browser, to open its property sheet or perform an impact and lineage analysis on it.

The following operations are available by right-clicking an object in the Results List:
- *Properties* - to open the object's property sheet (see Object Properties [page 107]).
- *Impact and Lineage Analysis* - to analyze the object (see Impact and Lineage Analysis [page 489]).
- *Find in Browser* - to highlight the object in the Browser (see The Browser [page 37]).
- *Find in Diagram* - to open the diagram (see Diagrams [page 185]) with the symbol centered. If the object has symbols in multiple diagrams, you see a list. If the object has no symbol, a warning message appears.
- *Copy* - to copy the object (seeMoving and Copying Objects [page 103]). Select a destination in the Browser or a diagram and select *Edit* > *Paste* or *Edit* > *Paste as Shortcut*.

> **i Note**
>
> For information about searching in the repository and in PowerDesigner Web, see Finding Objects in the Repository [page 292] and Searching for Diagrams or Objects.

1.4.10.1 Find Parameters

The Find Objects dialog contains tabs that let you precisely specify your search criteria.

Name and Location Tab

The *Name & Location* tab provides basic search parameters:

Parameter	Description
Look in	Specifies the scope of the search. You can select the entire workspace, a project, folder, model, or package.
Model type	Specifies the type of PowerDesigner model to search. The options available in this list are affected by your choice in the *Look in* field.
Object type	Specifies the type of model objects to be searched. The options available in this list are affected by your choice in the *Model type* field.
Include shortcuts	Instructs PowerDesigner to include object shortcuts that match your criteria in the results.

Name	Specifies the object name to search for. You can use the following special characters:
	• `*` - none to any number of characters. For example:
	  ○ `W*` finds "Work" and "Washington"
	  ○ `*96` finds "01/11/96" and "26/08/96"
	• `?` – exactly one character. For example:
	  ○ `*_emp_???` finds "Div_emp_idn" but not "Div_emp_ident"
	• `\` - escapes `*`, `?`, or `\`. For example:
	  ○ `\?\\` - finds ?\
	• true/false - Boolean value (True is when the check box is selected)

Code	Code of the object. You can type the exact code of the object or use a string expression.
Case sensitive	Specifies that the results must match the case of the criteria.

User and Date Tab

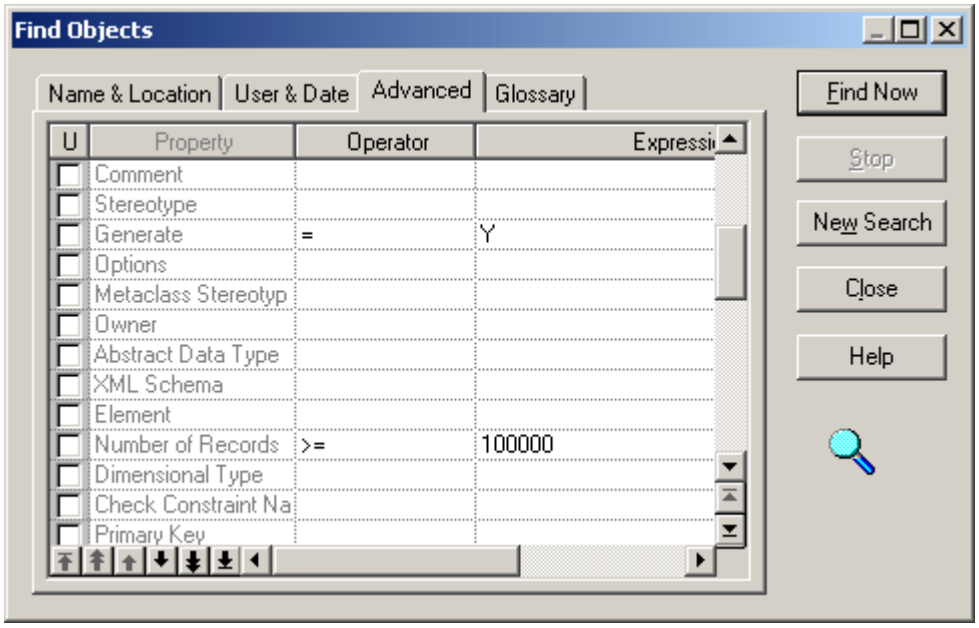
The *User & Date* tab lets you constrain your search based on when and by whom the object was created and modified:

Parameter	Description
Created	Enables searching on creation parameters. You can search against the name of the user who created the object and against the creation date using the following options: • Before the specified date • After the specified date • Between the two specified dates • In the specified number of days since today
Modified	Enables searching on modification parameters.

Advanced Tab

The *Advanced* tab allows you to specify additional search constraints for each property of the selected object type in the following format:

Parameter	Description
U (Used)	Specifies a property on which to apply the search. Select this option for a property without entering an expression to find objects for which the property is null.
Operator	Specifies an operator to use for the search. Click in the *Operator* column to display the list of available operators.
Expression	Specifies a string expression to search for. For detailed information about the operators and the expression syntax, see Defining a Filter Expression [page 127].



> **Note**
>
> If you do not select an object type on the *Name & Location* tab, then you can only search on the name and code properties on this tab.

Glossary Tab

The *Glossary* tab allows you to search against any glossary objects (see The Glossary [page 38]) with which objects may be associated:

Parameter	Description
Search by	Specifies the string that must be contained in the glossary object's name.
Search Range	Specifies the glossary objects to search on. You can select one or more of *Keywords* and *Terms* [default] with which the object is related, and *Related Terms* and *Synonyms* that are associated with the term (see Term Properties [page 41]).
Case sensitive	Specifies that the results must match the case of the criteria.

1.4.11 Business Rules

A business rule is a written statement specifying what a system must do or how it must be structured. Rules can be derived from a government-imposed law, a customer requirement, or an internal guideline. You can

attach rules to your model objects to complement your diagrams with information that is not easily represented graphically.

For example, a rule stating that "An employee belongs to only one division." can help you graphically build the link between an employee and a division. Rules often start as simple observations that develop, during the design process, into more detailed expressions. You may, for example, develop rules to explicitly define what information a customer supplies when placing an order, or how much a customer can spend based on a credit limit.

Rules can be developed from procedures that the system must respect, specifications dictating the scope of the project, and external constraints. In the case of the PDM and OOM, you can generate business validation rules attached to domains as check parameters.

Creating a Business Rule

You can create a business rule from the *Model* menu, from the Browser or from another object's property sheet:

- Select *Model* > *Business Rules* to access the List of Business Rules, and click the *Add a Row* tool.
- Right-click the model or package in the Browser, and select *New* > *Business Rule*.
- Open the property sheet of the object to which you want to apply the rule, click the *Rules* tab, and click the *Create an Object* tool.

Business Rule Properties

You can modify an object's properties from its property sheet. To open a business rule property sheet, double-click its Browser entry in the Business Rules folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for business rules. The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.

Property	Description
Type	Specifies the nature of the business rule. You can choose between: • Constraint – a check constraint on a value. In a PDM, constraint business rules can be generated in the database. For example, "The start date should be inferior to the end date of a project." • Definition – a property of the element in the system. For example; "A customer is a person identified by a name and an address". • Fact – a certainty in the system. For example, "A client may place one or more orders". • Formula – a calculation. For example, "The total order is the sum of all the order line costs". • OCL constraint [OOM only] – An Object Constraint Language expression. • Requirement – a functional specification. For example, "The model is designed so that total losses do not exceed 10% of total sales". • Validation – a constraint on a value. For example, "The sum of all orders for a client must not be greater than that client's allowance".

The following tabs are also available:

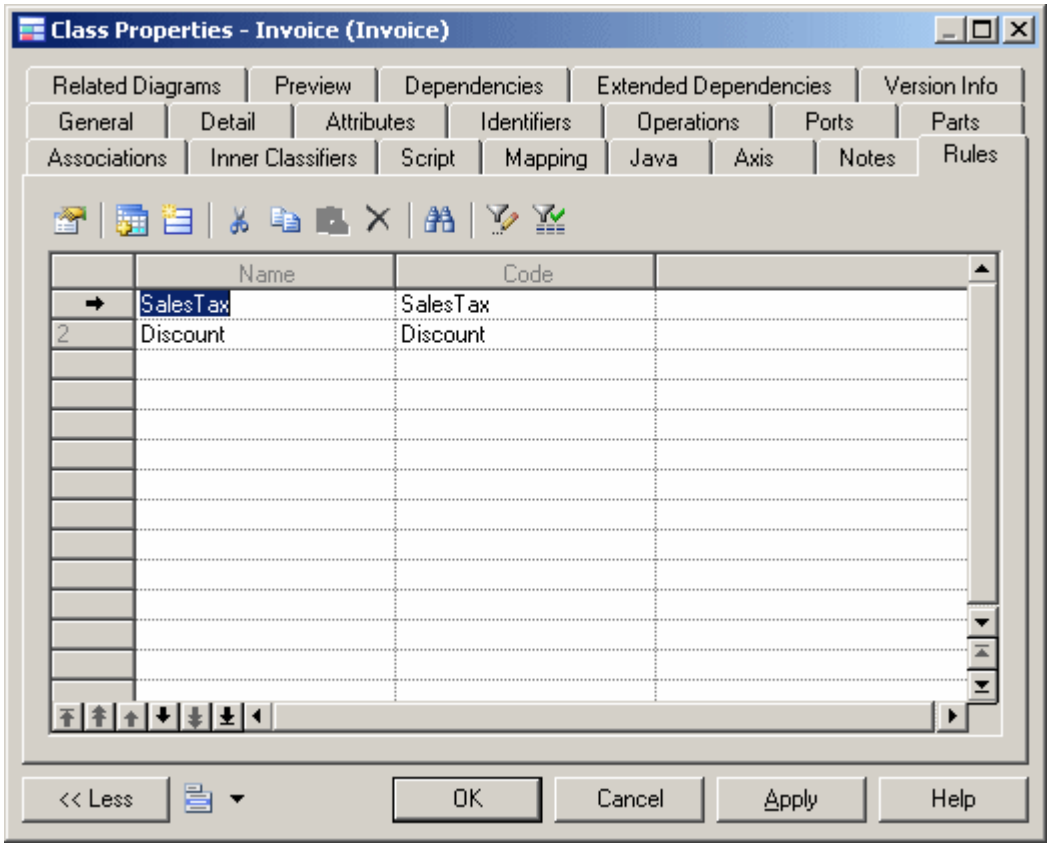
- *Expression* - Though business rules typically start out as descriptions, as you develop your model and analyze your business problem, you can enrich them by adding technical expressions on this tab. Expressions are used primarily in CDMs and PDMs. Each rule can include two types of expression, which you define on the appropriate sub-tab:
  - *Server*
  - *Client*
- *OCL Constraint* - This tab is only available for business rules with a type of OCL Constraint. Enter your OCL expression in the text field using the Object Constraint Language / UML expression language.

1.4.11.1 Attaching a Business Rule to an Object

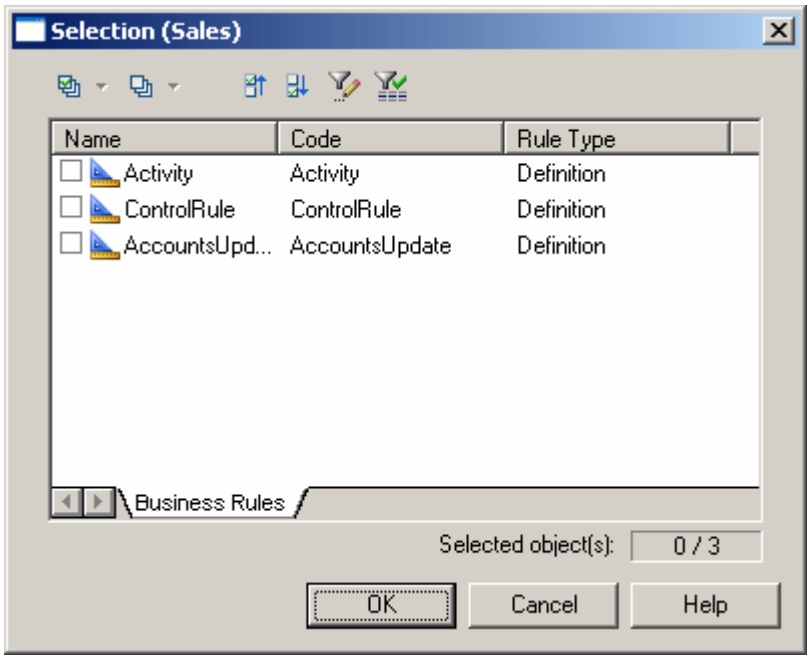
You can attach business rules to your objects on the *Rules* tab of the object's property sheet.

Procedure

1. Open the property sheet of an object, and then click the *Rules* tab:



2. Click the *Add Rules* tool to open a selection window listing all the business rules available in the model:



3. Select the business rules you want to add to the object, and then click *OK* to return to the object's property sheet.
4. Click *OK* to close the object property sheet and return to the model.

> **i Note**
>
> When you apply a business rule to an object, its U (Used) column in the List of Business Rules is automatically checked. This column allows you to see what rules are unused, and delete them if necessary.

1.4.12 File Objects

You can embed any application file in your model or provide a link to it. For example you could include a detailed description of your problematic by pointing to a text or Word file from your model. In an OOM, you could associate a class to a Java file. The file itself can be embedded in the model or referenced by it.

File objects can be simply included in the model or can be attached to other objects via traceability links.

To open a file in its associated editor, double-click its diagram symbol or browser entry.

> **i Note**
>
> To change the associated editor, right-click the file object and select *Open With* > *Choose Program*.

Creating a File Object

You can create a file object through drag and drop, or from the Toolbox, Browser, or *Model* menu:

- Drag a file from Windows Explorer and drop it in the diagram or Browser to automatically create a file object pointing to the file in the filesystem.
- Use the *File* tool in the Toolbox.
- Select *Model* > *Files* to access the List of Files, and click the *Add a Row* tool.
- Right-click the model or package in the Browser, and select *New* > *File*.

File Object Properties

To view or edit a file object's properties, right-click its diagram symbol or Browser or list entry and select *Properties*. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator. The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Location type	Specifies the nature of the file object. You can choose from the following: • Embedded file – the file is stored within the model and is saved when you save the model. If you subsequently change the type to external, you will be warned that the existing contents will be lost. • External file – the file is stored in the Windows file system, and you must enter its path in the Location field. If you subsequently change the type to embedded, you will be prompted to import the contents of the file into the model. • URL – the file is on the web and you must enter its URL in the Location field
Location	Specifies the path or URL to the file.
Extension	Specifies the extension of the file object. By default, the extension is set to `txt`.
Generate	Specifies to generate the file object when you generate the model to another model.
Artifact	Specifies that the file object is not a piece of documentation, but rather forms an integral part of the application. If an artifact has an extension that is defined in the Editors page in the General Options dialog linked to the <*internal*> editor, a Contents tab is displayed in the artifact property sheet, which allows you to edit the artifact file in the PowerDesigner text editor.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

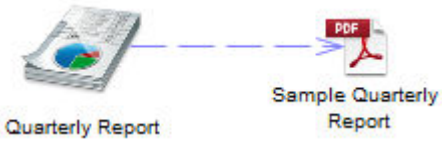
1.4.12.1 Attaching a File Object to a Model Object

You can attach file objects to other model objects via traceability links. Click the *Link/Traceability Link* tool in the toolbox to draw a link from the object symbol to the file object symbol.

Alternatively, you can right-click an object symbol in the diagram and select *Edit* > *File* > *Add File*, or use the *Add Objects* tool on the *Traceability Links* tab of the object property sheet (see Traceability Links [page 146]).

The connection between the object and the file object can be seen:

- As a traceability link in a diagram (if both the file and the object have symbols in the diagram):



- On the *Dependencies* tab of the file object's property sheet.
- On the *Traceability Links* tab of the object's property sheet.
- By right-clicking the object symbol and selecting *Edit* > *File* > `<filename>`.

1.4.13 Packages

Packages can be used to organize your model into different tasks and subject areas, and to assign parts of it to different development teams. Each package has a default diagram, and you can add further diagrams, model objects, and sub-packages. You can create as many packages as you need in a model and decompose a package into other packages, and so on.

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Default diagram	[package only] Specifies the diagram to display by default when you open the package.

Property	Description
Use parent namespace	[package only] Specifies that the package does not represent a separate namespace from its parent and thus that objects created within it must have names that are unique within the parent container. If this property is not selected, then the package and its parent package or model can both contain objects that are called Object A.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.4.13.1 Controlling the Namespace of a Package

PowerDesigner applies uniqueness checks on the names of objects. The namespace defines an area in which the name and the code of an object of a given type must be unique. You can decide whether an individual package must represent a separate namespace or not with the *Use parent namespace* property.

Context

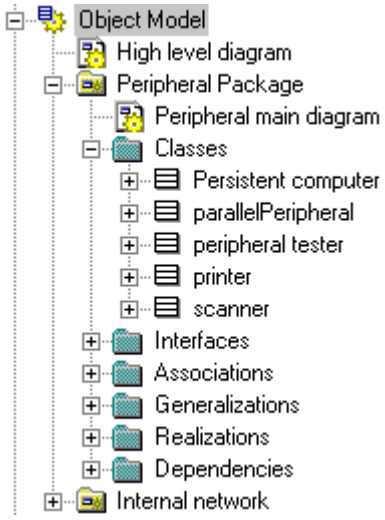
Depending on the type of model where you create a new package, this option may be selected by default:

- For the EAM, CDM, LDM, PDM, DMM, and FEM - the entire model is, by default, a single namespace, and all the packages have the *Use parent namespace*. PowerDesigner applies checks on uniqueness at the model level.

  > **i** Note
  >
  > The Glossary and XSM do not support packages. For information on namespaces in the XSM, see *XML Architecture* .

- For the OOM, RQM and BPM – each package is, by default, a separate namespace and the property is not selected. PowerDesigner applies checks on uniqueness at the package level.

In the following example, the class Printer is located in the Peripheral Package of an OOM where each package is, by default, a separate namespace:



1.4.14 Areas

An area is an abstract object that can group together other objects. The objects do not belong to the area and are just grouped in it.

Areas are available in the following types of diagrams:

- EAM - All diagrams except organization charts
- FEM
- CDM, LDM, PDM
- DMM - Top-level diagrams
- OOM - All diagrams except sequence and interaction overview diagrams

Creating an Area

You can create an area from the Toolbox, Browser, or *Model* menu, or on the *Sub-Areas* tab of an area.

Area Properties

To view or edit an area's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations - [EAM only] Lists the organization units, people, and roles associated with the area, and the type of role they play in relation to it.
- Attached Objects – Lists the objects that are associated with the area.
- Sub-Areas – Lists the areas contained within the area.

1.4.15 Traceability Links

Traceability links provide a flexible means for creating a connection between any object in any type of model and any other object in the same model or any other model open in your workspace. Traceability links have no formal semantic meaning, but can be followed when performing an impact analysis or otherwise navigating through the model structure.

To create a traceability link between objects in the same diagram, select the *Link/Traceability Link* tool in the Toolbox. Click inside the symbol of the object that is dependent and, while continuing to hold down the mouse button, drag the cursor and release it on the symbol of the object on which it depends. In this example, the **`Work`** entity is shown as being dependent on **`School`** through a traceability link:

To create a traceability link to any object in any model that is open in the Workspace, open the property sheet of the dependent object, click its *Traceability Links* tab, and click the *Add Objects* tool. Use the *Model* list to select a different model, select the object to point to and click *OK* to create the link and return to the dependent object's *Traceability Links* tab. You can optionally specify a type for any traceability link in the *Link Type* column.

Click the *Types and Grouping* tool to perform various actions on this tab:

- To make a link type available for selection in the *Link Type* column, click the *Types and Grouping* tool and select *New Link Type*. Enter a *Name* for the link type and, optionally, a *Comment* to explain its purpose, and then click *OK*.

> **Note**
>
> Traceability link types created in this way are stored as stereotypes in an extension file embedded in the model. To work directly with this file click the *Types and Grouping* tool and select *Manage Extensions*. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files* .

- To control the display and grouping of links, click the *Types and Grouping* tool and select:
  - *No Grouping* - to display all the links in a single list.
  - *Group by Object Type* - to display links to different types of objects on separate sub-tabs. To add a link to a new object type, click the plus sign on the leftmost sub-tab.
  - *Group by Link Type* - to display different link types on separate sub-tabs. To add a new link type, click the plus sign on the leftmost sub-tab.

> **Note**
>
> To see all of the objects that point to an object via traceability links, open its property sheet, click its *Dependencies* tab, and click the *Incoming Traceability Links* sub-tab.

1.4.16 Extended Objects, Sub-Objects, and Links

Extended objects, sub-objects, and links let you model concepts that are not natively supported by PowerDesigner. For example, you can add extended objects to a PDM to model new database objects. You can specify generated files and templates in the definition of your extended objects in order to enable their generation and reverse-engineering.

Extended objects and extended links are available by default in the free model (see The Free Model (FEM) [page 241]), and can be added to any other type of model through the Profile category in the model's resource file or in an extension file (see *Customizing and Extending PowerDesigner > Extension Files*).

To view or edit an extended object's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Source	[extended links only] Specifies the name of the origin object of the extended link.
Destination	[extended links only] Specifies the name of the destination object of the extended link.

Property	Description
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

> **i** Note
>
> You can use the CanLinkKind event handler to restrict the kind of objects you want to link together. This event handler is called when you create a link using the Toolbox tool or when you try to modify the ends of a link from its property sheet. For more information, see *Customizing and Extending PowerDesigner > Extension Files > Event Handlers (Profile)* .

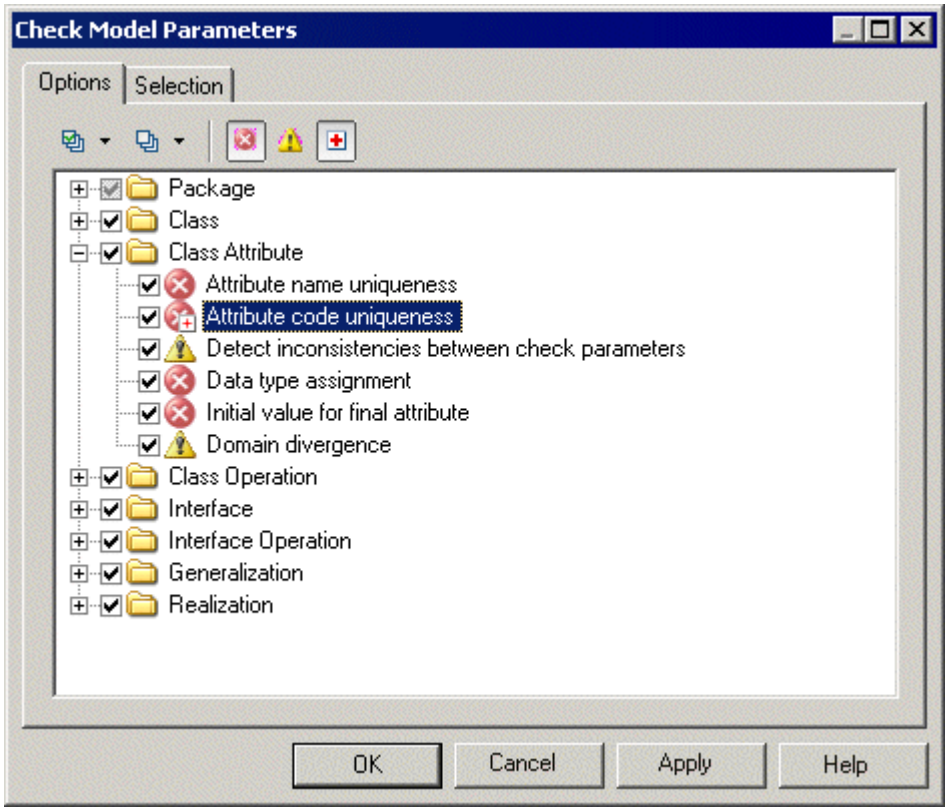
1.4.17 Checking Models

You can check the validity of your model at any time. We recommend that you check your model before generating code or another model from it. The Check model option is enabled by default in the Generate dialog box and, if an error is found, the generation is stopped.

Procedure

1. Press F4, select *Tools* > *Check Model*, or right-click the diagram background and select *Check Model* to open the Check Model Parameters dialog.

   The *Options* tab lists the types of objects to be checked, and the individual checks to be performed are displayed with symbols indicating their severity:

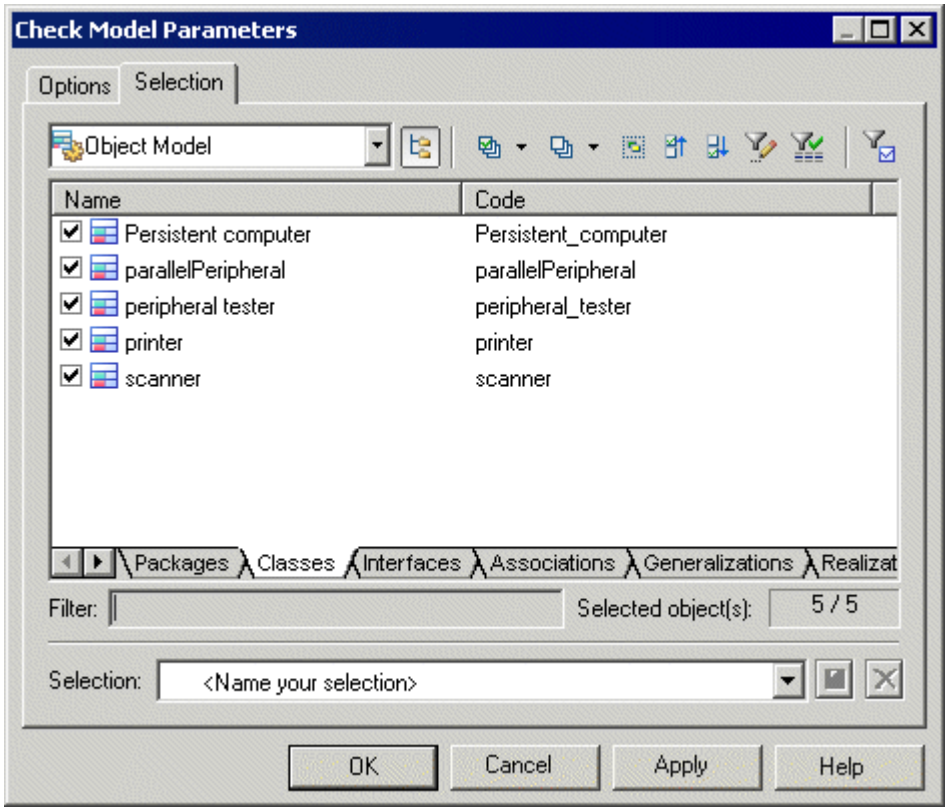


2. [optional] Select or deselect types of objects to check, and expand object nodes to enable, disable, vary the severity of, and enable or disable automatic correction of individual checks with the following tools:

Tool	Description
	Select All – Click the arrow to select all checks, all error checks, or all warning checks.
	Deselect All - Click the arrow to deselect all checks, all error checks, or all warning checks.
	Error – Sets the selected check to error level. When errors are encountered, any model generation is stopped.
	Warning - Sets the selected check to error level. Warnings do not allow model generation to proceed.
	Automatic correction – [if available for the selected check] Enables automatic correction for the selected check and places a small red cross on the bottom-right corner of the check icon. Automatic corrections may have unexpected effects, and you might want to disable them in some cases. For example, in a PDM, if a column code length is longer than the length specified in the DBMS `MaxColumnLen` entry, then PowerDesigner will truncate the code, but may also change the code to avoid duplicating an existing code.

> **Note**
>
> Right-click a check and select *Help* to display its documentation.

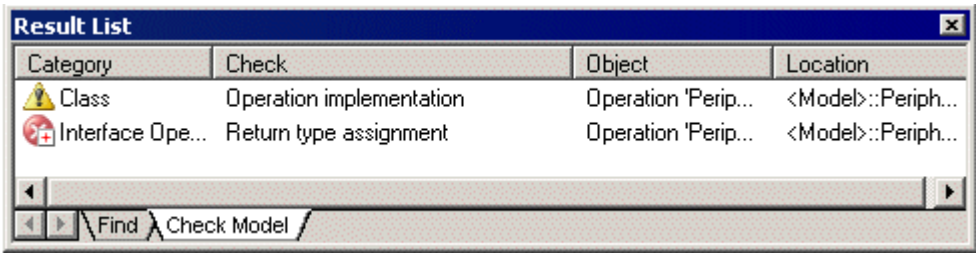
3. [optional] Click the *Selection* tab, and select or deselect individual objects for checking. Sub-tabs are available for each type of object:



> **Note**
>
> If you had previously selected symbols in your diagram, you can select them for checking with the *Use Graphical Selection* tool. For detailed information about these tools, see Adding an Item from a Selection List [page 129].

4. Click *OK* to begin the model check.

   The Check Model Result List displays errors and warnings based on the check options you have defined. For information about how to correct errors, see Correcting Errors in the Check Model Result List [page 151].



1.4.17.1 Correcting Errors in the Check Model Result List

When errors and warnings are encountered during model checking, they are listed in the Check Model Result List pane. You can correct the problems either by invoking a automatic correction (if available) or by opening the property sheet of the affected object and correcting it manually.

The following tools are available to assist you in correcting problems with your model. If this toolbar is not displayed, select *Tools* > *Customize Toolbars*, select *Check*, and click *OK*.

Tool	Description
	Correct error – Opens the property sheet of the affected object to allow you to correct the error.
	Display help – Provides documentation for the error or warning.
	Check again – Re-performs the check, to allow you to verify your correction.
	Automatic correction – Only available if an automatic correction is defined for this kind of error. Performs the automated correction.
	First error – Goes to the first error in the list.
	Previous error - Goes to the previous error in the list.
	Next error - Goes to the next error in the list.
	Last error - Goes to the last error in the list.

> **Note**
>
> These and other options are also available by right-clicking an item in the Check Model Result List.

1.4.17.2 Checking Generic Objects

Generic objects are available in all types of models and have standard checks defined for them. For model-specific checks, see the appropriate modeling guide.

1.4.17.2.1 Business Rule Checks

PowerDesigner provides default model checks to verify the validity of business rules.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Unused business rule	The business rule you have created is not used in the model. • Manual correction: Apply the business rule to an object in the model • Automatic correction: None

1.4.17.2.2 Extended Object/Link Checks

PowerDesigner provides default model checks to verify the validity of extended objects/links.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.

1.4.17.2.3 File Checks

PowerDesigner provides default model checks to verify the validity of files.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.

Check	Description and Correction
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Existence of external file location	External file objects should have a valid path location. • Manual correction: Define a valid path location • Automatic correction: None

1.4.17.2.4 Replication Checks

PowerDesigner provides default model checks to verify the validity of replications.

Check	Description and Correction
Partial replication	A replica object is partially synchronized with its replicated object. • Manual correction: Modify the list of replicated attributes from the replication property sheet • Automatic correction: Enforces the replication of desynchronized attributes of the replica object in the replication property sheet

1.4.17.3 Checking Glossary Compliance

When you enable the use of glossary terms for your object naming conventions, additional model checks help you monitor glossary compliance for object names and automate the replacement of unapproved synonyms with approved glossary terms. You can restrict the scope of your compliance efforts by disabling the checks for certain object types on a model-by-model basis.

The following checks are enabled for all model objects when you select *Use glossary for autocompletion and compliance checking* for your naming conventions (see Naming Conventions [page 155]):

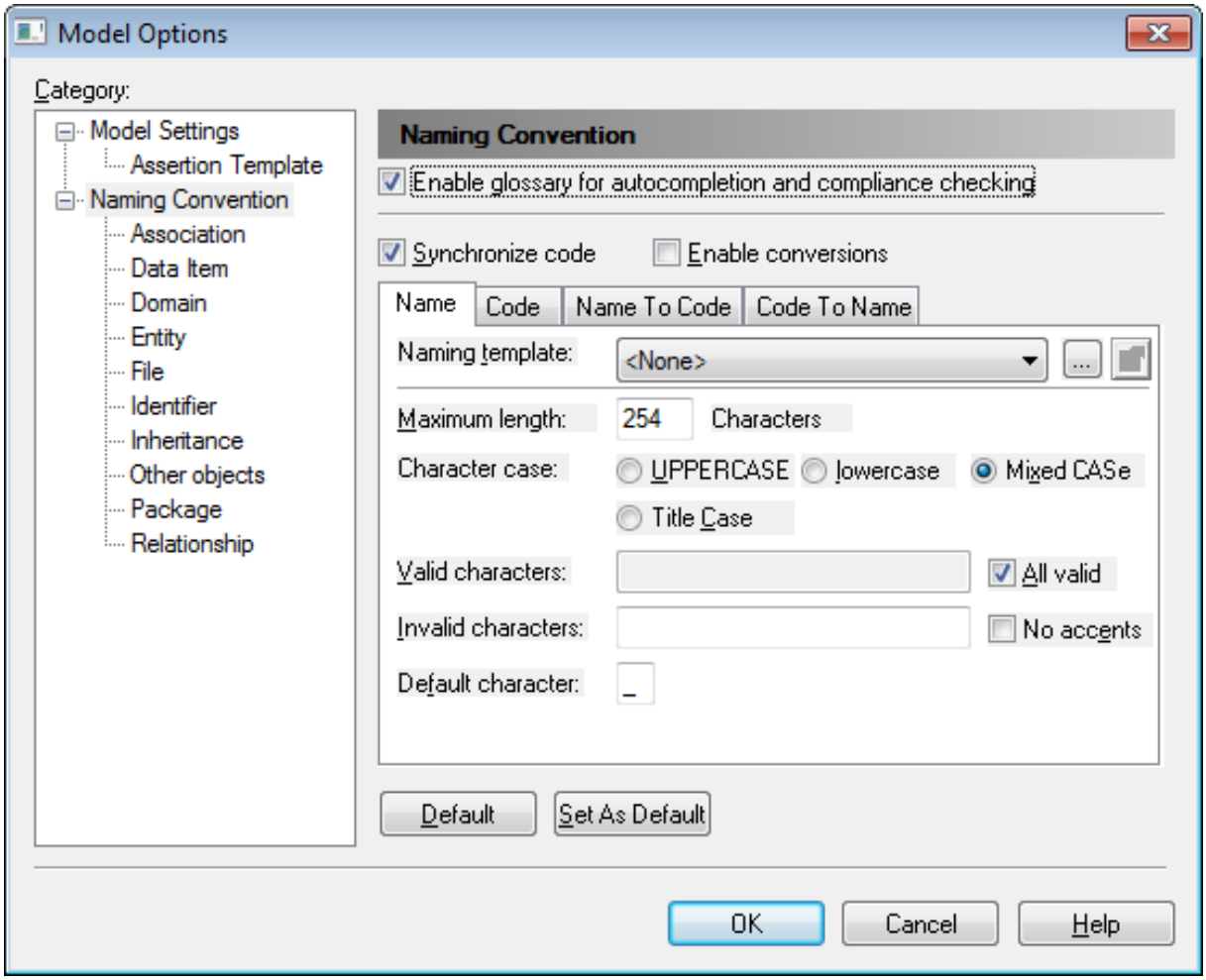
Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None. To disable this check for a given object type, clear its check box in the tree in the *Check Model Parameters* dialog. To reduce its severity from Error to Warning, right-click the check and select *Warning*. The check icon changes to a warning sign.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms. To enable the automatic replacement of synonyms with approved glossary terms for a given object type, right-click the check in the *Check Model Parameters* dialog, and select *Automatic Correction*. A small overlay with a red cross appears at the bottom-right of the check icon.

In addition, the following check is enabled for models and packages to allow administrators to add terms used in object names to the glossary:

Check	Description and Correction
List of missing terms in object names/codes in package	[if glossary enabled] This special check lists all the terms that are used in the names or codes of all the objects contained in the model or package. • Manual correction: An administrator with write permission on the glossary can double-click the line for this check in the *Result List* to open a merge window in which you can select to add some or all of these missing terms, as appropriate, to the glossary. • Automatic correction: None.

1.4.18 Naming Conventions

You can use naming conventions to specify a case, maximum length, and valid characters for object names and codes, and enforce compliance with an enterprise glossary. You can also invoke conversion scripts and conversion lists to generate appropriate object codes from names (or names from codes).



To set naming conventions to control the names and codes of all objects, select *Tools* > *Model Options* and click the *Naming Convention* node. To set naming conventions for a particular kind of object, select its node under the *Naming Convention* node.

These options are available only on the *Naming Convention* node and apply to all objects:

Option	Description
Enable glossary for autocompletion and compliance checking	Enables the use of the enterprise glossary in the model. When this option is selected, auto-completion is enabled for object names and codes using the glossary terms (see The Glossary [page 38]), and additional checks are made to ensure that all model objects use only approved glossary terms in their names and codes (see Checking Glossary Compliance [page 154]). For information about administering the glossary, see Deploying an Enterprise Glossary [page 540].

Option	Description
Synchronize code	Specifies that the code of the object is synchronized with the name of the object by applying the naming conventions specified on the *Code* subtab. This option is enabled by default unless or until you reverse-engineer or import a database, business process, XML or OO code, when it will be unselected to allow the free editing of object names without affecting the underlying code. You can reselect the option at any time. To disable name to code synchronization for an individual object, click to release the = button to the right of the *Code* field on the *General* tab of its property sheet. When names and codes are not synchronized, you can generate a name from a code or a code from a name by clicking the = button to the right of field that you want to change.
Enable conversions	Enables the application of the conversion scripts and conversion tables specified on the *Name to Code* tab when synchronizing object codes with names (see Name and Code Conversion Scripts [page 159]).

These options are available on both the *Name* and *Code* subtabs of the *Naming Convention* node (where they apply to all objects) and on each subnode (where they apply to the selected object):

Option	Description
Naming template	Specifies a naming template to supply the naming conventions for the name or code. A naming template contains the same fields as on this tab, but in a format that can be reused for other objects (see Creating a Naming Template [page 158]). Click the ellipsis button to the right of this field to open the List of Naming Templates, or click the *Properties* tool to view and edit the selected template.
Maximum length	Specifies the maximum number of characters permitted in a name or code. In a PDM, you can set the maximum here and in the DBMS definition file. If you specify both, then PowerDesigner applies the stricter constraint. For example, if you set 128 here and the DBMS file specifies 30, PowerDesigner applies the 30 character limit.
Character case	Specifies the case to apply to the name or code. You can choose between: • UPPERCASE • lowercase • Mixed CASe • Title Case (Name only) • UpperCamelCase (Code only) • lowerCamelCase (Code only)

Option	Description
Valid characters	Specifies the list of characters permitted in the name or code. Enter character ranges between single quotes, separated by a dash. Enter individual or multiple characters between double quotes. Separate valid items by a comma. By default, PowerDesigner allows the following valid characters for codes: `'a'-'z','A'-'Z','0'-'9',"_"` Select *All valid* to the right of this field to permit any character.
Invalid characters	Specifies the list of characters not permitted in the name or code. By default, PowerDesigner excludes the following characters for names: `"+-*/\.,!:;"` Select *No accents* to the right of this field to remove accents from accented characters.
Default character	Specifies the character that is used to replace any invalid characters that are entered.

1.4.18.1 Creating a Naming Template

You can create a naming template to specify naming conventions for names or codes and reuse it for multiple types of objects.

Procedure

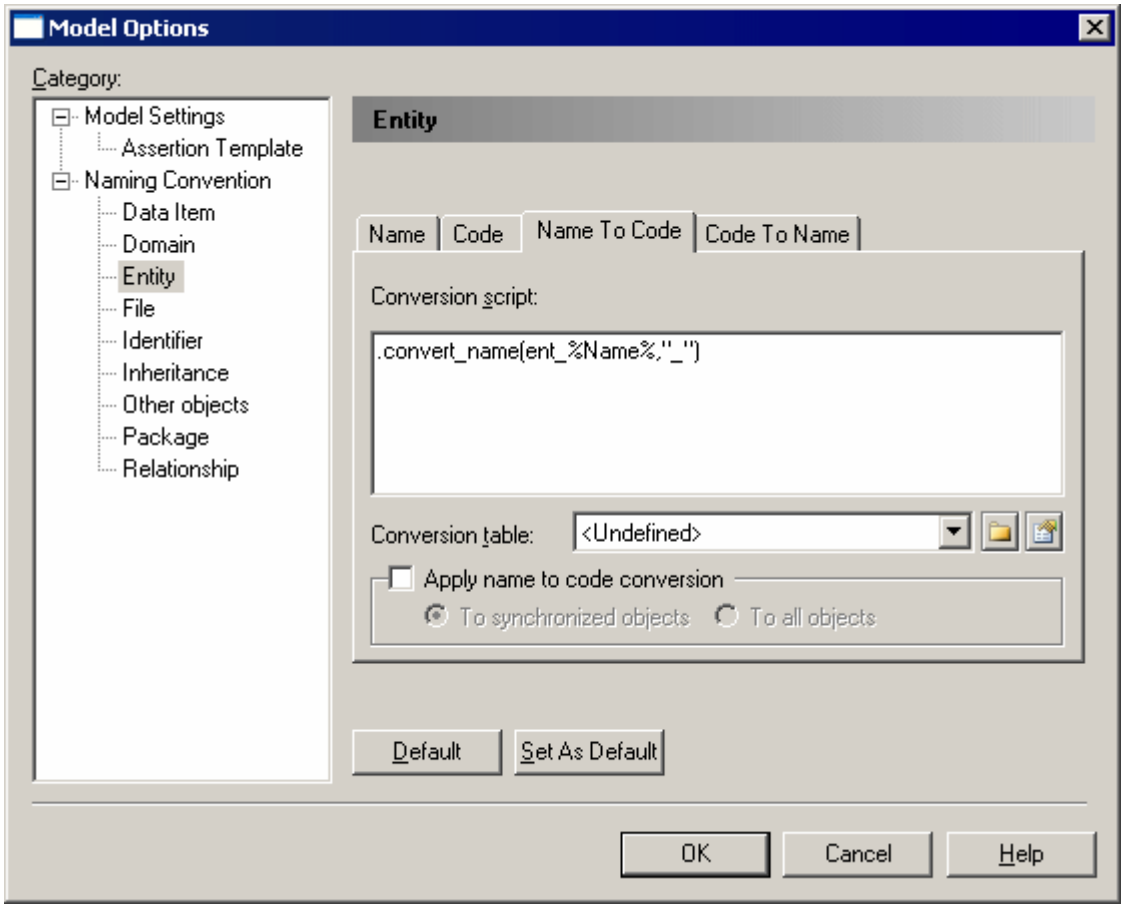
1. Select *Tools* > *Model Options* and click the *Naming Convention* node in the Category tree.
2. Click the Ellipsis button to the right of the *Naming Template* field to open the List of Naming Templates.
3. Click the *Add a Row* button and enter a name for the new naming template.
4. Click the *Properties* tool to open the template property sheet, complete all the appropriate properties (see Naming Conventions [page 155]), and then click *OK* to return to the list.
5. Click *OK* to return to the Model Options page.

   Your template is now available for selection in the *Naming Template* field. Click the *Properties* tool to the right of this field to modify the selected template.

1.4.18.2 Name and Code Conversion Scripts

Conversion scripts permit more complex transformations, including adding prefixes or suffixes and the use of conversion tables. You specify the script on the *Name to Code* or *Code to Name* tab, depending on which direction you want to perform the conversion.

> **i** Note
>
> To enable the use of your conversion script, you must select *Enable name/code conversions* in the upper part of the *Naming Convention* page of the Model Options dialog.



The following options are available on each of these tabs:

Option	Description

Conversion script	The default Name to Code conversion script is:
	`.convert_name(%Name%,"_")`
	This script takes the value of the *Name* field (represented by the GTL variable **%Name%**) and generates a *Code* by replacing any nonalphanumeric characters with an underscore. It also calls a conversion table (if one is selected in the *Conversion table* field) to perform conversions on specific strings encountered in the name or code.
	You can modify the conversion script. For example, to insert the prefix **tbl_** before the code of each table, use:
	`.convert_name(tbl_%Name%,"_")`
	If the **stdnames** conversion table is selected then, for a table with the name **Customer**, PowerDesigner automatically provides a code **tbl_CUST**.
	You can enter any valid GTL code (including macros such as **.foreach_part**, **.lowercase**, **.uppercase**, **.replace**, and **.delete**), but only the **.convert_name** and **.convert_code** macros call a conversion table if one is selected.
	For detailed information about GTL, see *Customizing and Extending PowerDesigner > Customizing Generation with GTL* .

Conversion table	Specifies the conversion table to use to perform conversions on specific strings encountered in the name or code (see Conversion Tables [page 161]). PowerDesigner provides an example conversion table called **stdnames.csv**. For example, an object name **Customer_1** is transformed into the object code **CUST_1** if **stdnames** is selected. To use your glossary term names and codes (see The Library [page 42]) as a conversion table, select **glossary terms**.
Apply name to code conversion/ Apply code to name conversion to all objects	Applies the conversion script (and conversion table) when you click *OK*: • *To synchronized objects* - those objects in which the object code is currently synchronized with the name. • *To all objects* - all objects in the model regardless of whether the object code is synchronized with the name. The option on the *Code to Name* tab always applies your changes to all objects in the model.

> **i Note**
>
> When generating one model from another, you can use name to code conversion to provide appropriate codes for the target model objects (see the *Convert Names into Codes* options on the Model Generation Options [page 425]).

1.4.18.3 Conversion Tables

Conversion tables are .csv files containing a list of strings that may appear in object names matched with equivalent (generally shorter) strings with which to replace them when generating object codes. For example, each instance of the word **customer** in object names in your model could be converted to **cust** in their codes.

Context

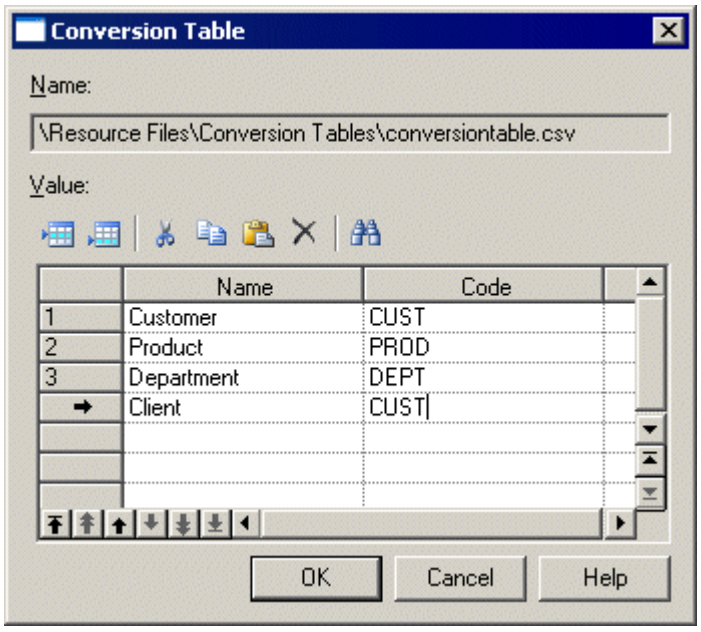
You can add your existing .csv files containing name/code pairs to the conversion table folder and create conversion tables directly within PowerDesigner.

> **i Note**
>
> To review the example conversion table provided with PowerDesigner navigate to and open `<install_dir>/Resource Files/Conversion Tables/stdnames.csv`.

You can check conversion tables into the repository `Library` folder to make them available to all users (see Sharing Resource Files via the Library [page 551]).

Procedure

1. Select *Tools* > *Resources* > *Conversion Tables* to open the List of Conversion Tables.
   To review an existing conversion table, select it in the list and click the *Properties* tool. To change the folder where PowerDesigner looks for conversion tables to populate the list, click the *Path* tool.
2. Click the *New* tool, enter a name and click *Save* to create the file and open the Conversion Table dialog.
3. Click the *Add a Row* tool and then enter a name and the code into which you want to convert it:



You can create as many conversion Name-Code pairs as required. The values in these columns are not, by default, case-sensitive, but you can turn on case-sensitive matching in your conversion script (see .convert_name and .convert_code Macros [page 162]).

> **i Note**
>
> It may be that several different names are converted to the same code. For example, **client** and **customer** may both be converted to **CUST**. If a table containing these lines were used to convert codes to names, then the first value encountered in the Name column is always used and there is a risk that multiple objects will be given the same name.

4. Click *OK* to return to the list of conversion tables, and then click *Close* to return to your model.

   Your conversion table is now available to be selected in the Model Options dialog *Naming Convention* page (see Naming Conventions [page 155]).
5. [optional] Check the conversion table into the repository `Library/Resources/Conversion Tables` folder to make it available to other users (see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

1.4.18.4 .convert_name and .convert_code Macros

These macros convert the object name to its code (or vice versa) and can search for and apply values found in an associated conversion table.

Use the following syntax to convert a name to a code:

```
.convert_name (<expression>[,"<separator>","<delimiters>",<case>,<sensitive>])
```

Use the following syntax to convert a code to a name:

```
.convert_code (<expression>[,"<separator>","<delimiters>",<sensitive>])
```

The following parameters are available:

Parameter	Description
`<expression>`	Specifies the text to be converted. For .convert_name, this is generally the `%Name%` variable and may include a suffix or prefix. Type: Simple template
`<separator>`	[optional] Character generated each time a separator declared in `<delimiters>` is found in the code. For example, "_" (underscore). Type: Text

Parameter	Description
<delimiters>	[optional] Specifies the different delimiters likely to exist in the input code or name, and which will be replaced by <separator>. You can declare several separators, for example "_ " and "-" Type: Text
<case>	[optional for .convert_name only] Specifies the case into which to convert the code. You can choose between: • firstLowerWord - First word in lowercase, first letters of subsequent words in uppercase • FirstUpperChar - First character of all words in uppercase • lower_case - All words in lowercase and separated by an underscore • UPPER_CASE - All words in uppercase and separated by an underscore
<sensitive>	[optional] Controls case-sensitivity when looking up names to convert via a conversion table. You can choose between: • 0 [default] - The first match found in the conversion table, irrespective of the case, is used. The names **account**, **Account**, and **ACCOUNT** are treated as identical, and the first one encountered in the list is matched. • 1 - Matching in the conversion table is case-sensitive and **account**, **Account**, and **ACCOUNT** are treated as three different names.

Examples

In this example, the template is applied to class name Corporate Customer:

```
.convert_name (cls_%Name%[,,,FirstUpperChar])
```

Result:

```
clsCorporateCustomer
```

In this example, the template is applied to the entity name Current Account, and a conversion table is provided. The macro specifies that case-sensitive matching should be used:

```
.convert_name (%Name%[,,,,1])
```

The conversion table contains the following lines:

```
account, AC
Account, ACC
ACCOUNT, ACCT
```

Result:

```
CURRENT_ACC
```

Examples

In this example, the template produces a list of column names before and after conversion:

```
.foreach_item(Columns)
 %Name%,
 .foreach_part(%Name%)
  .convert_name(%CurrentPart%,,,lower_case)
 .next()
.next(\n)
```

Result:

```
Column 1, column_1
 Column 2, column_2
 Column 3, column_3
```

1.4.18.5 .delete and .replace Macros

These macros delete or replace all instances of the given string in the text input.

```
.delete (<string>)
    <block-input>
.enddelete
```

```
.replace (<string>,<new-string>)
    <block-input>
.endreplace
```

The following parameters are available:

Parameter	Description
`<string>`	Specifies the string to be deleted. Type: Text
`<new-string>`	[.replace only] Specifies the string with which to replace `<string>`. Type: Text
`<block-input>`	Specifies the text to be parsed for instances of the `<string>` to delete or replace. Type: Complex template

Examples	Result

`.delete(Get)`
`  GetCustomerName`
`.enddelete`
	`CustomerName`

Examples	Result

`.replace(Get,Set)`
`GetCustomerName`
`.endreplace`
	`SetCustomerName`

`.replace(" ", _)`
`Customer Name`
`.endreplace`
	`Customer_Name`
	i Note
	Surrounding quotes are optional for strings except whitespace.

1.4.18.6 .foreach_part Macro

This macro iterates over the parts of a string divided by a delimiter using the special `%CurrentPart%` local variable.

```
.foreach_part (<input> [,"<delimiter>" [,<head> [,<tail>]]])
    <output>
.next[(<separator>)]
```

The following parameters are available:

Parameter	Description
`<input>`	Specifies the text over which to iterate. Type: Simple template
`<delimiter>`	Specifies the sub-string that divides the input into parts. You can specify multiple characters including ranges. For example `[A-Z]` specifies that any capital letter acts as a delimiter. By default, the delimiter is set to `' -_,\t'` (space, dash, underscore, comma, or tab). **i Note** The delimiter must be surrounded by single quotes if it contains a space. Type: Text
`<head>`	[optional] Specifies text to be generated before the output, unless there is no output. Type: Text
`<tail>`	[optional] Specifies text to be generated after the output, unless there is no output. Type: Text

Parameter	Description
<output>	Specifies the text to output for each part in the input. Type: Complex template
<separator>	[optional] Specifies text to be generated between each part of <output>. Type: Text

For example:

Examples

This template is applied to **My class**:

```
.foreach_part (%Name%)
%.FU:CurrentPart%
.next
```

Result:

```
MyClass
```

This template is applied to **My class**:

```
.foreach_part (%Name%,' -_',tbl_)
%.L:CurrentPart%
.next(_)
```

Result:

```
tbl_my_class
```

This template is applied to **MyClass**:

```
.foreach_part (%Name%,[A-Z])
%.L:CurrentPart%
.next(-)
```

Result:

```
my-class
```

1.4.19 Spell Checking

You can spell check your PowerDesigner models if you have MS Word 2000 or higher installed on your machine, and if your system code page is consistent with the model language. For example, for the spell checker to work properly on a Chinese model, your system code page should be Chinese.

Context

> **i Note**
>
> You must enable the Spell Checker add in before use. To do so, select *Tools* > *General Options*, click the Add-Ins category, and select the Spell Checker add in in the list, then click *OK* to return to your model. To set preferences, select *Tools* > *Spell Checking Options* and choose the model objects properties for which you want to check the spelling. If you select *Check sub-objects*, then all the child objects of any object you check will also be checked. You can also specify the dictionary language to use when spell checking.

Procedure

1. Right-click an object or a model node in the Browser and select *Spell Check*. If an error is found the Spell Checking dialog opens.
2. For each error, you can:
   - Click *Change* to accept the suggested replacement word.
   - Type your own replacement and click Change
   - Click *Change All* to apply the change to the entire object or model.
   - Click *Add* to add the word to your custom dictionary

   A message is displayed to inform you that spell checking is successful.

1.4.20 Extending Objects

PowerDesigner includes powerful tools to extend and customize your modeling objects. Many extensions are defined in the resource files that are provided with PowerDesigner to support a particular language or DBMS, and you can add your own to add new attributes and attribute lists to objects, create entirely new objects, and customize object generation.

> **i Note**
>
> Many extended attributes are displayed in standard or custom property sheet tabs, but if any are not so assigned, an *Extended Attributes* tab will be displayed, listing all the extended attributes in alphabetical order and giving their datatypes, values, and a column, *[R]edefined*, in which the checkbox is selected if you have modified the attribute's default value.

For detailed information about working with extensions, see *Customizing and Extending PowerDesigner > Extension Files* .

1.4.20.1 Adding New Properties to an Object

To quickly add new attributes and collections to an object or to customize its property sheet, use the commands available from the property sheet menu. Extensions added to an object are immediately available for all other objects of that type in the model.

1.4.20.1.1 Adding a New Attribute to an Object

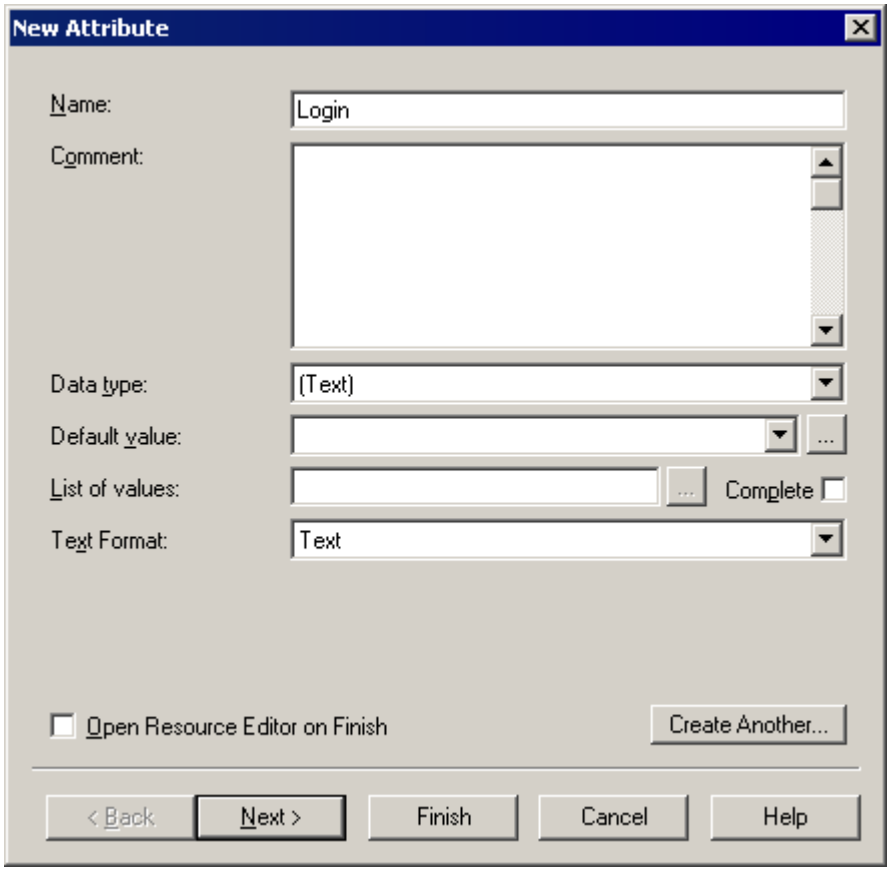
The PowerDesigner metamodel specifies the attributes available by default for an object (for example, name and comment). To quickly extend the metamodel and add a new attribute from an object property sheet, click the menu button, and select *New Attribute*. The new attribute is immediately available on all objects of this type.

Context

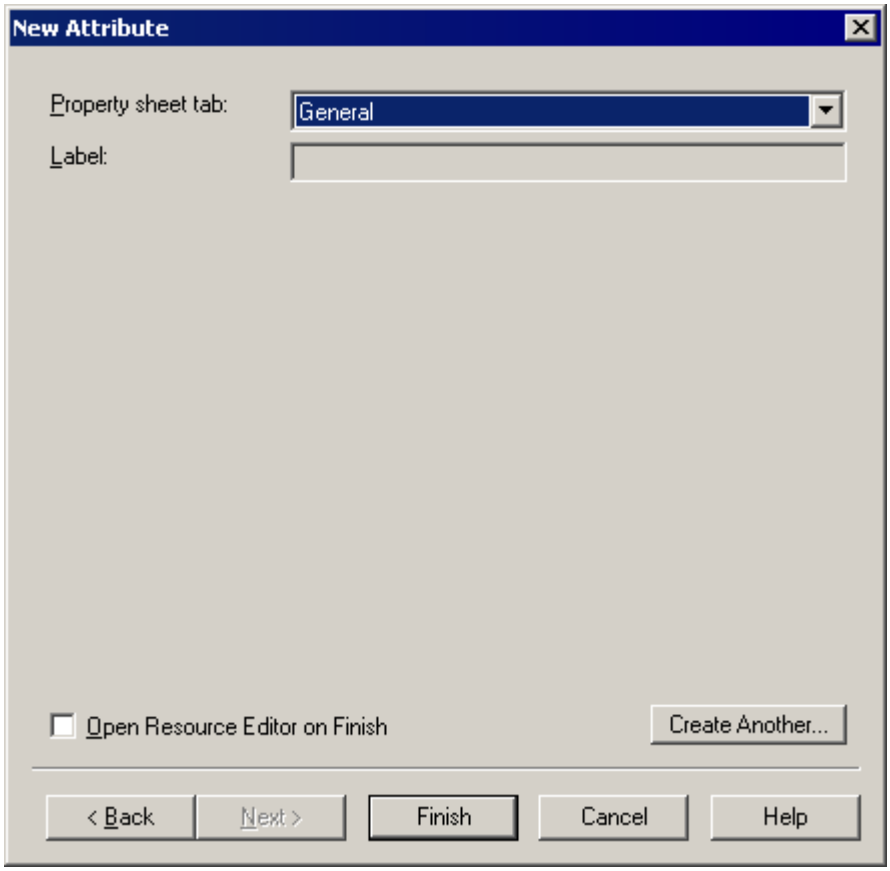
> **i** Note
>
> The definition of the new attribute is stored in a PowerDesigner extension file. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files* .

Procedure

1. Open the property sheet of the object you want to extend, click the menu button, and select *New Attribute* to open the New Attribute dialog:



2. Enter a *Name* and *Datatype* for the new attribute and complete any other appropriate fields.
3. [optional] Click *Next* to specify the property sheet tab on which you want the attribute to appear. If you choose to create a new property sheet tab, you must enter a tab label to display.



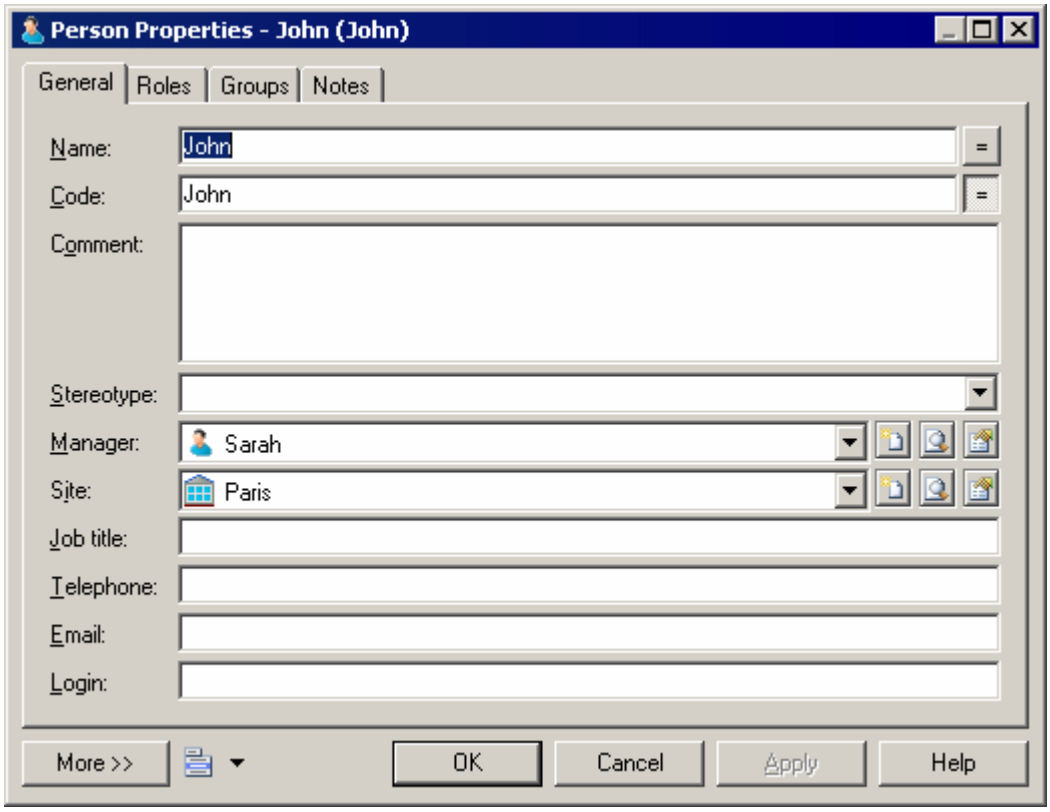
By default, the attribute is inserted on the tab from which you launched the New Attribute dialog, if the tab is editable.

4. [optional] If you want to view the new attribute in the PowerDesigner Resource Editor, select the *Open Resource Editor on Finish* checkbox.

   For information about working with the Resource Editor, see *Customizing and Extending PowerDesigner > PowerDesigner Resource Files*

5. Click *Finish* to close the dialog and create the new attribute.

   The attribute appears on the specified property sheet tab:



1.4.20.1.2 Adding a New Collection to an Object

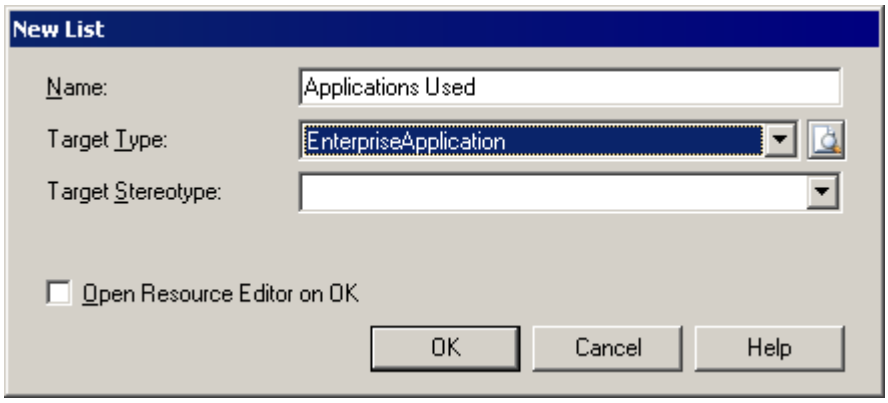
The PowerDesigner metamodel specifies the collections of related objects available by default for an object (for example table columns or class attributes). To quickly extend the metamodel and add a new collection from an object property sheet, click the menu button, and select *New List of Associated Objects*. The new collection is immediately available on all objects of this type.

Context

> **i** Note
>
> The definition of the new collection is stored in a PowerDesigner extension file. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files* .

Procedure

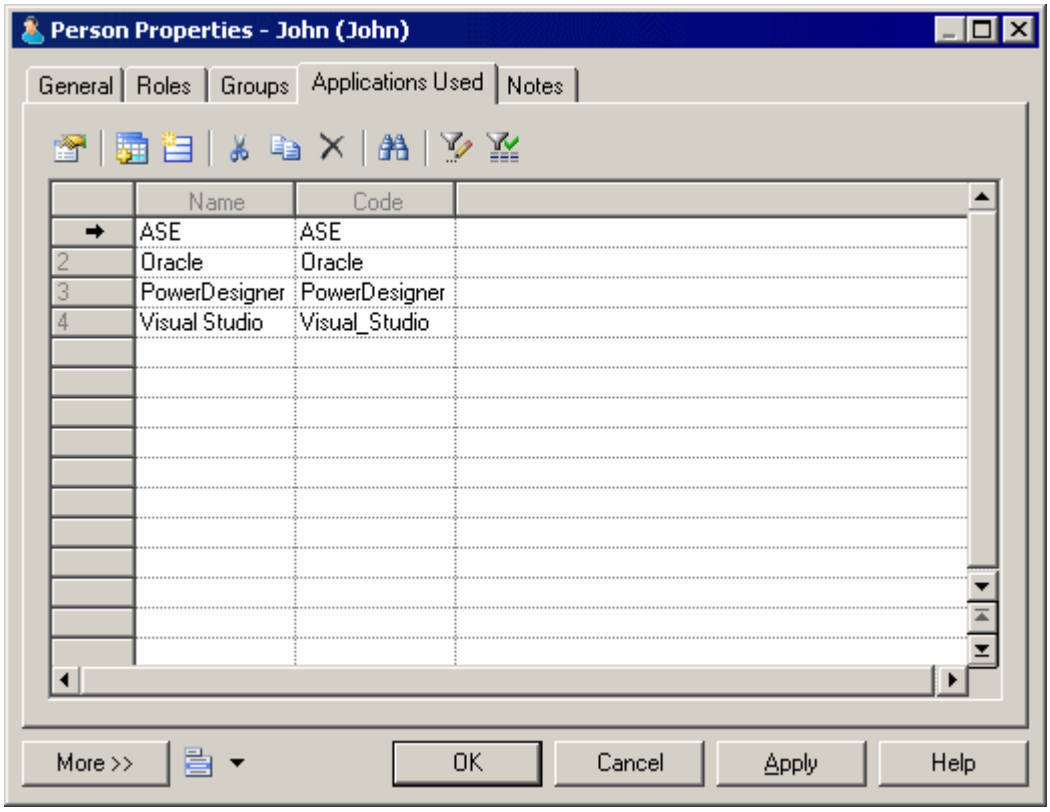
1. Open the property sheet of the object you want to extend, click the menu button, and select *New List of Associated Objects*.



2. Enter a *Name* for your list. This name is used as the name of the property sheet tab on which the list appears.
3. Select the type of associated object that you want your list to display. You can choose from any of the object types available in your model.
4. [optional] To limit the objects of the selected type that your object can be associated with, specify a stereotype that objects must bear in order to appear in the list.
5. [optional] If you want to view the new collection in the PowerDesigner Resource Editor, select the *Open Resource Editor on OK* checkbox.

   For information about working with the Resource Editor, see *Customizing and Extending PowerDesigner* > *PowerDesigner Resource Files*
6. Click *OK* to exit the dialog and create the new list.

   The list appears as a new tab in your property sheet, containing tools that let you associate existing and new objects of the specified type with your object:



1.4.20.1.3 Customizing an Object Property Sheet Tab

The PowerDesigner interface includes a standard property sheet for each type of object. To quickly customize a property sheet tab, click the menu button, and select *Customize Current Tab*. The customized tab is immediately available on all objects of this type.

Context

> **i** Note
>
> The definition of the customized tab is stored in a PowerDesigner extension file. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files* .

Procedure

1. Open the property sheet of the object whose property sheet you want to customize, select the tab to change, click the menu button, and select *Customize Current Tab*.

2. [optional] Modify the *Name* that will be displayed as the tab name.
3. Use the *Add Attribute* and *Add Collection* tools in the Form tab toolbar to add new properties to your form. You can reorder controls in the form control tree by dragging and dropping them.

   For detailed information about these and other form tools and properties click the *Help* button, or see *Customizing and Extending PowerDesigner > Extension Files > Forms (Profile)* .
4. Click *OK* to close the dialog and return to the modified property sheet.

1.4.20.1.4 Managing Extensions

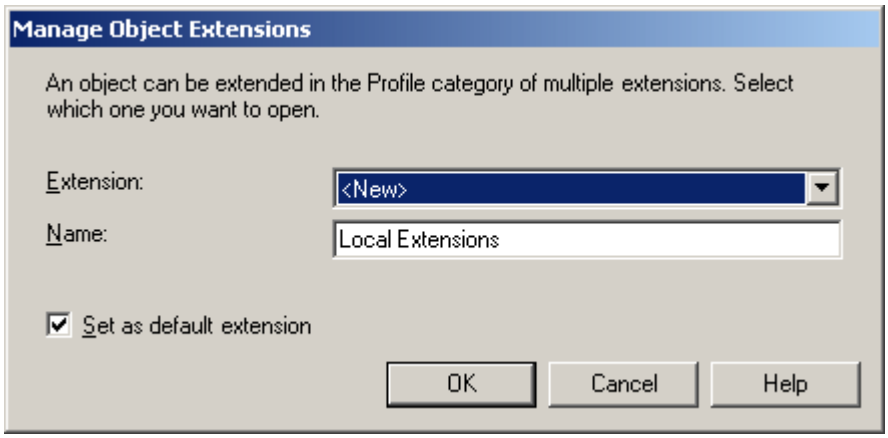
To review your object extensions in the Resource Editor, select *Manage Object Extensions* from the property sheet menu.

Context

You can create basic new properties and lists using the wizards available from the property sheet menu. To perform more advanced editing of extensions, you must open the appropriate extension.

Procedure

1. Open the property sheet of the object, click the menu button, and select *Manage Object Extensions*.



2. Select the *Extension* file that you want to open from the list of files attached to the model, or select **<New>** to create a new embedded extension file.
3. [optional] Select the checkbox to set the chosen extension file as the default. The default extension file is used to contain the extensions created from the property sheet menu
4. Click *OK* to open the chosen extension in the Resource Editor.

   For detailed information about editing these files, see *Customizing and Extending PowerDesigner > Extension Files* .

1.4.20.2 Customizing the Generation of Files for an Object

In PowerDesigner, you can define a generated file on a selected metaclass (or stereotype or criterion), in this case a file is generated for each instance of the metaclass existing in your model.

The generated files mechanism is defined in *Customizing and Extending PowerDesigner > Extension Files > Generated Files (Profile)* .

You can modify the default generation of files using artifacts. Artifacts are used to generate files only for selected instances of a metaclass in order to:

- Design a source file that includes the code of several objects in a single file.
- Generate only for selected instances of a metaclass.
- Customize the generated file name and path.

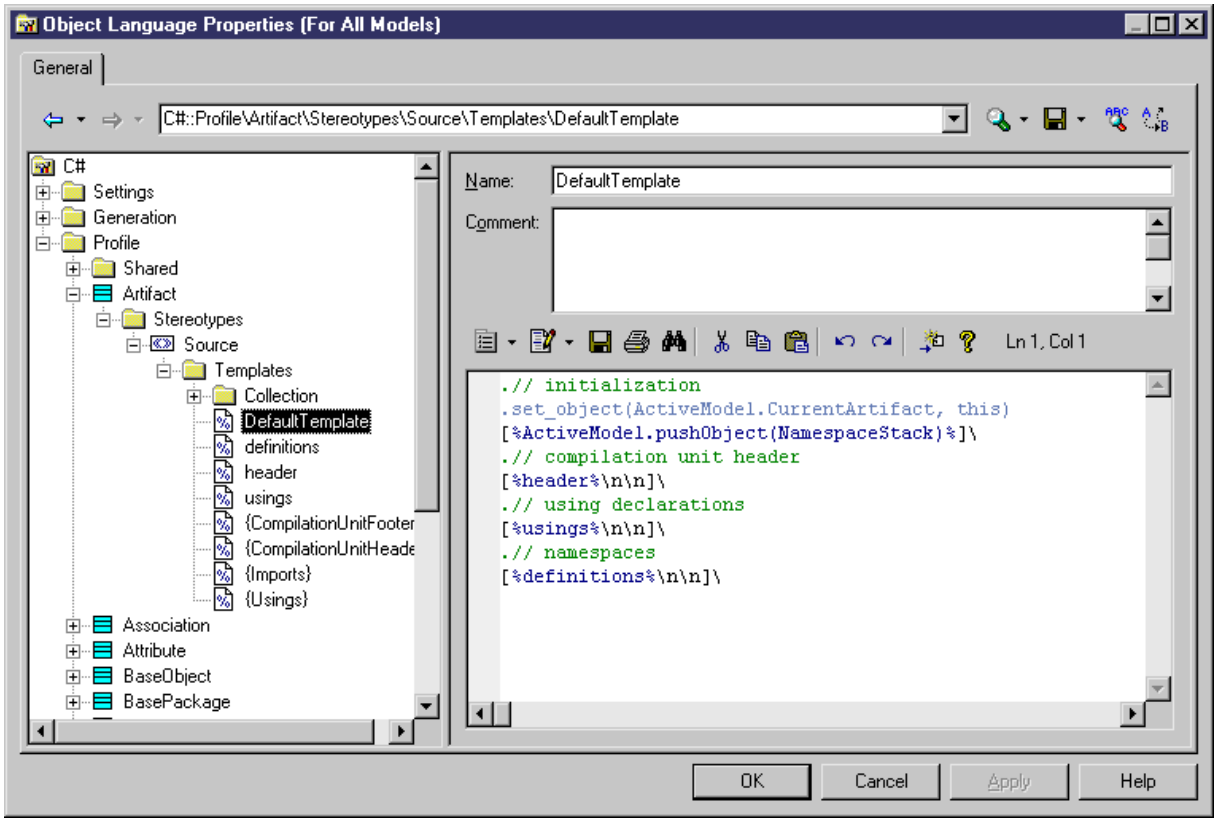
C# and VB .NET Reverse Engineering Use Case

Artifacts are visible in the C# and VB .NET object languages. This is to support round-trip engineering for these languages: when you reverse engineer C# or VB .NET code, each source file in the source code becomes an artifact in PowerDesigner. The artifact allows you to re-generate the same collection of objects while preserving file structure.

1.4.20.2.1 Using Artifacts in a Model

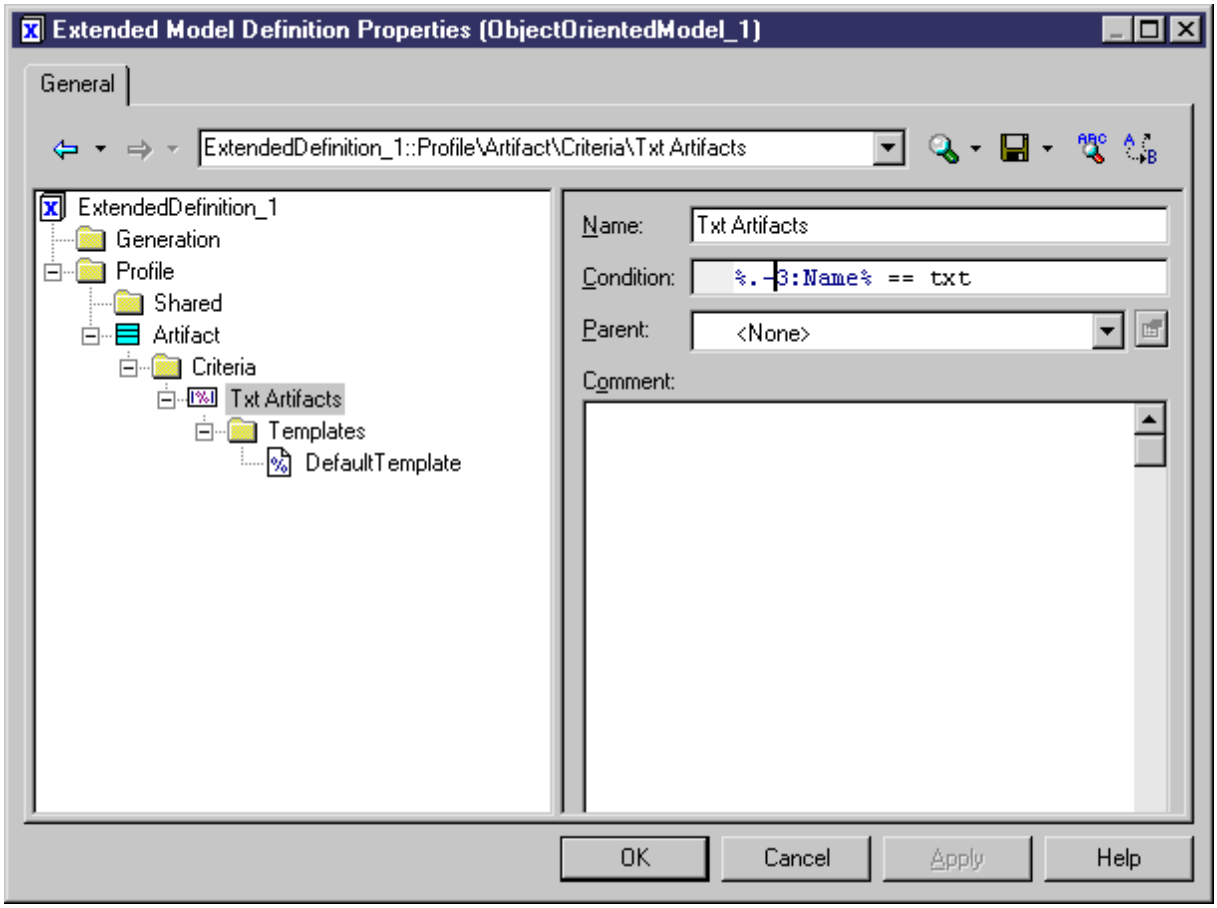
To be able to use an artifact in a model, you have to declare it in a resource file attached to your model. You must create a template called "DefaultTemplate" under an artifact stereotype or criterion bearing the name of the corresponding file type to generate.

In the C# resource file, if you expand the Artifact folder, you can see that the stereotype "Source" and the template "DefaultTemplate" are defined by default. This means that when you create a new artifact and assign the Source stereotype, this artifact inherits the default template defined below:



Example

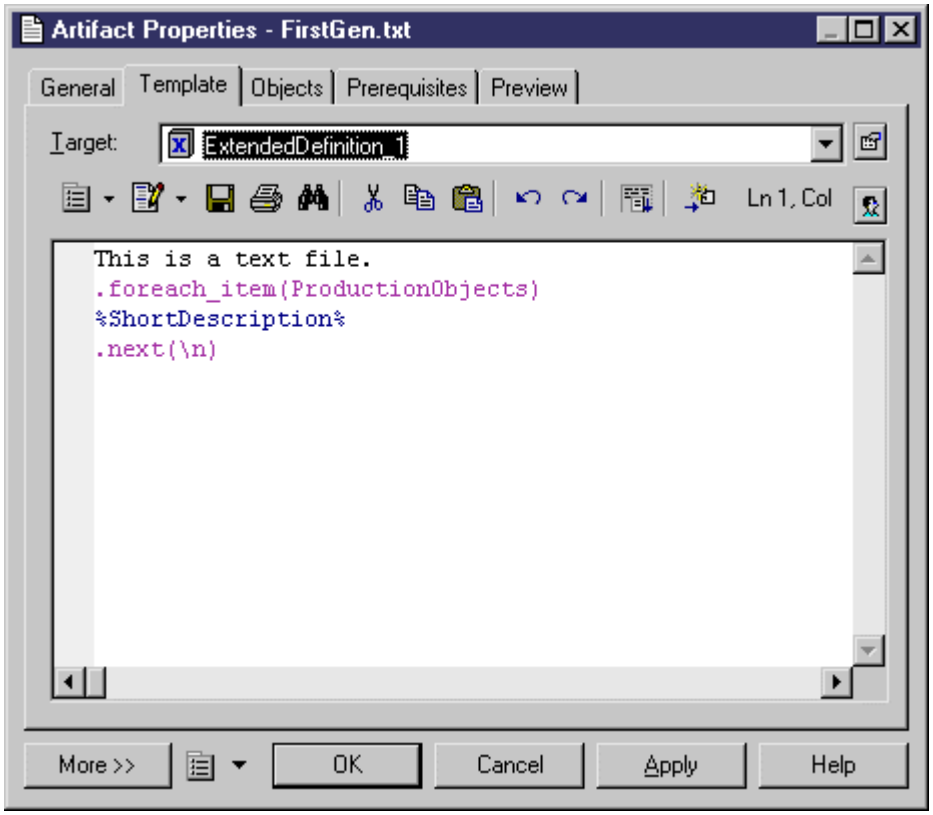
You define the following artifact criterion in an extension:



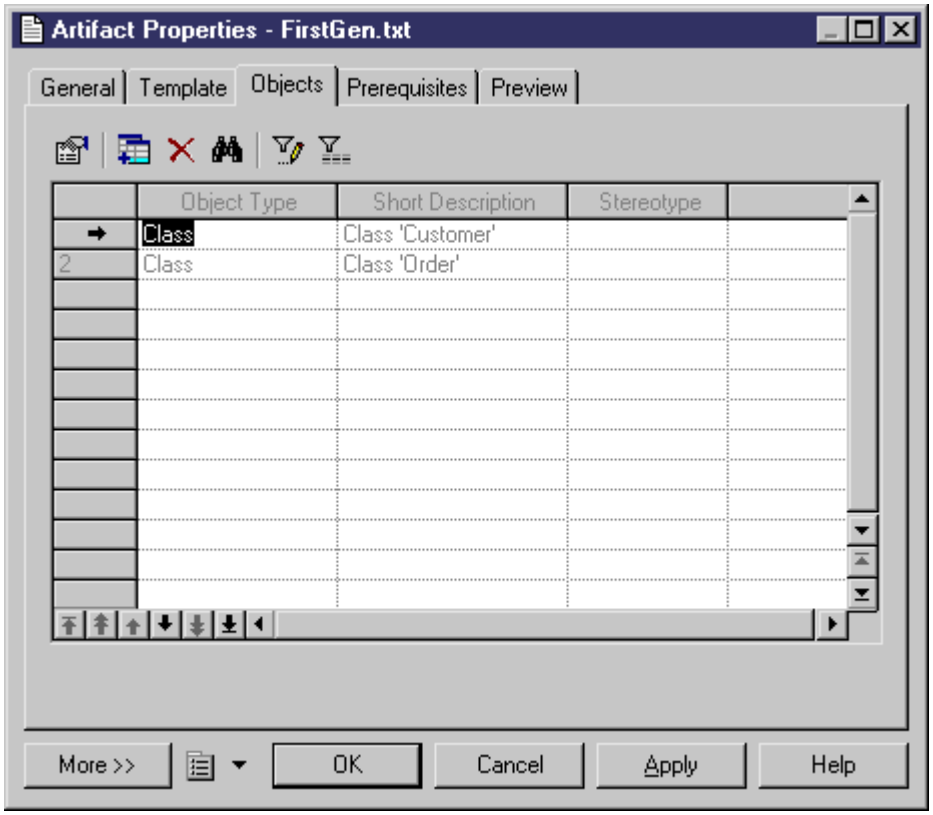
This criterion verifies that the last 4 characters of the artifact name are .TXT. You also need to create the default template in order to define the content of the generated file. In this example, DefaultTemplate is defined as follows:

```
This is a text file.
.foreach_item(ProductionObjects)
%ShortDescription%
.next(\n)
```

You can now create artifacts in the model using the *New* > *Artifact* command in the model contextual menu. If you create an artifact with the .TXT extension and select the correct target in the Template tab of the artifact property sheet, the default template is automatically assigned to this artifact. This means that the short description of each instance of object associated with the current artifact will be written in the generated file:



You can now define instances of objects you want to include in the file generated from the current artifact:



1.4.20.2.2 Defining an Artifact

Artifacts can replace the standard generation of files; by default, an artifact generates the same code as the generated file but only for selected instances of a metaclass.

Context

An artifact has the following properties:

Property	Description
Name	Name of the artifact.
Comment	Descriptive comment for the artifact.
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined.
Encoding	Allows you to modify the default file encoding of the files to reverse engineer.

An artifact definition also includes the following properties:

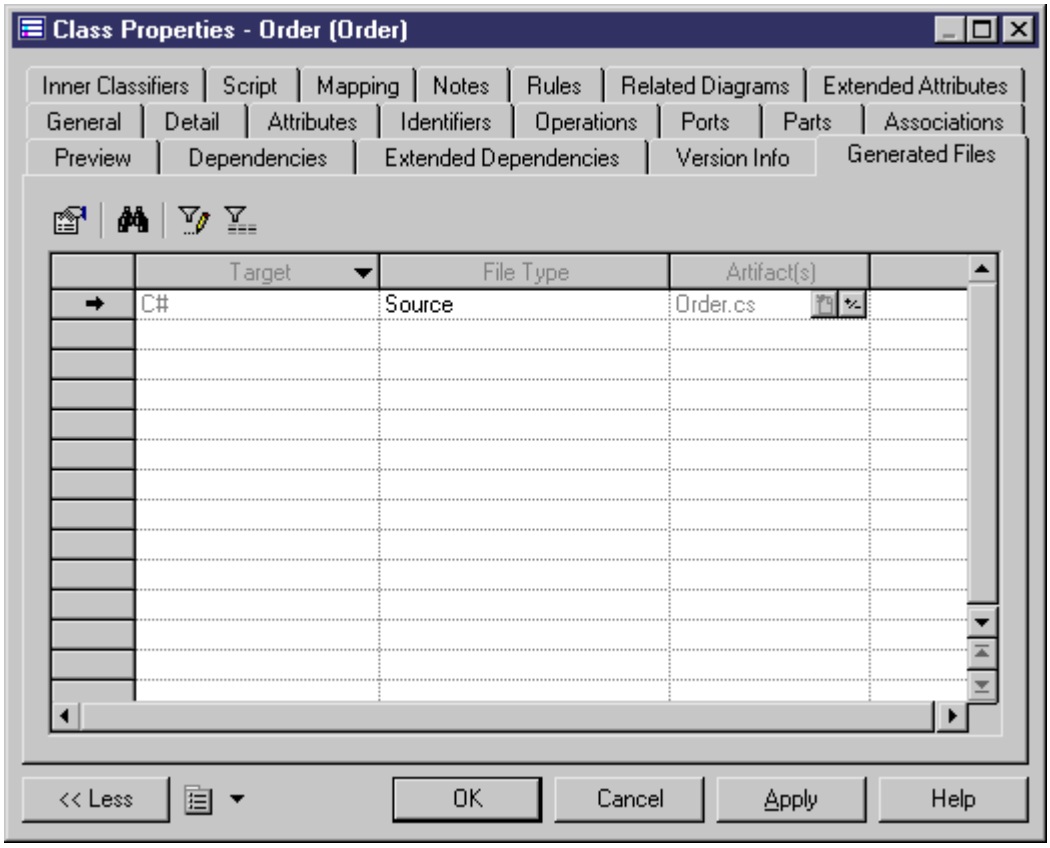
Property	Description
Template	Template used to generate the content of the generated file.
Objects	List of objects associated with the current artifact.
Prerequisites	List of artifacts that must be generated before the current artifact.
Preview	Allows to visualize the generated code of the artifact.

Procedure

1. Double-click the symbol of an object with generated files, for example a class in C# language.
2. Click the Generated Files tab.

   The Target column displays the resource file where the generated file is defined and the File Type column displays the type of the generated file.

   Each row in the list corresponds to a generated file type available for the current instance of a metaclass, you can customize the generation of this file using an artifact.
3. Click the Create tool in the Artifact(s) column to add an artifact. This artifact will replace the standard file generation for the current object.

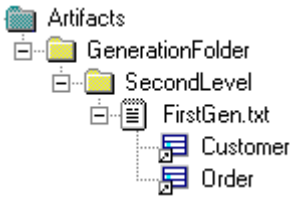


4. Click Apply and click the Properties tool to define artifact properties.
5. Click OK in both property sheets.

1.4.20.2.3 Managing Artifacts

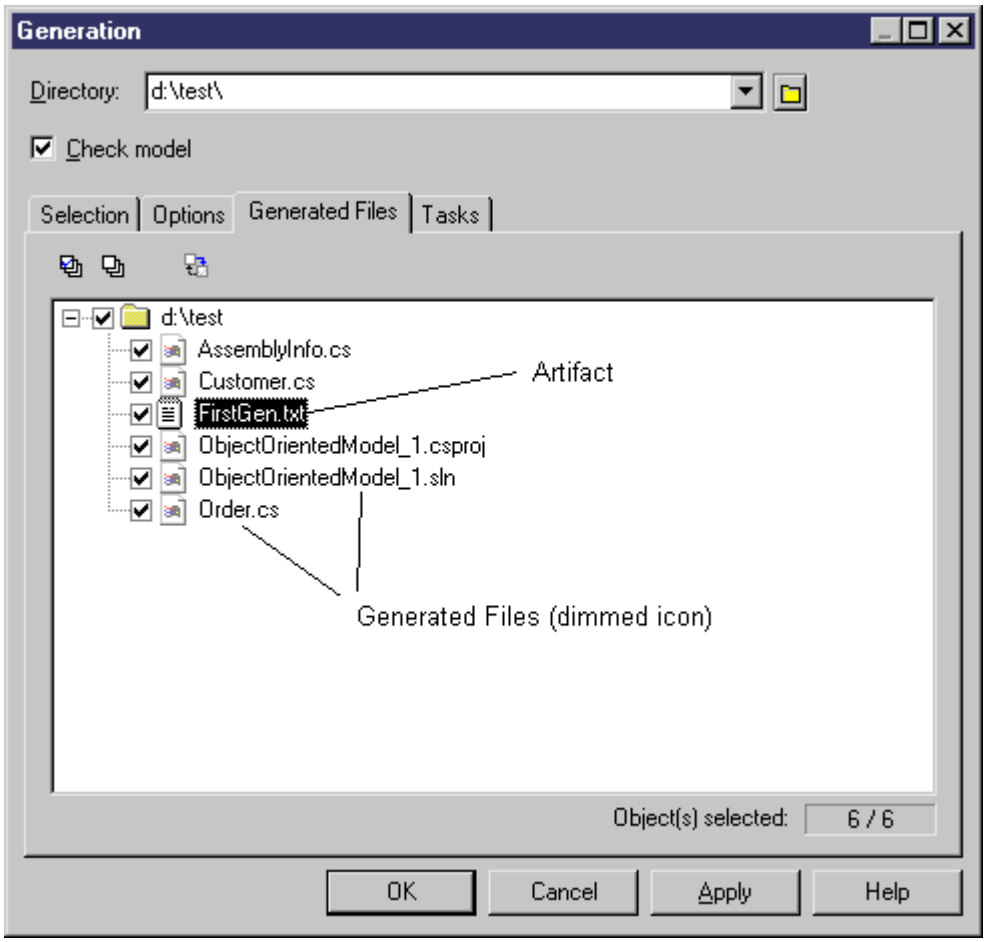
Artifacts appear in the Artifact category of the Browser. You can perform the following actions to customize an artifact. You can drag and drop authorized objects from the diagram or the Browser onto an artifact in order to add the code of this instance to the artifact.

You can use folders to create an artifact hierarchy for generation by right-clicking the Artifact category or a sub-folder and selecting *New* > *Artifact Folder*. You can create as many folder levels as necessary and create or place artifacts at any level.



1.4.20.2.4 Generated Files Tab

In the Generation dialog box, the Generated Files tab displays a checkbox tree with generated files (with a dimmed icon) and artifacts. You can select or deselect the files you want to generate.

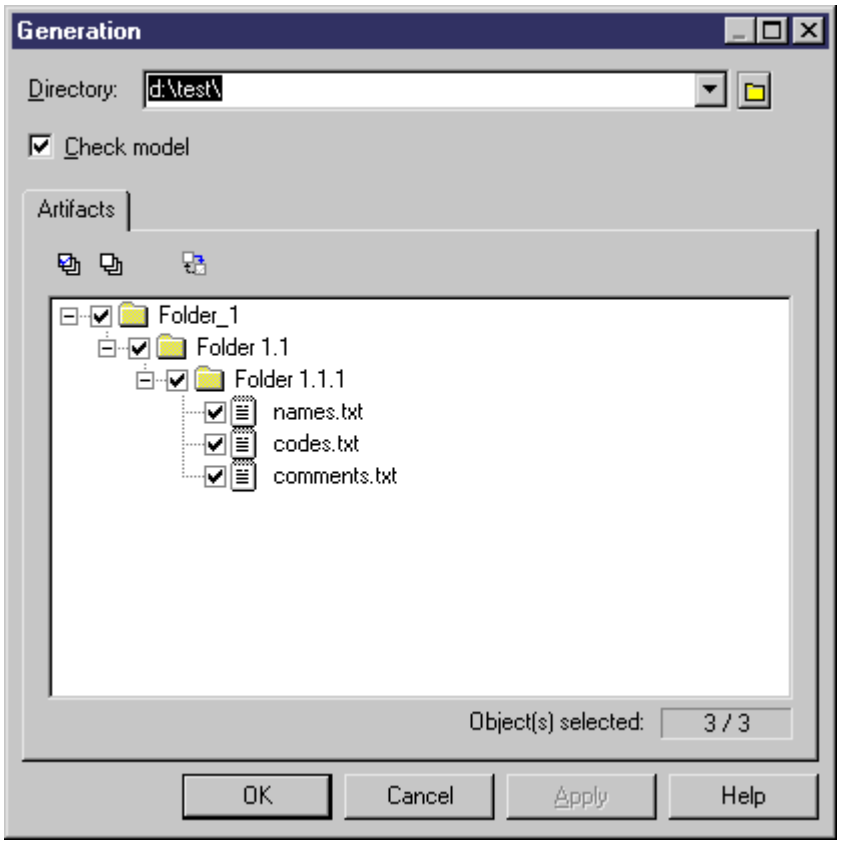


You can modify generation options from this dialog box, and you can also check artifact generation completeness: if an exclamation mark is displayed on the artifact icon it means that one or several prerequisite artifacts are missing.

If you click the Enforce Dependencies tool in the upper part of the dialog box, the artifacts that are prerequisites of other artifacts are automatically selected in the tree view in order to properly generate artifacts.

Generate from an Artifact Folder

If you click the Generate command in the artifact folder contextual menu, a Generation dialog box is displayed to let you manage the generation of a given artifact folder. This dialog box also displays a checkbox tree with children artifacts of the selected artifact folder.



You can enforce dependencies in the Generation dialog box to make sure all prerequisites are selected for generation.

1.4.21 Applying Transformations to Objects

Transformations are used to perform generic modifications to your model objects. You can apply transformations to your model as a sort of design pattern. You can design a pattern using the transformation feature and play it in your model in order to modify objects. For example, in an OOM, you could use a transformation to convert all analysis classes with the `<<control>>` stereotype into components in order to add an implementation layer to your model.

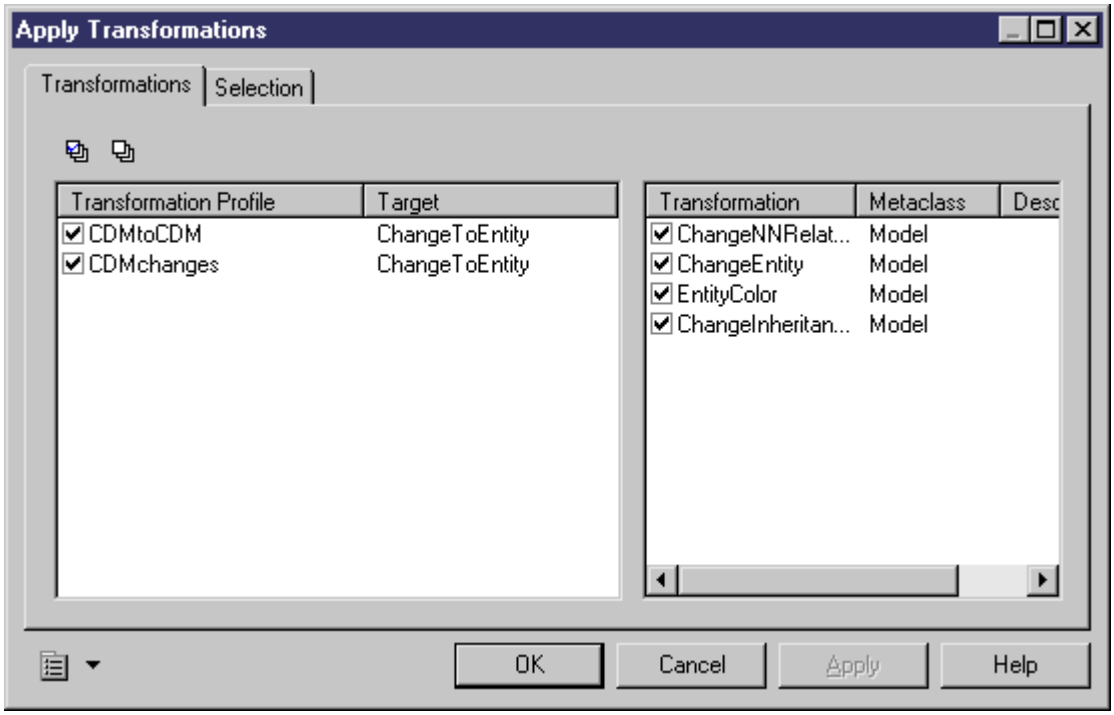
Context

In order to apply transformations, you must have either defined transformation profiles in your model's language definition file (for PDM, OOM, BPM, or XSM) or in an extension file (see *Customizing and Extending PowerDesigner > Extension Files > Transformations (Profile)*).

> **i Note**
>
> Pre- and post-generation transformations can be applied during model-to-model generation (see Applying Pre- and Post- Generation Transformations [page 430]). You can only invoke post-generation transformations outside of a generation.

Procedure

1. Select *Tools* > *Apply Transformations*.
2. Select the profiles and transformations to be applied. If you deselect a profile, none of its transformations will be executed. You can drag and drop profiles to modify the order of execution of transformations.



3. [optional] Click the *Selection* tab and deselect any objects that you want to exclude from the transformation.
4. Click *OK* to apply the transformations.

   The object tree of the model is traversed from the model object down and, for each type of object for which one or more post-generation transformation profiles is defined, each profile is applied in the order in which they appear on the tab, with each transformation being applied in the order in which they are defined in the profile.

1.4.22 Exporting Objects to Troux Semantics

The Troux Upload extension files provide a framework for mapping PowerDesigner metadata to Troux Semantics domains. An extension file with the export mechanism is provided for each type of PowerDesigner model along with instructions for how you can define mappings to export your model objects in the Troux Upload XML format. Example mappings are provided to map certain EAM objects to the Troux Application and Software domains

The extension files (along with developer and user documentation) are installed at `<install_dir>\Resource Files\Extended Model Definitions\Troux`, and should be copied into the parent `Extended Model Definitions` directory in order to be developed and used.

1.5 Diagrams, Matrices, and Symbols

Most PowerDesigner models contain diagrams, in which your model objects are represented by symbols. You can add a dependency matrix to any model to show relationships between objects.

1.5.1 Diagrams

A diagram is a graphical view of a model or a package. All models and packages have at least one diagram, and you can add additional diagrams as necessary. Objects can appear in multiple diagrams, and their symbols can show different kinds of information in each diagram.

For example, in a Publishing model, different diagrams could define the different activities involved in this industry: the printing diagram, the sales diagram, the accounting diagram, the book selection committee diagram.

When you save the model you also save all the diagrams it contains. You cannot save a diagram separately.

> **i** Note
>
> In requirements models (see *Requirements Management*), there are no diagrams, but rather views.

1.5.1.1 Creating a Diagram or a Diagram from a Selection

By default, any model or package opens with a default diagram. You can create as many diagrams as you want in a model or in a package. You can also select two or more symbols in an existing diagram and create a new diagram from your selection.

Context

You can create a diagram in any of the following ways:

- Select *View* > *Diagram* > *New Diagram* > `<Diagam type>`. If you have previously selected any object symbols, then the selected symbols will be copied into the new diagram. Alternatively, if you have selected two or more symbols in a diagram, you can right-click a selected symbol and choose *Create Diagram from Selection*.

  > **i** Note
  >
  > Since PowerDesigner objects can appear in multiple diagrams, the selected objects are not duplicated in the Browser, but they are now present in both the original and the new diagram.

- Right-click the background of your diagram and select *Diagram* > *New Diagram* > `<Diagram type>`.
- Right-click the model node in the Browser and select *New* > `<Diagram type>`.

In each case you will be invited to specify a name, code, and optional comment for the new diagram.

1.5.1.2 Diagram Properties

You can modify an object's properties from its property sheet. To open a diagram property sheet, right-click its Browser entry and select Properties from the contextual menu, or right click the diagram background and select *Diagram* > *Properties* from the contextual menu.

The *General* tab contains the following properties:

Property	Description
Name	Specifies the name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	Specifies 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 concerning the diagram
Parent	Specifies the name of the parent model or package

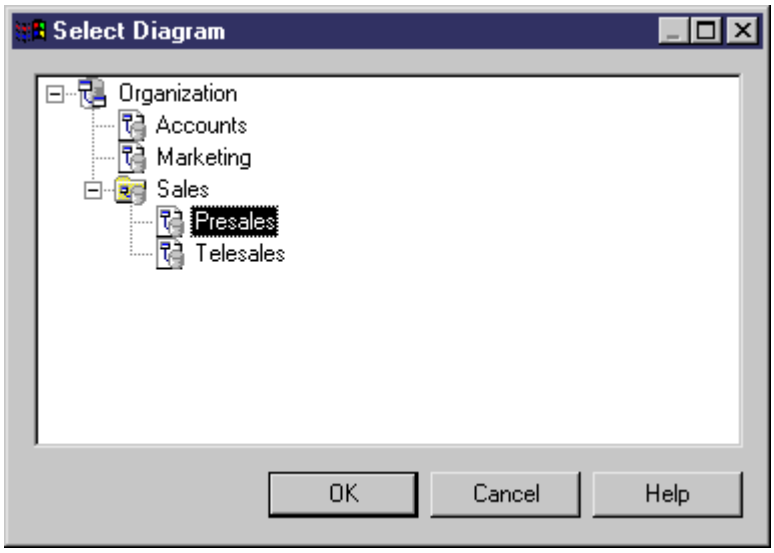
Property	Description
Stereotype	Specifies a stereotype for the diagram. For example, a statechart diagram can serve to model page flows in JSF and other web frameworks. You can use a profile to provide special processing for diagrams and other objects carrying stereotypes. For more information, see *Customizing and Extending PowerDesigner > Extension Files* .
Page scale	Sets a default display for page scale. The page scale percentage lets you define a page size according to your modeling needs. If you have a lot of objects on several pages, you can reduce the page scale percentage in order for all the objects to fit on a single printable page
Default Diagram	Diagram by default. This check box is automatically selected if the diagram is the first created diagram

1.5.1.3 Opening and Viewing Diagrams

You can open a diagram from the Browser or from the View menu, and move between open diagrams using the tabs above the canvas or by pressing Ctrl + Tab or Ctrl + Shift + Tab.

Context

- Double-click the diagram entry in the Browser.
- Press Ctrl + D, or select *View* > *Diagram* > *Select Diagram* to open the Select Diagram dialog box, select a diagram node in the tree and click *OK*.



- To access diagrams beneath package or other composite object symbols:
  - Select the *Open Diagram* tool from the Toolbox and click the symbol.

- Press Ctrl and double-click the symbol.
- Right-click the symbol and select *Diagram* > *Open Diagram*.

The following diagram viewing tools are available in the Toolbox or from the *View* menu:

Tool	Description
	Zoom In (F6) - Select the Zoom In tool and click anywhere in the diagram. The point clicked on is centered. Alternatively, you can select *View* > *Zoom In*, or turn your mouse scroll wheel away from you while holding the Ctrl key. To zoom in to a particular area, select the Zoom In tool and click and drag a rectangle around the area to be displayed. When you release the mouse button, the diagram zooms to the selected area.
	Zoom Out (F7) - Select the Zoom Out tool and click anywhere in the diagram. The point clicked on is centered. Alternatively, you can select *View* > *Zoom Out*, or turn your mouse scroll wheel towards from you while holding the Ctrl key.
	View the whole diagram (F8) - Double-click the Global View tool or select *View* > *Global View*.
[none]	View actual size (F5) - Select *View* > *Actual Size*.
[none]	View the current (printable) page (Ctrl + F10) - Select *View* > *Page View* > *Current Page*.
[none]	View all pages that contain symbols (F10) - Select *View* > *Page View* > *Used Pages*.
[none]	View all pages in the diagram - Select *View* > *Page View* > *All Pages*.
[none]	Center on selected symbols - Select *View* > *View Selection*.
[none]	Return to previous view (F9) - Select *View* > *Previous View*. This and the Next View option allows you to toggle back and forth between various selections and zooms you have used to navigate in your diagram, for example between a limited view and a global view of the diagram.
[none]	Go to next view (Shift + F9) - Select *View* > *Next View*.
[none]	Refresh View (Shift + F5) - Select *View* > *Redisplay*.

1.5.1.4 Finding an Object Symbol in the Diagram

You can locate any object with a symbol in a diagram or among several diagrams using Find in Diagram (or, for an RQM, Find in Document View). Objects without graphical symbol such as domains cannot be found in the diagram.

Context

If the object has only one symbol, the appropriate diagram is opened with the object symbol centered. If the object has several symbols, a dialog box opens to allow you to select one.

This feature can be very useful if you are looking for the target object of shortcut symbols, as you can access the target object from the shortcut property sheet and then locate the target object in the diagram.

Find in Diagram (or Find in Document View) is available from:

- The Browser — Right-click an object in the Browser and select Find in Diagram (or Find in Document View) from the contextual menu.
- The Result List — Right-click an object in the Result List and select Find in Diagram (or Find in Document View) from the contextual menu.
- The object property sheet dropdown menu — Open an object property sheet and select Find in Diagram (or Find in Document View) from the dropdown menu at the bottom-left corner.
- An objects list — From an objects list, select an object in the list and click the Find Symbol in Diagram tool in the list toolbar.

1.5.1.5 Specifying Diagrams as Related Diagrams

You can use related diagrams to further define the behavior and implementation of objects and to view them from different angles and in terms of semantic relationships. You can associate any type of diagram open in the workspace with an object, including diagrams from other packages or models.

To specify that a diagram is a related diagram of an object, open the property sheet of the object and click the *Related Diagrams* tab. If this tab is not visible, click the *More* button. The following tools are available:

Tool	Description
	Properties - Opens the property sheet of the selected diagram.

Tool	Description
	Open Diagram - Opens the selected diagram. **i Note** If the object is not composite (and so does not have any child diagrams), you can navigate to the first related diagram in the list from an object symbol by pressing *Ctrl* and double-clicking the symbol. To navigate to any related diagram in the list, including for composite objects, right-click the symbol and select *Related Diagram* > `<Diagram Name>`.
	Add Objects - Opens an object selection dialog to select existing diagrams and add them to the list.
	Create an Object - Creates a new diagram, and adds to the list as a related diagram. You must specify the location (model) and type of diagram to create and give it an appropriate name. **i Note** To create a new diagram and associate it to the object from the object symbol, right-click the symbol and select *Related Diagram* > `<New>`.

1.5.1.6 Deleting a Diagram

When you delete a diagram, you delete a view of a model or a package. This action does not affect the objects in the model or package.

Context

You can delete a diagram in any of the following ways:

- Select the diagram node in the Browser and press the *Del* key.
- Right-click the diagram window background and select *Diagram* > *Delete* from the contextual menu.
- Select *View* > *Diagram* > *Delete*.

1.5.1.7 Printing Diagrams

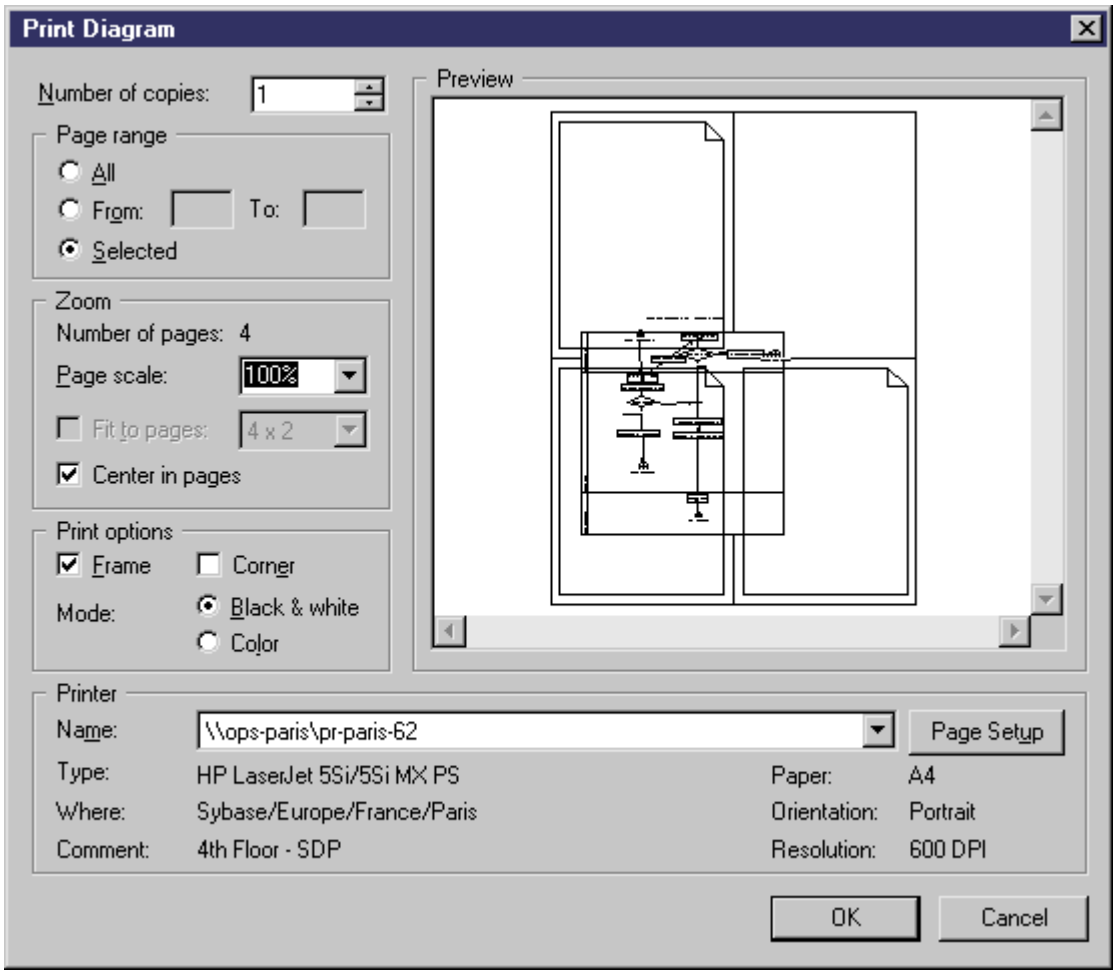
You can print the currently selected diagram at any time. You can print the whole diagram, a selection of pages, or a selection of objects.

Context

> **i** Note
>
> To print detailed information about model objects, create a model report (see Reports [page 295]).

Procedure

1. [optional] Select symbols in the diagram in order to print them and exclude the others.
2. Select *File* > *Print*, or click the *Print* tool to open the Print Diagram dialog, which displays default print options and the number of printed pages needed for the diagram.



3. [optional] Specify the pages to print in the *Page range* groupbox or by clicking in the *Preview* pane (see Print Diagram Options [page 192]). Only pages with an overlaid page frame will be printed.
4. [optional] Specify a page scale or set of pages to fit to (see Print Diagram Options [page 192]). By default, diagrams are printed at 100% scale on as many pages as necessary.
5. [optional] Click the *Page Setup* button to open the Page Setup dialog and specify your page layout (see Page Setup Options [page 193]).
6. Click *OK* to start printing.

1.5.1.7.1 Print Diagram Options

You can preview the selection to be printed and specify various options in the Print diagram dialog box.

Option	Description
Number of copies	Specifies the number of copies to print.
Page range	Specifies which pages to print. You can choose between: • All - Prints all the pages of the diagram. • From / To - Prints a range of pages covered by the diagram. The corresponding pages are framed in the Preview window. • Selected - Prints the pages you select in the Preview window.
Number of pages	Displays the number of pages required based on your zoom options.
Page scale	Specifies the scale at which the diagram will be printed.
Fit to pages	Reduces the scale of the diagram to print it on the number of pages specified, for example 1x3. The *Number of pages* option is updated and the *Center in pages* option is unavailable. To apply these changes to your diagram, select *Symbol* > *Fit to Page File* and *File* > *Page Setup* for paper orientation.
Center in pages	Centers the diagram in the pages that it covers. The *Fit to pages* option is unavailable.
Frame	Draws a solid line border around the image on all pages.
Corner	Specifies the printing of crop marks in each corner to help align multiple pages.
Mode	Specifies whether the diagram will be printed in Black & White or Color.
Printer	Select a printer from the Name list. Click the *Page Setup* button to modify printer parameters.

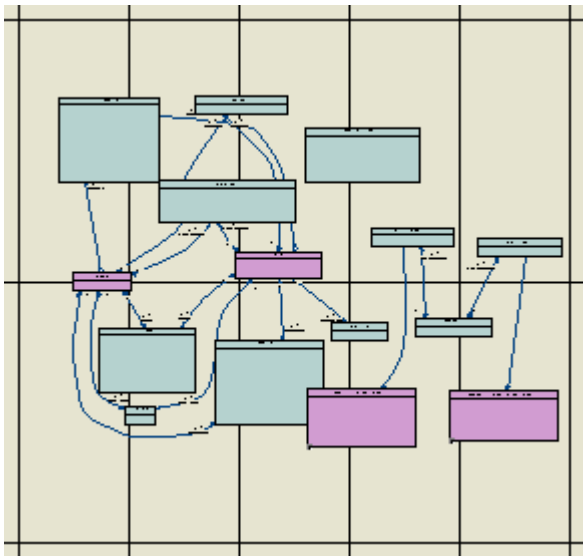
1.5.1.7.2 Page Setup Options

The Page Setup dialog allows you to modify standard printer parameters, such as Paper, Orientation and Margins, and to add various kinds of information to the Header, Footer and Page of your printed diagram.

You can arrive at the Page Setup dialog by selecting *File* > *Page Setup*, or by clicking the *Page Setup* button in the Print Diagram dialog (see Printing Diagrams [page 191]).

The following fields let you specify the content of the header and footer:

Option	Description
Header	Specifies the content of the page header. You can insert fixed text and use the arrow button to the right of the field to insert one or more of the following variables: • Current Date - inserts the date • Current Time - inserts the time • Current Page - inserts the content of the *Page* field • Last Page - inserts the number of the last page • Diagram - inserts the diagram name • Location - inserts the file path • Model - inserts the model name • Package - inserts the package name • Module - inserts the model type • Application Name - inserts "PowerDesigner"
Footer	Specifies the content of the page footer. You can insert fixed text and use the arrow button to the right of the field to insert the same variables as for the page header.

Option	Description
Page	Specifies the page number format to use in the header and footer. You can insert fixed text and use the arrow button to the right of the field to insert one or more of the following variables: • Page Row - inserts the diagram pagination grid row number. This can be useful to help assemble large diagrams printed on multiple pages. You may need to zoom out to see the diagram pagination grid:  • Page Column - inserts the diagram pagination grid column number • Page Number - inserts the standard page number

1.5.1.8 Converting a Diagram to a Package

PowerDesigner lets you convert a diagram to a package. You can move all the objects in the diagram to the new package or specify only certain objects. Other objects will stay in their original package and be represented via shortcuts in the new package.

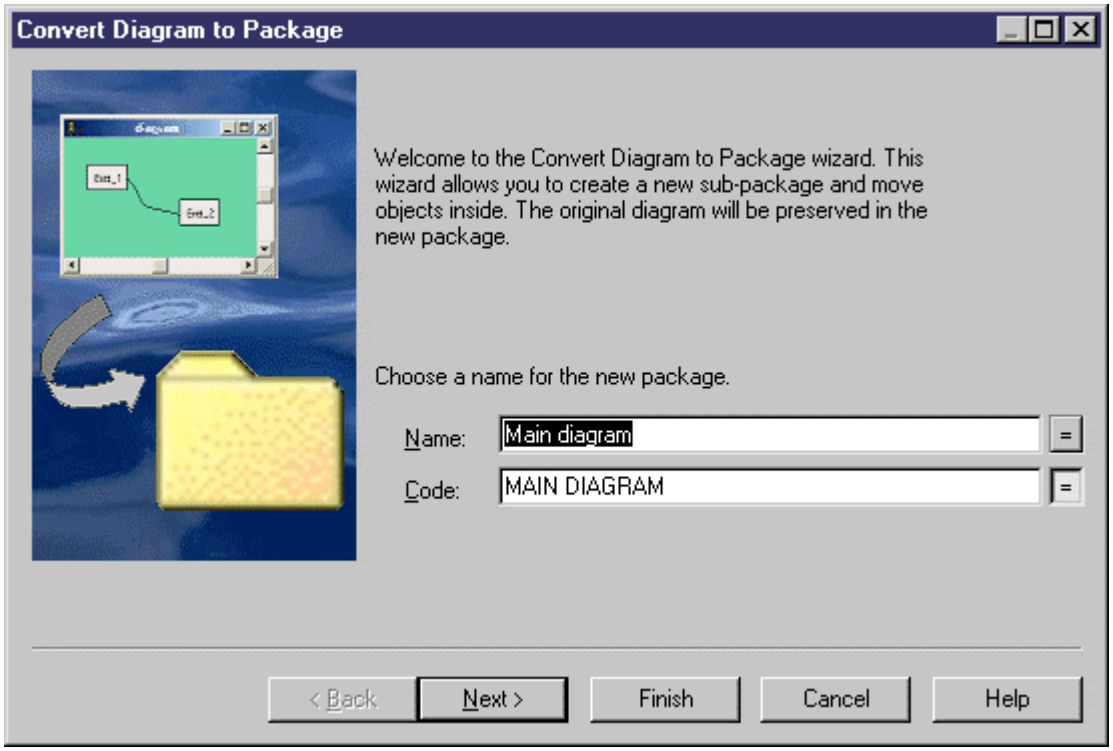
Context

The linking objects that you move keep their links in the target package and a shortcut is usually created in the source package. The general rule being that conceptual modeling must be preserved.

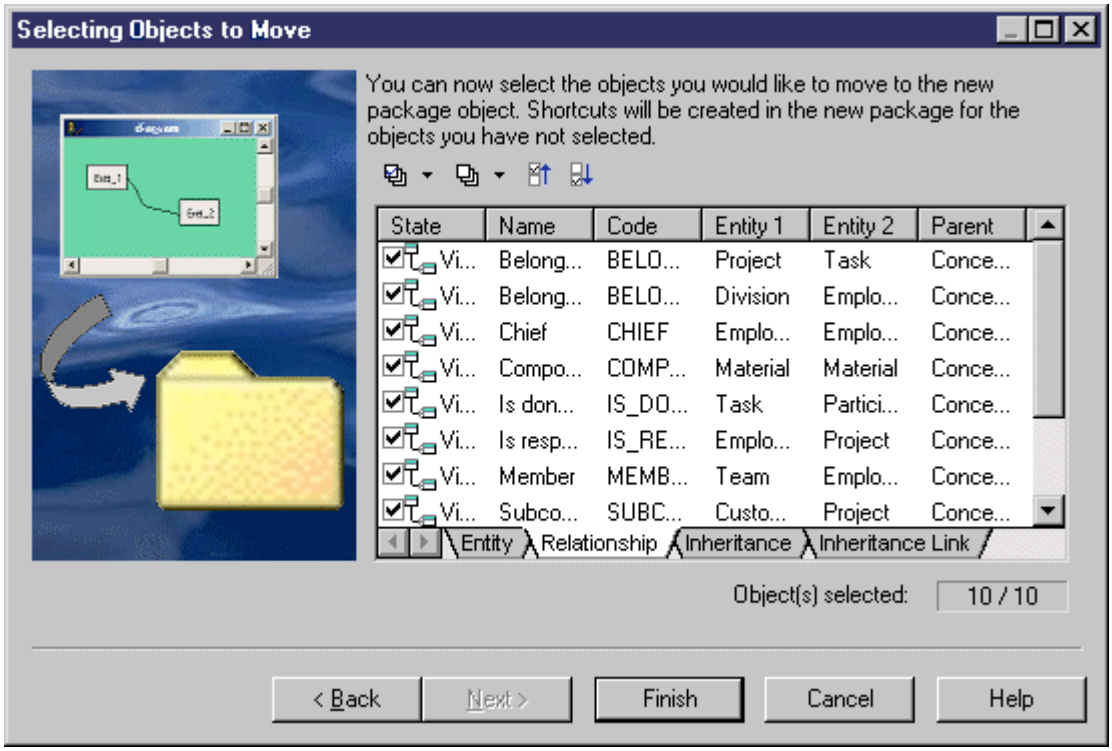
Shortcuts creation rules in PowerDesigner also apply to moving objects between packages (see Shortcuts and Replicas [page 443]).

Procedure

1. Select *View* > *Diagram* > *Convert to Package* (or right-click the diagram background window and select *Diagram* > *Convert to Package* or right-click the diagram node in the Browser and select *Convert to Package*) to open the Convert Diagram to Package wizard. By default the package takes the name of the diagram.



2. Click *Next* to open the Selecting Objects to Move page, which lists all the objects in the diagram available to move to the new package. Objects are organized by object type, with a sub-tab for each object type. By default, all the objects are selected.



3. [optional] Deselect any objects you do not want to move to the new package. Objects deselected here will remain in the original package and be represented in the new package via shortcuts.
4. Click *Finish* to create the new package and move the selected objects to it.

   The new package and diagram are added in the Browser.

1.5.1.9 Moving a Diagram to a Package

In some cases, you may want to move a diagram and some or all of the objects it contains into another package or composite object, such as a process or activity.

Context

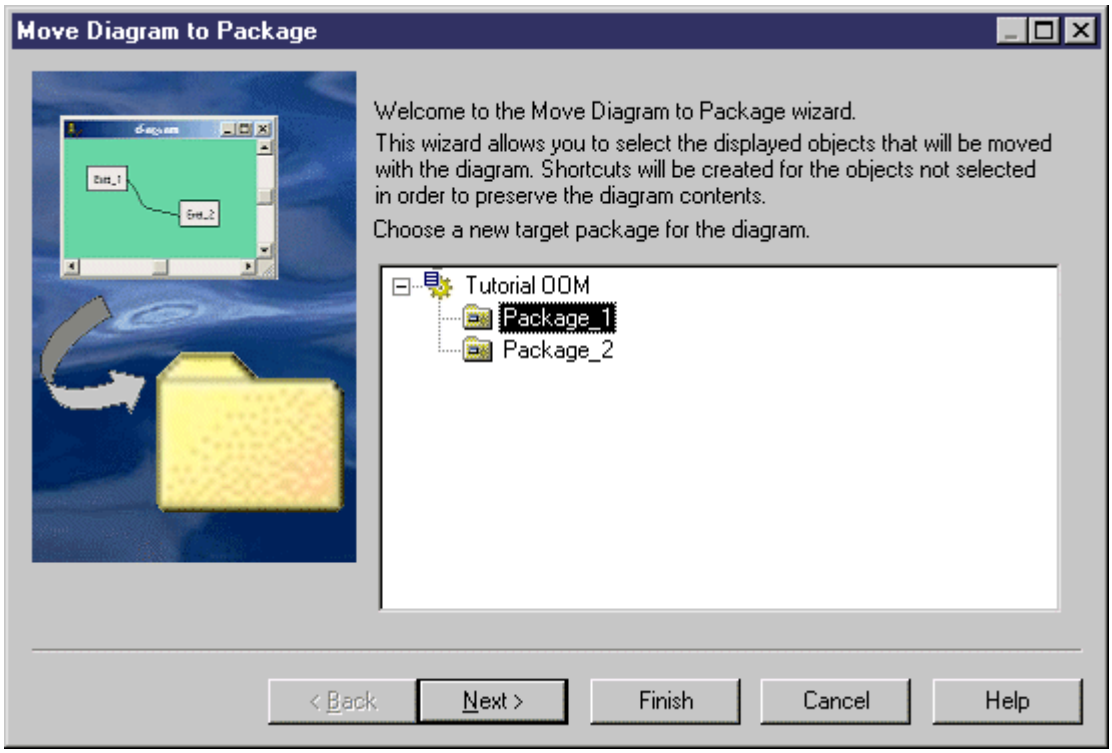
The linking objects that you move with the diagram keep their links in the target package and a shortcut is usually created in the source package. The general rule being that the design of the original diagram must be preserved.

Shortcuts creation rules in PowerDesigner also apply to moving objects between packages (see Shortcuts and Replicas [page 443]).

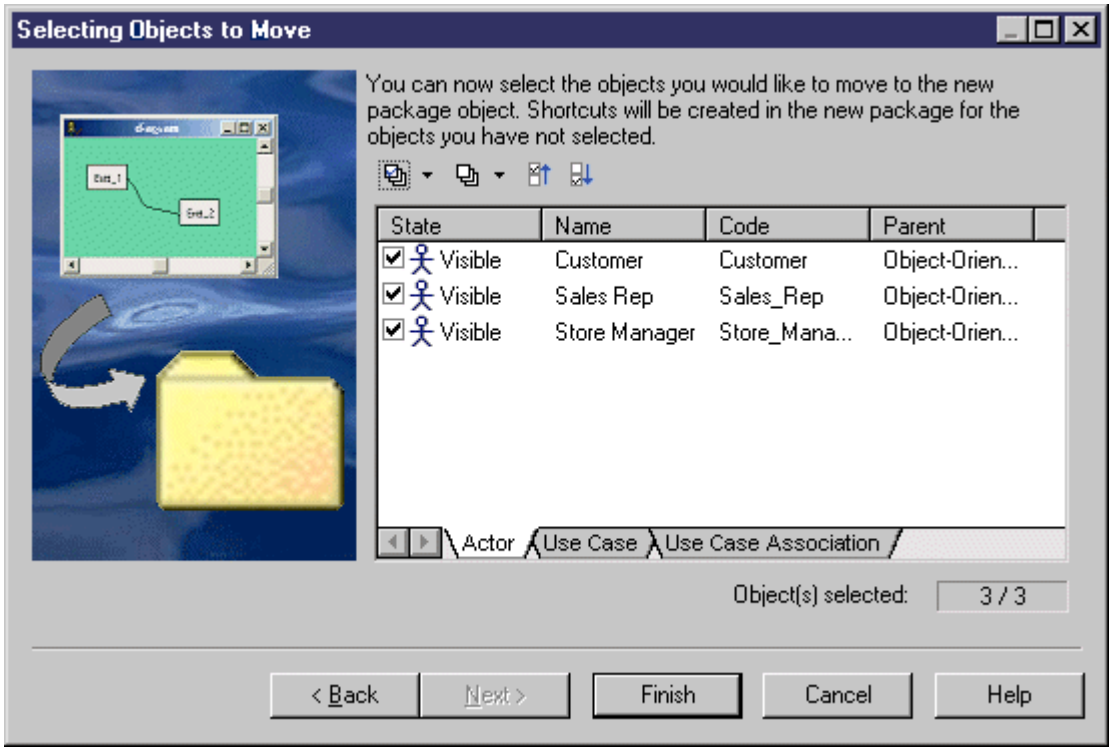
You can simply drag and drop the diagram from one package to another in the Browser to move all the objects in the diagram to the target package. To control which objects are moved, use the wizard.

Procedure

1. Select *View* > *Diagram* > *Move to Package* (or right-click the diagram background window and select *Diagram* > *Move to Package* or right-click the diagram node in the Browser and select *Move to Package*) to open the Move Diagram to Package wizard.



2. Select the target package to which you want to move the current diagram and click *Next* to open the Selecting Objects to Move page, which lists all the objects in the diagram available to move to the new package. Objects are organized by object type, with a sub-tab for each object type. By default, all the objects are selected.



3. [optional] Deselect any objects you do not want to move to the new package. Objects deselected here will remain in the original package and be represented in the new package via shortcuts.
4. Click *Finish* to move the diagram to the new package.

   If the last diagram is moved or deleted from a package, a new diagram is automatically created as all packages must contain at least one diagram.

1.5.1.10 Moving Entities Between Packages in a CDM

In a CDM, when moving entities containing data items from one package to another, different rules apply depending on where the data items are used.

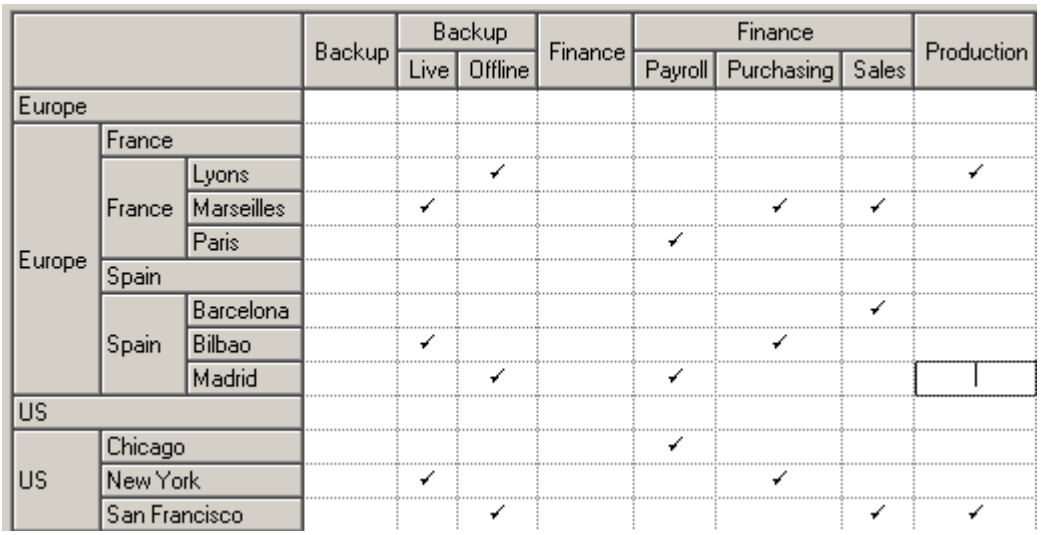
Data items	Namespace	Move result
Only used by selected entity	Move within the same name-space	The data items are moved with the entity.
Reused among different entities	Move within the same name-space	Shortcuts of data items are created for reused data items.
Used only by one entity or reused among different entities	Change namespace	The data items are copied in the other namespace.

1.5.2 Dependency Matrices

You can create dependency matrices to review and create links between any kind of objects. You can create an individual matrix from the Browser or define a matrix in a resource file for reuse.

Context

In the following enterprise architecture model (EAM) example, systems are listed along the top of the matrix, and linked to the sites where they are in use, via traceability links:

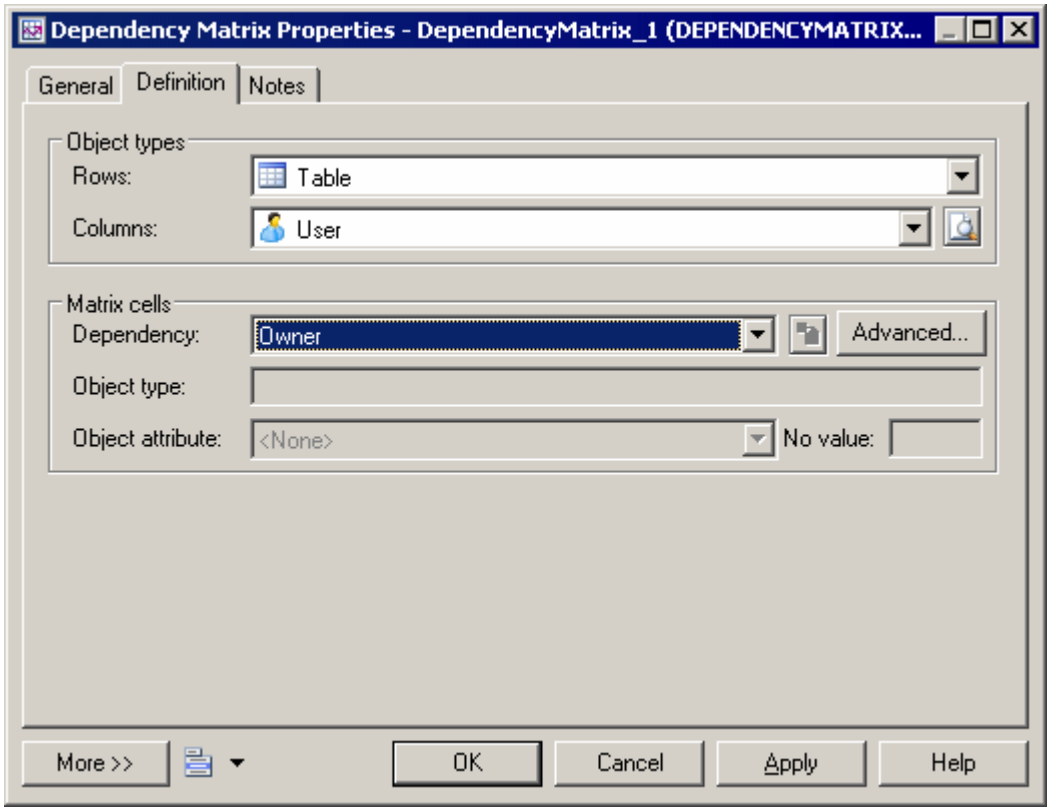


			Backup	Backup		Finance	Finance			Production
				Live	Offline		Payroll	Purchasing	Sales	
Europe										
Europe	France									
	France	Lyons			✓					✓
		Marseilles		✓				✓	✓	
		Paris					✓			
	Spain									
	Spain	Barcelona							✓	
		Bilbao		✓				✓		
		Madrid			✓		✓			
US										
US	Chicago						✓			
	New York			✓				✓		
	San Francisco				✓				✓	✓

Procedure

1. Right-click a model or package node in the Browser and select *New* > *Dependency Matrix* to open the matrix property sheet to the *Definition* tab.
2. Select an object type from the current model to populate your matrix rows and an object type from the current or another model type to populate the columns.
3. Specify how the rows and columns of your matrix will be associated by selecting a dependency from the list.

   In the following example, PDM tables are associated to users by the **Owner** dependency:



Only direct dependencies are available from the list. To specify a more complex dependency, click the *Advanced* button to open the Dependency Path Definition dialog (see Specifying Advanced Dependencies [page 201]).

4. For certain dependencies, the *Object type* on which the dependency is based will be displayed, and you can select an *Object attribute* to display in the matrix cells along with the *No value* symbol, which is displayed if that attribute is not set in any particular instance.
5. Click the *General* tab and enter a name for the matrix (for example **`Table Owners Matrix`**).
6. Click *OK* to complete the definition and open your matrix.

Results

> **i** Note
>
> For information about defining a dependency matrix in a resource file (such as a DBMS definition file, object language file or extension file) and for detailed information about its properties, see *Customizing and Extending PowerDesigner > Extension Files* .

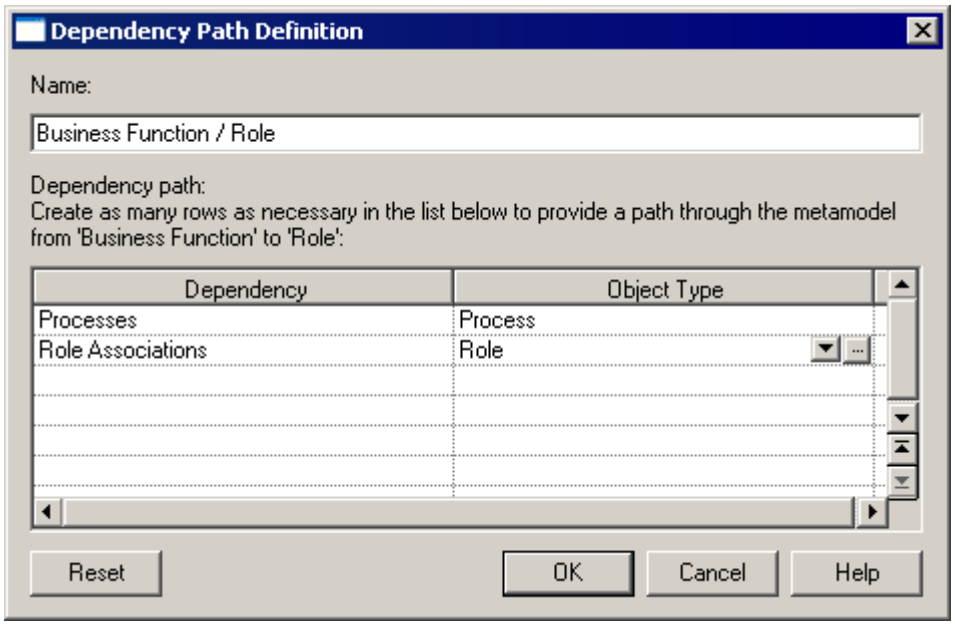
1.5.2.1 Specifying Advanced Dependencies

You can examine dependencies between two types of objects that are not directly associated with each other, using the Dependency Path Definition dialog, which is accessible by clicking the Advanced button on the Definition tab, and which allows you to specify a path passing through as many intermediate linking objects as necessary.

Each line in this dialog represents one step in a dependency path:

Property	Description
Name	Specifies a name for the dependency path. By default, this field is populated with the origin and destination object types.
Dependency	Specifies the dependency for this step in the path. The list is populated with all the possible dependencies for the previous object type.
Object Type	Specifies the specific object type that is linked to the previous object type by the selected dependency. This field is autopopulated if only one object type is available through the selected dependency.

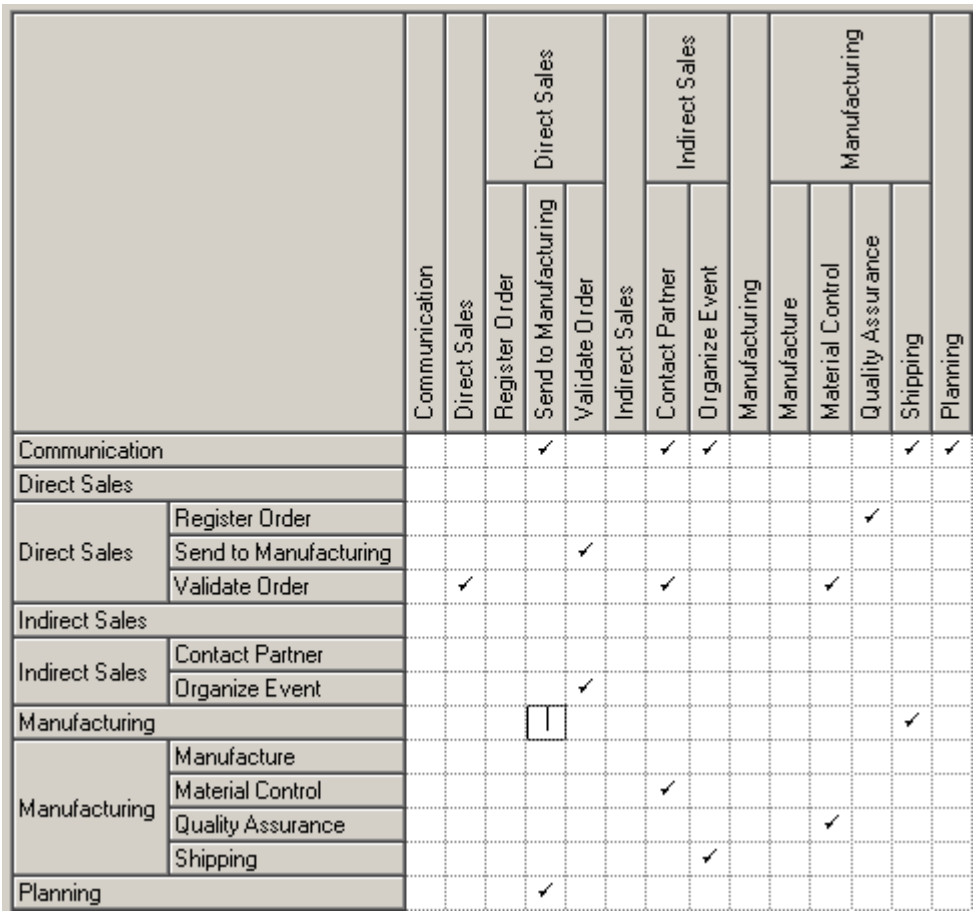
In the following example, a path is identified between business capabilities and roles, by passing from the business capability through the processes it contains, to the role linked to it by a role association:



1.5.2.2 Working with Dependency Matrices

The dependency matrix displays the connections between the objects of the types specified in your definition. You can add and delete connections in the matrix as well as filter the rows and columns displayed and print or export the matrix.

In the following example, the same hierarchy of business processes is shown in the row and column headers in order to analyze the dependencies between them:



		Communication	Direct Sales	Direct Sales			Indirect Sales			Manufacturing				
				Register Order	Send to Manufacturing	Validate Order	Indirect Sales	Contact Partner	Organize Event	Manufacturing	Manufacture	Material Control	Quality Assurance	Shipping	Planning
Communication					✓			✓	✓					✓	✓
Direct Sales															
Direct Sales	Register Order												✓		
	Send to Manufacturing					✓									
	Validate Order		✓					✓				✓			
Indirect Sales															
Indirect Sales	Contact Partner														
	Organize Event					✓									
Manufacturing														✓	
Manufacturing	Manufacture														
	Material Control							✓							
	Quality Assurance											✓			
	Shipping								✓						
Planning					✓										

You can modify the links in the following ways:

- To add a link, click in the appropriate cell, and then click the *Create link* button at the bottom of the matrix (or press the `Spacebar` or `V`).
- To delete a link, click in a cell containing a link, and then click the *Delete link* button at the bottom of the matrix (or press the `Spacebar` or `Del`).
- To select all the cells in a row or column, click the appropriate header. To select multiple cells, rows, or columns, use the `Ctrl` key. To select ranges, use the `Shift` key or drag the cursor over the required area.

The following tools are available above the dependency matrix:

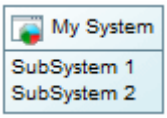
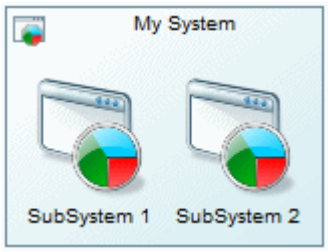
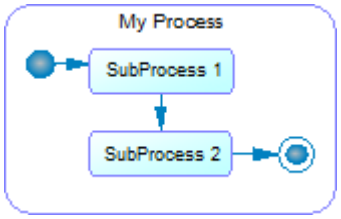
Tool	Description
	Properties – opens the property sheet of the object associated with the selected row, column, or cell.
	Copy – Copies the entire matrix for pasting into a CSV environment such as Excel or a plain text file.
	Select Rows/Columns (Ctrl + N) – Opens a selection box which allows you to select rows and columns to display in the matrix either by hand or by defining a filter (see Adding an Item from a Selection List [page 129]).
	Display only Non-Empty Rows/Columns (Ctrl + R) – Filters the display to show only rows/columns that are populated.
	Display only Empty Rows (Ctrl + E) – Filters the display to show only empty rows.
	Vertical/Horizontal Column Header - Toggles between vertical and horizontal orientation of column headers.
	Shrink to Fit - Shrinks row and column headers to fit their contents.
	Show Parents in Row Header - Displays the hierarchy of objects that are ancestors to the row objects.
	Show Parents in Column Header - Displays the hierarchy of objects that are ancestors to the row objects.
	Show Packages in Headers - Displays the hierarchy of packages that contain the row and column objects.
	Refresh - Refreshes the display of dependencies in the matrix.
	Export to Excel - Exports the matrix as an MS Excel file. If the specified file already exists, you will be given the option to overwrite it or append a new worksheet in the file.

> **i Note**
>
> You can modify the definition of the dependency matrix at any time by right-clicking its Browser node, selecting *Properties*, and clicking the *Definition* tab.

1.5.3 Symbols

PowerDesigner gives you wide control over the formatting and arrangement of your object symbols.

The following symbol modes are available:

Mode	Details
Icon: 	Most EAM and DMM objects support icon or picture symbols that display a name or other limited information beneath them. To switch between icon and detail mode, right-click the symbol and toggle the *Show Detail* command or press *CTRL+Q*. To change any object symbol to an icon, select a *Custom Shape* of type Bitmap in the Symbol Format or Display Preferences dialog.
Detail:  	All objects support detail symbols (rectangles or other shapes) that can contain extensive information about the object inside the symbol, including lists of sub-objects. To control the information displayed, use the *Content* tab in the Symbol Format or Display Preferences dialog.
Editable Composite:  	Some EAM objects (and CDM and LDM entities using the Barker notation) support editable composite symbols that allow you to create or drag and drop sub-objects on the symbol. To switch between composite and detail mode, right-click the symbol and use the *Composite View* submenu.
Read-Only Composite:  	Decomposed BPM processes and OOM activities and packages in all types of models support read-only composite symbols that display the object's sub-diagram. To descend into the sub-diagram for editing, press `Ctrl` and double-click the symbol. To go back up to the parent diagram, press `Ctrl` + `U`.

- To change the format or content of one or more individual symbols, select them, then right-click in the selection, and select *Format*. Use the commands *Get Format* and *Apply Format* to copy the format of one symbol to another.
- To change the format or content of all the symbols in a diagram, select *Tools* > *Display Preferences*.
- To arrange and manipulate groups of symbols, use the *Symbol* menu.

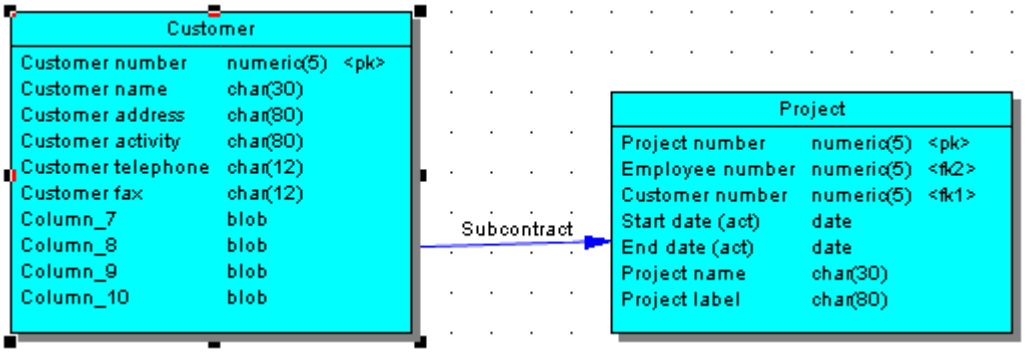
1.5.3.1 Manipulating Symbols

You can select symbols in a PowerDesigner diagram using standard gestures. You can edit properties of the selected symbol's object or it's sub-objects, and resize the symbol by clicking and dragging on its handles.

Click on a symbol with the default *Pointer* tool to select it. To select:

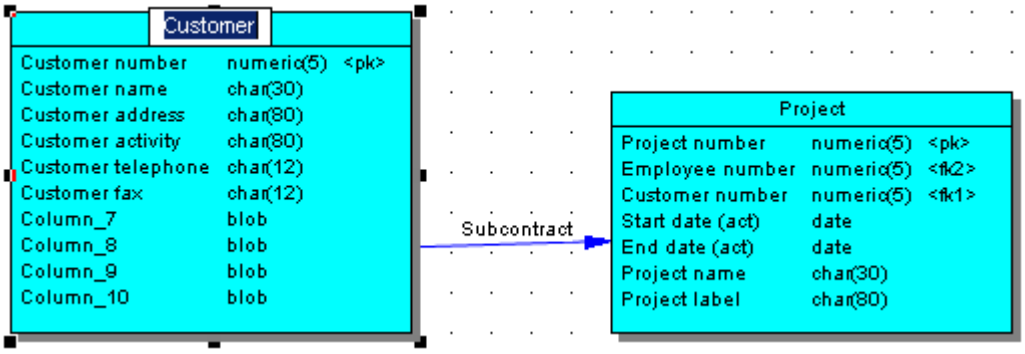
- Additional symbols, press and hold `Shift` while clicking them.
- All symbols in an area, click and hold while dragging the pointer over the area.
- All symbols connected to a symbol, click the symbol and then select *Edit* > *Select Connected Symbols*
- All symbols in the diagram, press `Ctrl` + `A`, click the *Grabber* tool, or select *Edit* > *Select All*

In the following example, the **`Customer`** table is selected, and can be moved by dragging or resized by dragging on one of its handles:



> **Note**
>
> To resize all the symbols in a diagram at once, click the *Grabber* tool and then drag one of the handles.

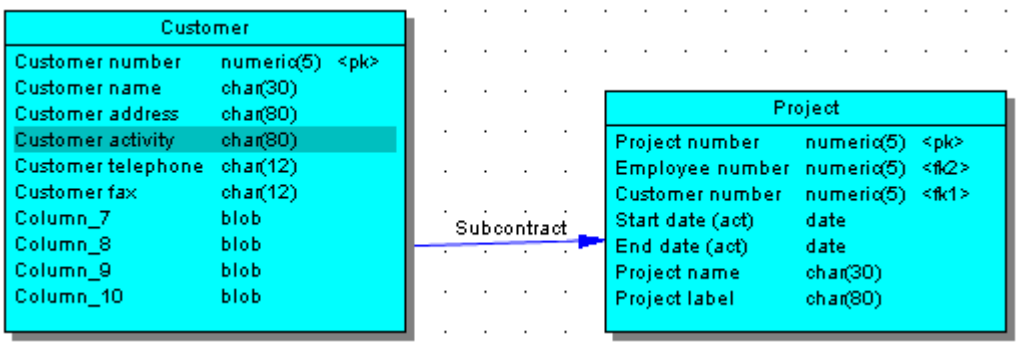
When a symbol is selected, clicking on one of its object properties lets you edit its value. Now, the table name is selected for editing:



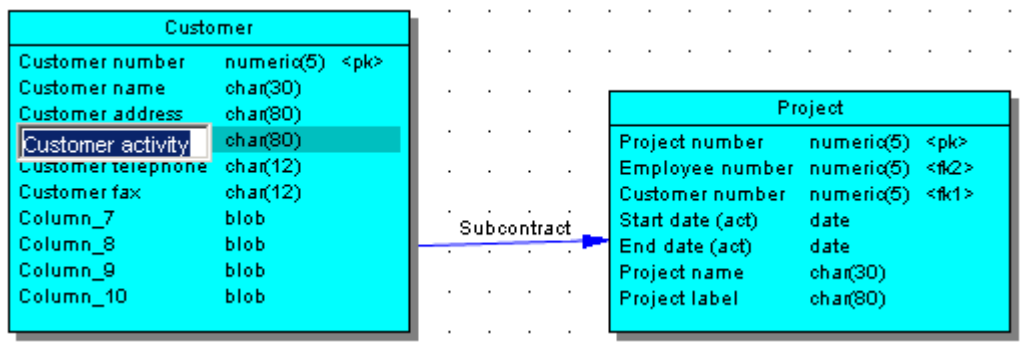
When an object property is selected for editing, you can navigate through its editable properties and on into its sub-object properties by pressing the `Tab` or `Shift` + `Tab` keys.

> **Note**
>
> You can control the object and sub-object properties that appear on a symbol using display preferences (*Tools* > *Display Preferences*).

When a symbol is selected, clicking on one of its sub-objects selects that sub-object. Now, the **`Customer activity`** column sub-object is selected:



You can navigate up and down in the list of sub-objects by pressing the *Up Arrow* or *Down Arrow* keys. When a sub-object is selected, clicking on one of its properties lets you edit its value. Now the **`Customer activity`** column name is selected for editing:



To create a new sub-object at the end of the list, press `Ctrl` + `N`. To insert a new sub-object above the selected sub-object, press `Ctrl` + `I`. To delete the selected sub-object, press `Del`. Right-clicking a sub-object opens a contextual menu offering useful commands.

When a sub-object property is selected for editing, you can navigate right and left along the line of its editable properties by pressing the `Tab` or `Shift` + `Tab` keys.

> **Note**
>
> The following kinds of properties are not editable via object symbols:
>
> - Boolean properties, such as the *Abstract* property on OOM class symbols.
> - Custom indicators, such as those used to identify keys and indexes on PDM table symbols.
> - Certain calculated properties, such as *Domain* or *Datatype* on PDM table symbols.

1.5.3.2 Manipulating Link Symbols

You can add and remove corners to and from link symbols, and change the objects that they connect.

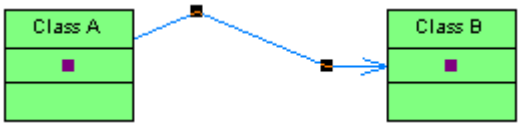
Procedure

1. Draw a link between two objects:

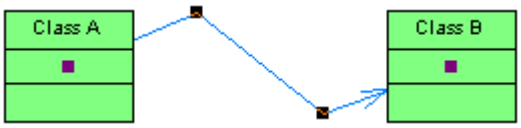
2. Press *Ctrl* and click a point on the line to create a handle where you want to insert a corner:

3. You can add a second handle, or as many as you need:

4. To create a corner, click and hold the handle and then drag it to where you want the corner to be:



5. You can drag the other handles too:



6. To remove a handle (and corner), press *Ctrl* and click on the handle to remove:

> **i** Note
>
> To move an extremity of a link symbol from one object to another, click to select the link, and then drag the end handle to the new object.

1.5.3.3 Formatting, Aligning, Ordering, Grouping, Hiding, and Protecting Symbols

The *Symbol* menu provides various options for formatting and arranging your symbols.

Menu Item	Function
Format (Ctrl + T)	Opens the *Symbol Format* dialog, which allows you to control appearance and content of the selected symbols (see Symbol Format Properties [page 214]).
Get Format	Copies the format of the selected symbol, making it available for applying to other symbols. Only available if a single symbol is selected.
Apply Format	Applies the format copied with *Get Format* to the selected symbols.
Restore Format	Restores the default formats to the selected symbols. These may be the standard formats or any custom symbol formatting defined in an extension file (see Controlling the Format of Custom Symbols [page 218]).
Restore Format For All Diagrams	Restores the default formats to all symbols in all diagrams in the model.
Shadow (Ctrl + W)	Applies the standard shadow effect specified in the *Symbol Format* dialog to the selected symbols.
Adjust to Text (Ctrl + J)	Expands or shrinks the width of the selected objects to fit the length of their names.
Normal Size	Applies the default size specified in Display Preferences to the selected objects.
Fit to Page	Opens the *Fit to Page* dialog, which gives the number of pages currently used and allows you to specify a display scale, the number of pages to use, or to center the symbols on pages.
Auto-Layout	Rearranges the symbols in the diagram (see Auto-Layout [page 210]).
Align	Provides various means for aligning the selected symbols (see Aligning Selected Symbols [page 211]).

Menu Item	Function
Disposition	Arranges the selected symbols in various ways: • Horizontal (Ctrl + H)/Vertical (Ctrl + L) – Straighten selected link objects and makes them horizontal or vertical where possible. • Flip Horizontal/Flip Vertical – Reverses the relative positions of two or more selected symbols. In the case where **Table_1** is directly above **Table_2**, selecting them and then choosing Flip Horizontal, will reverse their places so that **Table_2** is now directly above **Table_1**. • Arrange Symbols – Spaces the selected symbols evenly. • Arrange Connectors – Straightens the selected link symbols and centers their endpoints in the objects that they connect. • Arrange Attach Points - Centers the endpoints of the selected link symbols in the objects that they connect. • Arrange Attached Text – Reverts labels associated with the selected link symbols to their default positions.
Order	Promotes or demotes the selected symbols in relation to other overlapping symbols in the diagram. You can: • Bring to Front • Send to Back • Bring Forward • Send Backward By default, when you place a free symbol such as a note on a design object symbol such as a table, the free symbol is placed behind the object, as free symbols are usually used as backgrounds. When symbols overlap, it may not be possible to select the symbol in the background, even if its handles are visible, without moving the symbol in front of it.
Group	Groups selected symbols, allowing them to be selected, moved, and resized as a single block.
Ungroup	Separates selected objects that were previously grouped together.
Hide	Hides (makes invisible) the selected symbols. You may want to do this to make a large model more readable, or to focus on only a particular part of a model.
Show / Hide Symbols	Opens the *Show Symbols* dialog, which lets you select symbols in the diagram to show or hide (see Showing, Hiding, Protecting, and Unprotecting Symbols [page 212]).
Protect	Protects the selected symbols, making them impossible to select and edit.
Unprotect All	Unprotects the selected symbols, making them available to select and edit.
Protect / Unprotect Symbols	Opens the *Protect Symbols* dialog, which lets you protect and unprotect individual symbols in the diagram

1.5.3.3.1 Auto-Layout

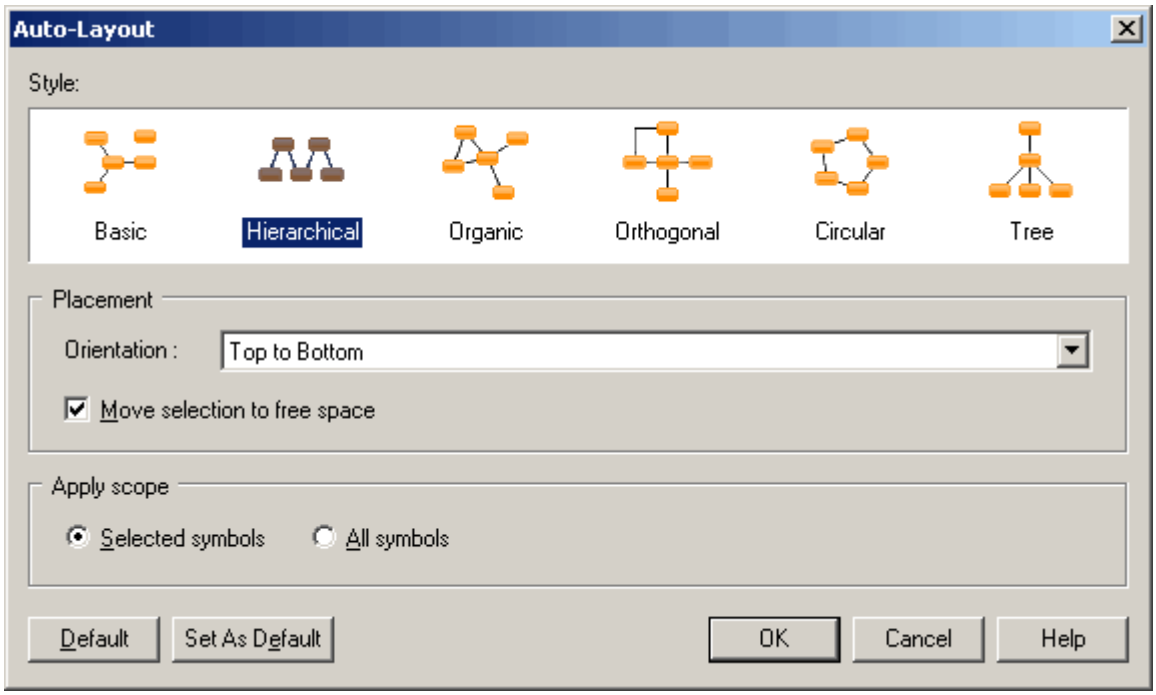
The auto-layout command automatically rearranges symbols in diagrams in order to simplify the routing by minimizing node and link overlaps and reducing the distances between related nodes.

Context

> **i** Note
>
> Auto-layout is not available for the OOM sequence diagram or any diagram containing swimlanes.

Procedure

1. Select *Symbol* > *Auto-Layout* to open the Auto-Layout window.



2. Only the styles relevant for the current diagram can be selected. Choose one of the available styles:
   - Basic – Provides minimal rerouting for any diagram style.
   - Hierarchical – Highlights the main direction or flow within a directed graph. You can additionally specify an orientation for the flow within the graph.
   - Organic – For undirected graphs.
   - Orthogonal - For undirected graphs. You can additionally specify an orientation for the flow within the graph.

- Circular – Produces interconnected ring and star topologies to emphasize group and tree structures within a network. You can additionally specify a cycle or radiation shape.
- Tree – For directed or undirected trees. You can additionally specify an orientation for the flow within the graph.

3. [if you have selected one or more symbols] Specify if you want to apply the auto-layout to all or only the selected symbols.

   When applying Auto-Layout to a selection, you can additionally check the *Move selection to free space* option to extract the selected symbols from the main body of the diagram and place them in free space. PowerDesigner may create graphical synonyms (see Creating Graphical Synonyms for Object Symbols [page 220]) for certain symbols in (or adjoining) the selection to minimize the length of connecting links.
4. Click *OK* to apply the auto-layout and return to the diagram.

> **i Note**
>
> You can, at any time, click the *Default* button to revert to the default auto-layout settings. Click the *Set As Default* button to set the currently selected style as the default.

1.5.3.3.2 Aligning Selected Symbols

The Alignment toolbar and *Symbol Align* submenu both provide multiple tools for aligning symbols selected in the diagram.

Tool	Action
	Left - Aligns left borders of selected symbols with leftmost selected symbol
	Center on vertical axis - Centers selected symbols on the most central selected symbol
	Right - Aligns right borders of selected symbols with rightmost selected symbol
	Same width - Stretches selected symbols to the width of the selection area (from the leftmost symbol to the rightmost symbol)
	Evenly space horizontally - Assigns equal space between three or more symbols on a horizontal axis
	Top - Aligns tops of selected symbols with topmost selected symbol
	Center on horizontal axis - Centers selected symbols on the most central selected symbol
	Bottom - Aligns bottom borders of selected symbols with lowest selected symbol
	Same height - Stretches selected symbols to the height of the selection area (from the topmost symbol to the lowest symbol)

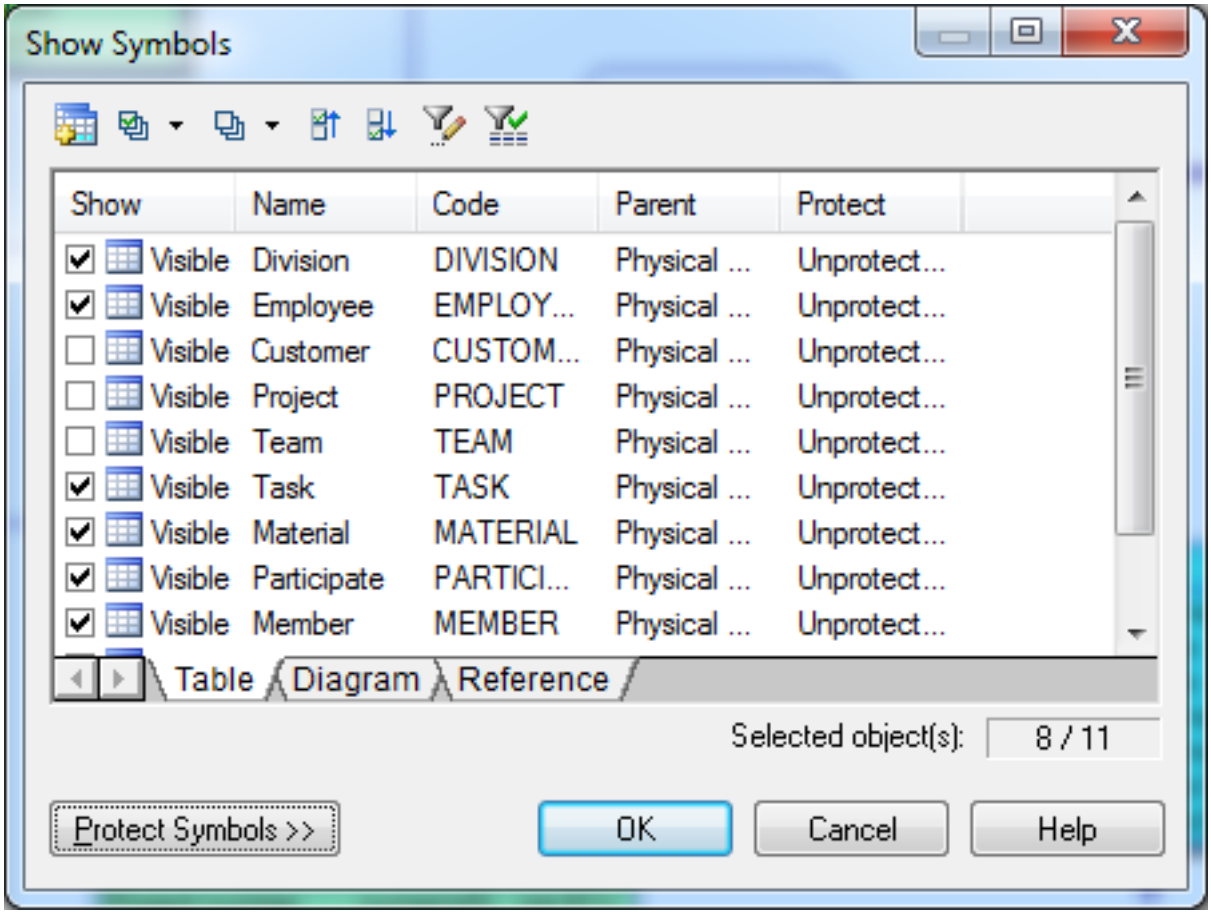
Tool	Action
	Evenly space vertically - Assigns equal space between three or more symbols on a vertical axis

1.5.3.3.3 Showing, Hiding, Protecting, and Unprotecting Symbols

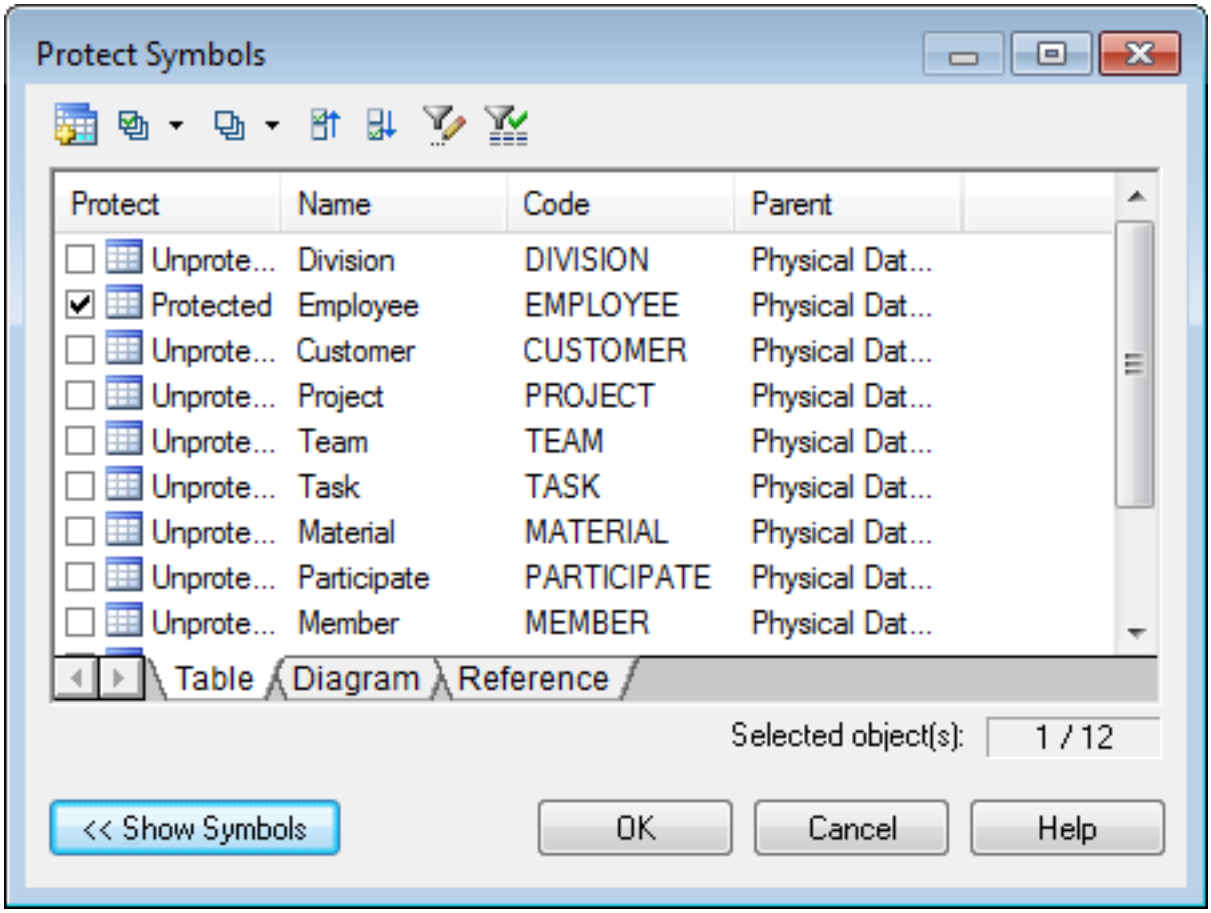
For PowerDesigner objects that have symbols, a symbol is displayed in the diagram when you create the object. You can hide symbols to improve the focus or readability of the diagram without deleting its object. You can protect diagram symbols to prevent them from being moved or deleted.

To hide or protect one or more symbols in the diagram, select them and then select *Symbol* > *Hide* or *Protect*.

To show symbols, drag their objects from the Browser or from another diagram into the diagram or use the Show Symbols dialog. To open the Show Symbols dialog, select *Symbol* > *Show / Hide Symbols* or right-click the diagram background and select *Diagram* > *Show Symbols*:



To open the Protect Symbols dialog, select *Symbol* > *Protect / Unprotect Symbols* or click the *Protect Symbols* button at the bottom-left of the Show Symbols dialog:



In both dialogs, object symbols are organized by type in sub-tabs. If shortcuts, traceability links and/or free symbols are present in the package, they appear on their own sub-tabs. You show and hide symbols by selecting or deselecting them.

The following tools are available for selecting symbols:

Tool	Description
	Opens a dialog to select objects from other packages or open models to display them in the current diagram.
	Selects or deselects all objects in the current tab. To select or deselect all objects on all tabs, press the `Ctrl` key as you click or click the arrow and select *All Pages*.
	Moves all selected object to the top or bottom of the list.

> **Note**

When you hide or show links or objects with links, the following rules apply:

- Hide an object - Links attached to the hidden object are also hidden
- Show an object - Links attached to the object are shown if the object at their other extremity is already visible.
- Show a link - The objects at both ends are also shown.

In a CDM, the following rules apply:

- Hide a parent entity - Inheritances for which the hidden entity is the parent are also hidden.
- Show an association - Entities and association links attached to the association are also shown.
- Show an association link - Entities and associations attached to the association link are also shown.
- Show an inheritance - Parent and child entities and inheritance links attached to the inheritance are also shown.
- Show an inheritance link - Parent and child entities and inheritances attached to the inheritance link are also shown.

1.5.3.4 Symbol Format Properties

You can change the size, line style, fill, shadow, font, alignment, shape and content of symbols in the Symbol Format dialog. Some items may not be available if your modeling methodology restricts the modification of symbol format or content.

You may arrive at the Symbol Format dialog when changing the format of:

- One or more individual symbols – select the symbols and then press Ctrl + T, select *Symbol* > *Format* >, or *Format* from the contextual menu.
- All symbols - via display preferences (see Display Preferences [page 228]).
- Custom symbols defined for extensions – see *Customizing and Extending PowerDesigner* > *Extension Files* .

> **Note**
>
> When defining the format of custom symbols in the Resource Editor, each tab in the Symbol Format dialog contains two additional options, *Apply* `<format>` *to symbols*, and *Allow users to modify symbol* `<format>` , which allow you to control the default format of symbols and whether users can modify them to a very fine degree. For more information on these options, see Controlling the Format of Custom Symbols [page 218].

Size Tab

The Size tab controls the size of the symbol and how the size can be manipulated:

Property	Description
Current size	Specifies the width and height (in pixels) of the symbol.
Normal size	[read-only] Specifies the default width and height (in pixels) for symbols of this kind.
Auto-adjust to text (Ctrl + J)	[Default] Auto-resizes the width of the symbol to display all the text it contains up to the value specified for word wrapping in the diagram (see General Settings Display Preferences [page 229]). This option is disabled if you select the *Keep size* preference or manually resize the symbol and can be reapplied by selecting the symbol and pressing Ctrl + J.
Keep aspect ratio	Maintains the ratio of height to width when manually resizing the symbol.
Keep center	Maintains the center position of the symbol when resizing it.
Keep size	Prevents the resizing of symbols through: • Manual resizing • The *Auto-adjust to text* preference (which is mutually exclusive with this preference). • The creation of sub-object symbols in editable composite symbols (see Using Composite View to Display Sub-Objects within a Symbol [page 220]). Selecting this option prevents the symbol from growing to accomodate the addition of sub-symbols and may cause their display to be truncated due to lack of space.

Line Style Tab

The *Line Style* tab controls the color, size and format of lines (for link and other one-dimensional symbols) and borders (for two-dimensional symbols, such as classes or tables). You can modify the line style of any symbol in the model.

Property	Description
Color	Specifies the color of the line or border.
Width	Specifies the thickness of the line or border.
Style	Specifies the format of the line or border, such as invisible, solid, dashed or dotted.
Corners	[Link and line symbols only] Specifies the format of corners, such as sharp or rounded right-angles and free angles.

Property	Description
Arrow	[Link and line symbols only] Specifies the format of the link or line symbol at its Beginning, Center, and End.
Use perpendicular arrow	[Link and line symbols only] Specifies that the link or line object always touches objects at its ends at a right angle. This option allows you to use free angles for the body of a link or line object, while retaining a connection at right angles for its beginning and end.

> **i** Note
>
> These options may be overridden by the design semantics of a link symbol (such as a PDM reference).

Fill Tab

The *Fill* tab controls the color, content, and effects for symbol filling.

Fill effects	Description
Fill color	Specifies the color to use for the fill. The Fill color check box is available for free symbols, packages, interaction fragments, and swimlanes, and when cleared will create a transparent symbol. This can be useful when you wish to use an "in-box" UML representation, for packages for example. Note that transparent objects can only be selected by clicking near their borders.
Image	Specifies a graphic file to display within the symbol. Click the *Modify* button to select a file.
Display Mode	Specifies the alignment of the image within the symbol.
Gradient	Specifies the gradient fill options. Click the *Modify* button to open a dialog allowing you to control Start and End colors, End Color Luminosity, Shading Style and Gradient Mode.

Shadow Tab

The *Shadow* tab allows you to add a standard, 3D effect or gradient shadow to objects in a diagram.

Font Tab

The *Font* tab allows you to define the display preferences for the font, size, style, and color of text associated with symbols in the model. When you modify font preferences, they apply to all existing and new symbols.

Text Alignment Tab

The *Text Alignment* tab allows you to define the alignment of text in text boxes and rectangles, ellipses, rounded rectangles, and polygons.

> **i** Note
>
> You can only control the text alignment for these shapes, and not for model object symbols. When working with RTF, all the options but *Vertical* are disabled.

Parameter	Description
Center	Centers the text horizontally and vertically.
Word wrapping	Auto-wraps text in the symbol.
Horizontal	Aligns text to the left, center, or right on the horizontal axis.
Vertical	Aligns text to the top, center, or bottom on the vertical axis

Custom Shape Tab

The *Custom Shape* tab allows you to define a new symbol shape for most non-link symbols.

> **i** Note
>
> Custom shapes with the *Content position* set to `Under` cannot display collections or attribute icons.

Property	Description
Enable custom shape	Enables or disables the customization of a symbol shape
Shape Type	Specifies the type of shape to be used. Choose one of the following values, and then click the *Browse* button to select an image (see Browsing for Images [page 222]): • Predefined symbol - Default shape assigned to symbols in PowerDesigner. • Metafile (EMF, WMF) - Representation using geometrical formulas allowing for resizing and stretching. • Bitmap (DIB, RLE, JPG, JPEG, TIF, TIFF, PNG) - Representation of a graphics image consisting of rows and columns of dots. • Icon (ICO) - Small picture representing an object, usually smaller than standard PowerDesigner symbols.
Shape Name	List of the available shapes of the selected type. Click the *Browse* button to select shapes.

Property	Description
Content position	Specifies where the object name and any other properties should be displayed on (or under) the symbol (Center option is not available for icon shapes).

Content Tab

The *Content* tab allows you to specify the information that you want to display on the symbol. The properties that are listed on this tab as being available for selection are controlled by the content display preferences (see Content Display Preferences [page 232]).

Sub-Objects Tab

The *Sub-Objects* tab is only available when you open the Format tab after having selected a single object. It allows you very fine control over the sub-objects (such as PDM table columns or OOM class attributes and operations) that you want to display inside your object symbol. For each individual sub-object, you can decide whether to display it or not, and what font to use for its display.

> **i** Note
>
> For a collection of sub-objects to be available for selection and customization here, the collection must be selected for display in the object's content display preferences (see Content Display Preferences [page 232]). Where the number of lines to display for a collection is limited in the display preferences, that limit will take precedence over any choices you make here.

Each collection of sub-objects that is enabled in the display preferences has its own sub-tab. For each sub-object, you can:

- Select to display or hide it in the parent object symbol by selecting or deselecting its checkbox in the *[D]isplay* column.
- Apply a specific font to its display by clicking the *Select font tool* or the ellipsis button in the *Specific Font* column.

> **i** Note
>
> When not all sub-objects in a collection are selected for display, the parent object symbol will display ellipses to indicate that more items are available.

1.5.3.4.1 Controlling the Format of Custom Symbols

When defining custom symbols for your object extensions, you may want to enforce certain aspects of the symbol format, while allowing users some liberty to change others. If you open the Symbol Format dialog from

a custom symbol in the Resource Editor, you can control the default format options for the symbol, and whether users can edit them, on a per-tab basis.

For example, you could extend the PDM table metaclass with a criteria to specify a custom symbol with a red border for tables that are anticipated to contain more than 1m records. You may want to enforce this border style for all such tables while allowing users to change the font, size, content, and other aspects of the symbol format.

When you access the *Symbol Format* dialog from a custom symbol in the Resource Editor, the following additional options are available at the top of each tab:

Option	Description
Apply `<format>` to symbols	Specifies that the format options on the tab will be applied to custom symbols at creation time or whenever the appropriate criteria are met or the specified stereotype applied. If this option is not selected, the standard display preferences for the format options on the tab will be applied and users are free to modify them.
Allow users to modify symbol `<format>`	Specifies that users can modify the format options for the symbol and that any changes that are made will be respected in the event that the definition of the custom symbol is updated. If *Apply* `<format>` *to symbols* is selected and this option is unselected, then the format options specified on this tab cannot be changed by the user.

Click the *Advanced* button to open the *Customize Content* dialog to add or removed properties from this tab (see Customizing Content Display Preferences [page 234]).

For detailed information about extending objects and creating custom symbols for them, see *Customizing and Extending PowerDesigner > Extension Files* .

1.5.3.5 Finding Symbols from the Browser, Object Lists, and the Result List

You can find all the symbols representing an object listed in the Browser, in a model list, or in the Result List window by right-clicking the object and selecting *Find in Diagram*.

Context

If the object has only one symbol associated with it, the appropriate diagram is opened in the canvas with the symbol centered.

If the object has several symbols in the present package or there are symbols representing shortcuts to the object (see Shortcuts and Replicas [page 443]) in other packages or models open in the Workspace, the Select Symbol dialog opens listing all the available symbols with their parent model and diagram. To show only the symbols in the present package in this dialog, deselect the *Show symbols from other packages* option. Select a symbol and click *OK* to open its diagram in the canvas.

1.5.3.6 Using Composite View to Display Sub-Objects within a Symbol

Many objects (packages, processes, classes, activities, states, EAM objects, etc.) have the capability to display sub-objects within their symbols through one or more composite view modes. Depending on the type of object, the sub-objects can be displayed either as a static sub-diagram within the symbol or as dynamically editable symbols that you can create and arrange directly from the parent diagram.

Context

To toggle between the different composite view modes, right-click a symbol and select *Composite View* and then one of the following commands:

- None - display the parent symbol only without any sub-objects
- Read-only (Sub-Diagram) - display sub-objects in a non-editable sub-diagram that can be resized as necessary. To access the sub-diagram, press *Ctrl* and double-click the symbol
- Editable - allow the creation and arrangement of sub-objects directly within the symbol in the parent diagram.
- Adjust to read-only view - resizes the symbol to display all the objects in the read-only view.

> **i** Note
>
> Not all modes are available for all objects that support composite view. Certain objects (for example, processes in the BPM) must be decomposed before you can access composite view.

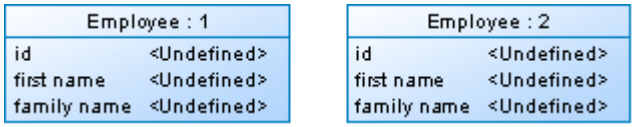
For information about the specific objects supporting composite view, see the appropriate modeling guide.

1.5.3.7 Creating Graphical Synonyms for Object Symbols

Graphical synonyms let you depict an object more than once within a diagram. This can improve readability by reducing the length or complexity of links when you have an object that is linked to many other objects. You create a graphical synonym by right-clicking a symbol and selecting *Edit* > *Create Graphical Synonym*. You can create as many graphical synonyms as you want within the same diagram, including graphical synonyms of graphical synonyms and of shortcuts. Each graphical synonym displays the name of the object followed by a colon and the number of the synonym.

> **i** Note
>
> A graphical synonym is merely a second symbol representing an object within the same diagram. You can add the same object to other diagrams in the same package or model by dragging and dropping it to them to create an ordinary symbol. If the object must be referenced in another model (or in another package in the model), you need to create a shortcut (see Shortcuts and Replicas [page 443]).

In the following example, **`Employee : 1`** and **`Employee : 2`** both represent the **`Employee`** table:



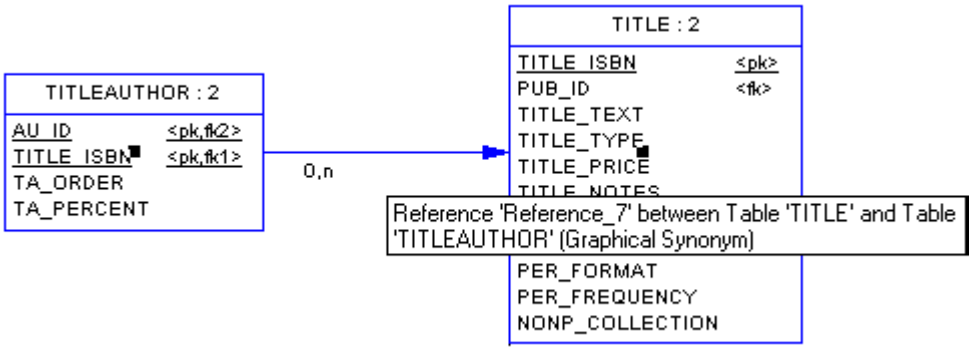
If you move an object for which you have created graphical synonyms from one package or model to another, shortcuts are created for the graphical synonyms in the original package.

> **i Note**
>
> To find any graphical synonyms of a symbol, right-click the symbol, select *Edit* > *Find Graphical Synonym*, and then select a graphical synonym from the list. The graphical synonym is centered and selected in the diagram window.

You can create graphical synonyms for link objects but only if both they and the symbols at both their extremities (which will also be duplicated) support them. The following object symbols do not support graphical synonyms:

- CDM - inheritances and inheritance links
- OOM - swimlanes, synchronizations, decisions, transitions, instance links, messages, association class links, and interaction frames, fragments, and references
- BPM - swimlanes, synchronizations, decisions, and flows
- All - free symbol line

> **i Note**
>
> Though you cannot visually distinguish a graphical synonym of a link from a normal link (unless you display the name attribute), when you select a link symbol, it is identified as a graphical synonym in its tooltip:

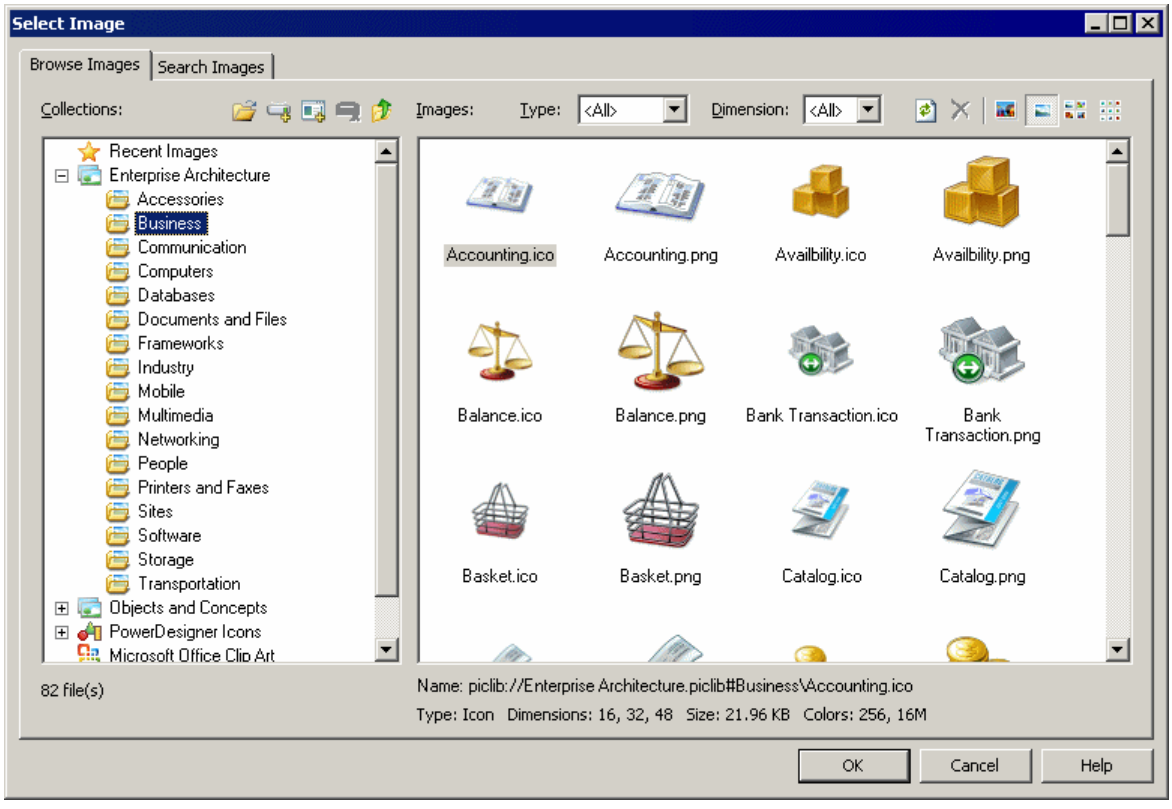


1.5.3.8 Browsing for Images

The Select Image dialog lets you manage the images that you use in your model diagrams, and to insert them into your models as object symbols, diagram backgrounds, etc.

Procedure

1. Open the Select Image dialog in any of the following ways:
   - Right-click a diagram symbol and select *Change Image*.
   - From the Symbol Format dialog *Custom Shape* tab, click the *Enable Custom Shape* check box, and then click the *Browse* button next to the *Shape type* property (see Symbol Format Properties [page 214]).
   - From the Symbol Format dialog *Fill* tab, click the *Modify* button in the *Fill Effects* group box (see Symbol Format Properties [page 214]).
   - Select *Edit* > *Import Image* (see Importing and Exporting Model Graphics [page 240]).



2. Click a collection in the *Collections* pane to display its available images in the *Images* pane.

   The following tools are available for this pane:

Tool	Description
	Open Image File - Lets you browse to an image file to select it directly without passing by an image collection.
	Add Directory – Adds a directory to the list of collections. When you add a directory, any subdirectories containing images are also added along with any intermediate directories required to represent the directory tree.
	Add Executable or DLL - Adds an executable or DLL and all the images associated with it to the list of collections.
	Remove Directory, Executable, or DLL – Removes a directory from the list of collections. Predefined directories, such as Recent Images or Microsoft Office Clip Art cannot be removed.
	Up – Moves up a level in the collections tree.

3. [optional] In the *Images* tab, select an image type and/or dimension to filter by from the *Type* and *Dimension* lists.

   The following tools are available for this pane:

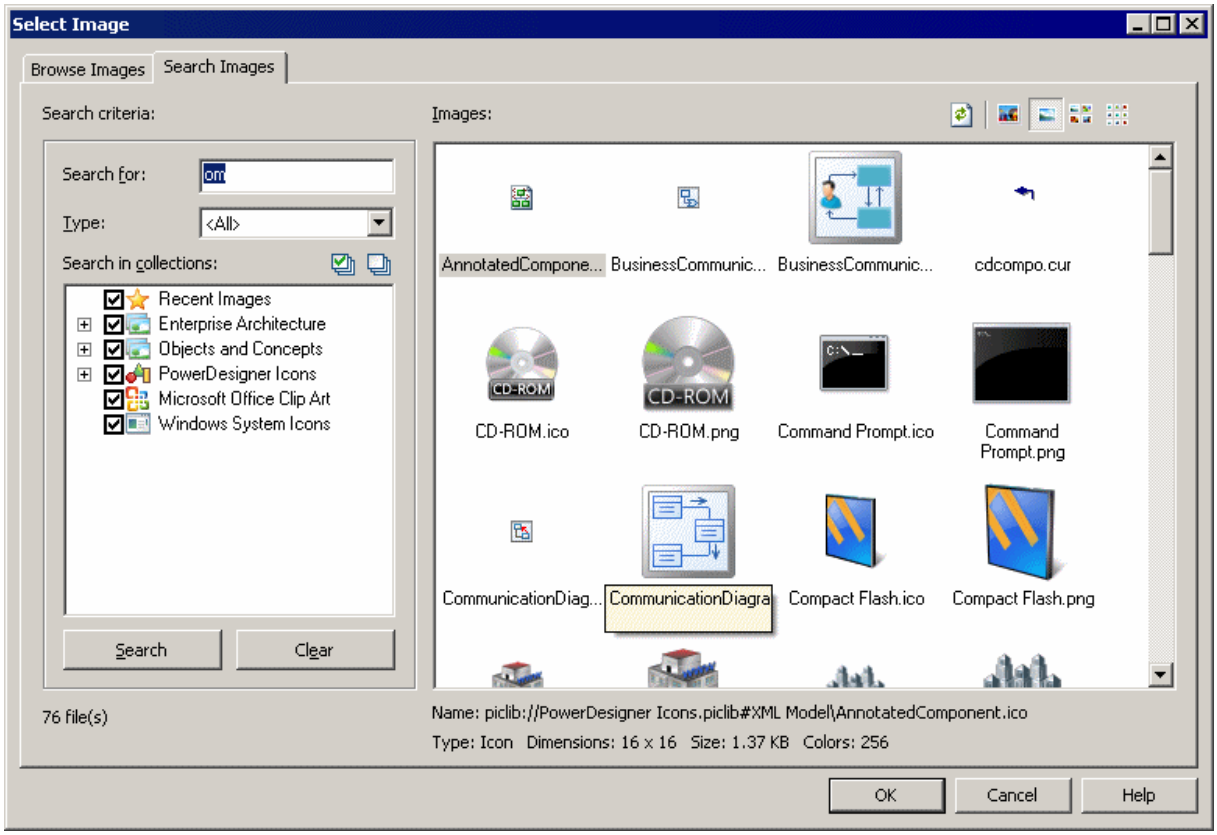
Tool	Description
Type	Filters images by type, such as PNG, Bitmap, Cursor, etc.
Dimension	Filters images by dimension, such as 16, 48, 128, etc.
	Refresh Preview – Refreshes the list of available images.
	Delete Recent Image – Deletes a selected recent image.
	Extra Large / Large / Medium / Small Icons – Change the display size of images in the Images pane.

4. Select an image and click *OK* to return to the Symbol Format dialog or to display the image in the diagram.

1.5.3.8.1 Searching for Images

You can search for images among your collections on the *Search Images* tab of the Select Image dialog.

Context



Procedure

1. Specify the appropriate search criteria in the *Search Criteria* pane. You can specify:
   - *Search for* – Enter all or part of a filename to search against. Wildcards are automatically applied so that for example, entering "ec" will return an image named "vector.png".
   - *Type* – Select the type of image to search for, such as PNG, Bitmap, Cursor, etc.
   - *Search in collections* – Specifies the scope of the search by selecting or deselecting collections to search in.
2. Click the *Search* button to display the result of the search in the Images pane.
3. Select an image in the *Images* pane, and click *OK* to return to the Symbol Format dialog or to display the image in the diagram.

   The *Clear* button resets the search criteria to the default values.

1.5.3.9 Decorative Symbols

Lines, ellipses, rectangles, and other decorative symbol tools available in the Toolbox have no technical meaning, but can help with readability. You can use them to surround parts of a model to, for example, distinguish domains of activity.

With each of the following tools, click and hold to draw, and release to complete the shape. To resize a completed shape, click it and drag a handle.

Tool	Description
	Insert an area where you can write free notes
	Draw either a graphical link between free symbols, a note link between a note and an object, or a traceability link between objects (see Traceability Links [page 146])
	Title - A diagram title box retrieves from the model properties and displays such information as: the model and the package to which the diagram belongs, the name of the diagram itself, the author and version of the model and the date of modification. If no Author is specified in the model property sheet, the user name specified in the Version Info page is used. You can choose to display the repository version number of the model or a user-defined version number on the Title display preferences page.
	Text - You can insert free text in your model independently of any shape, and can select and move it like any symbol.
	Line
	Arc
	Rectangle - Press *Ctrl* while drawing to create a square.
	Ellipse - Press *Ctrl* while drawing to create a circle.
	Rounded Rectangle - Press *Ctrl* while drawing to create a rounded square.
	Polyline - Release the mouse button at each point where you want to create a corner. Right-click to finish.
	Polygon - Release the mouse button at each point where you want to create a corner. Right-click to finish and close the polygon.

> **i Note**
>
> To add text to the symbol, double-click it to open an editor window. Lines and polylines support the display of text at their center, source, and destination (see Text Editor Tools [page 226]).

To change the format of the symbol, right-click it and select *Format* (see Symbol Format Properties [page 214]).

1.5.3.9.1 Creating a Graphical Link Between Any Two Symbols

You can create a graphical link between any two symbols in the diagram. This link is purely graphical and does not convey any semantics.

Procedure

1. Select the *Polyline* tool in the Toolbox.
2. Click inside the first symbol and while continuing to hold down the mouse button, drag the cursor to a second symbol. Release the mouse button inside the second symbol and right-click.

   A link is created. You can double-click the link symbol to open the *Link Symbol Text* dialog box and edit the link.

1.5.3.9.2 Text Editor Tools

PowerDesigner provides an internal rich text format (rtf) editor, and supports the use of external text editors.

The following tools are available on the editor toolbar (the *Edit With* command and various formatting commands and tools are only available in RTF editors):

Tools	Description
	Editor Menu (Shift + F11) - Contains the following commands: • *New* (Ctrl + N) - Reinitializes the field by removing all the existing content. • *Open...* (Ctrl + O) - Replaces the content of the field with the content of the selected file. • *Insert...* (Ctrl + I) - Inserts the content of the selected file at the cursor. • *Save* (Ctrl + S) - Saves the content of the field to the specified file. • *Save As...* - Saves the content of the field to a new file. • *Select All* (Ctrl + A) - Selects all the content of the field. • *Find...* (Ctrl + F) - Opens a dialog to search for text in the field. • *Find Next...* (F3) - Finds the next occurence of the searched for text. • *Find Previous...* (Shift + F3) - Finds the previous occurence of the searched for text. • *Replace...* (Ctrl + H) - Opens a dialog to replace text in the field. • *Go To Line...* (Ctrl + G) - Opens a dialog to go to the specified line. • *Format... [Ctrl+M]* - Opens a dialog to specify fonts and styles. • *Remove All Formats [Ctrl+L]* - Removes all formatting from the content of the field. • *Paragraph... [Ctrl+R]* - Opens a dialog to specify paragraph indentation, spacing, and alignment. • *Tabs... [Ctrl+B]* - Opens a dialog to specify tab stop positions and alignment.
	Edit With - Opens the content of the field in an external editor. Click the down arrow to select a particular editor or *Choose Program* to specify a new editor. Editors specified here are added to the list of editors available at *Tools* > *General Options* > *Editors*.
	Save (Ctrl + S) - Saves the content of the field to the specified file.
	Print (Ctrl + P) - Prints the content of the field.
	Find (Ctrl + F) - Opens a dialog to search for text.
	Cut (Ctrl + X), *Copy* (Ctrl + C), and *Paste* (Ctrl + V) - Perform the standard clipboard actions.
	Undo (Ctrl + Z) and *Redo* (Ctrl + Y) - Move backward or forward through edits.
	Format (Ctrl + M) - Opens a dialog to specify fonts and styles.
	Bold, *Italic*, and *Underline* - Format the selected text.
	Align Left, *Align Center*, and *Align Right* - Align the selected text.
	Bullets - Indent and bullet the selected text.

> **i Note**
>
> When editing free text on decorative symbols, use the radio buttons at the bottom of the dialog to switch between

- *Plain text* - text formatting is controlled by the display preferences for free symbols
- *RTF* – text formatting can be controlled directly in the text input window

1.5.4 Display Preferences

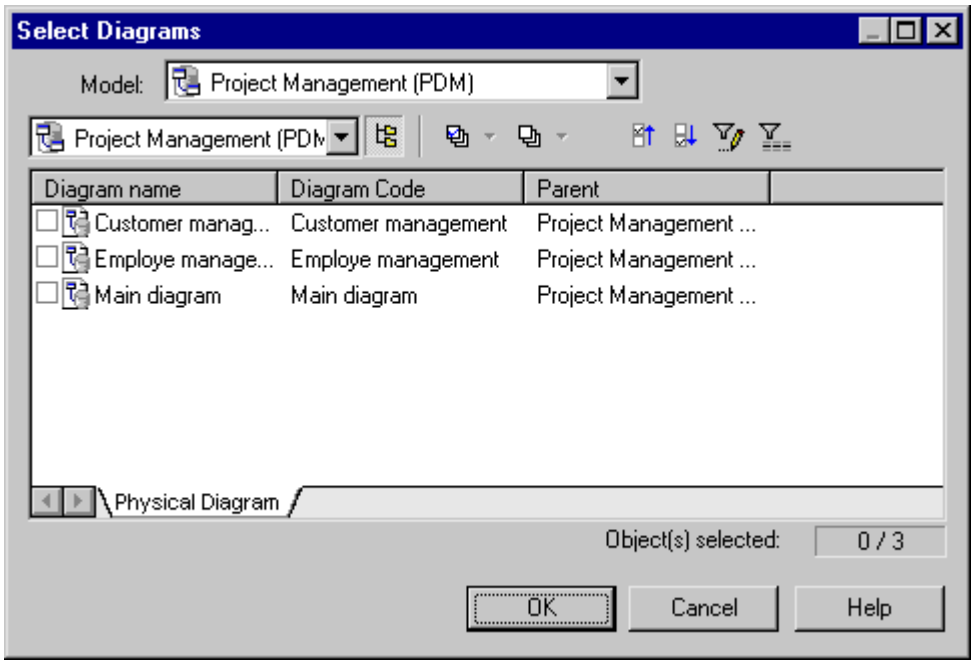
Display preferences control the format of the symbols in your diagrams and the information displayed on them. You can control the default display preferences for all objects, or set preferences for all, selected, or new objects in the current diagram.

Procedure

1. [optional] Select one or more symbols to which you want to apply format or content changes. If you do not select any symbols, you can choose to apply any changes to all objects or only to objects created after your changes.
2. Select *Tools* > *Display Preferences* to open the Display Preferences dialog.
3. Select an object in the Category list and change the preferences as appropriate in the right-hand panel. The General Settings category controls preferences for the diagram as a whole (see General Settings Display Preferences [page 229]).

   On each of the pages, you can, at any time click:
   - the *Default* button, which reverts any changes to their default values
   - the *Set As Default* button, which assigns the current values to defaults.

4. [optional] Click the *Apply To* button to open the Select Diagrams dialog, select the diagrams to which you want to apply your changes, and click *OK*:

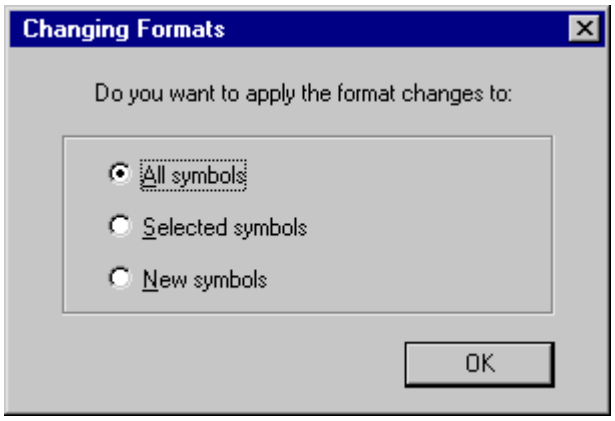


You will be asked whether you want to apply your changes to all the symbols in the selected diagrams. If you click:

- Yes – All existing and new symbols will reflect your changes to the Format, General and Content display preferences.
- No – New symbols only will reflect your changes. Only the General and Content display preferences are applied to existing symbols. The Format changes are canceled.
- Cancel – Your changes will not affect any symbols in any diagrams except the current one.

In each case, you return to the Display Preferences dialog.

5. On the Display Preferences dialog box, click *OK*. The Changing Formats dialog box will open:



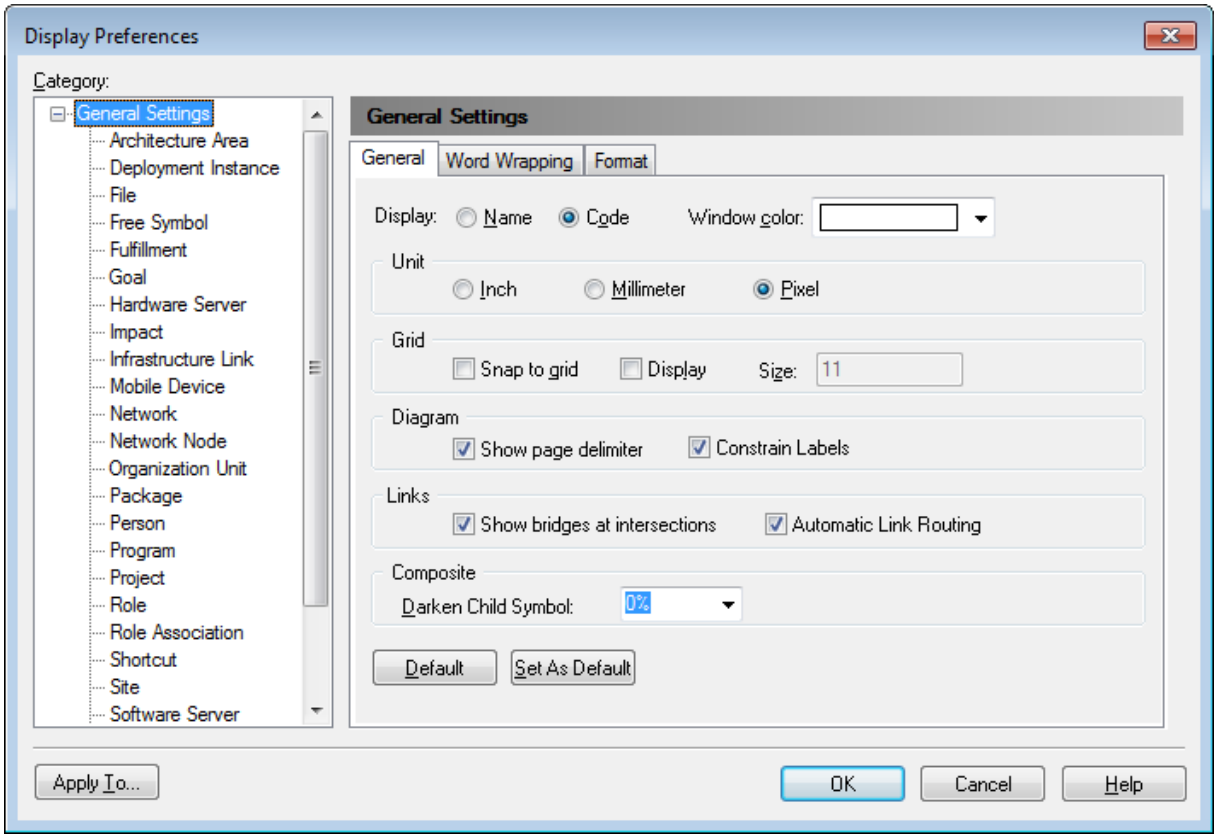
The choices in this dialog apply only to the present diagram and will not override your choices made in the Select Diagrams dialog.

6. Click *OK* to apply your changes

> **i Note**
>
> If you make changes in the Display Preferences dialog and these are not reflected in some or all of the symbols in your diagram, this may be because of display constraints enforced by your modeling methodology. For detailed information about how such constraints are defined, see *Customizing and Extending PowerDesigner > Extension Files* .

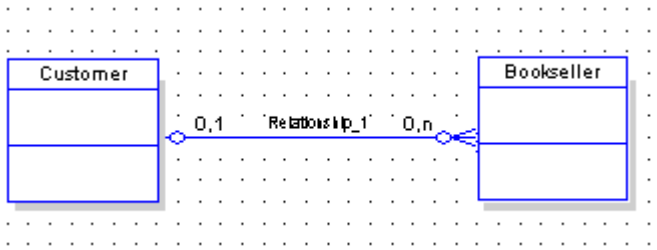
1.5.4.1 General Settings Display Preferences

The General Settings category allows you to control the overall look of the diagram.



The following preferences are available on the *General* tab:

Property	Description
Display	Specifies whether object names or codes are displayed in the Browser and on diagram symbols. You can toggle between the values of this setting with the *Display Name/Code* tool in the *Standard* toolbar.
Window color	Sets the background color for the diagram.
Unit	Specifies the measurement unit for symbol size. You can choose between inches, millimeters, and pixels.

Property	Description
Snap to grid	Automatically aligns all objects in the diagram to the anchor points of a grid. Control the visibility of the grid, and the number of anchor points per square inch with the *Display* and *Size* options: 
Show page delimiter	Displays lines showing the limit of pages on the diagram background.
Constrain labels	Limits the distance to which you can displace labels from link objects. Uncheck this option to be able to move link labels anywhere in your diagram.
Show bridges at intersections	Creates a bridge (or "line hop") when two links cross to show that they do not intersect. This preference has no effect on link symbols with free curve corners or CDM association links.
Automatic link routing	Calculates link paths to avoid crossing object symbols and to space links regularly. You can manually move routed links, and the algorithm will respect your choices and exclude such links from its calculations. To revert a manually moved link to its automatically routed position, right-click it and select *Reroute Link*. This preference has no effect on links with free curve or free angle corners, CDM association links, or any links in EAM organization charts, BPM process hierarchy diagrams, OOM sequence diagrams, or XSM XML diagrams.
Organization unit swimlane	[diagrams with swimlanes] Specifies that organization units will be displayed as swimlanes. You must also specify whether swimlanes have a *Horizontal* or *Vertical* orientation.
Darken Child Symbol	[diagrams with composite symbols] Specifies that the symbols of child objects are darker than those of their parents in order to make them stand out more.
Orientation	[diagrams with tree structures] Specifies whether branches expand in a *Horizontal* (left to right) or *Vertical* (top to bottom) orientation.

The following preferences are available on the *Word Wrapping* tab:

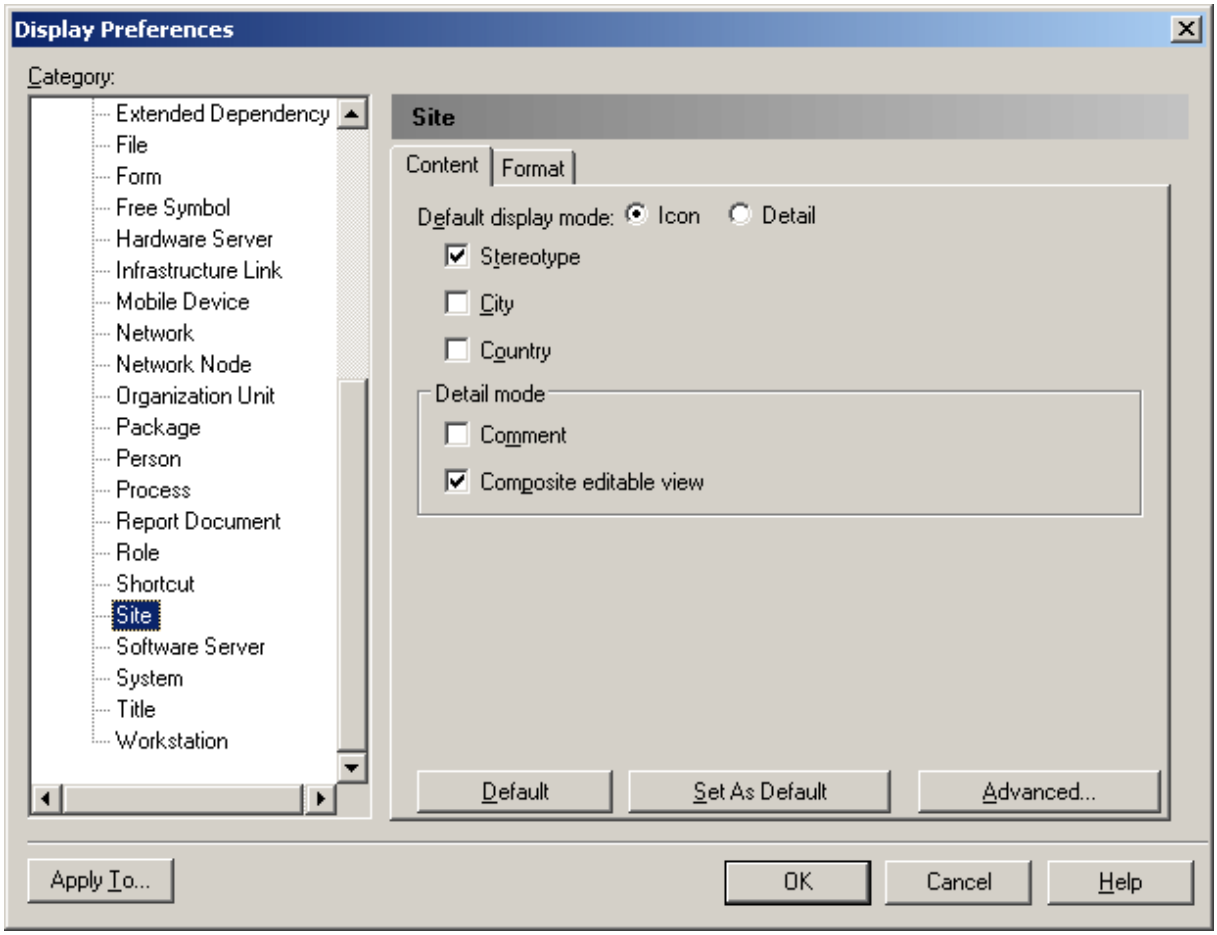
Preference	Description
Word wrapping	Controls the display of text on symbols and links. You can choose between: • None • Truncation - Truncates name (or code), to the specified length • Word wrapping - Wraps name and code text onto additional lines (up to the specified length) after the specified wrap character This preference is overriden if you manually resize a symbol or link text label. To reapply the wrapping specified here, select the symbol or link text label and press `Ctrl` + `J`.
Wrap characters	Specifies a set of characters after which word wrapping is possible.
Display overlay for non-local objects	Enables the display of an overlay on objects being reused in the same container but with distinct location. For instance an activity can be reused under another activity of the same package or an inner class can be referenced by another class of the same package. The overlay is a mirror image of the shortcut overlay (see Creating Shortcuts [page 446]).

The *Format* tab can be used to set the default size for all symbols in the diagram. Values entered here can be overridden for specific objects on the *Format* tab for each object type (see Format Display Preferences [page 234]).

1.5.4.2 Content Display Preferences

The Content category allows you to control what information is displayed in object symbols.

You can modify the information displayed on each type of object by selecting it in the list.



Choose the information that you want to display by selecting the appropriate checkboxes. You can customize the properties that are available to be displayed by clicking the *Advanced* button (see Customizing Content Display Preferences [page 234]).

The following special preferences are available for certain objects that can act as containers for other objects:

Preference	Description
Default display mode	Specifies the way the object will be displayed by default: • Icon • Detail – a rectangle permitting the display of additional information This preference is only available if the object supports both modes. **i Note** Multi-line fields and collections (such as table columns or class operations) cannot be displayed in icon mode or on link symbols

Preference	Description
Detail mode	Specifies the additional information that will be displayed in Detail mode: • Comment – from the *General* tab of the object's property sheet • `<[collections]>` – for example, sub-sites, applications, etc. • Composite view - displays child objects as symbols within the object symbol. If this option is present, it cannot (unlike the other preferences) be removed from this screen

1.5.4.3 Format Display Preferences

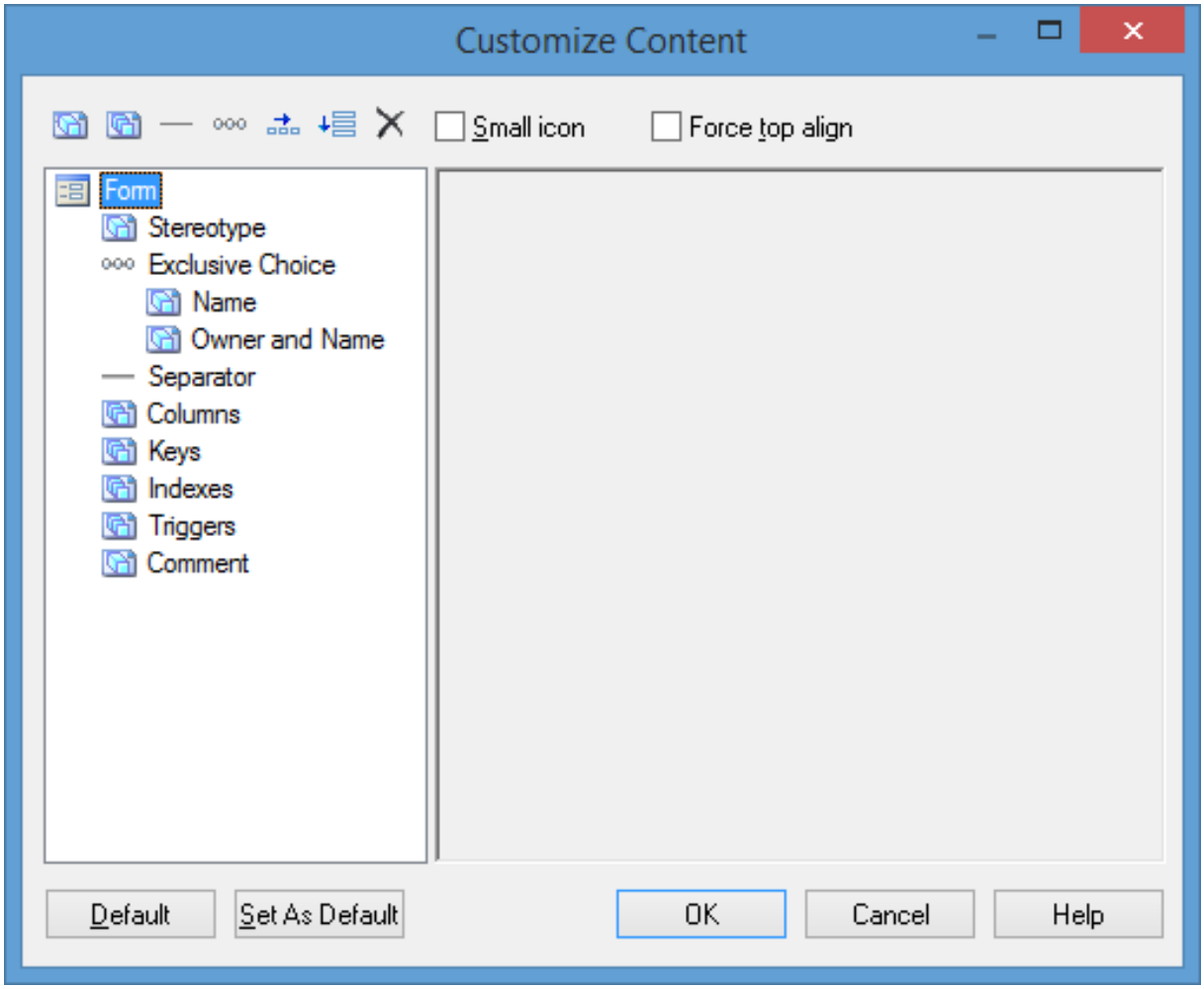
The *Format* tab allows you to control the look of object symbols and provides a preview for the preferences selected.

Click the *Format* tab of the *General Settings* category, to set the default horizontal and vertical size of all symbols in the diagram containing text (measured in units of 1/7200 of an inch).

For objects, click the *Modify* button to change the line style, fill, shadow, font, and custom shape of symbols (see Symbol Format Properties [page 214]).

1.5.4.4 Customizing Content Display Preferences

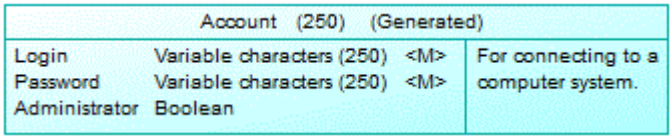
Click the *Advanced* button in the bottom-right corner of an object's *Content* tab to organize the information that is available for display on the symbol.



The list in the left hand pane shows the properties that are currently available to display on the symbol in the Display Preferences dialog. In the case of link symbols the list contains three sublists, Source, Center, and Destination, which allow you to specify different content to display on each of these parts of the link symbol.

You can drag and drop items to reorder the list, and use the following tools to modify it:

Tool	Description
	Add Attributes – allows you to select additional attributes to add to the list (see Configuring the Display of Attributes [page 236]).
	Add Collections – allows you to select additional collections (such as table columns, class operations, etc) to add to the list (see Configuring the Display of Collections [page 238]).
	Add Separator Line – inserts a separator line after the item selected in the list. Separators added to horizontal layouts display as vertical lines.
	Add Exclusive Choice - inserts an item under which you can group properties from which your users will choose one to display (see Configuring the Display of Exclusive Choices [page 239]).

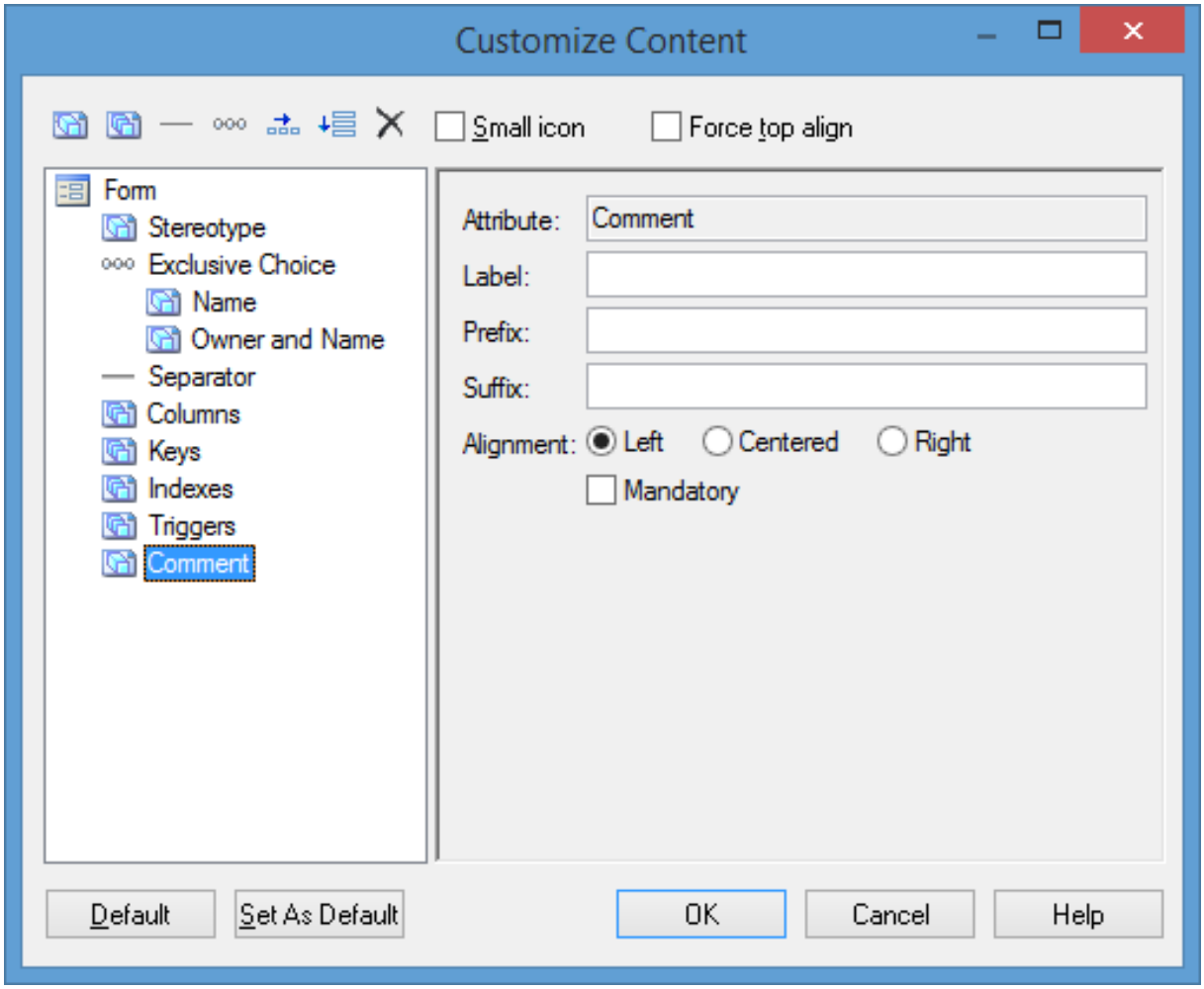
Tool	Description
	Add Horizontal Layout - inserts a horizontal layout. To arrange attributes to display side by side, drag them onto the horizontal layout in the list. 
	Add Vertical Layout - inserts a vertical layout. To arrange attributes to display one under the other, drag them onto the vertical layout in the list. Vertical layouts are often used in conjunction with a horizontal layout, to provide columns of attributes.
	Delete – removes the selected item from the list.

The following options control the overall look of symbols:

Tool	Description
Small icon	Places a small object icon in the top-left corner of the symbol in detail mode.
Force top align	Forces top alignment in the symbol for object attributes, such as Name. If this option is not selected, these properties are centered on the vertical axis.

1.5.4.4.1 Configuring the Display of Attributes

An attribute is any single value property. To enable the display of an attribute, click the *Add Attributes* tool, select attributes or extended attributes to display, and click *OK*.

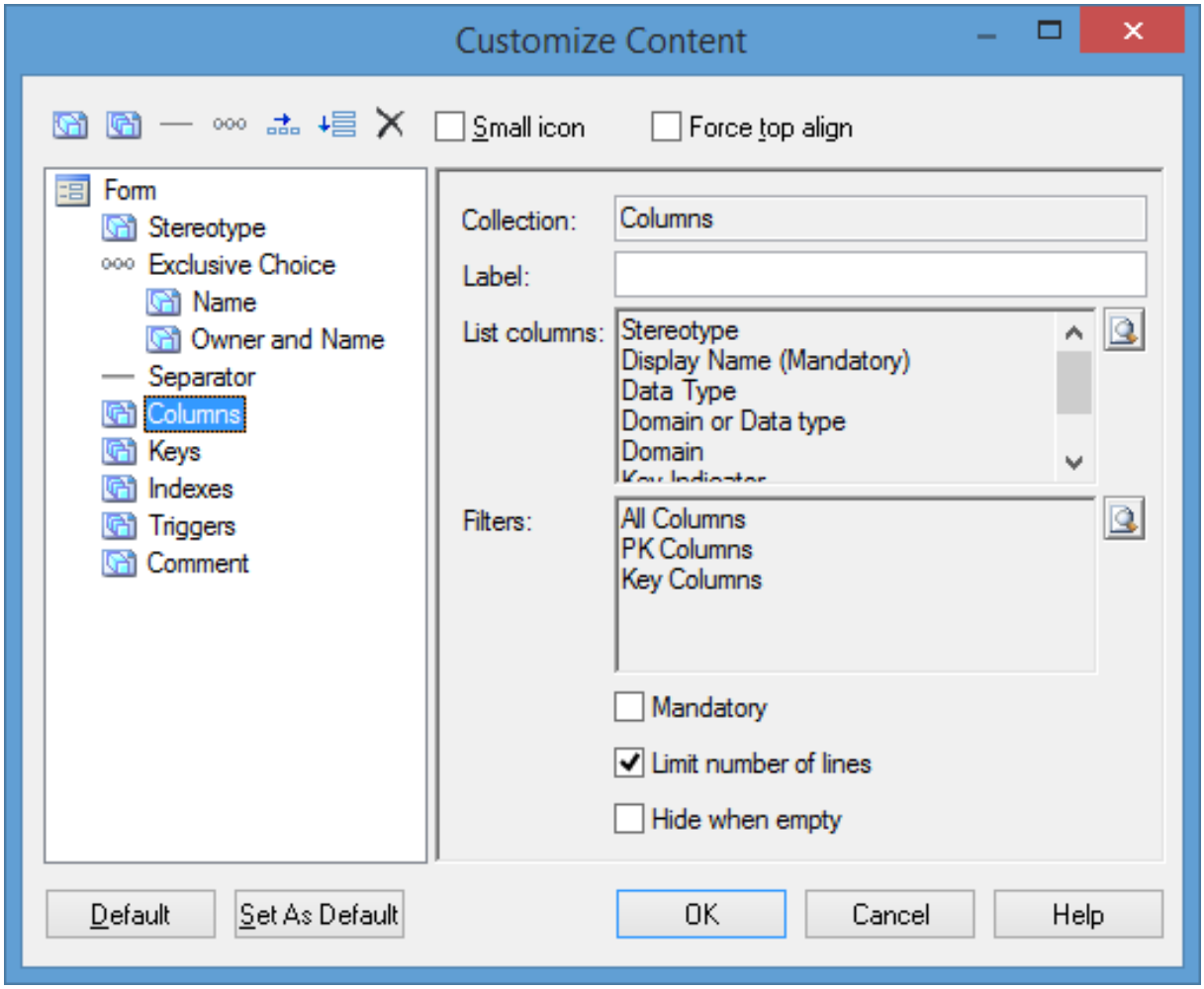


Click on an attribute in the list to set the following options:

Option	Description
Label	Specifies a label to use in the Display Preferences dialog in place of the attribute's name.
Prefix/Suffix	[text items] Insert a prefix and suffix before and after the value of the attribute.
Alignment	[text items] Specifies the alignment of the text.
True/False	[booleans] Specifies a value to display for true and false. For example, for an attribute **`Annotated`**, you could specify **`Annotated`** and **`Not Annotated`** for true and false. By default, boolean properties are displayed if they are true and do not appear if they are false.
Mandatory	Specifies that the attribute must be displayed on the symbol, and is not available for deselection in the Display Preferences dialog.

1.5.4.4.2 Configuring the Display of Collections

A collection is a group of sub-objects or related objects. To enable the display of a collection, click the *Add Collections* tool, select collections or extended collections to display, and click *OK*.



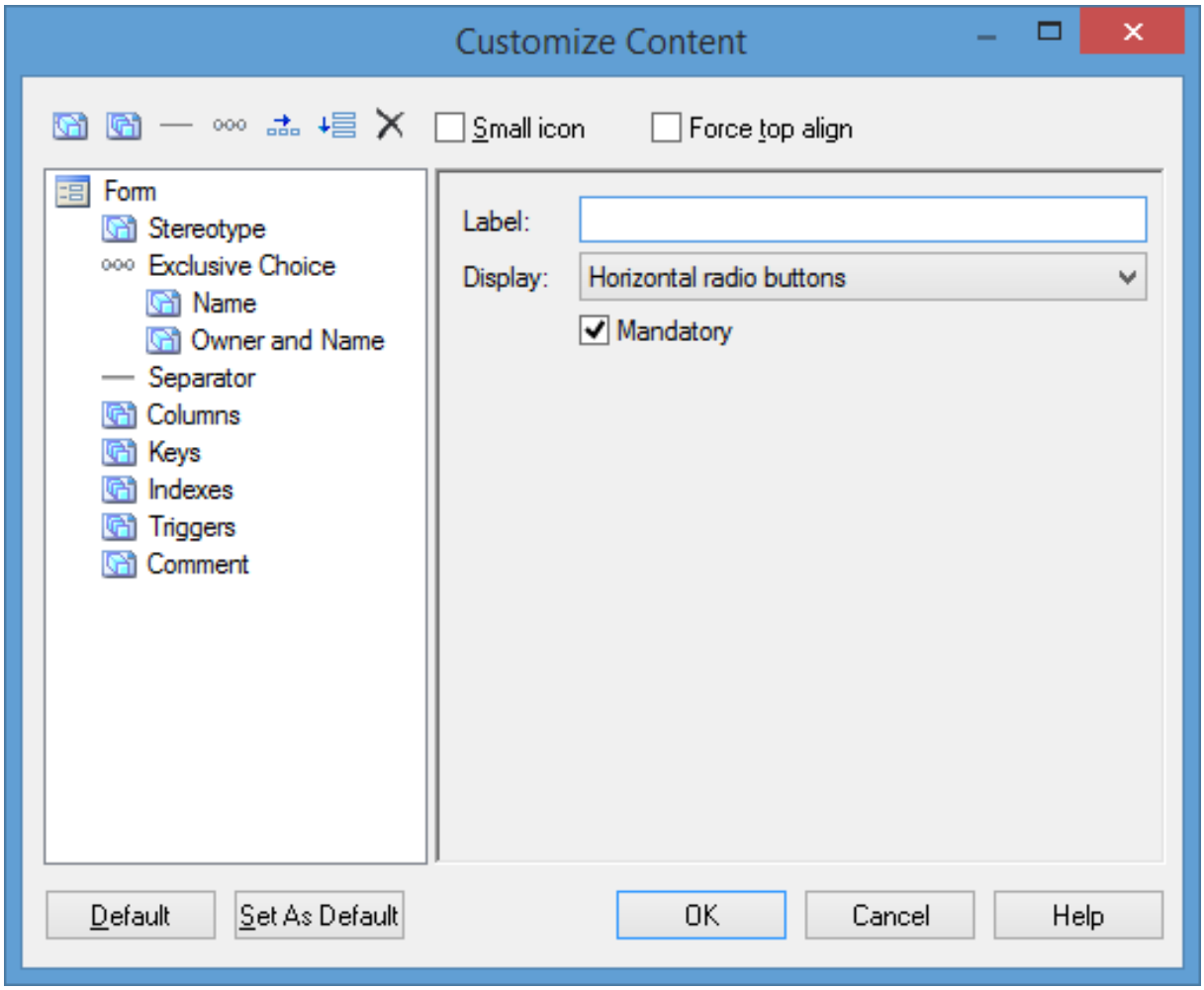
Click on a collection in the list to set the following options:

Option	Description
Label	Specifies a label to use in the Display Preferences dialog in place of the collection's name.
List columns	Specifies the properties displayed for each object in the collection. Click the *Select* tool beside the field to select attributes for display.
Filters	Specifies one or more filters to offer as radio buttons under the checkbox to select the collection in the Display Preferences dialog. Click the *Select* tool beside the field to open the Define Available Collection Filters dialog, enter a name for the filter and then click the ellipsis button in the *Filter Expression* column to define the parameters of the filter (see Defining a Filter Expression [page 127]).

Option	Description
Mandatory	Specifies that the collection must be displayed on the symbol, and is not available for deselection in the Display Preferences dialog.
Limit number of lines	Enables a field in the Display Preferences dialog allowing the user to limit the display of collection members to the number that they enter.
Hide when empty	Specifies that the symbol compartment containing the collection is hidden when there is nothing to display.

1.5.4.4.3 Configuring the Display of Exclusive Choices

An exclusive choice groups two or more attributes in the Display Preferences dialog, from which the user can choose to display one To add an exclusive choice, click the *Add Exclusive Choice* tool, and drag attributes onto it.



Click on an exclusive choice in the list to set the following options:

Option	Description
Label	Specifies a label to use for the choice in the Display Preferences dialog.
Display	Specifies whether the choice displays as horizontal or vertical radio buttons or as a combo list.
Mandatory	Specifies that the choice must be displayed on the symbol, and is not available for deselection in the Display Preferences dialog.

1.5.5 Importing and Exporting Model Graphics

You can import graphics to and export graphics from your PowerDesigner models.

To import an image into PowerDesigner, select *Edit* > *Import Image* to open the Select Image window (see Browsing for Images [page 222]), select an image and click *OK* to display it in the diagram. Imported images are saved in the model.

To export one or more symbols from PowerDesigner, select them in the diagram and then click *Edit* > *Export Image* to open a Save As dialog, select a format in the *Save As Type* list, enter the filename and click *OK*.

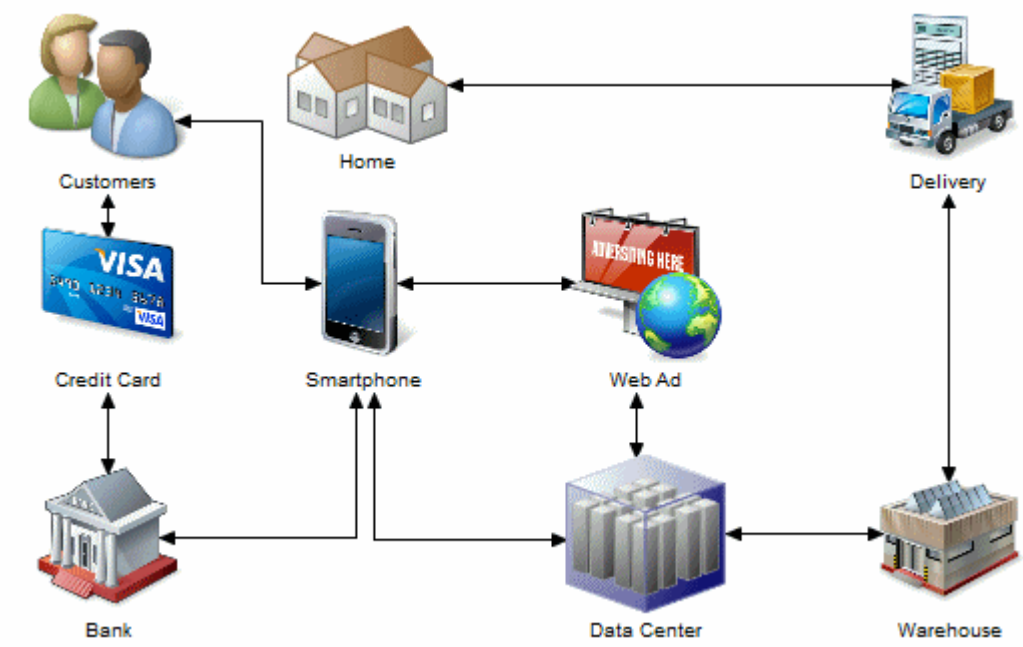
You can export symbols in any of the following file type formats:

File type	Extension
Enhanced Metafile	EMF
Bitmap	BMP, DIB, RLE
JPEG Compliant	JPG, JPEG
Portable Network Graphic	PNG
Graphics Interchanged Format	GIF
Tagged Image File Format	TIF, TIFF
Scalable Vector Graphics	SVG

> **i Note**
>
> If you export symbols or cut or copy them to the Clipboard, you must select *Edit* > *Export in Color* to retain them in color.

1.6 The Free Model (FEM)

A free model (FEM) provides a context-free environment for modeling any kind of objects or systems. You can use it to sketch simple visualizations or create entirely new forms of model by defining your own concepts and graphical symbols through an extension file.



To create a new, empty FEM - Select *File* > *New Model*, and choose *Free Model*.

For information about extending objects from their property sheets, see Extending Objects [page 167]. For detailed information about all the extension mechanisms available in PowerDesigner, see *Customizing and Extending PowerDesigner* > *Extension Files* .

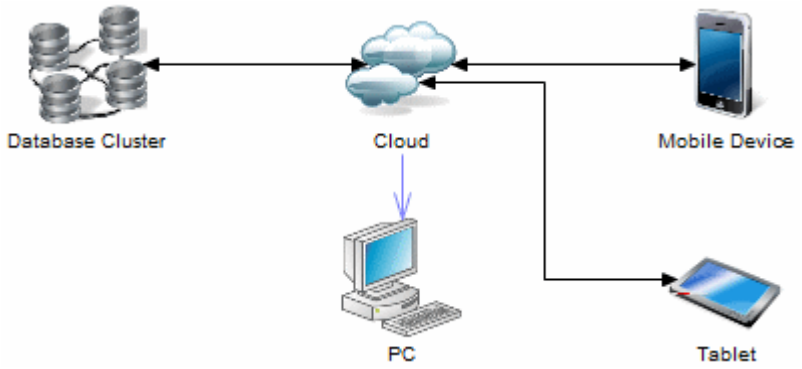
1.6.1 Sketching Objects with the Free Model

You can quickly sketch a visualization of any concepts by creating extended objects in a free model and giving them appropriate symbols.

Context

Procedure

1. [optional] Select *Tools* > *General Options* and select the *Edit in place after creation* tool to allow you to name objects immediately after creating them without opening their property sheets.
2. Select *File* > *New Model*, select *Free Model*, specify a model name, and click *OK*.
3. Use the *Extended Object* tool in the *Free Diagram* toolbox to create a basic object.
4. Right-click anywhere in the diagram to drop the tool and then right-click on your object and select *Change Image* to select a new image for the image (see Browsing for Images [page 222]).
5. Create additional objects as necessary and link them as appropriate using the*Extended Link* tool.



1.6.2 Creating New Types of Model with the Free Model

You can define a set of extensions and attach them to a free model to model entirely new concepts.

Context

Procedure

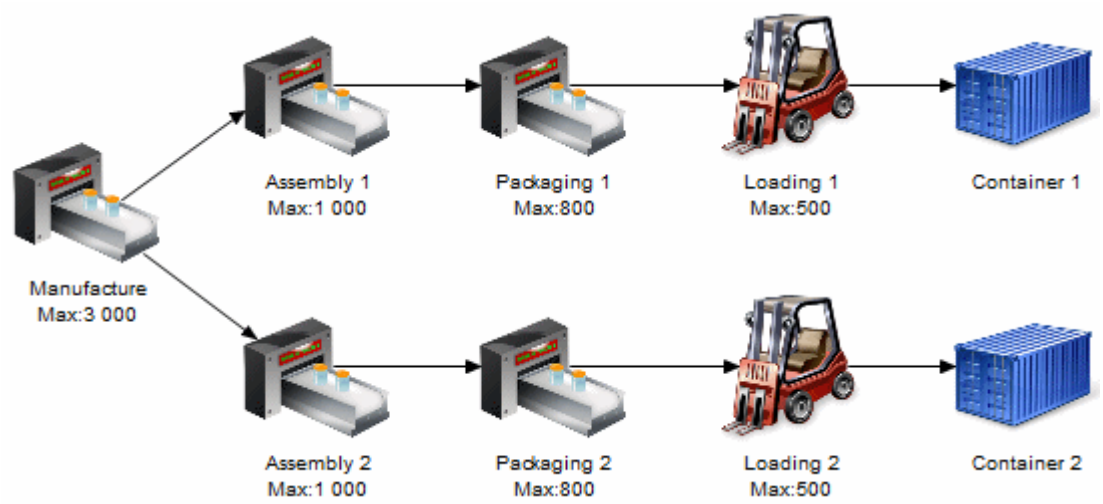
1. Select *Tools* > *Resources* > *Extensions* > *Free Models*.
2. Click the *New* button, enter an appropriate name, select to copy from the default template, and click *OK* to open your extension file in the resource editor.
3. Right-click the Profile category, select *Add Metaclasses*, click the *PdCommon* sub-tab, select the **`ExtendedObject`** metaclass, and click *OK*.
4. For each new concept that you want to define, right-click the **`ExtendedObject`** metaclass and select *New* > *Stereotype*, enter an appropriate name, and select the *Use as metaclass* and the *Toolbox custom tool* options.

Stereotypes with these options selected have their own object lists under the *Model* menu, their own Browser categories and a tool to create them in the Toolbox

5. Right-click a stereotype and define extended attributes (new properties) and other extensions under it as necessary.

   For detailed information about working with extensions, see *Customizing and Extending PowerDesigner* > *Extension Files* .

6. To model with your extensions, select *File* > *New Model* , select **`Free Model`**, specify a model name, click the *Select Extensions* button, select your extension to attach it to the model, and click *OK*.
7. Begin modeling with your new concepts. You can, at any time, double-click your extension file in the **`Extensions`** category under your model in the Browser to modify your extensions.



1.7 The Repository

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

> **i Note**
>
> Before you can access the repository, an administrator must install and configure it and provide you with a user name and password.

The following sections provide information about installing and administering the repository:

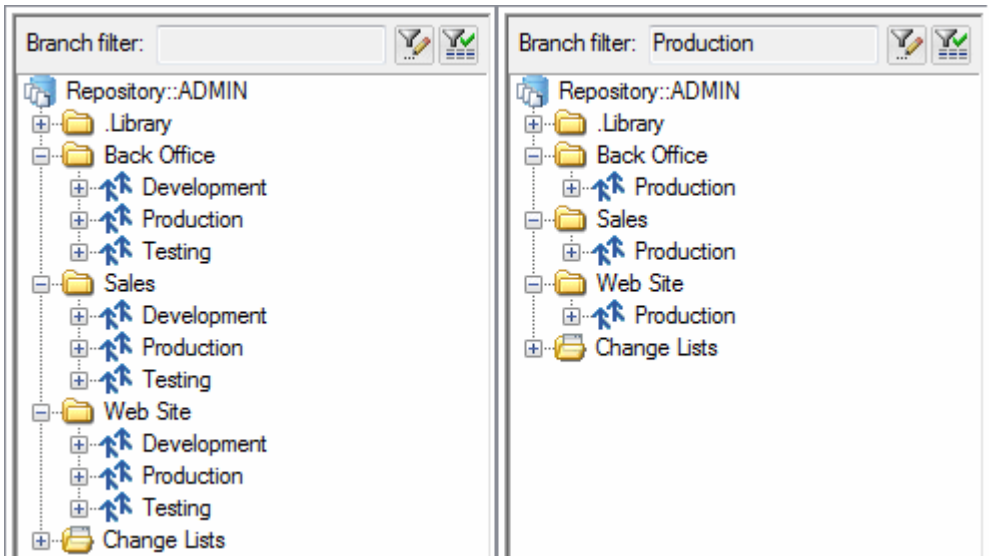
- Installing and maintaining the repository (see *Installation Guide* > *Installing the Repository*)
- Providing access to the repository by managing rights and permissions (see Controlling Repository Access [page 510])
- Providing customization of the PowerDesigner interface through permission, UI, and general profiles (see Customizing the PowerDesigner Interface [page 554])

- Deploying an enterprise glossary and library of reference models and other documents to be shared with all users (see Deploying an Enterprise Glossary and Library [page 540])
- Adding resource files to the repository and managing their sharing (see Sharing Resource Files via the Library [page 551])

1.7.1 Browsing the Repository

The Repository tab of the Browser displays the contents of the repository.

To expand all nodes, press the numpad plus sign (+). To collapse all nodes, press the numpad minus sign (-). To filter the Browser tree by branch (see Branching Version Trees [page 281]), click the *Customize Branch Filter* tool and select the branches you want to view in the *Branch Filter Selection* dialog. The filter works simply by name, so that branch names that are reused in different folders are treated as the same. You can toggle the use of the filter with the *Enable/Disable Branch Filter* tool:



> **i Note**
>
> You can only see repository objects if you have at least List permission on them (see Granting Access Permissions on Repository Items [page 534]).

A typical hierarchy of objects in the Browser *Repository* tab may include

- Root - [when connection is established] Displays the repository definition name, the user login, and the current branch (see Repository Root Properties [page 247]).
- Folders - Used to organize documents in the repository (see Repository Folders [page 245]).
- Library - A folder containing enterprise models that are pushed automatically to all users when they connect to the repository (see The Library [page 42]).
- Branches - Allow concurrent development of multiple versions of documents (see Branching Version Trees [page 281]).
- Projects - Container for models and other files (see Projects and Frameworks [page 46]).

- Documents - Models, multi-model reports, resource files, and external application files such as MS Office files, or graphics.
- Packages - Model packages (see Packages [page 143]).
- Diagrams, object categories, and objects - Are displayed only if the *Show objects* general option is selected (see Repository Options [page 396]).

1.7.1.1 Repository Folders

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

> **i Note**
>
> A PowerDesigner project (see Projects and Frameworks [page 46]) acts as both a repository document and a folder. Use the project permissions to control working with the project file itself (project diagram and any framework or dependency matrices) and the folder permissions to grant access to the models contained within the project, as you would for a repository folder.

You create a folder by right-clicking the root node, a project, or a folder, selecting *New* > *Folder*, and entering a name for the folder in the property sheet. Click the *Permissions* tab to add permissions for all appropriate users and groups for the folder and its contents (see Granting Access Permissions on Repository Items [page 534]).

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

You can move folders and documents in the repository by drag and drop if you have **`Full`** permission on the folder or model to be moved, and **`Write`** permission on the target folder (see Moving Documents in the Repository [page 258]).

You can delete a folder if you have **`Full`** permission on it and all its contents (see Deleting Documents, Document Versions, and Folders [page 295]).

1.7.1.2 Repository Document Properties

Double-click any item in the repository to see its repository property sheet.

The *General* tab displays read-only information about the object:

Property	Description
Name/Code/ Comment	Display the name, code, and comment of the object. Repository folders have an editable name, but no code or comment.
Type	[not available for folders] Specifies whether the item is a PowerDesigner model or package, or another type of document or object.

Property	Description
Location	[folders, documents, and model objects] Displays the full path to the folder or document. [model packages]: Displays the hierarchy of packages in the model.
Status	[not available for folders] Indicates the status of the document, which can be Updatable, Frozen (see Freezing and Unfreezing Document Versions [page 275]), or Locked (see Locking and Unlocking Documents [page 276]).

Version Info Tab

The *Version Info* tab displays information about the current version of the object:

Property	Description
Creation	Displays the name of the user who created the item in the repository, and the date of creation.
Last modification	Displays the name of the user who last modified the item, and the date of this modification.
Generated From Origin Object	[model objects only] When the current object has been generated from another object, 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	[not available for folders] Displays the number, branch name, and check-in comment for the current version.

Versions Tab

The *Versions* tab displays a tree of the versions of the document including any versions under branches. For more information about this tab, see Document Versions [page 273].

The following tabs are also available:

- *Definition* [models, packages, and model objects only] - contains annotations and descriptions of the object (see Definition Tab [page 110])
- *Attributes* [models, packages, and model objects only] - displays additional properties for these items.
- *Collections* [models, packages, and model objects only] - displays a list of collections or related objects (but not sub-objects) for the item. For example, the Collections tab in the property sheet of a CDM entity may display the business rules attached to the entity, but will not display entity attributes.
- *Permissions* [models, packages, folders and other documents only] – lists those users and groups with permissions on the document (see Granting Access Permissions on Repository Items [page 534]).

1.7.1.3 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.

In addition, the following tabs are available:

- *LDAP* - lists LDAP configuration parameters to enable remote LDAP users to access the repository (see Connecting to an LDAP Server for User Authentication [page 513]).
- *SMTP* - lists parameters for connecting the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]).
- *Passwords* - lists the requirements of the policy which controls the qualities of user passwords for the repository (see Defining a Password Policy [page 518]).
- *Permissions* – lists those users and groups with permissions on the entire repository (see Granting Access Permissions on Repository Items [page 534]).
- *Version Info* – specifies when the repository was created and last modified (see Version Info Tab [page 115]).

> **i Note**
>
> In addition to the standard version properties, the repository root also has a *Repository ID*, specifying the unique identifier of the repository, which is used to identify the associated glossary (see The Glossary [page 38]) and library (see The Library [page 42]).

1.7.2 Checking Documents into the Repository

You check documents into the repository to share them with others. You must have at least the **`Submit`** permission on a document (or on one of its packages) to be able to check it into a change list for approval or at least the **`Write`** permission to make changes directly to the repository version.

Procedure

1. In the *Local* tab of the Browser, select the documents to check in, then right-click and select *Check In* to open the:
   - Check In Project window - if you selected a project (see Checking In a Project [page 248]).
   - Check In Document window - if you selected a single model or file (see Checking In a Model, Resource File, or Other Document [page 249]).

- Check In Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see Checking In Multiple Documents [page 251]).

2. Select whether to check into a change list (see Change Lists [page 259]). You can choose from:
   - **<None>** - [if you have **Write** permission on the target folder] Allows you to check the documents in directly without review.
   - **<New change list>** - To create a new change list to contain the checked in documents.
   - **Change list_<xxxx>** - To add the documents to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list.

     > **i Note**
     >
     > If you are checking in a new version of a document that is already contained in a draft change list, then this field is read-only.

3. Enter a comment to explain the purpose of your check in, select any other appropriate parameters (see Check-In Parameters [page 252]), and click *OK* to check in the document.

   If you check into a change list, you can add other documents to it as necessary through other check ins before submitting it (see Creating, Building, and Submitting a Change List [page 260]). Once the change list has been approved by the reviewers, your changes are integrated into the official repository versions of the documents and made available to other users. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) you will receive an email at the moment of submission of your change list and when it is approved, rejected, or returned to you for further revision.

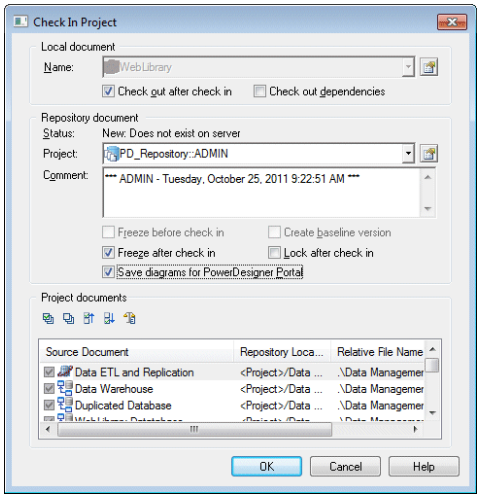
4. If you are checking in a document that is already stored in the repository, the Merge Changes to Repository Document dialog (see Resolving Conflicts During Check In [page 254]) may open to let you review the differences between your local document and the version in the repository. When you are satisfied, click *OK* to complete the check in.

   > **i Note**
   >
   > If you have changed the target (DBMS, OO language, etc) of the model outside of the target family (such as moving from Oracle 11g to SAP ASE 15.7), you will not be able to create a new version of the existing repository document and will need to save the model as a new document in the repository.

1.7.2.1 Checking In a Project

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

You arrive here by right-clicking a Project in the *Local* tab of the Browser, and selecting *Check In*. The dialog is populated with any project documents that have changes. The first time you check in a project, you must check in all its documents; subsequently, only documents with changes are listed and can be deselected if desired.



For information about the parameters available in this dialog, see Check-In Parameters [page 252].

1.7.2.2 Checking In a Model, Resource File, or Other Document

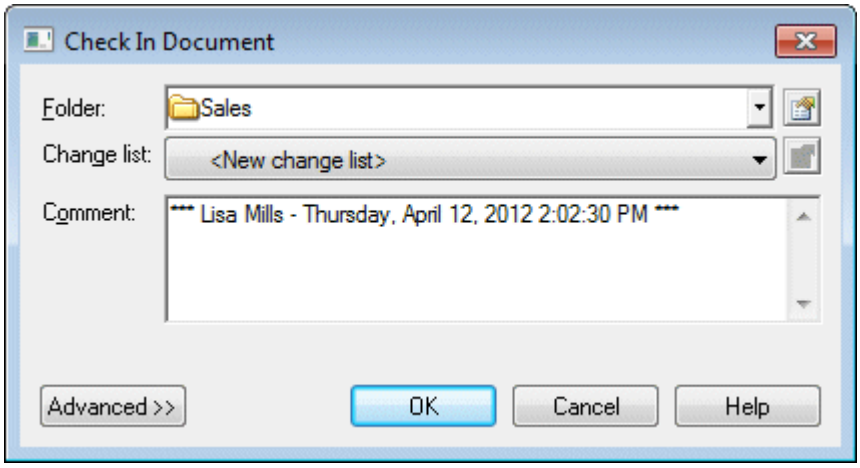
The Check In Document dialog guides you through checking in a PowerDesigner model, resource file, or other document.

You can arrive here by:

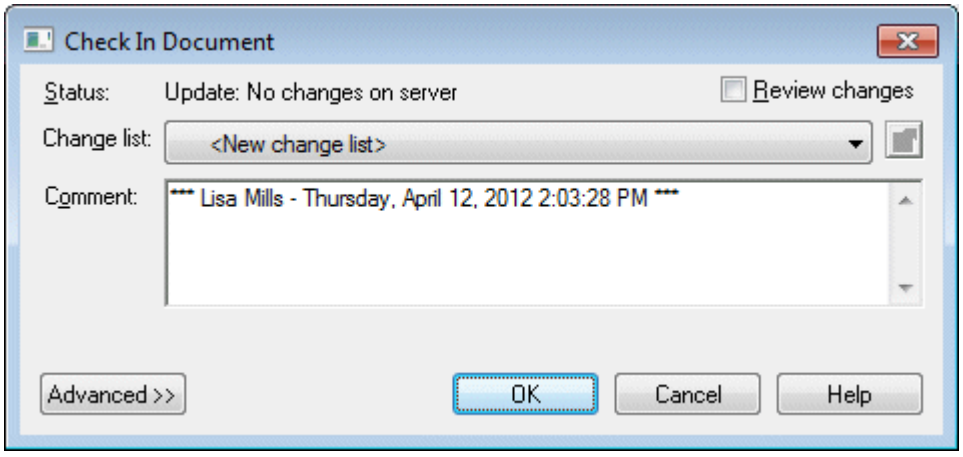
- Right-clicking a model or file in the *Local* tab of the Browser, and selecting *Check In*.
- Selecting a model or file in the Check In Multiple Documents dialog (see Checking In Multiple Documents [page 251]), and clicking the *Settings* tool.
- Clicking the *Add Document* tool in the Check In Multiple Documents window (see Checking In Multiple Documents [page 251]). You must specify the model or file to check in in the *Local Document* group box.
- Selecting a resource file, and clicking the *Check In* tool in the resource file list (see Sharing Resource Files via the Library [page 551]).

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

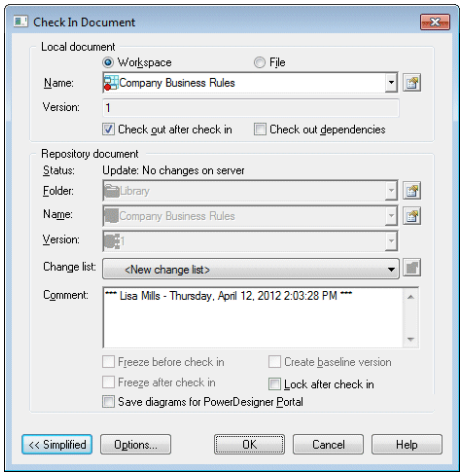
- Simplified (checking in a document for the first time) - Select a folder to check into (and a changelist if required or desired), enter a comment, and click *OK*:



- Simplified (checking in updates to a document already in the repository) - Choose whether to review your changes (see Resolving Conflicts During Check In [page 254]) (and a changelist if required or desired), enter a comment, and click *OK*:



- Advanced (when you click the *Advanced* button on either of the simplified dialogs):



For information about the parameters available in this dialog, see Check-In Parameters [page 252].

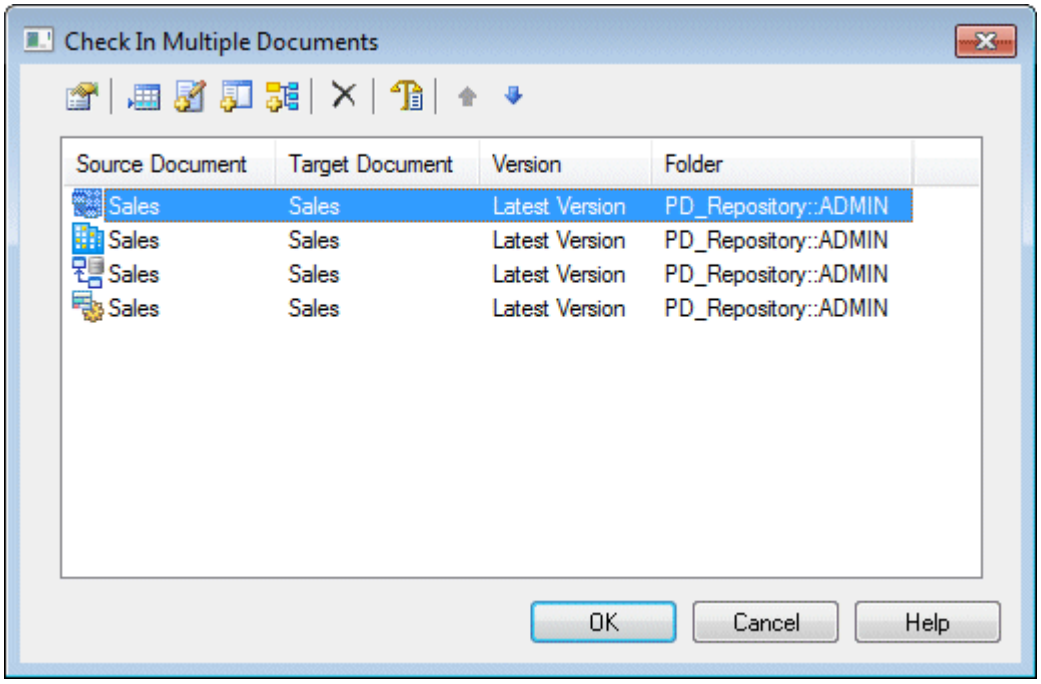
1.7.2.3 Checking In Multiple Documents

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

You can arrive here by:

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

> **i Note**
>
> You cannot select projects, which themselves contain multiple documents (see Projects and Frameworks [page 46]), to check in through this dialog.



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

Tool	Description
	Settings – Opens the Check In Document window to let you specify check in parameters for the selected document (see Checking In a Model, Resource File, or Other Document [page 249]).

Tool	Description
	Add Document - Opens the Check In Document window to let you to specify a document to add to the list (see Checking In a Model, Resource File, or Other Document [page 249]).
	Add Modified Documents in Workspace - Adds all the documents modified in the workspace during the current session to the list.
	Add All Documents in Workspace - Adds all the documents in the workspace (including closed documents) to the list.
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation or other links to the list. When you check in PowerDesigner models related by shortcuts, the models being referenced are checked in before the models containing the shortcuts.
	Delete – Removes the selected document from the list.
	Check In Options – Opens the Check In Options window to allow you to specify the update mode for changes (see Resolving Conflicts During Check In [page 254]).
	Move Up – Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

1.7.2.4 Check-In Parameters

These check-in parameters allow you to control details of your check in.

> **i Note**
>
> You can set default values for some of these parameters (see Repository Options [page 396]). Each check in is processed as one transaction. When you first check in a large model, the server creates large rollback logs and uses a large number of locks, which may require you to configure your server to allow such large transactions. Check ins of subsequent versions use much smaller transactions.

The *Local Document* group box contains the following parameters:

Parameter	Description
Workspace or File	[not for project] Specifies and displays the document name by its workspace name or file name and path.
Name	Specifies the name of the document to check in. Depending on whether you have selected Workspace or File, this is a: • Model name - click the *Properties* tool to open its property sheet • File name and path - click the *Browse* tool to open a file selection dialog.

Parameter	Description
Version	[not for project – read-only] Specifies the current version of the model or file in the repository, if available.
Change list	Specifies the change list to check the item into (see Change Lists [page 259]). You can choose from the following: • **<None>** - [if you have **Write** permission on the target folder] Allows you to check the documents in directly without review. • **<New change list>** - To create a new change list to contain the checked in documents. • **Change list_<xxxx>** - To add the documents to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list. **i Note** If you are checking in a new version of a document that is already contained in a draft change list, then this field is read-only.
Check out after check in	[models and resources only] Updates the local version after check in by checking out the repository version to ensure that you have the most up-to-date version where other users may be working concurrently.
Check out dependencies	[projects and models only] Checks out additional information about cross-model depencies during the check out after check in, to enrich dependencies and impact analysis information in future situations where you may be without a repository connection (see Obtaining Cross-Model Dependencies Without a Repository Connection [page 501]).

The *Repository Document* group box contains the following parameters:

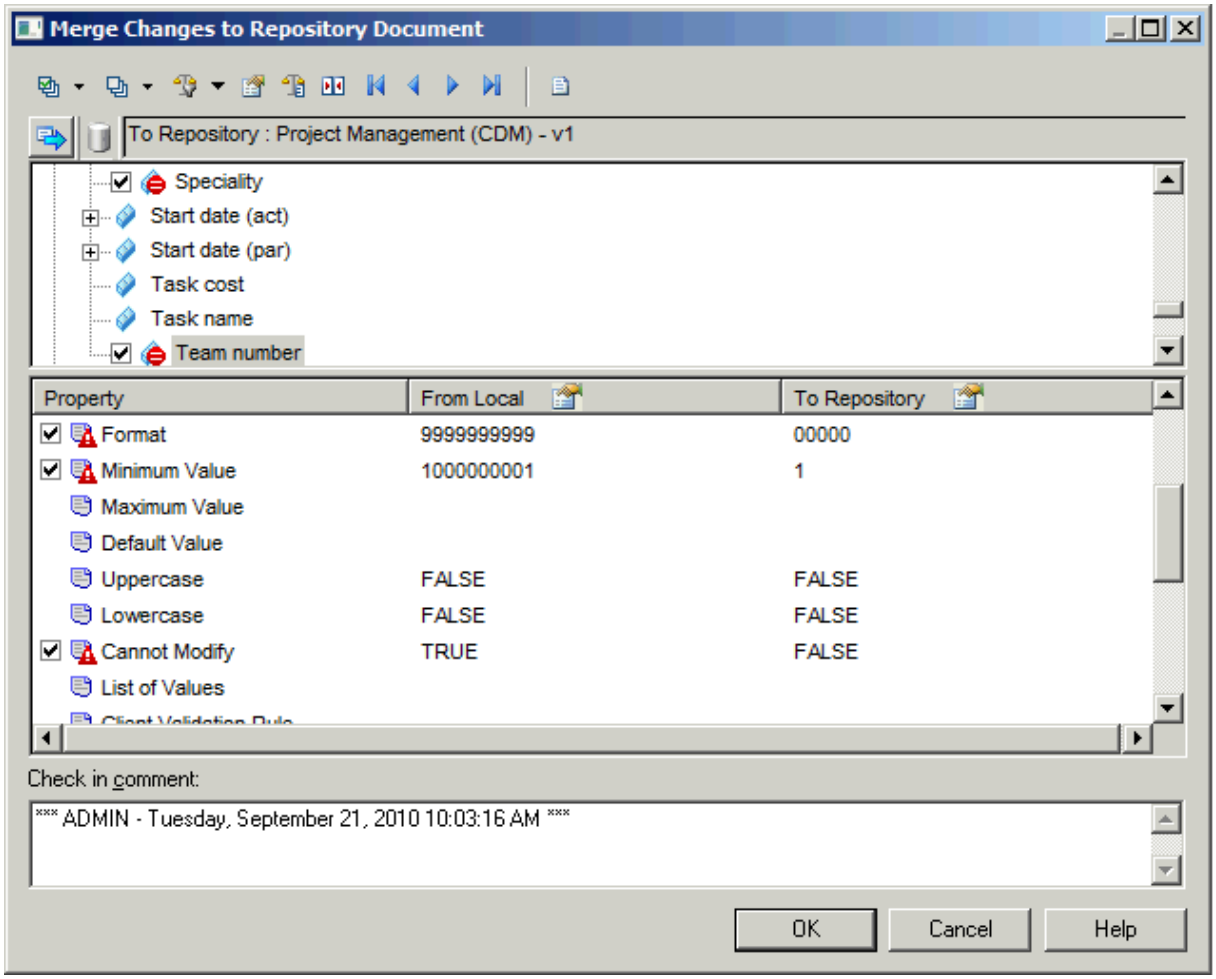
Parameter	Description
Folder/Project	[read-only after first check-in] Specifies the location where the document is checked in. Click the *Properties* tool to open the property sheet of the folder. If you do not specify a target folder, the document is checked in at the root of the browser. You should grant individual access permissions to this document.
Name	[not for project – read-only] Specifies the name of the document to apply your changes to. Click the *Properties* tool to open the property sheet of the document.
Version	[not for project – read-only] Specifies the version of the document to apply your changes to.
Comment	Specifies a comment to describe the purpose of the check in.

Parameter	Description
Freeze before check in	Freezes the previous versions of any modified objects in the repository, and creates a new version to accept the changes to be checked in (see Freezing and Unfreezing Document Versions [page 275]).
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 are made to new versions (see Freezing and Unfreezing Document Versions [page 275]).
Create baseline version	Creates a completely new version of the model, instead of simply versioning the objects that have changed. This option is available only if you have **`Full`** permission on the document. The Merge Changes to Repository Document dialog does not open. Baseline versions may be quicker to check in and out, but they require considerably more disk space than standard versions. Occasionally, the repository cannot process a standard version; in this situation, you should try to create a baseline version that can be used as a starting point for a new development step.
Lock after check in	Locks the document after check in (see Locking and Unlocking Documents [page 276]).
Save diagrams for PowerDesigner Portal	[not for resources] Prepares snapshots of the model diagrams that can be viewed in PowerDesigner Web (see *PowerDesigner Web*). Dependency matrices cannot be viewed in PowerDesigner Web.

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. Only project documents that are inside the project are listed here. Documents that are outside the project and only attached to it must be checked in separately.

1.7.2.5 Resolving Conflicts During Check In

The Merge Changes to Repository Document dialog lets you inspect and approve (or reject) the changes you are checking into the repository. To filter the tree and show only the changes, click the *Change Filter* tool and select *Show All Changes*. By default, all your changes are selected to be applied.



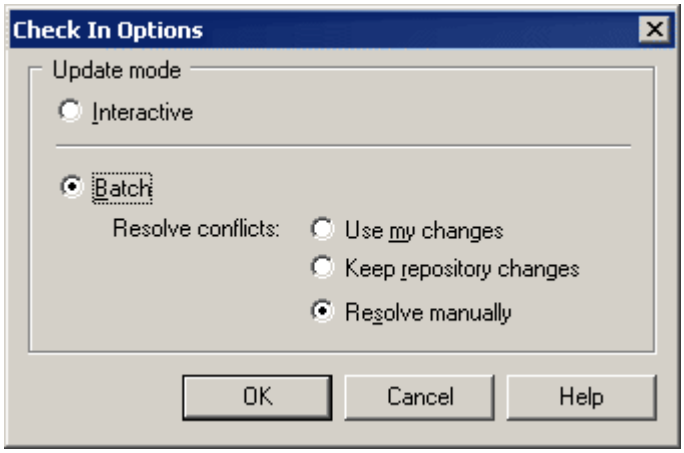
The window is divided in three parts:

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

For detailed information about using this dialog, see Merging Models [page 369].

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

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



The following modes are available:

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

1.7.2.6 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.

See also Checking Out Packages [page 272].

1.7.2.7 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.

1.7.2.8 Duplicating Models in the Repository

If you want to duplicate a model that has been checked into the repository, you must first check it out and then save it as a new model (with a new GUID).

Procedure

1. Select the model to duplicate in the repository, right-click it, and select *Check Out* to check it out into your local workspace.
2. Select the model in your local workspace and click *File* > *Save as New Model*.
3. Select an appropriate folder and filename and click *Save* to create the duplicate model.

   > **i Note**
   >
   > The duplicate model replaces the original model in the workspace, but you can verify that the duplication has succeeded by reading the new path in the PowerDesigner window title bar.

4. Right-click the duplicate model and select *Check In* to open the Check in Document dialog.
5. Specify the folder to check the model into and any other check-in options (see Check-In Parameters [page 252]), and click *OK*.

1.7.2.9 Undoing a Check In

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

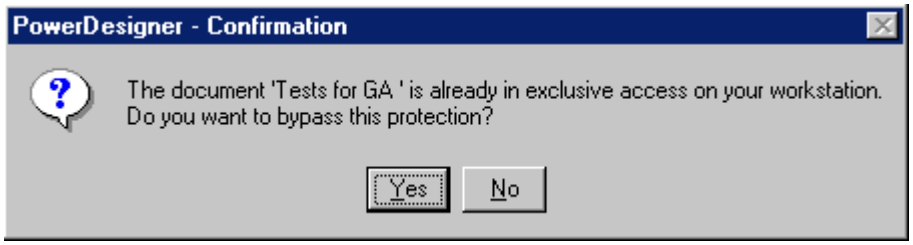
Since a check in to the repository does not necessarily create a new document version (see Freezing and Unfreezing Document Versions [page 275]), such a deletion may not always be possible.

1.7.2.10 Temporary Protection for Updates to Repository Documents

When you check in, freeze, lock, or delete a document in the repository, all the versions of the document are protected until the transaction is completed, and other users will receive an error message if they attempt to update the document. For example, you cannot lock a version that another team member is checking in, or freeze a version that someone is locking.

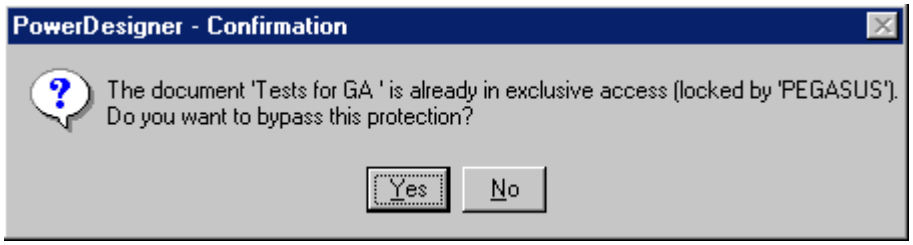
In rare cases, if the connection to the repository is interrupted during an update to a document, the document may remain protected, and you may not be able to update the document in the future until you bypass the protection or cancel the action:

- If the interruption occured during an operation launched from your workstation (PowerDesigner verifies the machine name, not the connected user), then you can bypass the temporary protection and continue your operation:



> **i Note**
>
> If you have two PowerDesigner sessions running on your workstation and are performing an operation on a document in one session, and want to perform another operation on the same document from the second session, then you should wait for the first operation to complete.

- If the interruption occured during an operation launched from another machine, you should ask an administrator to unblock it. The administrator will verify that no other users are currently updating the document and then perform an action on the document (for example, locking it) to display the message box and click *Yes* to release it:



> **⚠ Caution**
>
> Bypassing the protection when you are not entitled to may severely damage the document.

1.7.2.11 Moving Documents in the Repository

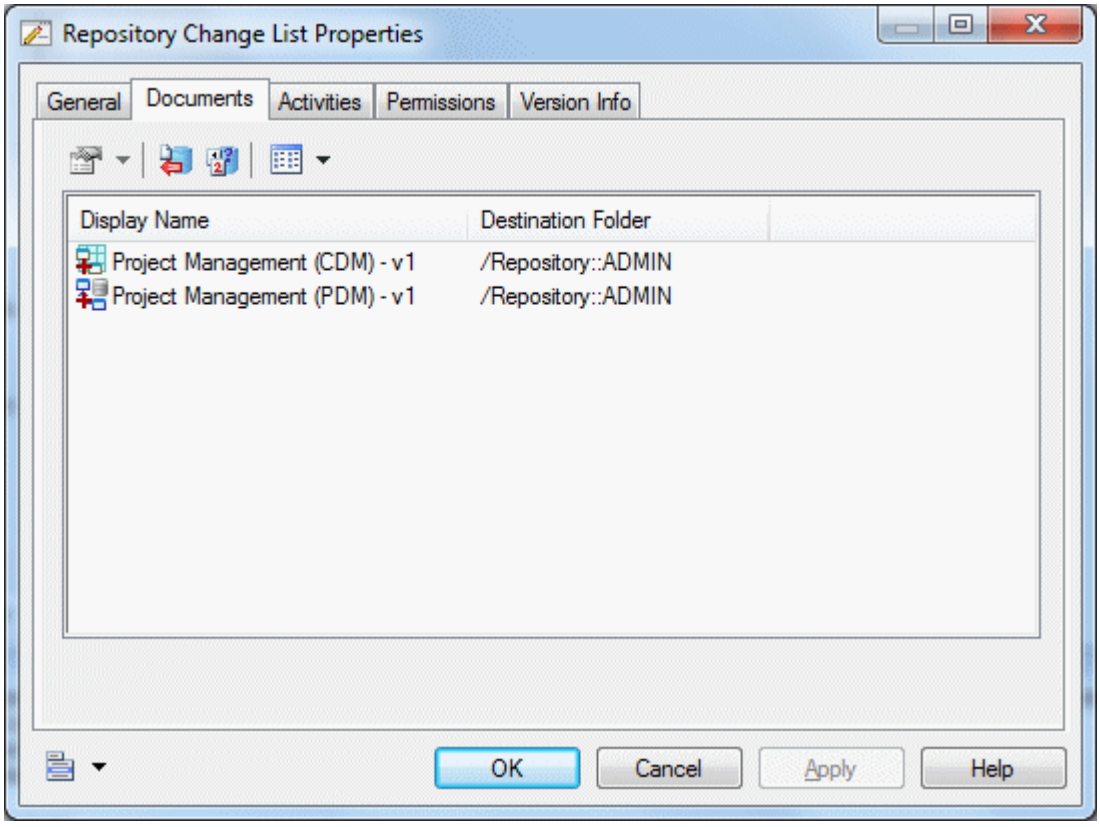
You can move a document from one folder to another in the repository by drag and drop. You cannot move a single document version nor can you move objects from one model to another.

To move a folder or a model you must have **`Full`** permission on the folder or model to be moved, and **`Write`** permission on the target folder.

> **i Note**
>
> You can move a document that is not in any branch structure (see Branching Version Trees [page 281]) to beneath a branch, but you cannot move a document from beneath a branch to another branch or to outside any branch. You cannot move a project (see Projects and Frameworks [page 46]) inside another project.

1.7.3 Change Lists

Users with the **`Submit`** permission on a repository folder or model cannot write directly to that location, but must instead check their changes into a change list, to be reviewed by one or more reviewers. You can add one or more models or other documents to a change list all at once or over several check ins. Documents checked into a change list are stored in a special folder in the repository and their contents are not available to other users until they have been approved.



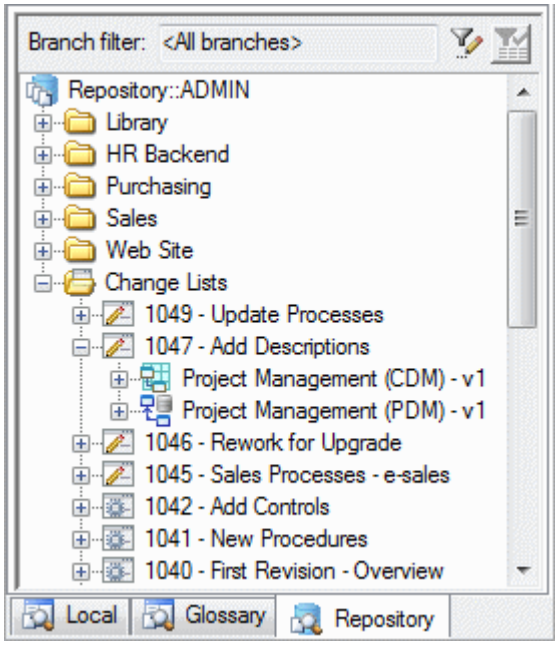
You can create a change list to propose:

- Check in of a new model or other document or changes to an existing model or other document (see Creating, Building, and Submitting a Change List [page 260]).
- Integration of repository documents from one branch to another (see Integrating Changes from One Branch to Another [page 284]).

An administrator will determine the policies by which documents can be checked into the repository on a global or folder-by-folder basis (see Controlling Repository Access [page 510]). PowerDesigner supports the following kinds of review policy:

- Simple review - Change lists submitted by users with the **`Submit`** permission are reviewed by users with the **`Write`** or **`Full`** permission.
- Peer review - Users with the **`Write`** or **`Full`** permission voluntarily submit change lists for review.
- Direct check in - The **`Submit`** permission and change lists are not used, and users all check in changes without review.

You can consult all the open change lists that you have created and those which are awaiting your review under the *Change Lists* folder in the Browser *Repository* tab:



To consult any change list, whether open or closed, containing models for which you have at least **`Read`** permission, select *Repository* > *Change Lists*.

1.7.3.1 Creating, Building, and Submitting a Change List

When you check one or more documents into a change list, the change list is created in the repository. You can check further changes to these documents into the change list or add other documents to it. Once you are satisfied with the set of changes contained in your change list, you can submit it for review. After submission, you can no longer edit it.

Context

> **Note**
>
> If you have only *Submit* permission on a document, you are required to check your changes into a change list, while if you have *Write* permission or higher, you can check the document in directly or choose to create a change list to have your changes reviewed by a peer.

Procedure

1. Choose one or more models, projects, or other documents in your local workspace, right-click them, and select *Check In* (see Checking Documents into the Repository [page 247]).
2. Choose **`New change list`** in the *Change list* field, enter a *Comment* to explain your changes, set any other appropriate options (see Check-In Parameters [page 252]), and click *OK*.

   If you are checking in a document that is already stored in the repository, the Merge Changes to Repository Document dialog (see Resolving Conflicts During Check In [page 254]) may open to let you review the differences between your local document and the version in the repository. When you are satisfied, click *OK* to complete the check in.

   The change list is created in the repository, and your model updates are saved to it. You can consult the change list under the *Change Lists* folder in the Browser *Repository* tab, and has a draft icon:

3. [optional] Check in further changes to the documents in your change list or add other documents to it by selecting it in the change list dialog, until you have a complete set of changes that you want to submit for approval.

   > **Note**
   >
   > The documents that you add to a change list must form a coherent set of changes that can be appropriately reviewed.You can only add documents to a change list if one or more other users has explicit *Write* permission on all the documents included. If you try to add a document that does not share at least one competent reviewer with the documents already included in the change list, then your check in will be canceled and the change list not updated.

4. Right-click the change list in the *Repository* tab of the Browser and select *Submit* to open its property sheet.
5. Enter a final comment, and click *Submit* to submit your change list for review.

   The icon of the change list changes to show that it has been submitted:

   You can no longer edit it or check further changes into it. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]), you will receive an email confirming the submission, and any users with **`Write`** or higher permission on all the documents

included in the change list will receive an email inviting them to review it. When the change list is approved, rejected, or returned to you for revision, you will receive a further email.

> **i Note**
>
> If you have *Write* permission or higher, and have voluntarily created a change list for peer review, you can, at any time, change your mind and approve your change list yourself.

1.7.3.2 Reviewing a Change List

If you have **`Write`** or higher permission on all the documents included in a change list submitted by another user, then you can review it and determine whether to approve the changes, reject them, or return the change list to the user requesting her to revise it.

Context

> **i Note**
>
> This procedure explains how to review a change list created by a user to propose changes to one or more repository documents. For information about reviewing a change list created by a user to propose integrating documents from one branch to another, see Reviewing an Integration Change List [page 288].

Procedure

1. When you connect to the repository a message in the Output window notifies you if you have change lists to review. Open the *Repository* tab of the Browser and expand the Change Lists folder to view all the change lists awaiting your review, which are identified by their "submitted" icon:

   > **i Note**
   >
   > If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]), you will receive an email when the user submits the change list, inviting you to review it.

2. Double-click a change list to view its properties (see Change List Properties [page 265]), and read the *Comment* field to understand the reason for its submission.
3. Click the *Documents* tab and review the documents that are included in the change list. The documents in the list display status icons to identify whether further changes have been checked into the mainline version by other users since the change list version was created:

Change List Document Status	No Locks	Mainline Locked by You (Green Padlock)	Mainline Locked by Other (Red Padlock)
Change list creates new document (red cross)		N/A	N/A
Changes in change list version (red checkmark)			
Changes in both change list and mainline versions (yellow exclamation point and red checkmark)			

4. [recommended] Right-click any models that are shown as having subsequent changes on the mainline and select *Update from Mainline* to merge these changes into the change list version (see Resolving Conflicts During Check In [page 254]).

   Merging modifications from the mainline ensures that you are reviewing the change list changes in the most up-to-date context, and will also simplify the approval process.
5. Perform the following actions on a document by right-clicking it or using the toolbar:
   - *Check Out* - Check the submitted version of the document out into your local workspace to review it in more detail (see Checking Documents Out of the Repository [page 266]).
   - *Compare* - Compare the submitted version of the document in the change list with the mainline version in the repository (see Comparing Models [page 366]).
   - *Find in Open Change Lists* - Open the List of Change Lists filtered to display only open change lists containing the document.
6. Note your agreement with the proposed changes or explain why you will reject them or return them to the submitter for modification in the *Comment* field of the change list, and then close it in preparation to approve, return, or reject it.
7. Right-click the change list in the Browser and select:
   - *Approve* - to approve the changes in the included documents and merge them into the mainline to make them available to all users with the necessary permissions. Enter any appropriate final comment and click the *Approve* button. For each PowerDesigner model included in the change list, an Integration dialog is displayed, in which you can review the changes one last time before integrating them into the mainline (see Resolving Conflicts During Check In [page 254]).

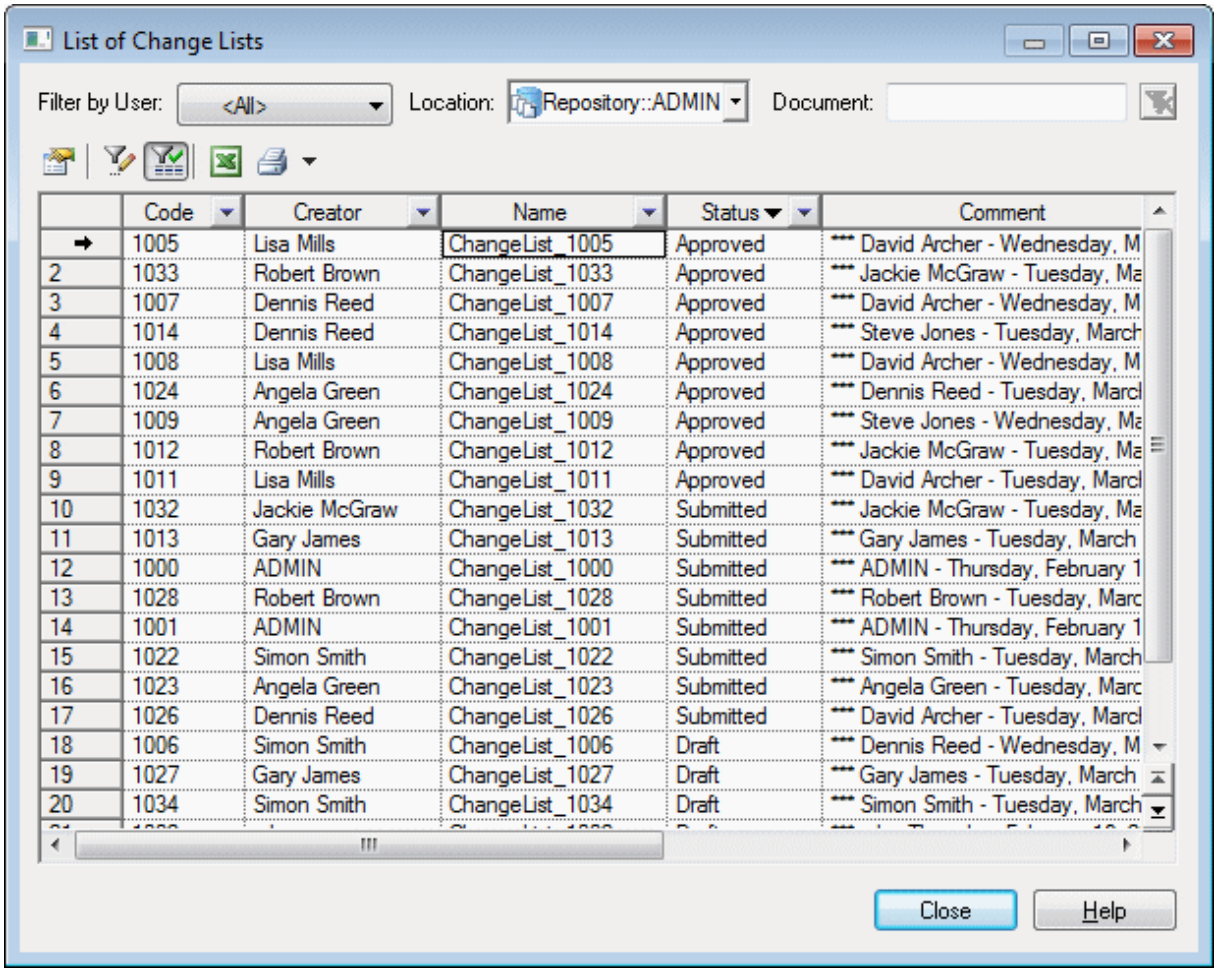
     > ⚠ Caution
     >
     > Once you begin the approval process, the mainline versions of all the documents in the change list are locked (see Locking and Unlocking Documents [page 276]) to enable the changes to be integrated as a coherent transaction. If you click *Cancel* before having integrated the entirety of the changes in each of the documents, then all the documents will remain locked until you complete the approval process. You can recommence the approval process at any time from the point at which you canceled it.

   - *Return for Revision* - to return the change list to the user with a request to make any changes required for you to approve it. Enter a comment explaining the changes to be made and click the *Return* button.
   - *Reject* - to reject the changes proposed in the change list. Enter a comment explaining your reasons for rejecting the change list and click the *Reject* button.

In each case, if an administrator has connected the repository to an SMTP server, the change list owner and any other potential reviewers receive emails notifying them of the approval, rejection, or return of the change list.

1.7.3.3 Consulting the List of Change Lists

To consult any change list, whether open or closed, containing models for which you have at least **Read** permission, select *Repository* > *Change Lists*.



	Code	Creator	Name	Status	Comment
➡	1005	Lisa Mills	ChangeList_1005	Approved	*** David Archer - Wednesday, M
2	1033	Robert Brown	ChangeList_1033	Approved	*** Jackie McGraw - Tuesday, Ma
3	1007	Dennis Reed	ChangeList_1007	Approved	*** David Archer - Wednesday, M
4	1014	Dennis Reed	ChangeList_1014	Approved	*** Steve Jones - Tuesday, March
5	1008	Lisa Mills	ChangeList_1008	Approved	*** David Archer - Wednesday, M
6	1024	Angela Green	ChangeList_1024	Approved	*** Dennis Reed - Tuesday, Marcl
7	1009	Angela Green	ChangeList_1009	Approved	*** Steve Jones - Wednesday, Ma
8	1012	Robert Brown	ChangeList_1012	Approved	*** Jackie McGraw - Tuesday, Ma
9	1011	Lisa Mills	ChangeList_1011	Approved	*** David Archer - Tuesday, Marcl
10	1032	Jackie McGraw	ChangeList_1032	Submitted	*** Jackie McGraw - Tuesday, Ma
11	1013	Gary James	ChangeList_1013	Submitted	*** Gary James - Tuesday, March
12	1000	ADMIN	ChangeList_1000	Submitted	*** ADMIN - Thursday, February 1
13	1028	Robert Brown	ChangeList_1028	Submitted	*** Robert Brown - Tuesday, Marc
14	1001	ADMIN	ChangeList_1001	Submitted	*** ADMIN - Thursday, February 1
15	1022	Simon Smith	ChangeList_1022	Submitted	*** Simon Smith - Tuesday, March
16	1023	Angela Green	ChangeList_1023	Submitted	*** Angela Green - Tuesday, Marc
17	1026	Dennis Reed	ChangeList_1026	Submitted	*** David Archer - Tuesday, Marcl
18	1006	Simon Smith	ChangeList_1006	Draft	*** Dennis Reed - Wednesday, M
19	1027	Gary James	ChangeList_1027	Draft	*** Gary James - Tuesday, March
20	1034	Simon Smith	ChangeList_1034	Draft	*** Simon Smith - Tuesday, March

In addition to the standard in-column filters, this list features the following meta-filters above the list:

- *User* - Shows only change lists for which the selected user is the creator or reviewer.
- *Location* - Shows only change lists containing models or other documents in the selected folder.
- *Document* - Shows only change lists containing documents beginning with the characters entered. The list filters dynamically after you have entered the first few characters.

You can filter on all three criteria in combination and your filter is remembered the next time you open the list. To remove all filters, click the *Clear* button.

> **i Note**
>
> Though the list displays all relevant results for direct connections to the repository hosted on most DBMSs, when connecting via the proxy or through certain DBMSs, an option is displayed below the list to allow you to restrict the number of results to improve performance:
>
>
>

1.7.3.4 Change List Properties

Change list property sheets provide information about the change list and give you tools to review the documents it contains.

The *General* tab displays information about the change list itself:

Property	Description
Name / Code	Specify the name and code of the change list. The name is editable, and the change list is listed in the repository browser as **<code> - <name>** (for example **1234 - My Changes**.
Comment / Comment History	The comment field contains your comments, including any comments entered when checking in models. The read-only comment history lists the comments entered by all users who have worked with the change list.
Location	Displays the full path to the change list in the repository.
Status	Displays the current status of the change list, which may be: • Draft - Not yet submitted for review, or returned for revision. • Submitted - Awaiting review. • Approved - Closed with changes integrated to mainline. • Rejected - Closed with changes rejected.
Owner	Displays the user who created the change list.

Documents Tab

The *Documents* tab lists the documents included in the change list. The following management tools are available:

Tool	Description
	Properties - Click the arrow to the right of the tool to open the property sheet of the : • *Original Mainline Version* - the version of the document stored in the mainline at the time of the change list's creation. • *Change List Version* - the version of the document included in the change list. • *Updated Mainline Version* - the version of the document stored in the mainline, which includes the changes merged from the change list version.
	Check Out - Checks out the change list version of the document (see Checking Documents Out of the Repository [page 266]).
	Compare - Opens the Select Models to Compare dialog, initialized to compare the change list version of the selected document with the current mainline version (see Comparing Models in the Repository [page 293]).

The following tabs are also available:

- *Activities* - Lists the actions taken on the change list (submission, approval, etc).
- *Permissions* – lists those users and groups with permissions on the change list (see Granting Access Permissions on Repository Items [page 534]). For draft change lists, the owner has **`Full`** permission. For submitted change lists, the owner has **`Read`** permission and reviewers have **`Write`** permission. For closed change lists, the permissions are restricted to the lowest permission that a user has on all of the included documents.
- *Version Info* - Displays the user who created the change list and the last user to modify it, along with the dates of creation and last modification.

1.7.4 Checking Documents Out of the Repository

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

Procedure

1. To check out a document not currently in your workspace, select the *Repository* tab of the Browser. If you are updating an existing file, select the *Local* tab.
2. Select the documents to check out, then right-click and select *Check Out* to open the:
   - Check Out Project window - if you selected a project (see Checking Out a Project [page 267]).

- Check Out Document window - if you selected a single model or file (see Checking Out a Document [page 268]).
- Check Out Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see Checking Out Multiple Documents [page 268]).

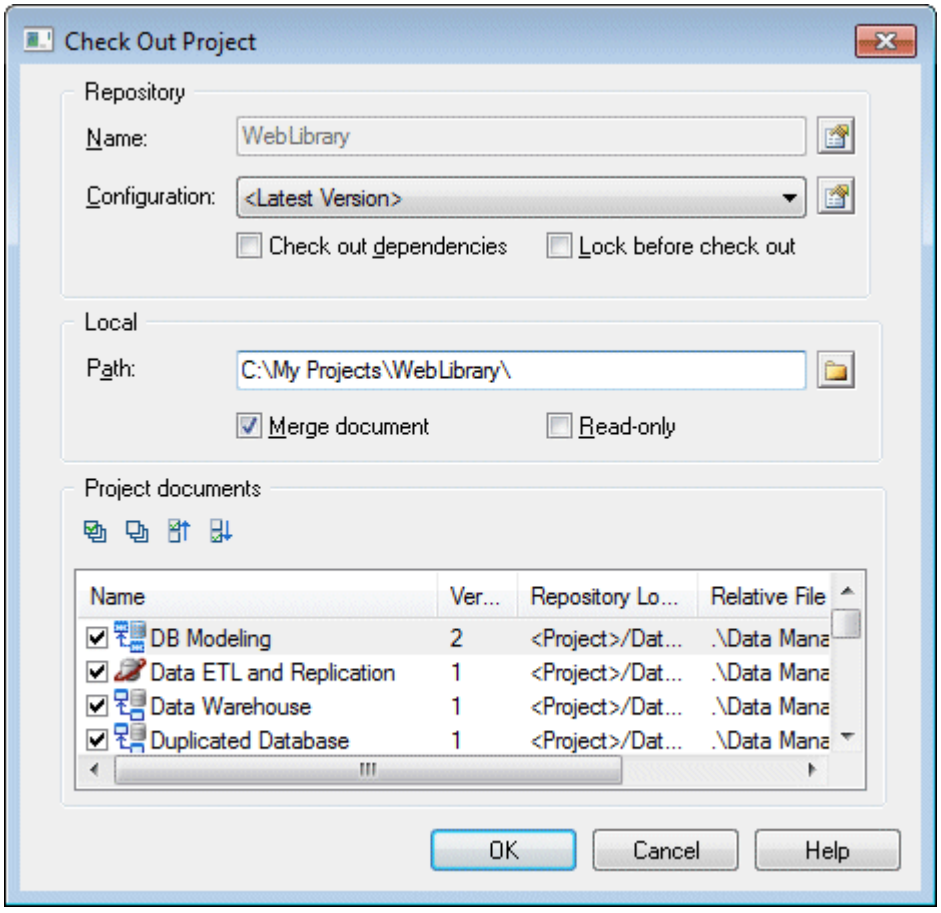
3. Select the appropriate parameters (see Checkout Parameters [page 270]), and click *OK* to check out the document.
4. If you are checking out a document that is already present in your workspace (and you have selected the *Merge document* option), the Check Out Document window (see Resolving Conflicts During Check Out [page 271]) will open to allow you to review the differences between the version in the repository and your local version. When you are satisfied, click *OK* to complete the check out.

1.7.4.1 Checking Out a Project

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

You can arrive here by:

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



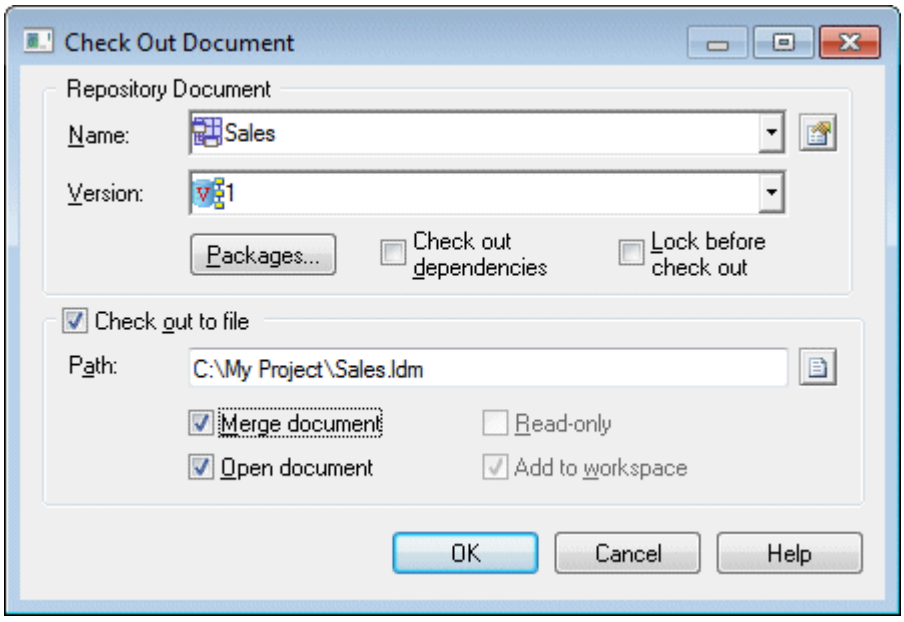
For information about the parameters available in this dialog, see Checkout Parameters [page 270].

1.7.4.2 Checking Out a Document

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

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

- Right-clicking a model or file in the *Repository* tab of the Browser and selecting *Check Out*.
- Right-clicking a model or file in the *Local* tab of the Browser and selecting *Update from Repository*.
- Selecting a model or file in the Check Out Multiple Documents window (see Checking Out Multiple Documents [page 268]) and click the *Settings* tool.
- Clicking the *Add Document* tool in the Check Out Multiple Documents window (see Checking Out Multiple Documents [page 268]). You must specify the model or file to check out in the *Local Document* group box.
- Select a resource file, and click the *Update from Repository* tool in a resource file list.



For information about the parameters available in this dialog, see Checkout Parameters [page 270].

1.7.4.3 Checking Out Multiple Documents

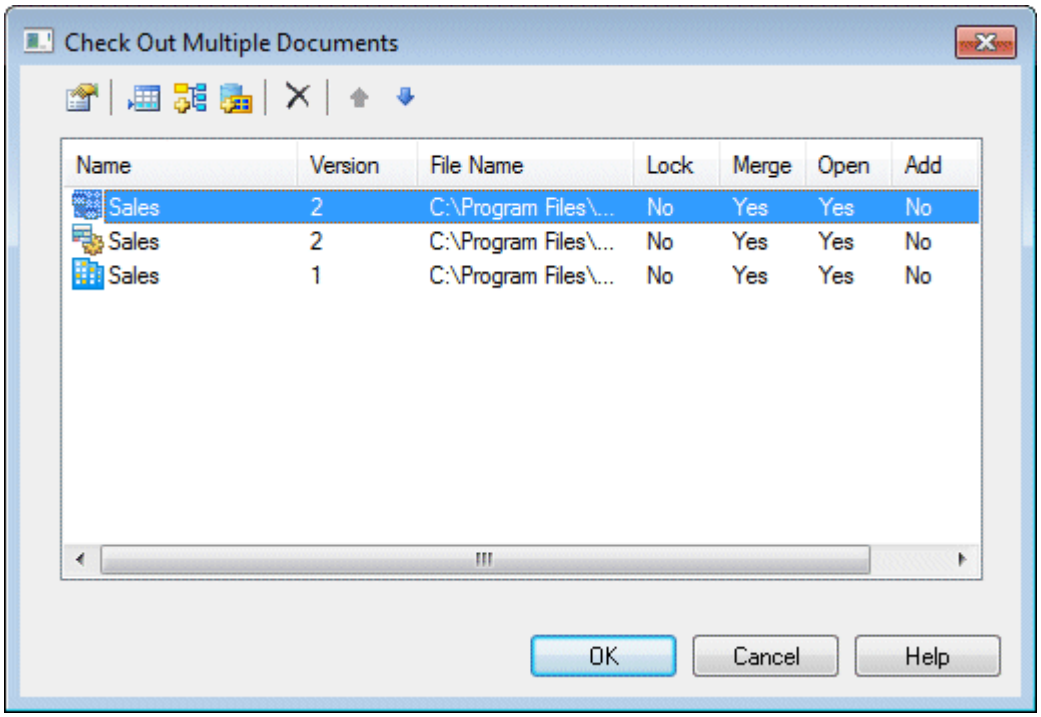
The Check Out Multiple Documents dialog is used when you select more than one document for check out.

You can arrive here by:

- Right-clicking the workspace root or a folder containing multiple documents (or manually selecting multiple documents in the *Repository* tab of the Browser, and selecting *Check Out*.

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

> **i Note**
>
> You cannot select projects, which themselves contain multiple documents (see Projects and Frameworks [page 46]), to check out through this dialog.



The following tools are available in this window:

Tool	Description
	Settings – Opens the Check Out Document window to let you specify check out parameters for the selected document (see Checking Out a Document [page 268]).
	Add Document Version - Opens the Check Out Document window to let you to specify a document to add to the list (see Checking Out a Document [page 268]).
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation links to the check out list.
	Add Document Versions from a Configuration – Opens a selection window allowing you to add all the document versions included in a configuration to the list (see Grouping Document Versions in a Configuration [page 278]).

Tool	Description
	Delete - Removes the selected document from the list.
	Move Up - Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

1.7.4.4 Checkout Parameters

The checkout parameters allow you to control details of your check out.

> **i Note**
>
> You can set default values for some of these parameters (see Repository Options [page 396]).

The *Repository (Document)* 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 Grouping Document Versions in a Configuration [page 278]).
Packages	[models only] Click the button to open a package selection dialog box to specify individual packages to check out instead of the entire model (see Checking Out Packages [page 272]).
Check out dependencies	[projects and models only] Checks out additional information about cross-model dependencies, to enrich dependency and impact analysis information in future situations where you may be without a repository connection (see Obtaining Cross-Model Dependencies Without a Repository Connection [page 501]).
Lock before check out	Locks the selected version of the document before checking it out, to prevent other users from modifying it (see Locking and Unlocking Documents [page 276]). This option is available only to users who have the **`Lock Versions`** right. The lock will be removed when you check the document back in unless you select the *Lock after check in* option (see Check-In Parameters [page 252]).

The *Check Out To File* or *Local* group box contains the following parameters:

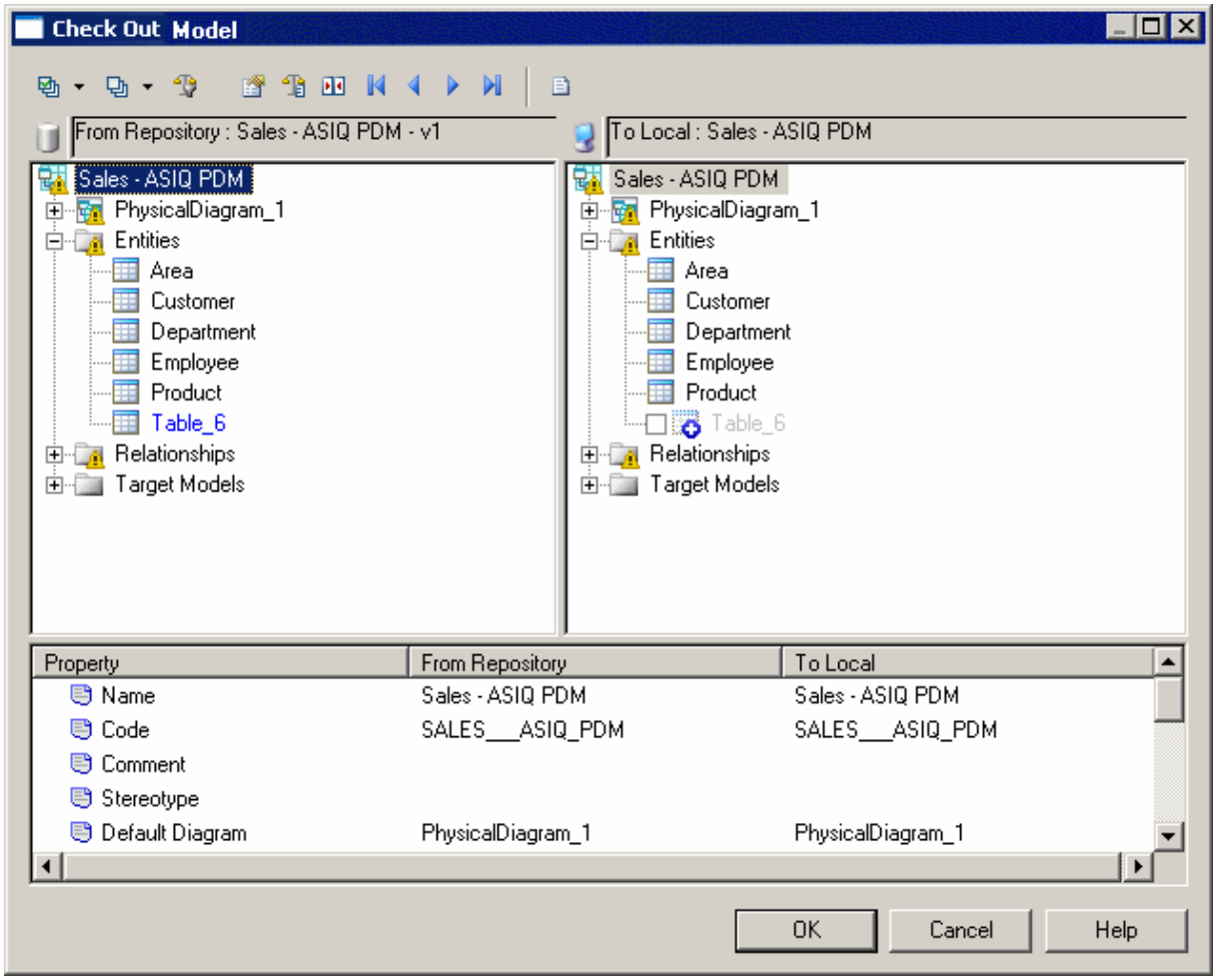
Parameter	Description
Check out to file	Checks out the model to a file on your local machine. This option can be unselected only for PowerDesigner models, which can be loaded in memory and added to your workspace without creating a file.
Path	Specifies the path of the file or (projects only) folder to which the document will be checked out. Click the *Select File* tool to the right of this field to browse for a location.
Merge document	Opens the Check Out Model window (see Resolving Conflicts During Check Out [page 271]), 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 available only 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 it out.
Open document	[models and files only] 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.

The *Project Documents* group box lists the model and file documents associated with the project, and lets you select them to be checked out with the project. 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.

1.7.4.5 Resolving Conflicts During Check Out

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

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



The repository version of the model that you are checking out is displayed in the left pane, and the local version is in the right pane. To filter the tree and show only the changes, click the *Change Filter* tool and select *Show All Changes*. By default, all the changes from the repository are selected to be applied. For detailed information about using this dialog, see Merging Models [page 369].

1.7.4.6 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.

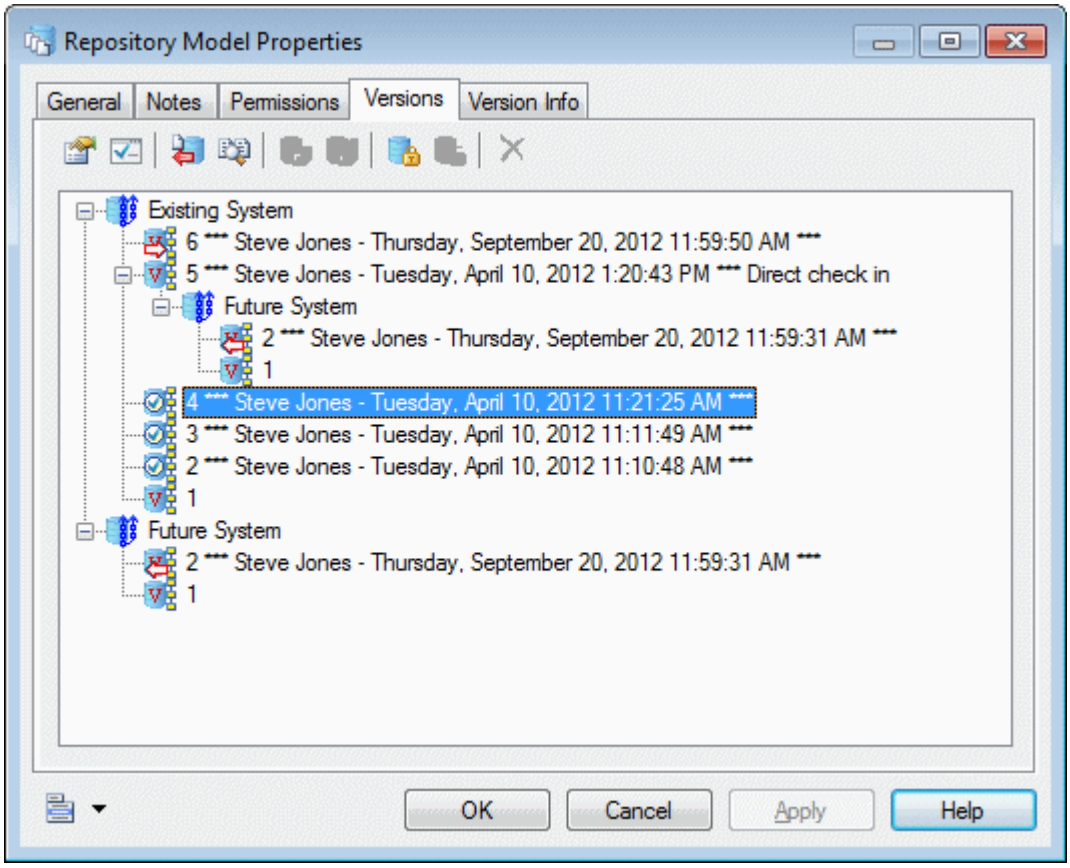
If you select a parent package, PowerDesigner automatically extends the selection to sub-packages. If you deselect some sub-packages in the package tree, the selection of the parent packages is grayed to indicate this. If you select a sub-package, PowerDesigner automatically extends the selection to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

> **i** 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.

1.7.5 Document Versions

The repository provides version tracking for changes made to documents. Details of the current version of the document are displayed on the *Version Info* tab of its repository object property sheet, and all the previous versions including versions on other branches appear as a tree on the *Versions* tab.



The following version icons can appear on this tab:

Tool	Description
	Branch - You create a new branch in the tree by branching (see Branching Version Trees [page 281]). **i Note** Branches are displayed twice; both beneath the document version from which the branch was created, and also as a root.
	Version from direct check in - Created directly in the repository without passing by a change list. You create a new version by freezing (see Freezing and Unfreezing Document Versions [page 275]).
	Version checked in/integrated via change list - Submitted in a change list (see Change Lists [page 259]) and approved by a reviewer.
	Version integrated from/to another branch (see Integrating Changes from One Branch to Another [page 284]).

In the example above, the document is first checked in directly, and then three versions are checked in via change lists Robert Brown checks in the first version of the document, and Steve Jones checks in a second version which is branched from the Existing System branch to become the first version in the Future System Branch.

The following tools are available on this tab:

Tool	Description
	Properties - Opens the property sheet of the selected version.
	Change List - Opens the property sheet of the change list from which the selected version was approved (see Change Lists [page 259]).
	Check Out - Checks out the selected version (see Checking Documents Out of the Repository [page 266]).
	Compare - Compares two versions (see Comparing Models in the Repository [page 293]).
	Freeze/Unfreeze - Freezes or unfreezes the selected version (see Freezing and Unfreezing Document Versions [page 275]).
	Lock/ Unlock - Locks or unlocks the selected version (see Locking and Unlocking Documents [page 276]).
	Delete - Deletes the selected version (see Deleting Documents, Document Versions, and Folders [page 295]).

> **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.

All repository documents and model objects can be versioned, but repository folders, 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.

1.7.5.1 Freezing and Unfreezing Document Versions

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

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

> **Note**
>
> To freeze a document, you must have the **`Freeze versions`** right (see Granting Privileges to Users and Groups [page 533]) and the appropriate permissions on the document.

You can freeze a document during check in (see Checking Documents into the Repository [page 247]) by selecting one or both of the following options:

- *Freeze before check in* - to preserve the state of the model before your changes.
- *Freeze after check in* - to preserve the state of the model including your changes (this option can be selected by default if you select the *Freeze after check in* general option (see Repository Options [page 396].

> **Note**
>
> PowerDesigner Web does not support checking in changes without creating a new version. When users publish changes through PowerDesigner Web the *Freeze before check in* and *Freeze after check in* options are both automatically applied to freeze the previous document state and create a new frozen document version.

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

You can freeze a document or unfreeze its latest frozen version at any time by:

- Right-clicking the document, folder, project, or root node and selecting *Freeze* or *Unfreeze*.

  > **Note**
  >
  > To freeze or unfreeze all the documents inside a folder or branch, right-click it and select *Freeze* or *Unfreeze*. You must have the appropriate permissions on all the contained documents.

- By clicking the *Freeze* or *Unfreeze* tool on the property sheet of a:
  - Document - on the *Versions* tab.
  - Configuration - on the *Members* tab.

> **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.

1.7.5.2 Locking and Unlocking Documents

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

Locked documents display a lock icon in the repository browser:

Icon	Description	Icon	Description
	Blue, if you have set the lock.		Red, if another user has set the lock.

> **Note**
>
> To lock a document, you must have the **`Lock versions`** right (see Granting Privileges to Users and Groups [page 533]) and the appropriate permissions on the document.

You can lock a document during check out (see Checking Documents Out of the Repository [page 266]) by selecting the *Lock before check out* option, and remove the lock during check-in (see Checking Documents into the Repository [page 247]) by selecting the *Unlock after check in* option.

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

You can lock or unlock a document at any time by

- Right-clicking the document and selecting *Lock* or *Unlock*.

  > **Note**
  >
  > To lock or unlock all the documents inside a folder or branch, right-click it and select *Lock* or *Unlock*. You must have the appropriate permissions on all the contained documents.

- Clicking the *Lock* or *Unlock* tool on the property sheet of a:
  - Document - on the *Versions* tab, if the latest version is selected.
  - Configuration - on the *Members* tab, if the document that is selected has its latest version included in the configuration.
- Selecting *Repository* > *Administration* > *Locks* and using the tools available on the List of Locks:

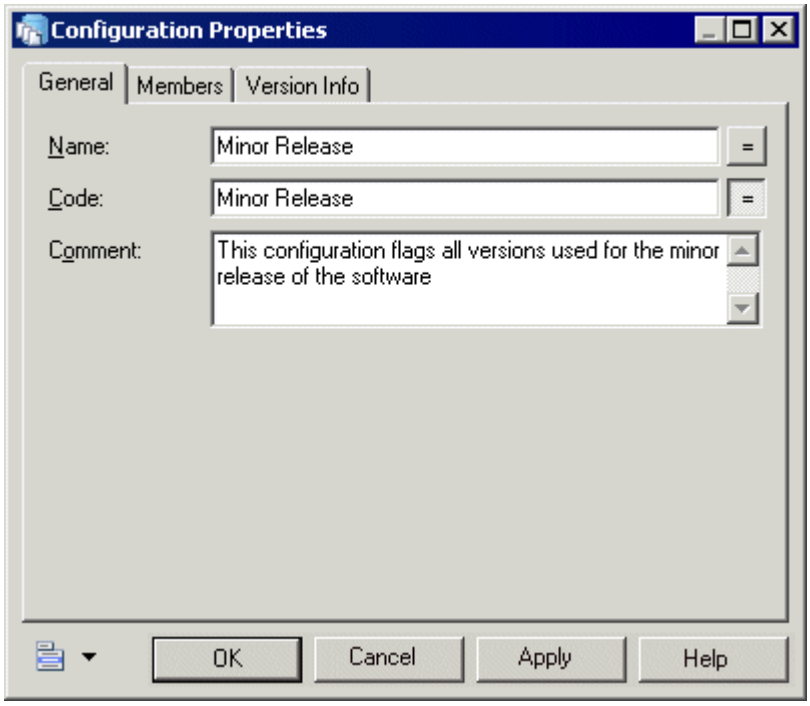
Tool	Description
	Settings – Opens the Lock Document dialog for the selected document, which shows details of the lock upon it.
	Add Lock – Opens the Lock Document dialog. Select a document to lock in the Name list and a comment, and click OK to lock it and return to the List of Locks.
	Add Related Document Versions - [models only] Locks the latest versions of 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. **Note** Documents contained within a configuration can only be locked if they are the latest versions.
	Delete – Unlocks the selected document, and deletes it from the List of Locks. Note that this tool does not delete the document.

1.7.6 Grouping Document Versions in a Configuration

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

Procedure

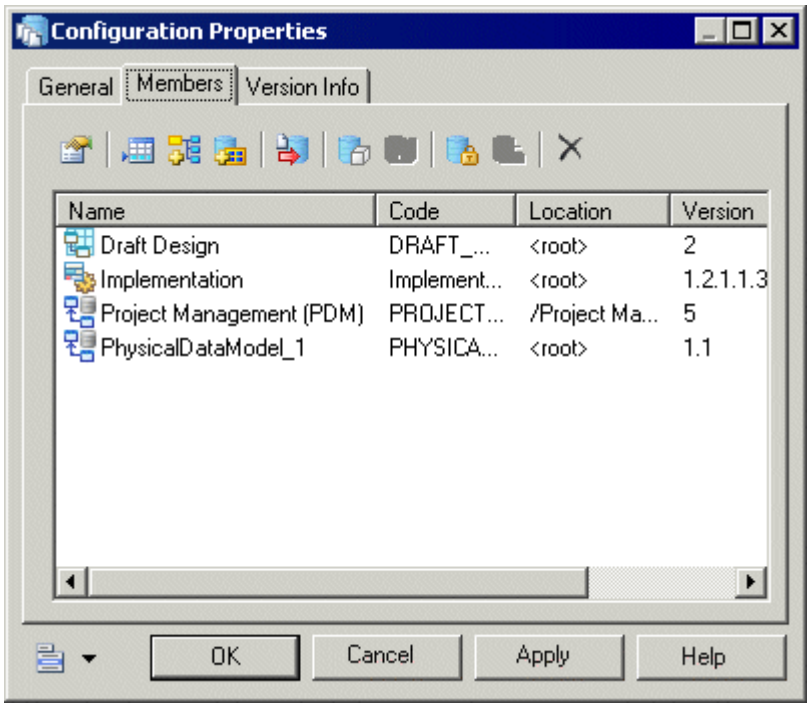
1. Select *Repository* > *Configurations* to open the List of Configurations.
2. Click the *Add a Row* tool to create a new configuration, and then click the *Properties* tool to open its property sheet.
3. Enter an appropriate Name, Code, and Comment to describe the configuration.



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

Tool	Description
	Properties – Opens the selected repository document property sheet.
	Add Document Version – Opens the Add Document to Configuration dialog, which allows you to specify the version of a document to add to the configuration.

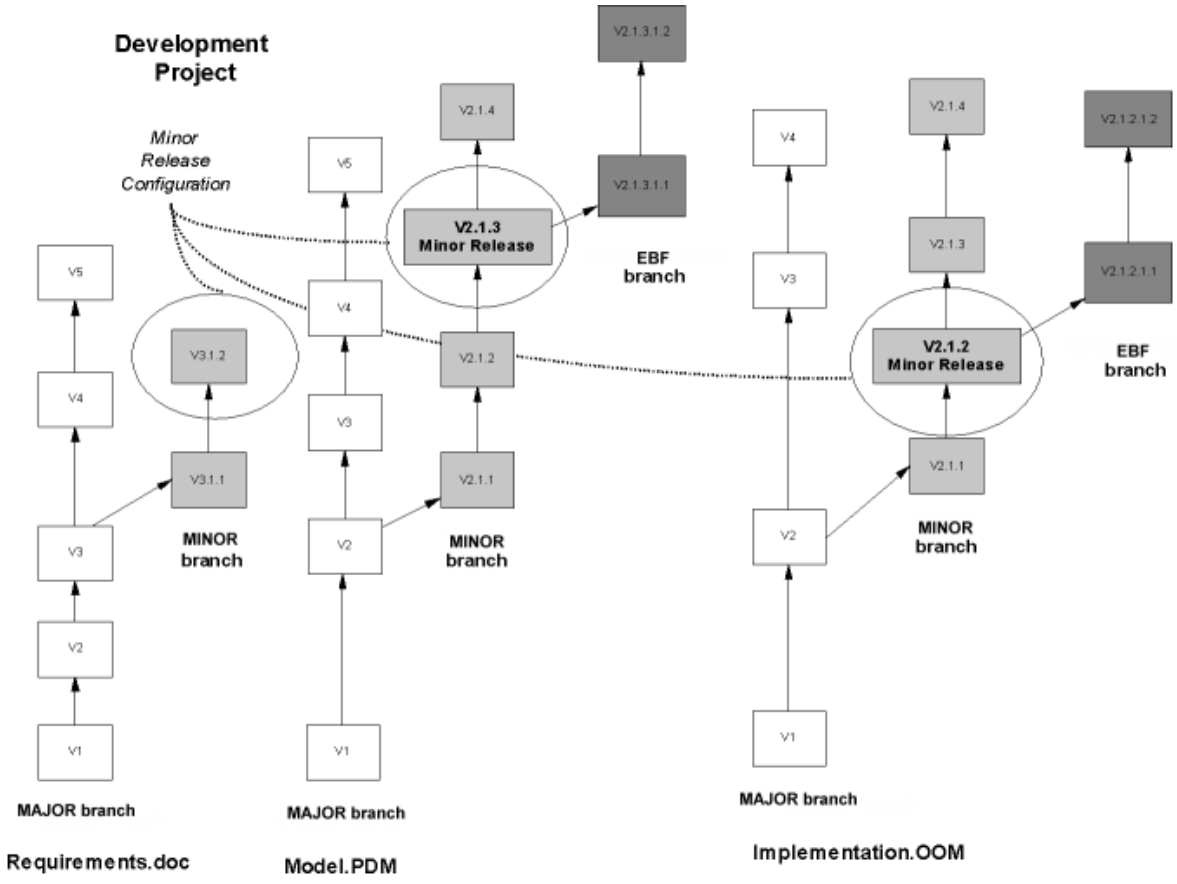
Tool	Description
	Add Related Document Versions – [models only] Adds any models that the selected model references through external shortcuts. The related models are automatically added to the configuration.
	Add Document Versions From a Configuration – Opens the Select Configuration dialog, which lets you select a configuration, whose members will be added to the configuration.
	Check Out – Checks out the selected version (see Checking Documents Out of the Repository [page 266]).
	Freeze/Unfreeze - Freezes or unfreezes the selected version (see Freezing and Unfreezing Document Versions [page 275]).
	Lock/ Unlock - Locks or unlocks the selected version (see Locking and Unlocking Documents [page 276]).
	Delete – Deletes the selected version from the configuration (see Deleting Documents, Document Versions, and Folders [page 295]).



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

Results

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



1.7.6.1 Creating a Configuration from a Project or Folder

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

Context

The property sheet of the new configuration opens. You can modify the default name and add a comment on the *General* tab and modify the document versions it contains on the *Members* tab (see Grouping Document Versions in a Configuration [page 278]).

1.7.6.2 Checking Out a Configuration

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

Procedure

1. Select *Repository* > *Check Out* to open the Check Out Multiple Documents dialog (see Checking Out Multiple Documents [page 268]).
2. Click the *Add Document Versions from a Configuration* tool, select the configuration to check out, and click *OK* to add the document versions it contains to the list.
3. Click *OK* to check out the documents.

> **i** 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.

1.7.7 Branching Version Trees

Users with the **`Manage Branches`** right can create branches in the repository to support concurrent development of multiple versions of documents. Standard permissions control read and write access to individual branches (including integrating changes made in one branch into another).

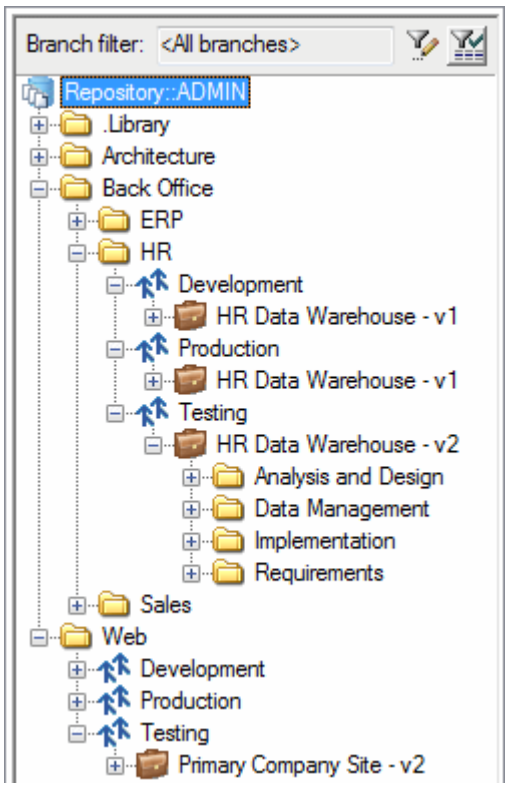
Context

> **i** Note
>
> Prior to v16.1, the repository browser only supported displaying a single branch at a time. When you upgrade an existing repository to v16.1 or higher, all branches will become, by default, visible for all users. You may wish to simplify viewing the repository by deleting documents and branches that are no longer used, by using permissions to hide certain branches from users or groups who have no need of them, or by migrating active models to a new repository (see Post-Upgrade Cleanup for a Repository Containing Branches).

You can branch the contents of any folder in the repository, but we strongly recommend creating at least one level of sub-folders beneath the repository root and creating your branches on one of these sub-folders.

> **⚠** Caution
>
> Once you have branched a folder, you cannot add a new folder directly beneath the branched folder as a sibling to the branches, because the entire contents of the folder is now branched and every document and

folder must necessarily belong to one or more of the branches. If you branch the repository root, you will not be able to create any other sub-folders at the root level and will be constrained to placing all of your repository documents inside a single global branch structure.



Branch names must be unique for the folder, but may be reused in another folder. In the example above, three branches (**`Development`**, **`Testing`**, and **`Production`**) are created in the folder `/Repository/Back Office/HR`, and branches with the same names are also created in the folder `/Repository/Web`

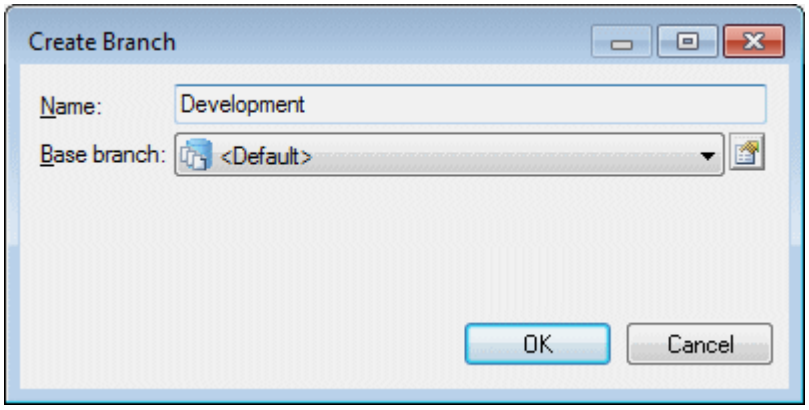
Users can filter the Browser *Repository* tab to display only the Branches that interest them (see Browsing the Repository [page 244]).

Procedure

1. Right-click the folder that you want to branch (and on which you have at least **`Write`** permission), and select *New* > *Branch*.

   > **i Note**
   >
   > You cannot create sub-branches under a branch and nor can you create branches inside a project.

2. Enter a name for the branch and select a base branch from which to copy the initial contents of the new branch:



The base branch can be any of the sibling branches (or **`Default`** if this is your first branch in this location), and all the contents of the base branch will be copied to the new branch. Alternatively, you can select **`None`** to create an empty branch.

The selected base branch is the default branch selected when you integrate items from the new branch to another branch (see Integrating Changes from One Branch to Another [page 284]).

3. Click *OK* to complete the creation of the branch.

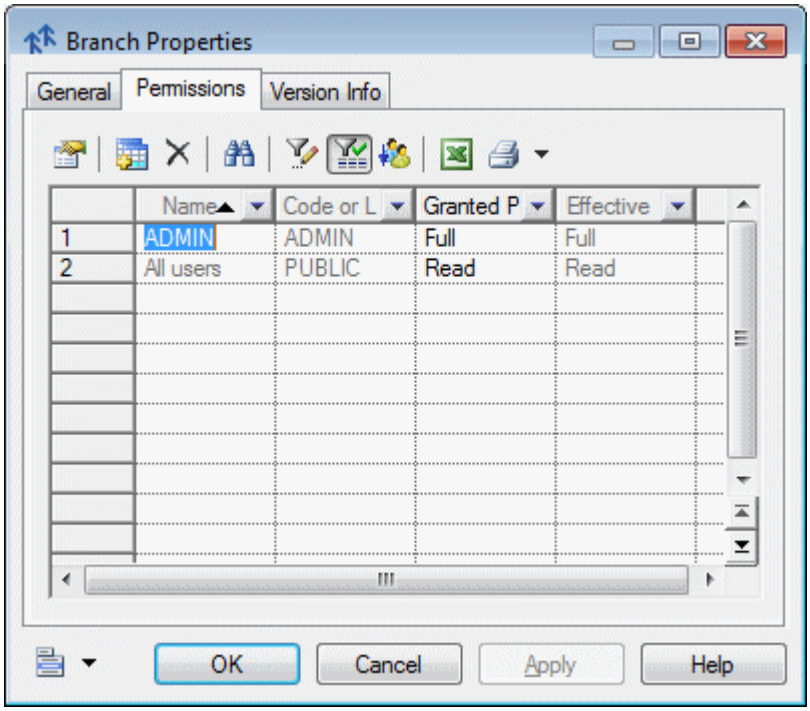
   The new branch is displayed in the Browser. If it is the first branch created in this location, then a **`<Default>`** branch is created next to the new branch. You can change the name by right-clicking the branch and selecting *Properties*.

1.7.7.1 Controlling Branch Permissions

Administrators and users with the **`Full`** permission on a folder containing branches can grant permissions on the branches from the *Permissions* tab of its repository property sheet.

Procedure

1. Right-click a branch in the repository browser and select *Permissions* to open the branch property sheet at the *Permissions* tab:



2. Click the *Add* tool to select any users or groups that should have access to the branch for adding to the list, and then click *OK*.
3. Set the *Granted Permission* for each user and group to an appropriate level (see Granting Access Permissions on Repository Items [page 534]).
4. Select any users or groups in the list that should not have even **`List`** access to the branch and click the *Delete* tool.

> **i Note**
>
> Users can filter the Browser *Repository* tab to display only the Branches that interest them (see Browsing the Repository [page 244]).

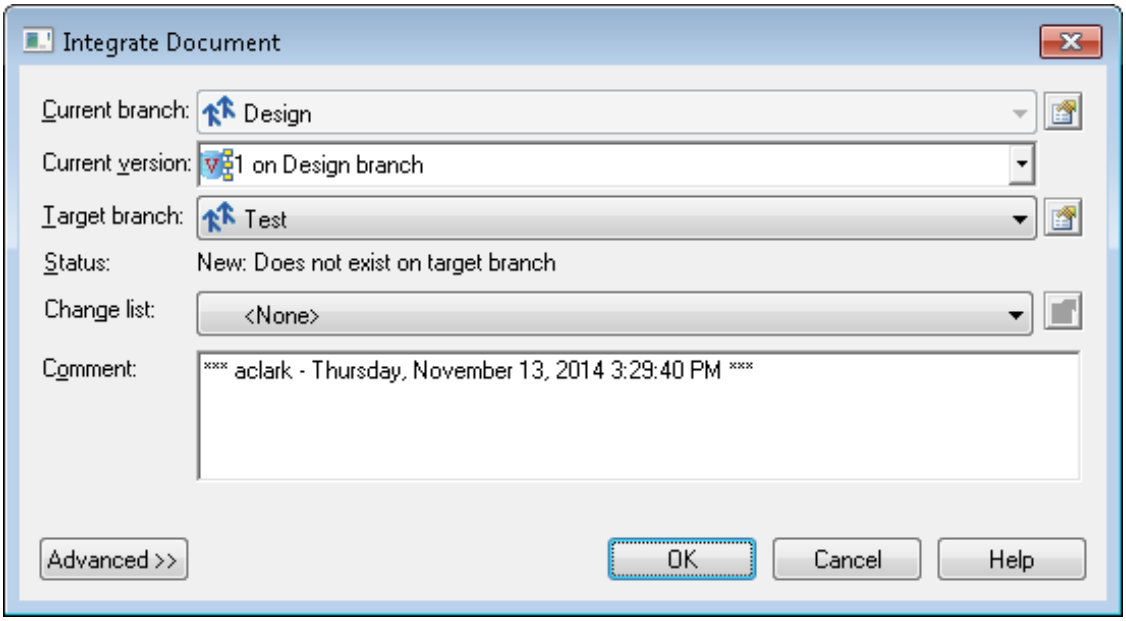
5. Click *OK* to complete the changes to the branch permissions.

1.7.7.2 Integrating Changes from One Branch to Another

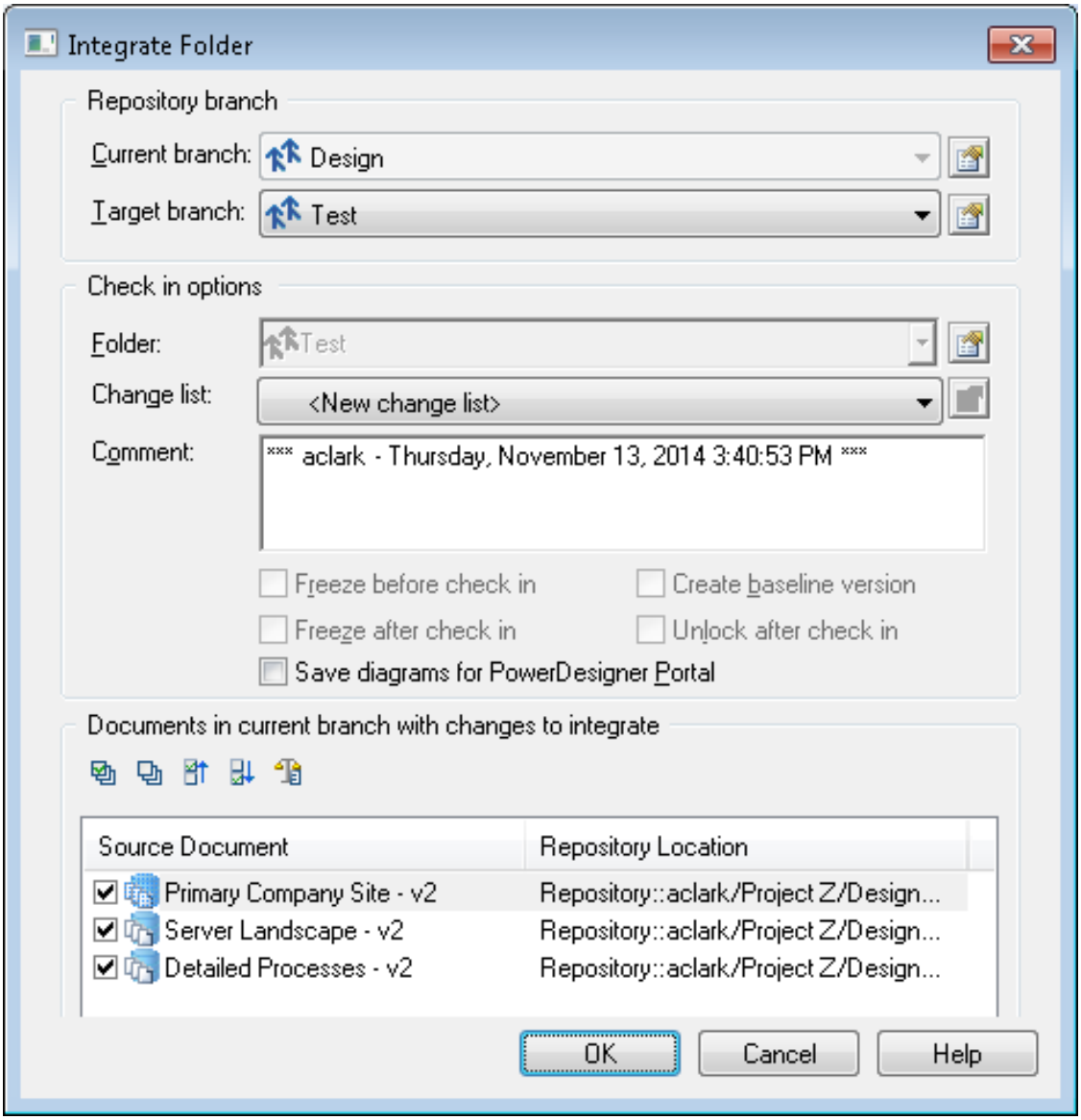
You can integrate changes made to a project, model or other document (or to the contents of a folder) in one branch to another branch. You must have **`Submit`** permission on the branch to which you want to integrate the changes to propose an integration via change list or **`Write`** permission to integrate directly. You can select individual changes to integrate on an object-by-object basis, and your choices are retained as defaults for subsequent integrations.

Procedure

1. Right-click the document, project, or folder in the branch containing the changes you want to merge to another sibling branch and select *Integrate*. Depending on the item selected, one of the following dialogs will open:
   - For a single document, you must select the version of the document you want to integrate and the target branch to integrate your changes to. You can optionally enter a comment, and click the *Advanced* button to access the same options as are available in the Check In Document dialog (see Check-In Parameters [page 252]):

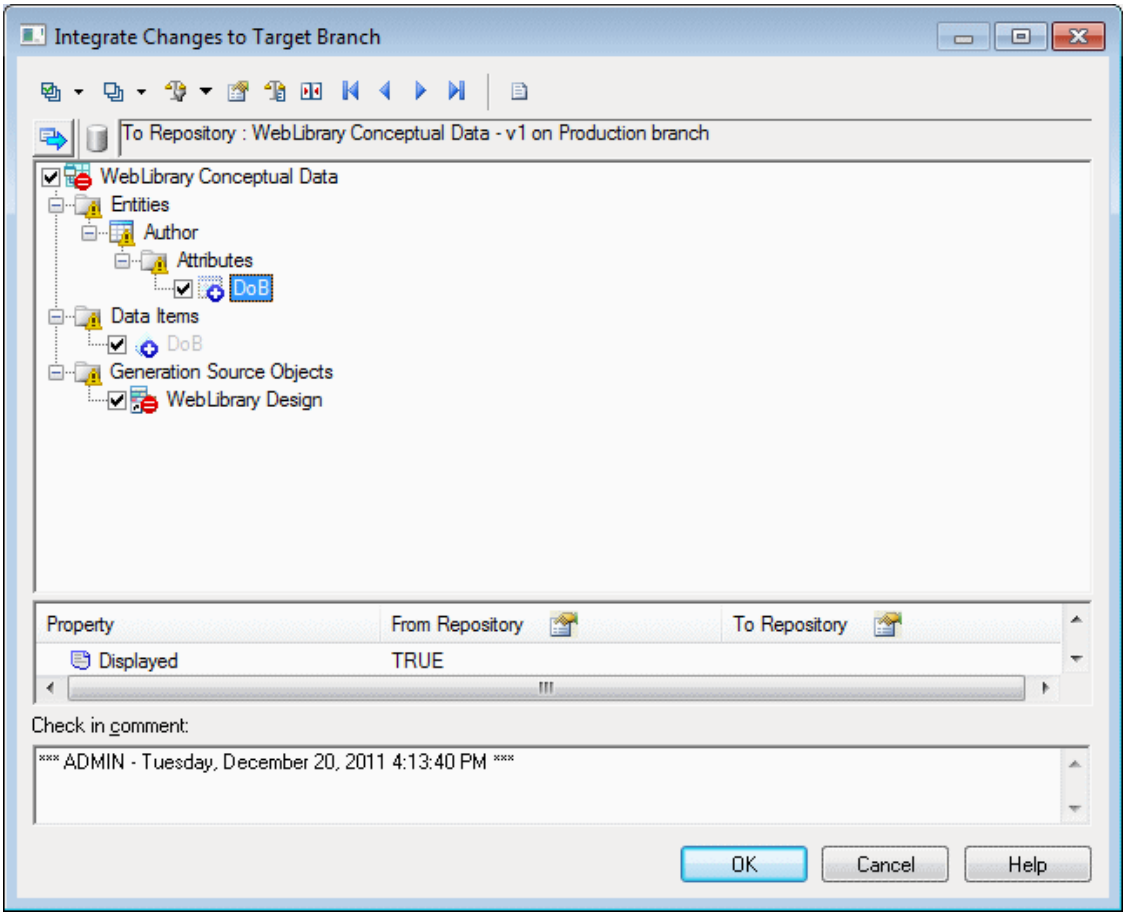


     If you have **Submit** permission on the target branch, you must select a change list. If you have **Write** permission or higher, you can check in directly or to a change list for peer review. You can choose between:
     - **<None>** - [if you have **Write** permission on the target branch] Allows you to integrate the documents to the branch directly without review.
     - **<New change list>** - To create a new change list to contain the documents to be integrated.
     - **Change list_<xxxx>** - To add the documents to be integrated to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list.
   - For a folder or project, you must select a target branch to integrate your changes to and can optionally enter a comment. Users with **Submit** permission on the target branch must select a change list. Users with **Write** permission or higher can check in directly or to a change list:



> **i** Note
>
> Only documents with changes since the creation of the branch (or the last integration) are listed and you can deselect documents in the list to exclude them from the integration.

2. Click *OK*:
   - If you did not select a change list and are integrating one or more PowerDesigner models, a merge dialog (see Merging Models [page 369]) opens to allow you to begin merging changes from the current branch to the target branch:



Select or reject each change as appropriate and then click *OK* to merge the selected changes from the current branch model to the target branch model.
If you are integrating a project or folder, an additional merge window will open for each of the models to be integrated, to allow you to inspect and select or reject their changes. When the final model merge window is closed, the integration is complete.

- If you selected a change list, you can add other documents to integrate as necessary before submitting it (see Creating, Building, and Submitting a Change List [page 260]). Once the change list has been approved by the reviewers, your changes are integrated to the target branch. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) you will receive an email at the moment of submission of your change list and when it is approved, rejected, or returned to you for further revision.

1.7.7.3 Reviewing an Integration Change List

If you have **`Write`** or higher permission on all the target branch documents included in an integration change list submitted by another user, then you can review it and determine whether to approve the integration, reject it, or return the change list to the user requesting her to revise it.

Context

> **i Note**
>
> This procedure explains how to review a change list created by a user to propose integrating documents from one branch to another. For information about reviewing a change list created by a user to propose changes to one or more repository documents, see Reviewing a Change List [page 262].

Procedure

1. When you connect to the repository a message in the Output window notifies you if you have change lists to review. Open the *Repository* tab of the Browser and expand the Change Lists folder to view all the change lists awaiting your review, which are identified by their "submitted" icon:

   > **i Note**
   >
   > If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]), you will receive an email when the user submits the change list, inviting you to review it.

2. Double-click a change list to view its properties (see Change List Properties [page 265]), and read the *Comment* field to understand the reason for its submission.
3. Click the *Documents* tab and review the documents that are included in the change list. The documents in the list display status icons to identify whether further changes have been checked into the mainline version by other users since the change list version was created:

Change List Document Status	No Locks	Target Branch Locked by You (Green Padlock)	Mainline Locked by Other (Red Padlock)
Change list creates new document (red cross)	+	N/A	N/A

Change List Document Status	No Locks	Target Branch Locked by You (Green Padlock)	Mainline Locked by Other (Red Padlock)
Changes in change list version (red checkmark)			
Changes in both change list and target branch versions (yellow exclamation point and red checkmark)			

4. Perform the following actions on a document by right-clicking it or using the toolbar:
   - *Check Out* - Check the submitted version of the document out into your local workspace to review it in more detail (see Checking Documents Out of the Repository [page 266]).
   - *Compare* - Compare the submitted version of the document in the change list with the target branch version in the repository (see Comparing Models [page 366]).
   - *Find in Open Change Lists* - Open the List of Change Lists filtered to display only open change lists containing the document.

   > **i Note**
   >
   > You will not be permitted to continue with the integration if any other submitted change list contains a proposed integration of any earlier version of any of the included documents. If this is the case, you must approve, return, or reject each change list in order so as to perform the requested integrations in a coherent order.

5. Note your agreement with the proposed integration or explain why you will reject it or return it to the submitter for modification in the *Comment* field of the change list, and then close it in preparation to approve, return, or reject it.
6. Right-click the change list in the Browser and select:
   - *Approve* - to approve the integration of the included documents and merge them into the target branch. Enter any appropriate final comment and click the *Approve* button. For each PowerDesigner model included in the change list, an Integration dialog is displayed, in which you can review the changes one last time before integrating them into the target branch (see Resolving Conflicts During Check In [page 254]).

     > **⚠ Caution**
     >
     > Once you begin the approval process, the target branch versions of all the documents in the change list are locked (see Locking and Unlocking Documents [page 276]) to enable the integration to be performed as a coherent transaction. If you click *Cancel* before having integrated all the documents, then all the documents will remain locked until you complete the approval process. You can recommence the approval process at any time from the point at which you canceled it.

   - *Return for Revision* - to return the change list to the user with a request to make any changes required for you to approve it. Enter a comment explaining the changes to be made and click the *Return* button.
   - *Reject* - to reject the integration proposed in the change list. Enter a comment explaining your reasons for rejecting the change list and click the *Reject* button.

In each case, if an administrator has connected the repository to an SMTP server, the change list owner and any other potential reviewers receive emails notifying them of the approval, rejection, or return of the change list.

1.7.7.4 Deleting a Branch

Administrators and users with the **`Manage Branches`** right can remove unused branches by right-clicking them and selecting *Delete*.

Context

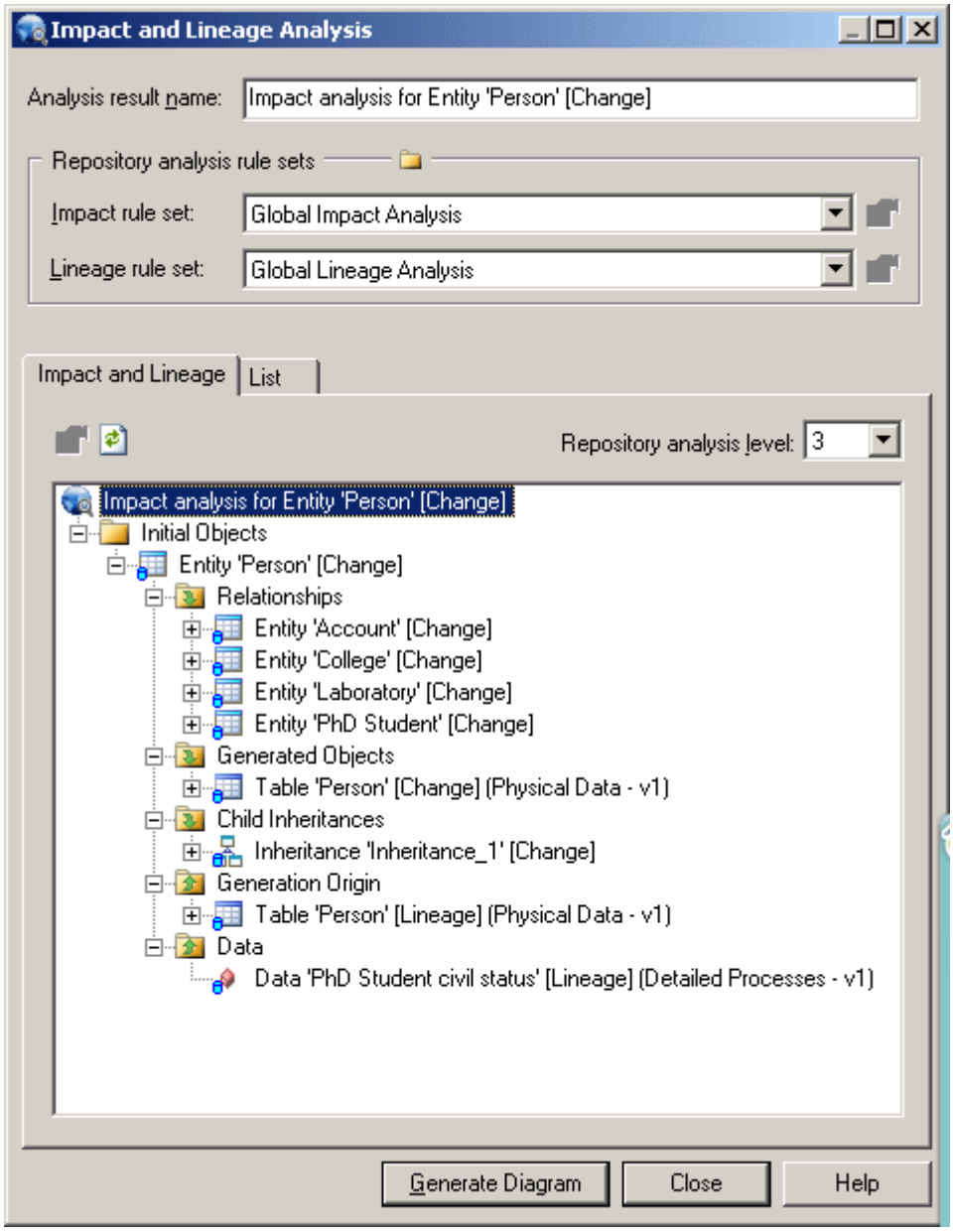
Before deleting a branch, you must delete all its contents (see Deleting Documents, Document Versions, and Folders [page 295]).

> ⚠ Caution
>
> Deleting a branch cannot be undone.

1.7.8 Impact and Lineage Analysis in the Repository

You can analyze the impact of a change to an object (impact analysis) and identify the objects that take part in the definition of the object (lineage analysis) on any model object in the repository by right-clicking it in the *Repository* tab of the Browser and selecting *Impact and Lineage Analysis*.

By default, the analysis descends to only three levels of dependencies, but you can select more or fewer levels by selecting a number in the *Repository analysis level* list.



For detailed information about using this dialog, see Impact and Lineage Analysis [page 489]. For information about the limitations of repository analyses, see Repository Rule Set Limitations [page 508].

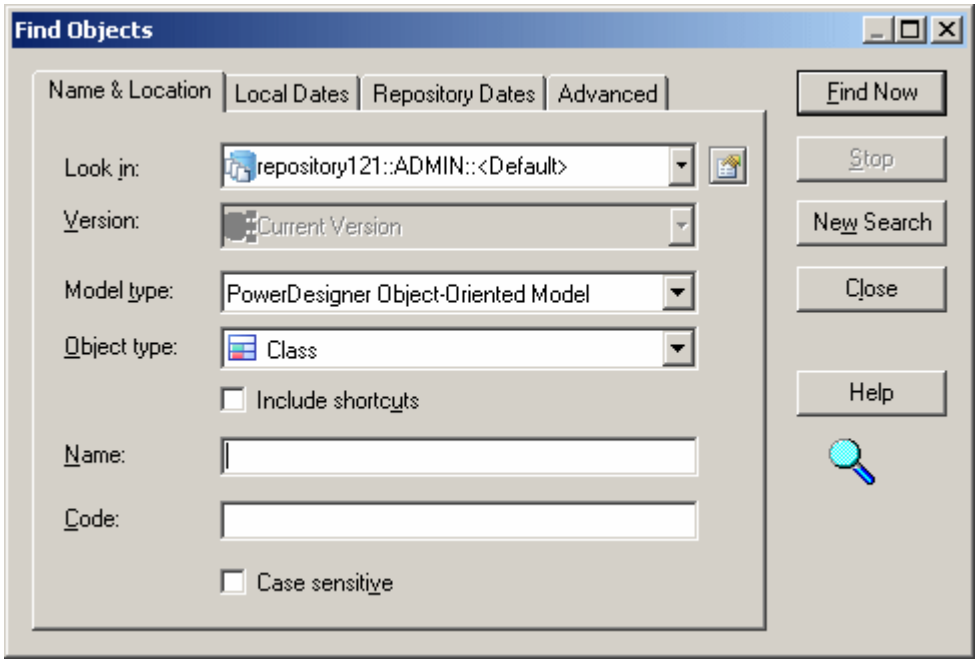
> **i Note**
>
> By default, analyses launched from the Repository tab use the standard **`Global Impact Analysis`** and **`Global Lineage Analysis`** rule sets delivered with PowerDesigner and available in the Library folder. You can check additional rule sets into the library or into another repository folder (see Checking Rule Sets into the Repository [page 508]). To change the folder where PowerDesigner looks for rule sets for selection in the dialog, use the *Choose Path* tool above the rule set fields.

1.7.9 Finding Objects in the Repository

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

Procedure

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

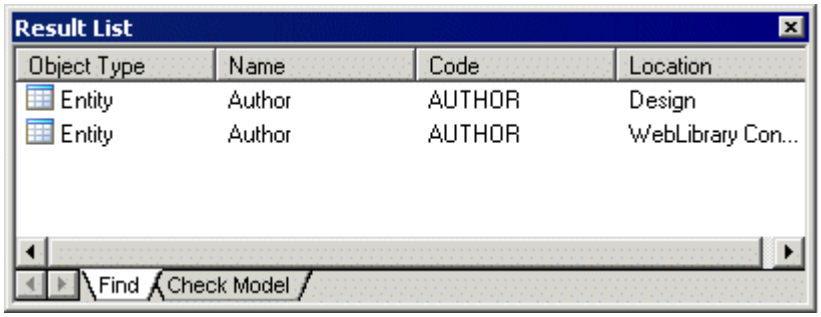


2. Specify the appropriate parameters in the different tabs of the dialog box.

   For detailed information about these tabs, see Find Parameters [page 135]. The repository Find Objects dialog allows you to search by both local and repository dates.

3. Click the *Find Now* button. You can stop the process at any time by clicking the *Stop* button.

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



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

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

1.7.10 Comparing Models in the Repository

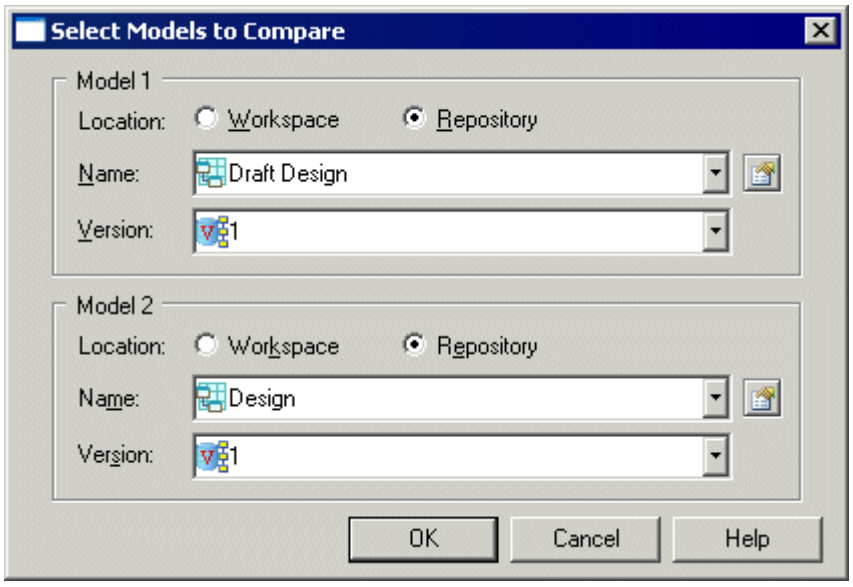
You can compare two PowerDesigner models to obtain a detailed list of the differences between them. You can compare two versions of the same repository model, two different repository models, or one repository model and one local model.

Context

> ⚠ Caution
>
> You can only compare models of the same type (for example, PDM, OOM, EAM).

Procedure

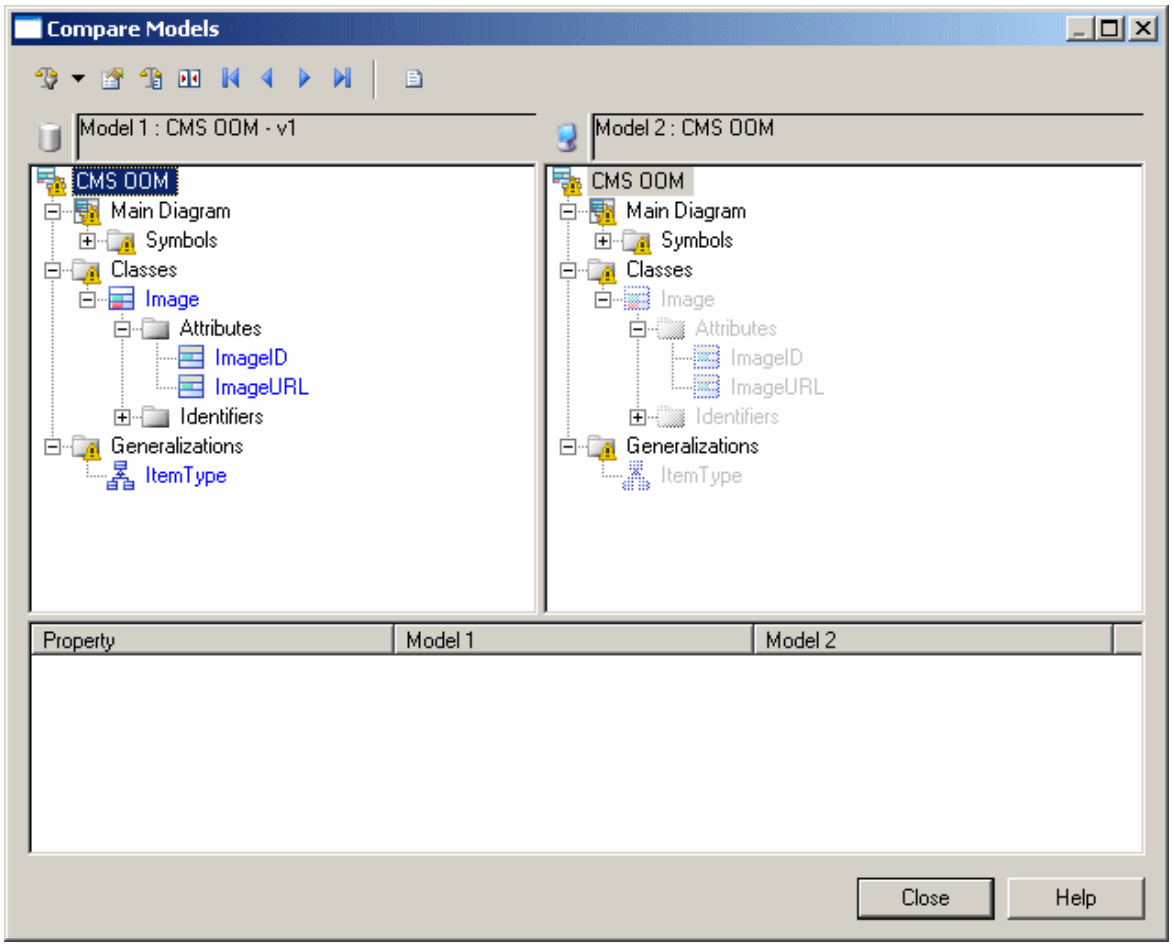
1. Press Ctrl + Alt + M, select *Repository* > *Compare*, or right-click a model in the Repository tab of the Browser, and select *Compare*.



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 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.

    > **i Note**
    >
    > The *Version* list includes only the versions in a particular branch (see Branching Version Trees [page 281]). To select a version in another branch, select the model in the *Name* list and then select a version.

3. Click *OK* to open the Compare Models dialog:



Results

For detailed information about using this dialog, see Comparing Models [page 366].

1.7.11 Deleting Documents, Document Versions, and Folders

If you have the **Full** permission on a document, you can delete the most recent version of the document or delete the document (and all its versions). When you delete a document, you also delete all the objects it contains. If you have **Full** permission on a folder and all its contents, you can delete it and all the versions of all the documents it contains.

> ⚠ Caution
>
> Deleting a version, document, or folder (see Repository Folders [page 245]) cannot be undone. If you delete a document that contains objects referenced by shortcuts in other models, these shortcuts will be shown as unresolved.

You can delete the latest version of a document by:

- Right-clicking the document in the Browser Repository tab and selecting *Delete Version*.
- Opening the repository document property sheet to the *Versions* tab (see Document Versions [page 273]), selecting the latest version of the document (or the latest version in a branch), and clicking the *Delete* tool.

To delete all the versions of a document in a branch, right-click the document and select *Delete Document*.

To delete a folder, right-click it and select *Delete*. You must have **Full** permission on the folder and all its contents.

1.7.12 Repository Web Client

PowerDesigner comes with a repository Web client, PowerDesigner Web.

For information about:

- Installation - See Installing the PowerDesigner Web Server
- PowerDesigner Web - See PowerDesigner Web

You can access the Web client at the following address:

- PowerDesigner Web: `<server-name>:<port-number>/powerdesigner-web`

> i Note
>
> The default port number is `3030`.

1.8 Reports

PowerDesigner reports allow you to publish information about your models in HTML or RTF format.

You can create the following types of report:

- A model report – provides information about all or some of the objects in a model, and is shown in the Browser in the Reports folder beneath its parent model (see Creating a Model Report [page 296]).

- A multi-model report – provides information about one or more models, and is shown in the browser as an independent item, saved as a .mmr file (see Creating a MultiModel Report [page 311]).
- A list report – provides information about a single type of object, and is shown in the Browser in the List Reports folder beneath its parent model (see Creating a List Report [page 303]).

This version of PowerDesigner supports two kinds of model and multi-model reports:

- Legacy reports - use the traditional Report Editor, which is deprecated and will be removed from a future version (see The Legacy Report Editor [page 334]).
- New format reports - use the new graphical Report Editor, which provides synchronized Design and Outline views, seamless inclusion of model extensions, simplified WYSIWYG formatting, and modernized HTML and RTF generation (see The Report Editor [page 312]).

> **i Note**
>
> The initial version of the new Report Editor has the following limitations:
> - No global selection of objects to display in the report - Define any appropriate selections on object books.

1.8.1 Creating a Model Report

You can create a model report using the Report Wizard or manually, refine it in the Report Editor, and generate it in HTML or RTF format, or print it at any time.

Procedure

1. Create the report in your model using the Report Wizard (see Creating a Report with the Report Wizard [page 296]) or from the list of reports (see Creating a Report from a Template or an Empty Report [page 302]).
2. [optional] Refine the content and format of your report in the report editor (see The Report Editor [page 312]).

   > **i Note**
   >
   > The report is saved as part of its model. If your model is very large, you may want to create the report as a multi-model report, which is saved separately (see Creating a MultiModel Report [page 311]).
3. Generate the report as HTML or RTF, or print it directly (see Generating a Model Report [page 356]).

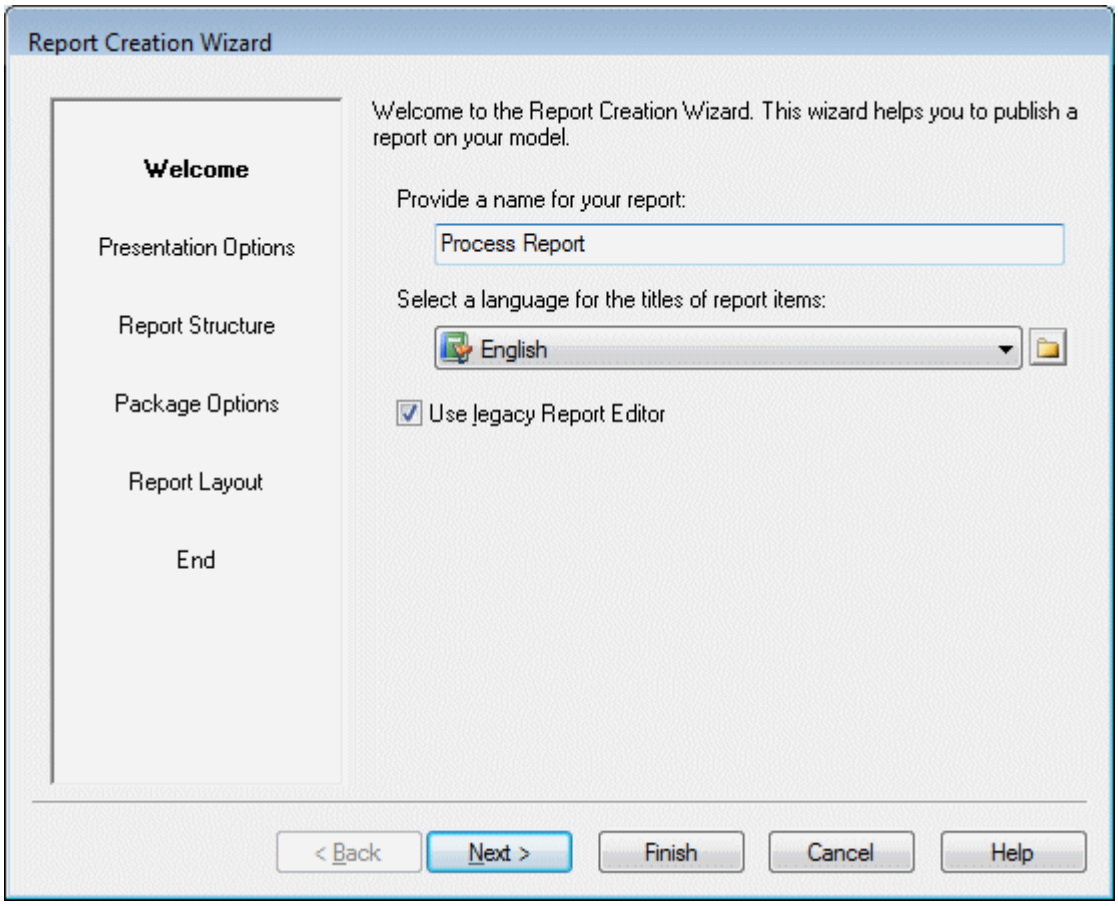
1.8.1.1 Creating a Report with the Report Wizard

The Report Wizard provides an easy way to create a report, select model objects and format them. The wizard can be launched from your model to create a new report, or from the Report Editor to configure an existing report. You can complete as much of the Wizard as you want. At each step, you click *Next* to advance to the next stage, or *Finish* to create a report based on your selections so far.

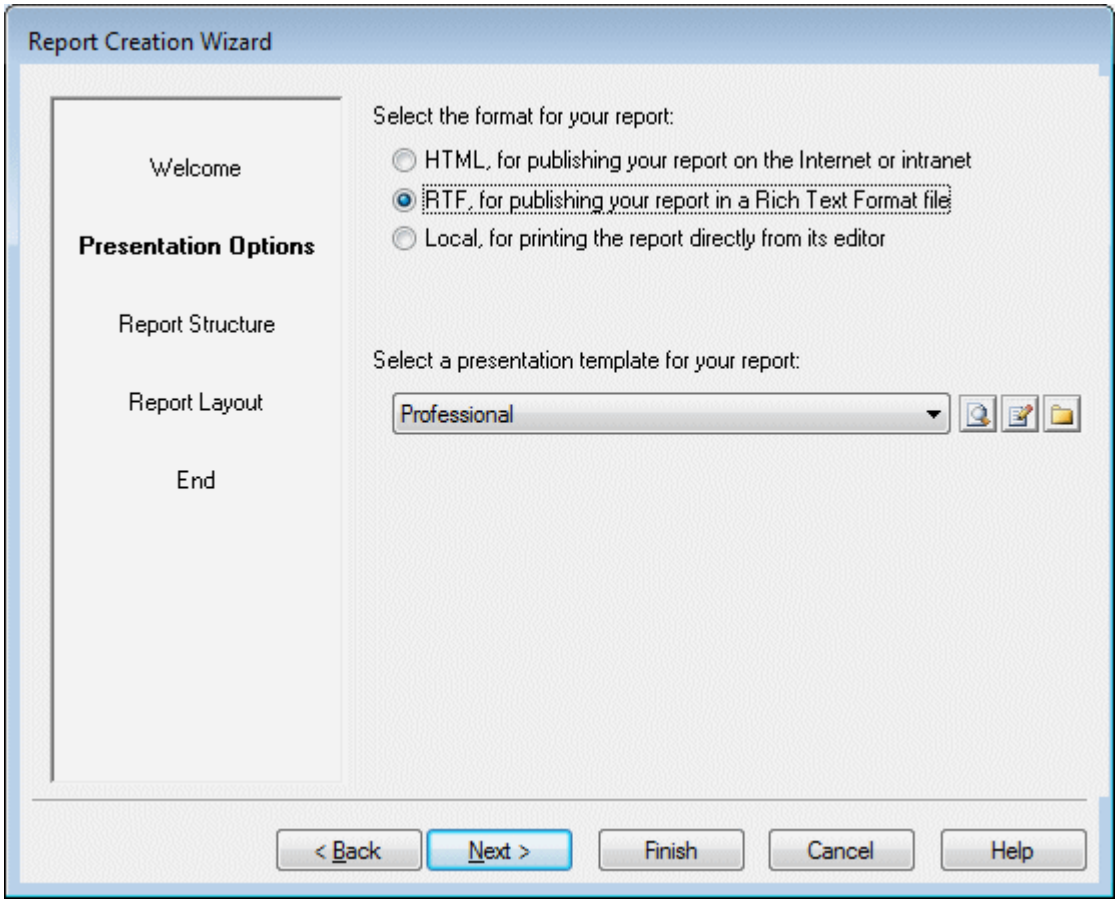
Procedure

1. Select *Report* > *Report Wizard* (or select *Report* > *Reports* to open the List of Reports, and then click the *Report Wizard* tool) to launch the wizard.
2. Specify a name for the report and the language in which you want its titles to appear, and then click *Next*.

> **i** Note
>
> [optional] To use the new report format and editor (see The Report Editor [page 312]), deselect *Use legacy Report Editor*.

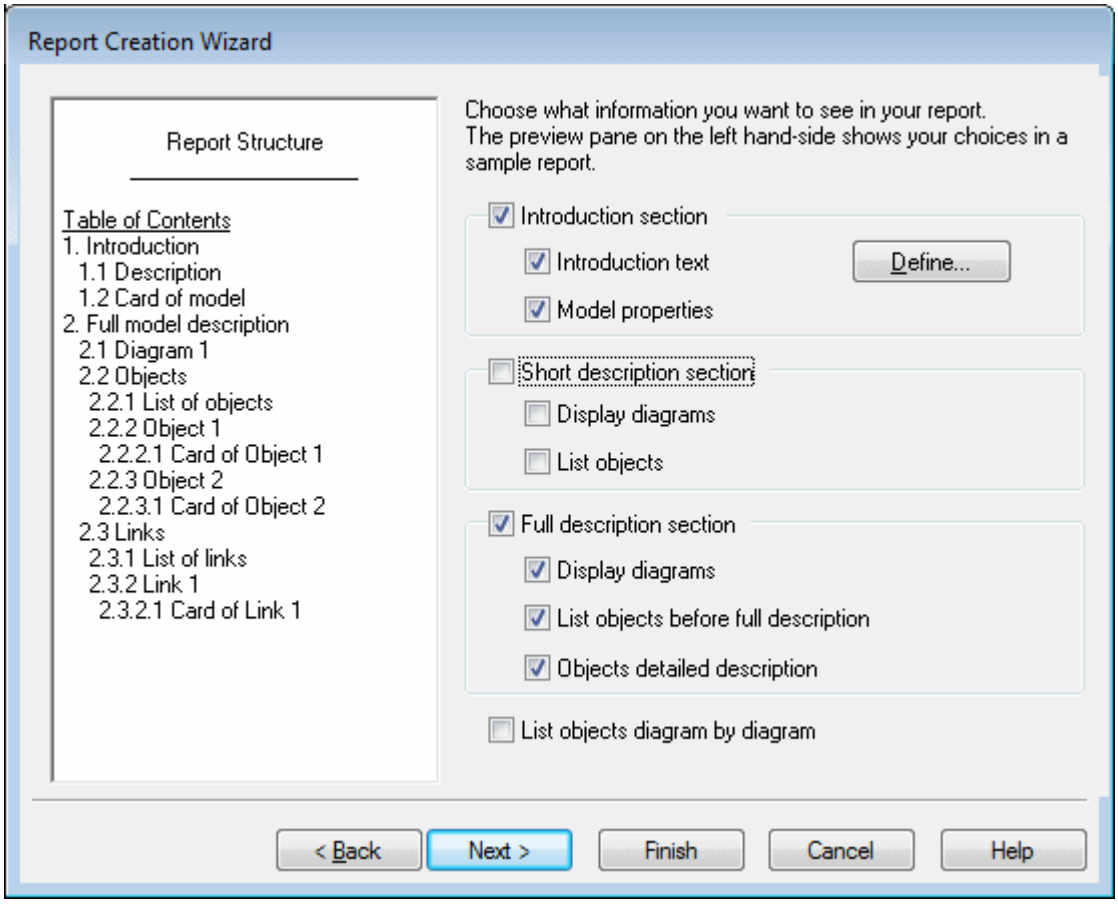


3. Select whether you want to prepare your report for publishing as HTML or RTF, or for direct printing, select a presentation template, and then click *Next*.



> **i** Note
>
> For HTML and RTF reports, you can use the tools to the right of the template field to preview and edit the template and to change the folder in which to search for templates.

4. Specify the types of information that you want to appear in your report, and click *Next*.



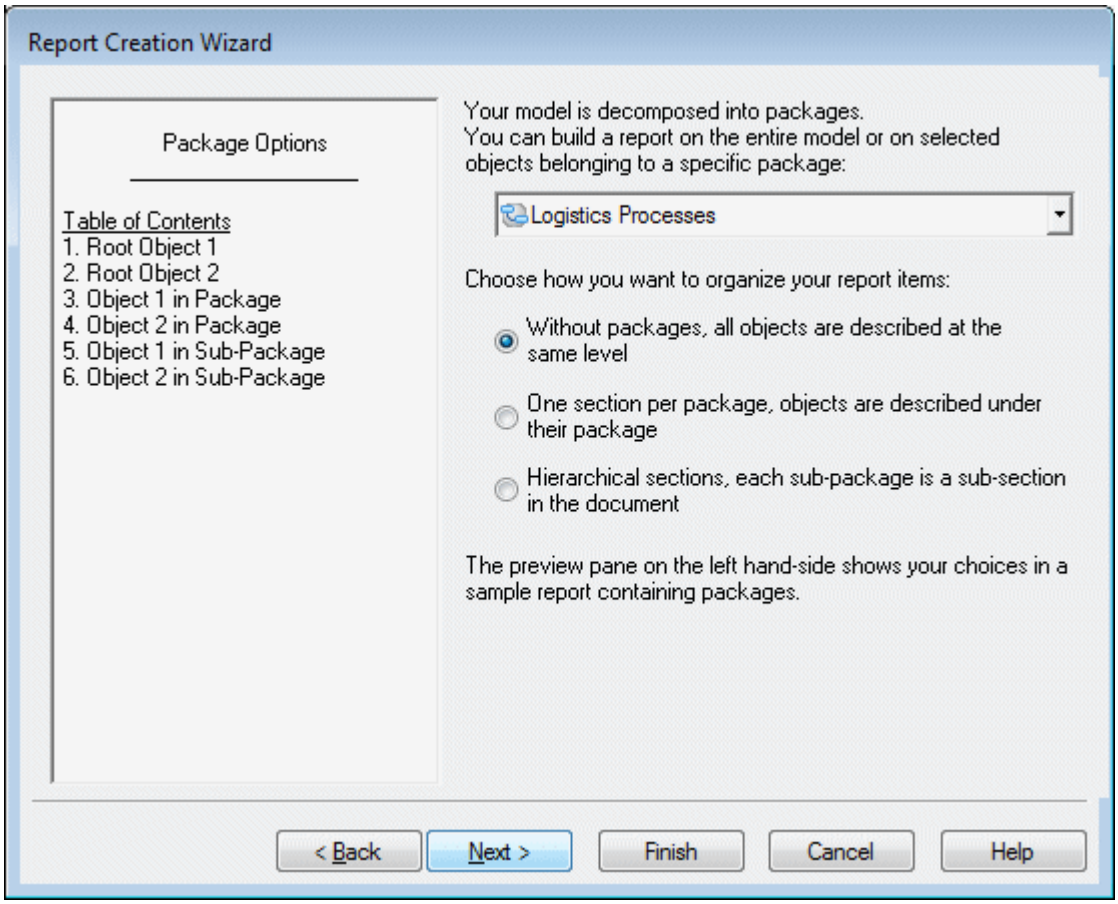
The preview pane on the left hand-side displays a sample report that changes dynamically according to your selections. You can select any or all of the following options:

- *Introduction section* – Inserts a page break followed by an introduction section, which can contain a paragraph of *Introduction text* (click the *Define* button to enter text), and a card listing the *Model properties*.
- *Short description section* – Inserts a page break followed by a short description section, which can *Display diagrams* (including the Diagram book with a picture of each diagram with its description and annotation) and *List objects* (with a list for each type of object in the model).

  > **i Note**
  >
  > RQM views are not diagrams and cannot be published graphically.

- *Full description section* – Inserts a page break followed by a full description section sub-heading, which can display all the elements from the short description section plus the *Objects detailed description* (a book item for each object type).
- *List objects diagram by diagram* - [except RQM] Instead of a flat display of object types, displays objects sorted by the diagram in which they are displayed. Global objects, such as business rules, do not belong to a specific diagram, and are therefore listed in their own section under the root.

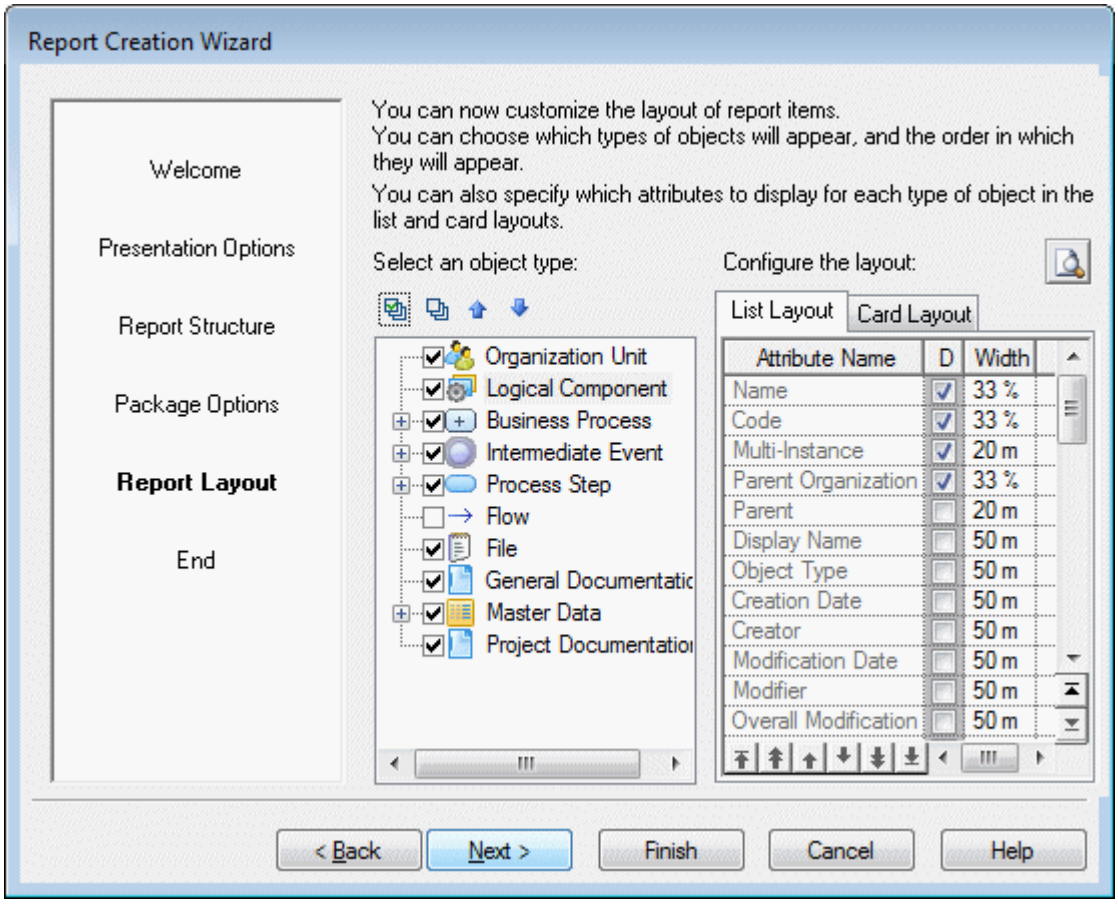
5. [if your model contains packages] Specify whether you will report on the whole model or only a package and whether you will ignore the package structure, or organize objects under a flat list of packages or a hierarchical list of packages (where sub-packages are children of their parent package), and then click *Next*.



The preview pane (on the left hand-side) changes dynamically in order to preview your selection.

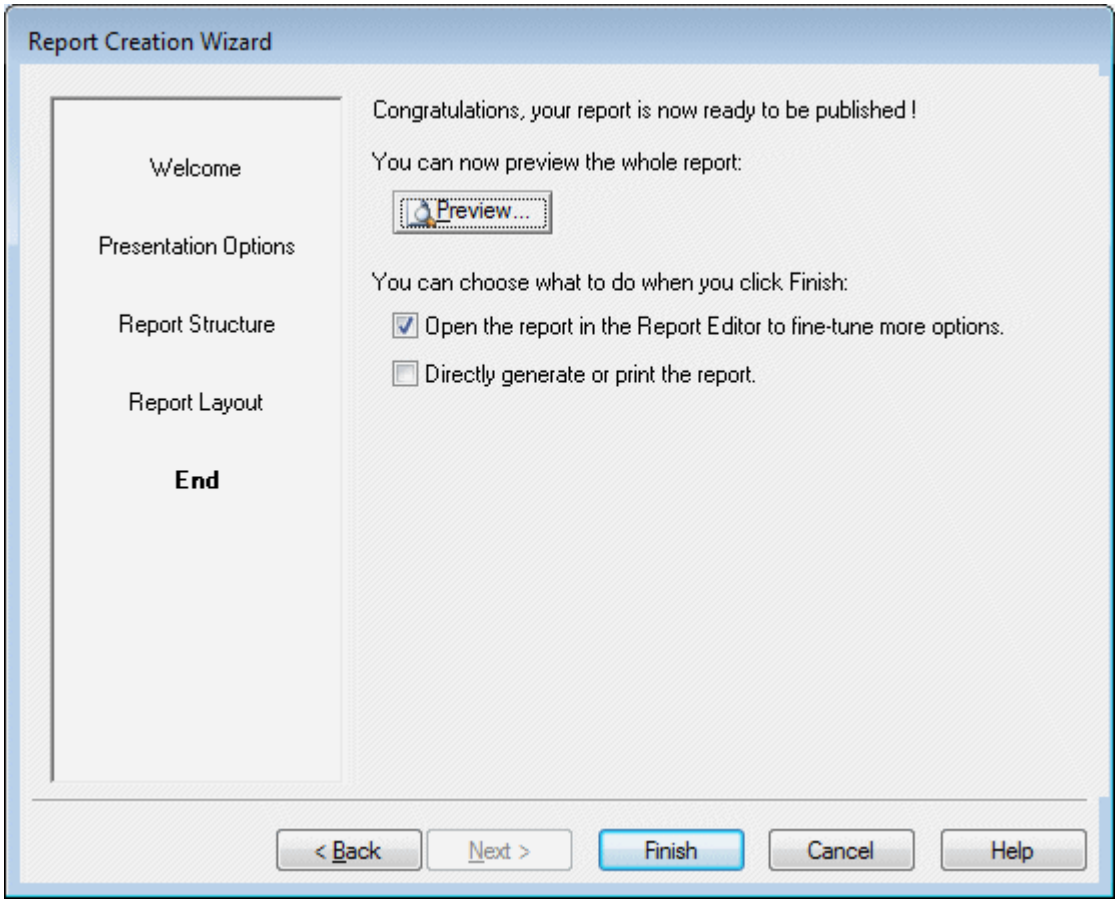
> **i Note**
>
> Global objects, such as business rules, are always included, even if you select a package, and composite objects, such as OOM activities or processes, are always displayed hierarchically, even if you have selected a different package option. Only the hierarchical structure is supported for HTML reports or RQM reports.

6. Specify the types of object that will be included in your report, along with the properties that will be displayed in each of their list and card report items, and then click *Next*.



- To include an object type in the report, select its check box in the list. You must select at least one object type. Click on the plus sign to the left of an object to select sub-objects. Use the *Move Up* and *Move Down* tools to reorder the list (sub-objects cannot leave their parent scope). New types of objects defined in extensions (stereotypes used as metaclasses) are listed after their parent object type.
- To control the properties included for an object type in lists and cards, select the *List Layout* or *Card Layout* tab and use the following columns
  - *Attribute Name* – Use the arrows at the bottom of the list to reorder the list.
  - *D[isplayed]* – Select the check box to display the attribute.
  - *Width* [List Layout] – Specify the width of the column either as a percentage or in millimeters.

7. Optionally preview your report, and select to open it in the report editor and or directly generate or print it, and then click *Finish*.



1.8.1.2 Creating a Report from a Template or an Empty Report

You can create a model report directly from one of the standard report templates without making any object or format selections. Whether you choose a template or create an empty report, you can modify it in the Report Editor.

Procedure

1. Select *Report* > *Reports* to open the List of Reports, and click the *New Report* tool.
2. Enter a *Report name*, and select a *Language* for the report's titles and other framing text.
3. [optional] Select a report template from the list to prepopulate the report. The following standard templates are available:
   - **`Full`**– provides lists of all the types of objects in the model, together with detailed information on each main and sub-object.

- **`List`** – provides lists of all the types of objects in the model.
- **`Standard`** – provides lists of all the types of objects in the model, together with detailed information on each main object.
- **`<None>`** – creates an empty report.

> **i** Note
>
> For more information about report templates, including how to create your own, see Report Templates [page 357].

4. [optional] To use the new report format and editor (see The Report Editor [page 312]), deselect *Use legacy Report Editor*.
5. Click *OK* to create the report and open it in the Report Editor (see The Report Editor [page 312]).
6. [optional] Generate or print the report by clicking the appropriate toolbar tool.

> **i** Note
>
> Report templates can also be applied at generation time (see Generating a Model Report [page 356]).

1.8.2 Creating a List Report

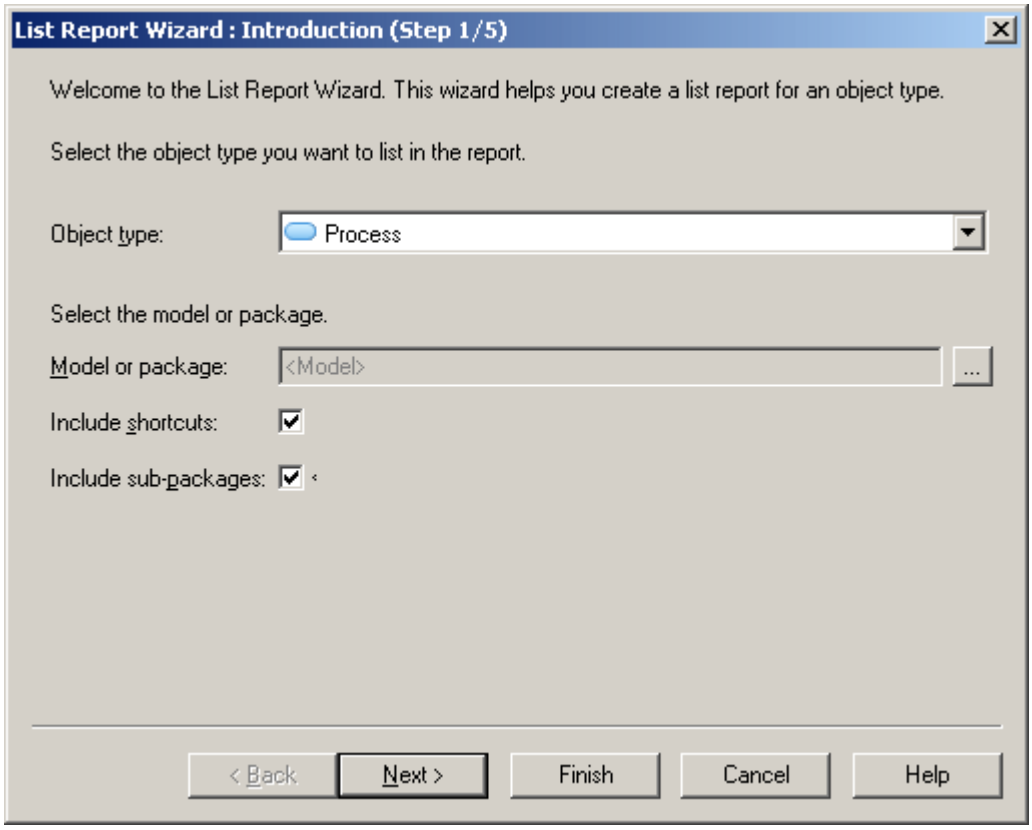
A list report documents a single object type within a model, and is displayed as a customizable list with columns and rows that you can filter as necessary. A list report is saved in the model and can be exchanged between models of the same type.

Context

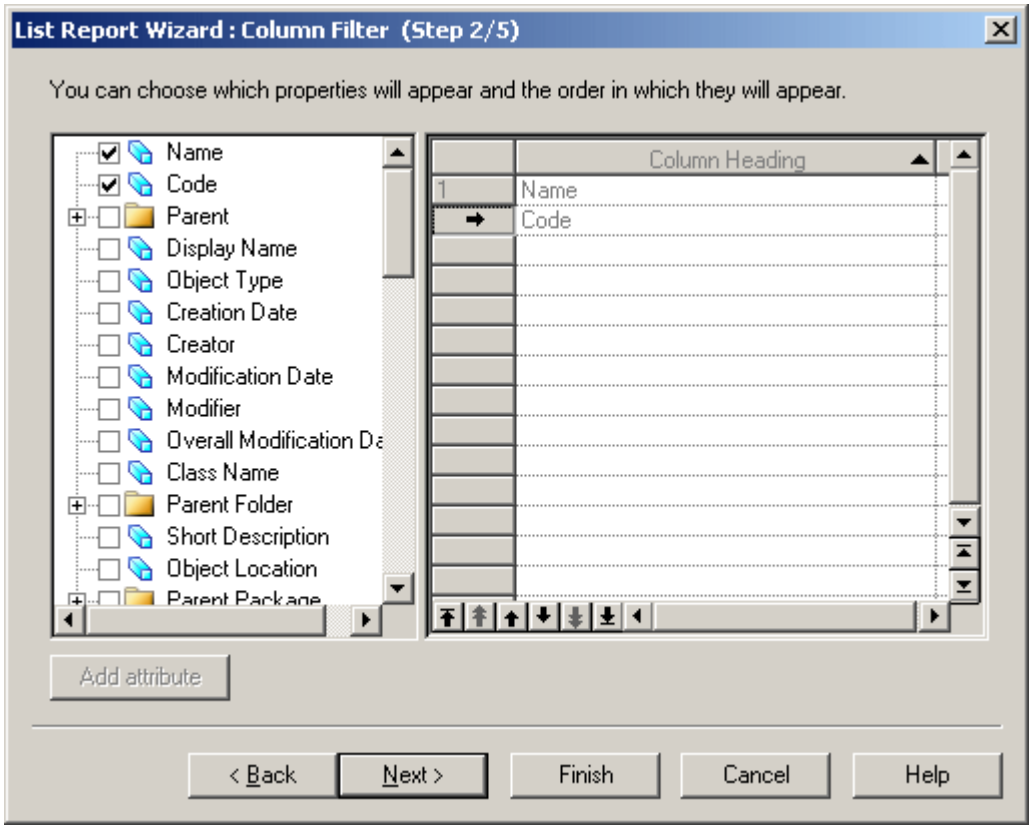
You can create snapshots (called result sets) of a list report to keep a history of the execution of a list report in your model. Result sets are listed in the browser beneath their parent list report, and are saved with the model (see Creating Result Sets [page 309]).

Procedure

1. Launch the List Report Wizard by selecting *Report* > *List Report Wizard*, and select the object for which you want to create the report. By default the list report is based on the model and excludes shortcuts and sub-packages:



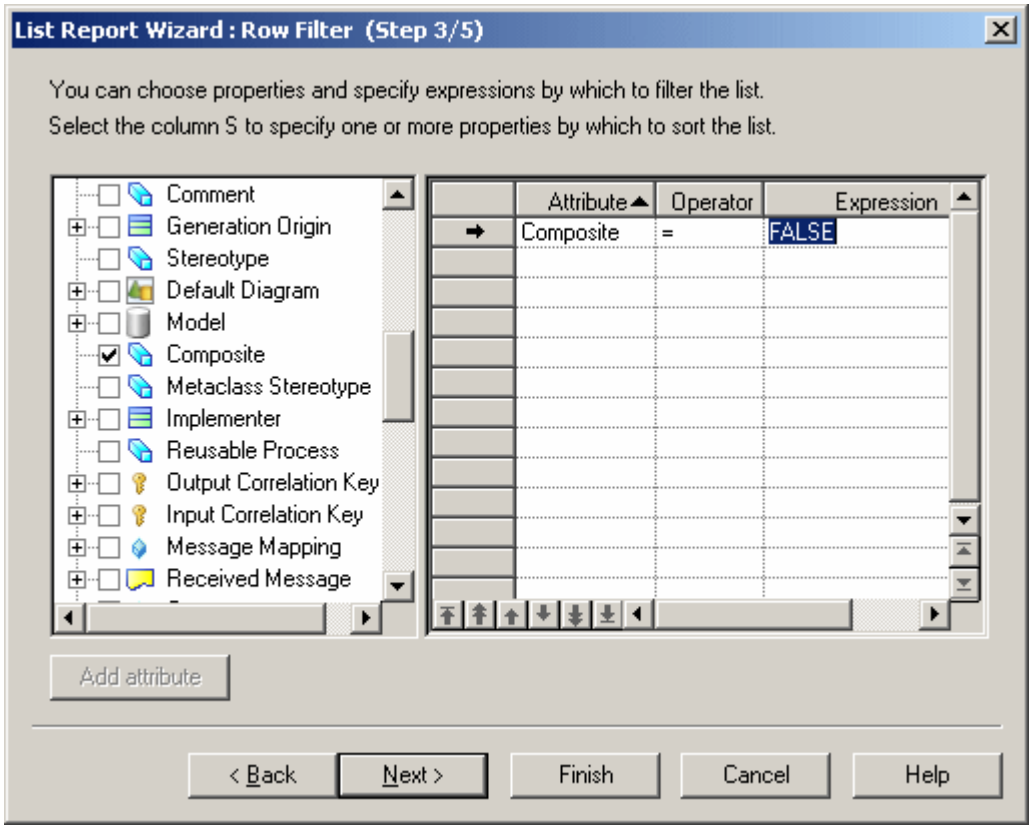
2. Click *Next* to go to the Column Filter page, which allows you to specify which of the object's properties will be included in the list report. Select a property in the left-hand pane to add it to the list in the right-hand pane. You can promote or demote properties in the list using the arrows at the bottom of the pane:



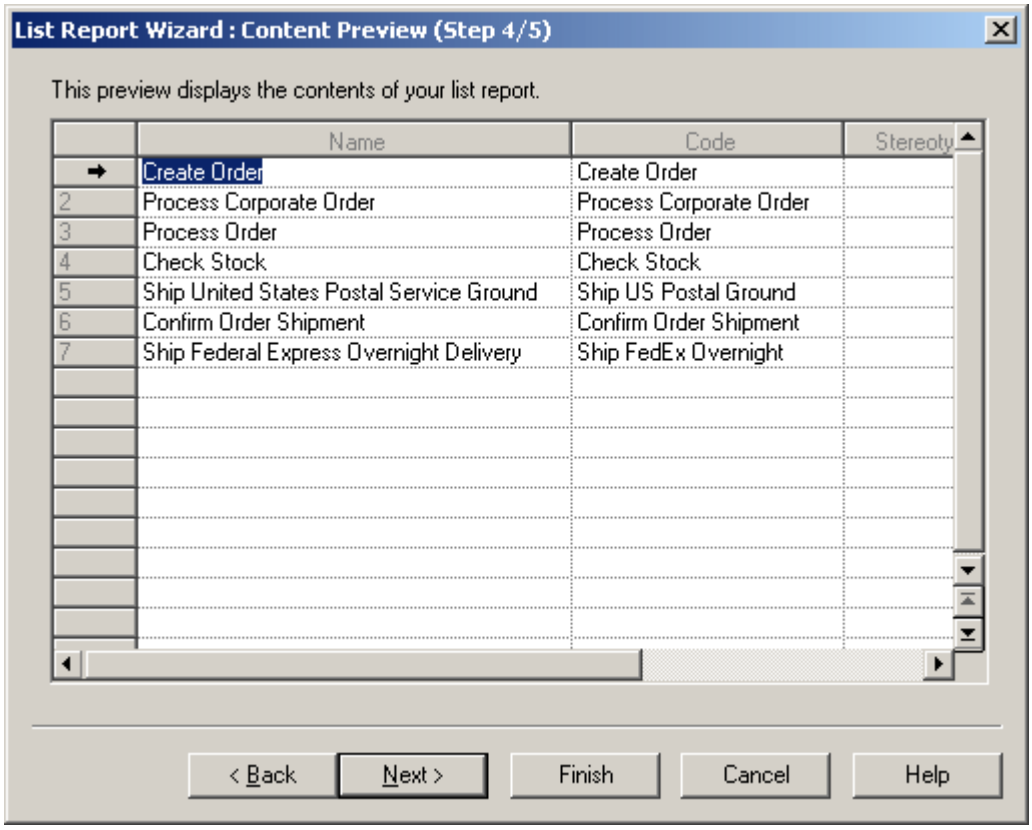
3. Click *Next* to go to the Row Filter page allows you to specify filters to restrict the objects that will be included in the list report. Select a property in the left-hand pane to add it to the list in the right-hand pane, select an operator, and then enter an expression to filter by (see Defining a Filter Expression [page 127]).

   You can specify to sort the list on the values of a property by selecting the S[ort] checkbox.

> ### i Note
>
> Some lists, such as a table's columns, have a natural order that is controled by the modeler and is separate from any property values. To retain this order, select the *Natural List Order* property and select the S[ort] checkbox.

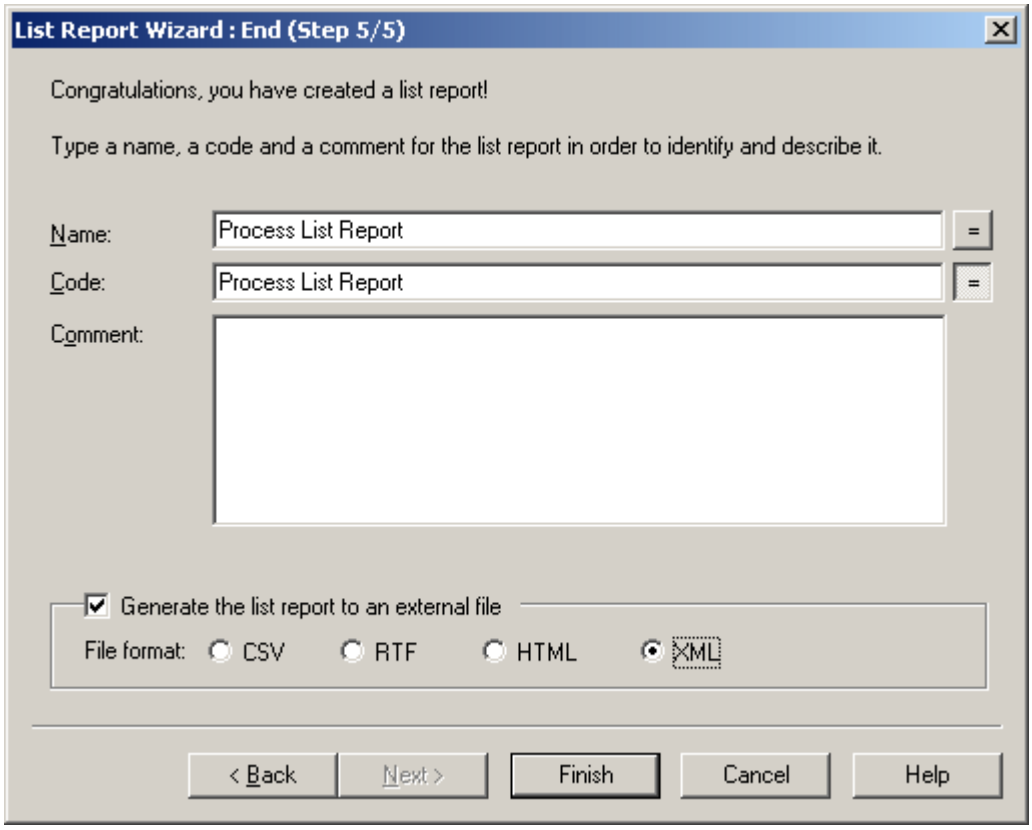


4. Click *Next* to go to the Content Preview page allows you to visualize the results of your choices so far:



To alter the definition click the *Back* button.

5. Click *Next* to go to the End page, which allows you to specify a name and code for the list report, and to add a comment. You can also specify whether to generate the report to an external file.



6. Click *Finish* to exit the wizard and create the report.

   The list report is added to the model and listed in the Browser under the List reports folder. If you have selected to generate the report to an external file, you will be asked to specify a file name.

1.8.2.1 List Report Properties

You can modify an object's properties from its property sheet. To open a list report property sheet, double-click its Browser entry in the List Reports folder or in the List of List Reports. By default, the property sheet of a list report always opens on the Content tab.

The *General* tab contains the following properties:

Property	Description
Name	Specifies the name of the list report.
Code	Specifies the code of the list report.
Comment	Descriptive comment for the object.

Property	Description
Object type	Specifies the type of the object the list report will be based on. You must select an object type to have any content in your list report.
Model or Package	Specifies the model or package from which the objects will be drawn. Package selection is unavailable for global objects such as users.
Include shortcuts	Specifies whether the list report includes shortcuts.
Include sub-packages	Specifies whether the list report includes sub-packages. This option is unavailable for global objects such as users.

The following tabs are also available:

- *Column Filter* - Lets you choose the object properties to be used as column titles in the list report and the order in which they will be displayed.
- *Row Filter* - Lists the filters that restrict the rows included in the list report. Select a property in the left-hand pane to add it to the list in the right-hand pane, select an operator, and then enter an expression to filter by (see Defining a Filter Expression [page 127]). You can specify to sort the list on the values of a property by selecting the S[ort] checkbox.
- *Content* - Displays the current values for the list report. Properties selected in the *Column Filter* tab are displayed as column headings and property values in rows must satisfy filter expressions defined in the *Row Filter* tab. You can open the property sheet of any object included in the list report by clicking the *Properties* tool.
  Note that data displayed on this tab does not dynamically change when you perform a change to your model that may affect your list report. To update the data, you must click the *Refresh List Report* tool.
  Use the generation tools to generate a list report result set as an RTF, HTML, XML, or XLS/CSV file (see Generating a List Report [page 310]).
- *Result Sets* - lists the result sets stored for the list report (see Creating Result Sets [page 309]). Click the *Properties* tool to open a result set for generation.

1.8.2.2 Creating Result Sets

A result list is a snapshot of the content of a list report at a given system date and time. You can create as many result sets as you want for the same list report, in order to keep a history of a given list report for future reference.

You can create a result set in any of the following ways:

- Open the property sheet of a list report, click the *Result Sets* tab, and then click the *Create Result Set* tool.
- Right-click a list report in the Browser and select *New* > *List Report Result Set* from the contextual menu.

Result sets are stored within the model and display under the list report to which they are related in the Browser:

Result Set Properties

You can modify an object's properties from its property sheet. To open a result set property sheet, double-click its Browser entry in the List Reports folder just beneath the list report to which it applies. The following sections detail the property sheet tabs that contain the properties most commonly entered for result sets.

The *General* tab contains the following properties:

Property	Description
List Report	Name of the list report to which the result set applies.
Report Date	Date and time when the result set was created.
Comment	Descriptive comment for the result set that allows you to identify it.

The Content tab shows the contents of the result set. Use the generation tools to generate the result set as an RTF, HTML, XML, or XLS/CSV file.

1.8.2.3 Generating a List Report

You can generate a list report to a CSV, XLS, RTF, HTML, or XML file.

Context

- Right-click the list report in the Browser and select *Generate* > `<Format>`.
- Open the property sheet of the list report, select the Content tab, and click the Generation *Format* tool.
- Select *Report* > *Generate List Report* to open the Generate List Report dialog box.

You can also right-click a result set in the Browser and select *Preview* or *Print*.

1.8.2.4 Importing and Exporting List Report Files

It can be useful to exchange list report files between models of the same type. For example, you may want to compare the properties of a certain subset of classes in two OOMs. You can import and export list reports to and from your model using the List of List Reports (*Report* > *List Reports*).

The following tools are available on the List of List Reports:

Tool	Description
	List Report Wizard - Launches the List Report Wizard (see Creating a List Report [page 303]).
	Import List Report from File - Opens a standard Open dialog, which allows you to browse to the list report file to import. A list report file has a .LRT extension.
	Export List Report to File - Opens a standard Save As dialog, which allows you to browse to the location to save the list report.

1.8.3 Creating a MultiModel Report

A multimodel report (MMR) is a PowerDesigner report that can document any number of models together and show the links between them. To create such a report, you must have at least one model open in the workspace, and you can add additional models at any time.

Procedure

1. Select *File* > *New Model* and choose Multi-Model Report.
   This item may be in the left or right hand pane, depending on the configuration of your New Model dialog (see Creating a Model [page 12]). It is not available if you do not have at least one model open in your workspace.
2. Enter a *Report name*, specify the *Language* to use for report titles and other framing text, and select a model from the *Model name* list as the basis for the first section of the report.
3. [optional] Select a template (see Creating a Report from a Template or an Empty Report [page 302]) for this first section from the Report Template list.

   > **i Note**
   >
   > If you choose a template in a language different from the one specified for the report, standard report titles and framing text will be presented in the report language and any user-defined text in the template will be in the template language.

4. [optional] To use the new report format and editor (see The Report Editor [page 312]), deselect *Use legacy Report Editor*.
5. Click *OK* to create the multimodel report and open it in the Report Editor.

6. [optional] To add a second model open in the workspace, select *Report* > *Add Section*, select the model to add (and, optionally, a template), and click *OK*.

   Each model in the report belongs to a separate section. You can only add models that are currently open in the workspace, but once a model has been added, you can edit the report even if the source model has been closed.

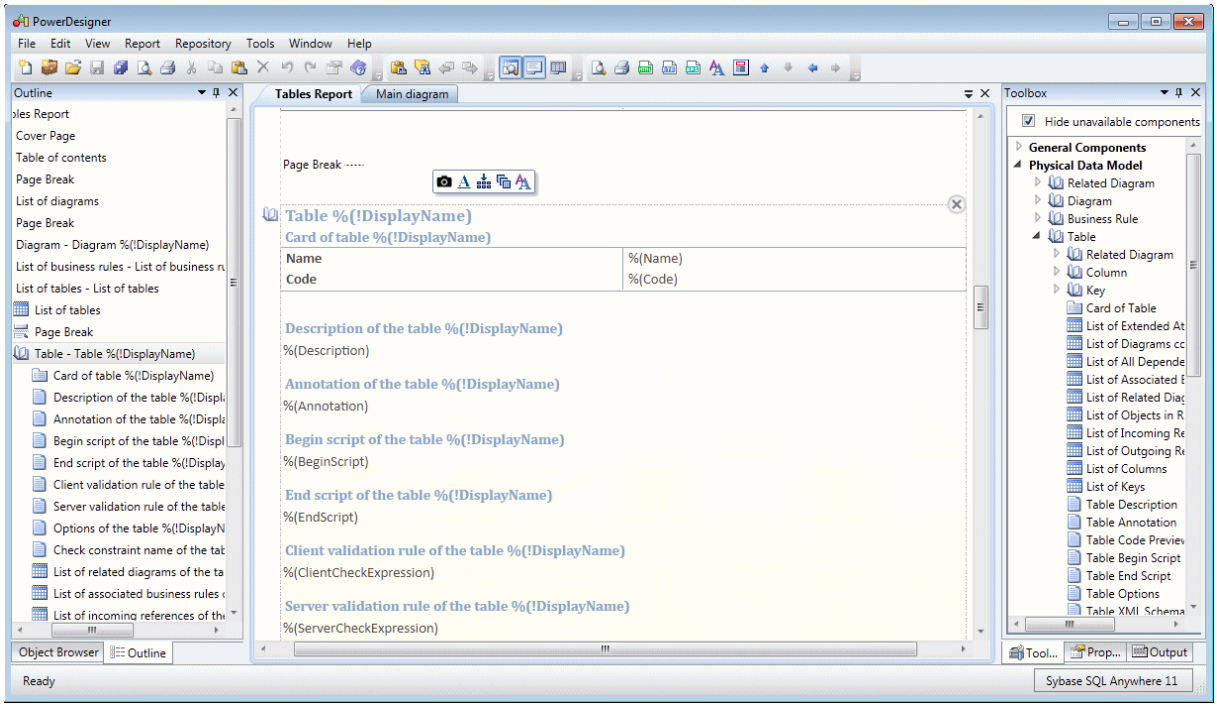
   > **i** Note
   >
   > To add a section to a legacy report, use the report property sheet *Sections* tab (see Report Properties [page 358]).

7. [optional] Launch the Report Wizard to populate and format the section (see Creating a Report with the Report Wizard [page 296]).
8. [optional] Edit the report in the Report Editor (see The Report Editor [page 312]).
9. [optional] Generate the report (see Generating a Model Report [page 356]).

1.8.4 The Report Editor

You can edit model reports, multi-model reports, and report templates in the Report Editor. To open a report in the editor, double-click its entry in the Browser or select *Report* > *Reports*, select your report in the list, and click the *Edit Report* tool.

Reports are composed of report items, which provide various ways of documenting model objects (books, cards, lists) and other report ornamentation such as a table of contents, page breaks, and free text (see Adding Items to a Report [page 314]).



The Report Editor provides four views:

- *Outline* (left) - Provides an overview of the tree structure of your report, and supports reordering by drag and drop and various other operations by right-clicking items.
- *Design* (center) - Provides a detailed graphical view of the report, and supports wysiwyg editing of item layout and formatting. The following tools are available on the command bar, which appears when you select a report item in the Design view or by right-clicking an item in the Design or Outline views):

Tool	Description
	Quick View - Previews the item (see Previewing a Report or Report Item [page 333]).
No Tool	*Up / Down* - Moves the item up or down one place in the report.
No Tool	*Promote / Demote* - Moves the item up or down one level in the report.
	Properties - Selects properties to display in cards or lists, and reorder them (see List and Card Report Items [page 318]).
	Collection - [books, cards, lists] Changes the type of object that will be published to the item.
	Selection - [books, cards, lists] Specifies the individual objects that will be published to the item (see Selecting Objects to Display for a Report Item [page 329]).
	Format/Styles - Modifies the specific format for the item or the default styles for all items of this type (see Specifying Global Styles and Local Formats for Report Items [page 327]).
No Tool	*Show Title* - Toggles the display of the item's title in the body of the report.
No Tool	*Hierarchical Display* - [composite object books] Toggles the organization of composite object book items, such as those representing packages, OOM activities, or BPM processes between List (where they are all displayed on one level as siblings) and Hierarchical (where child objects are displayed inside their parent).
No Tool	*Show Title in HTML TOC/Show Contents in HTML TOC* - Toggles the display of the book item title and the the book's sub-item titles in HTML report table of contents.
No Tool	*Delete* - Hover over an item and then click the X in its upper right corner to delete the item.

- *Toolbox* (right) - Lists the report items you can add to your report by double-clicking or dragging and dropping them onto the Design or Outline views (see Adding Items to a Report [page 314]). By default, only items associated with objects present in your model are displayed. To show all items, deselect the *Hide unavailable components* check box at the top of the Toolbox.
- *Properties* (right, hidden) - Lists the properties of the report item selected in the Design or Outline views.

The following tools are available from the Report toolbar:

Tool	Description
	Print Preview/Print - (see Previewing a Report or Report Item [page 333]).
	Generate HTML/RTF/XPS - Generates the report in the specified format.

Tool	Description
	Styles - Opens the Default Styles dialog (see Specifying Global Styles and Local Formats for Report Items [page 327]).
	Show/Hide Header/Footer - Toggles the display of the report header and footer (see Modifying Report Section Headers and Footers [page 332]).
	Up / *Down* - Moves the selected item up or down one place in the report.
	Promote / *Demote* - Moves the item up or down one level in the report.

1.8.4.1 Adding Items to a Report

You can add report items by double-clicking or dragging and dropping tools in the report Toolbox. Generic report items are available under the General Components section of the report Toolbox, while items instantiated for the objects present in your model are available under the Model section. Standard and extended objects and properties are all available for selection. You can insert the same item multiple times in a report.

To add an item to a report:

- Below the selected item - double-click the appropriate tool in the Toolbox.
- Anywhere in a report - drag the tool from the Toolbox and drop it in the appropriate position in the Design or Outline views. When the cursor is over an existing report item in the Design view, arrows appear to guide the placement of the new item. To place the item next to, above, or below the existing item, drop it on the left, right, top, or bottom arrow. To add the item as a child of the existing item, drop it on the center guide arrows.

The following tools are available to insert report items to document model objects:

Icon	Description
	Book – Groups together cards, lists, and text items to display all or most of the properties of an object and may contain sub-books to describe any sub-objects (see Book Report Items [page 317]). **i Note** To display packages or other composite objects (such as BPM processes that can contain other processes) hierarchically with children shown beneath their parent instead of all at a single level, add the appropriate book item, right-click it, and select *Hierarchical Display*. HTML reports always use hierarchical display.
	Card – Displays a configurable subset of properties of an object in a two-column table with the property name in the left column and the value in the right column (see List and Card Report Items [page 318]).

Icon	Description
	List - Displays a list of objects in a multi-column table with one property in each column. (see List and Card Report Items [page 318]).
	Text – Displays an object property or free text (see Text Property and Free Text Report Items [page 319]).
	Diagram - Inserts a PowerDesigner diagram (see Diagram Report Items [page 320]).
	Chart - Inserts a chart to visualize statistics about a collection of objects (see Chart Report Items [page 321]).
	Matrix - Inserts a dependency matrix (see Dependency Matrices [page 199]).
	Section - Inserts a section. Sections are only supported in multimodel reports, where each section contains a report relating to a specific model (see Creating a MultiModel Report [page 311]).

> **i Note**
>
> By default, a report item will generate information for every object of the appropriate type in the model. For example, the Table book will generate information for every table in your model. To generate for only a subset of objects defined by name or criteria, use the *Edit Selection* tool in the item's command bar (see Selecting Objects to Display for a Report Item [page 329]).

The following tools are available to add miscellaneous additional report items:

Icon	Description
	Cover Page - Inserts a set of text items followed by a page break to create a cover page:
	Image File - Displays an image, selected via a file chooser.
	Column Panel - Inserts a container element that holds two report items side-by-side. A column panel is automatically created when you drag an item onto the Design view and drop it on the left or right guiding arrow of an existing item.
	Page break – Inserts a page break. Right-click the inserted item and deselect Generate in HTML to disable it when generating HTML reports.
	Table of contents – Inserts a table of contents followed by a page break.
	Text – Inserts a text item. This can be a free text paragraph or, if you specify a data source, the value of a property.
	Text file – Inserts the contents of a text file, selected via a file chooser.
	Title - Inserts a titled container. Add items to the container, which inherits the scope of its parent item, as necessary.

> **i Note**
>
> You can move items from one report to another report of the same type of model by drag and drop. Press the `Ctrl` key while dragging and dropping to copy the item to the second report.

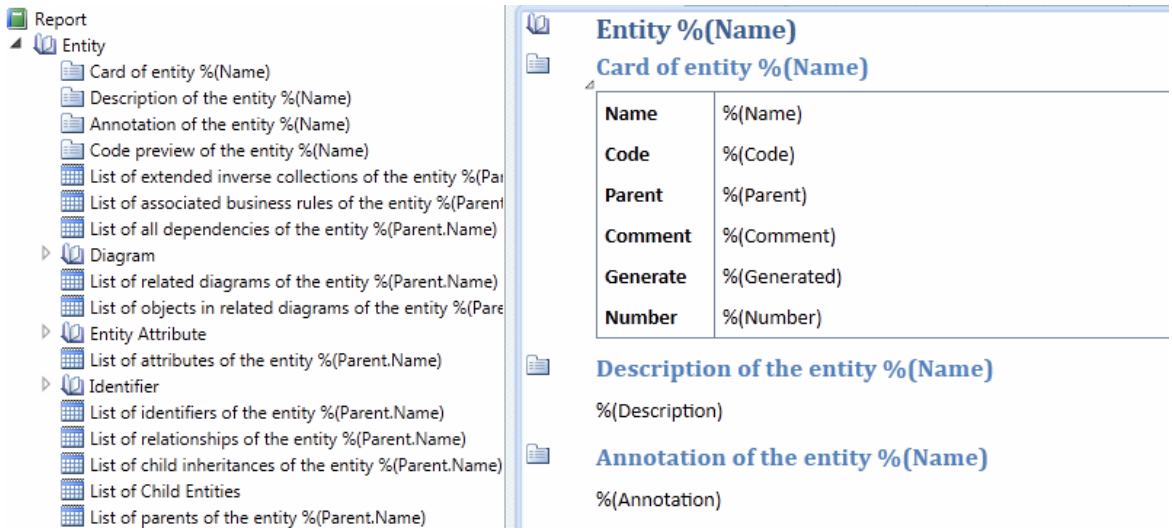
1.8.4.2 Book Report Items

Book report items can contain cards, lists, text items, and sub-object books to give detailed information about your model objects.

Procedure

1. To insert a book item, click in the Design or Outline view in the position where you want to add the item and then double-click or drag and drop the appropriate Toolbox item.

   A book is displayed in the Outline view as a node with a number of subitems, and the same series of items are shown in the Design view:



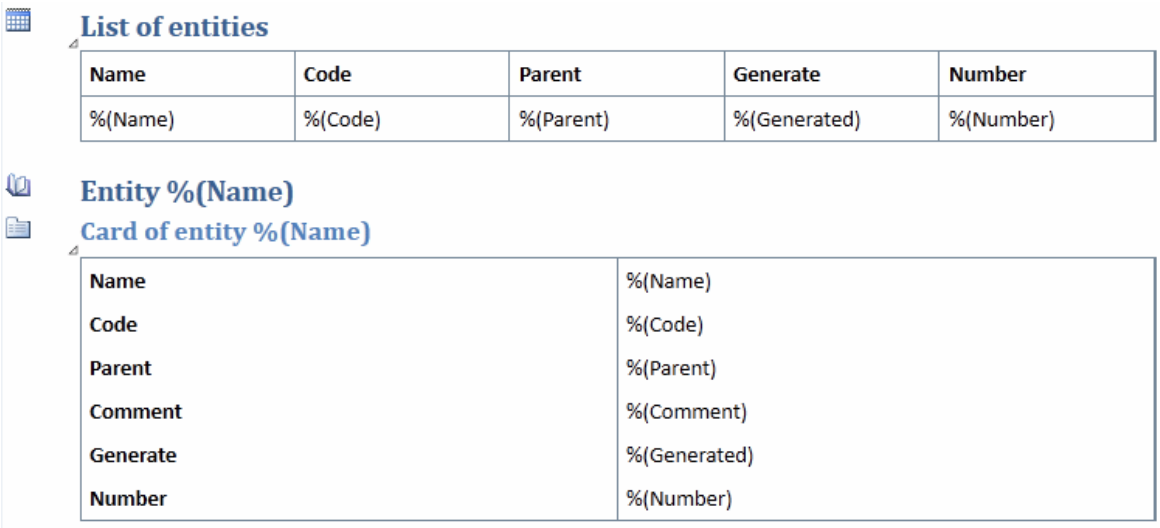
2. To remove unwanted sub-items from the book, select them in the Outline view and press *Delete* or hover over them in the Design view and click the X in the upper-right corner of the item.
3. To reorder sub-items inside the book, drag and drop them in the Outline or Design view.
4. For composite objects (such as packages that can contain other packages or BPM processes or OOM classes that can contain other processes or classes) you can choose between two presentations:
   - List Display - [default] Objects, their children, grandchildren, and subsequent descendants are all generated at the same level.
   - Hierarchical Display - Children are generated as sub-items inside the parent object book, grandchildren are generated as sub-items inside the child book, and so on to any level of decomposition. Select the *Hierarchical Display* option in the Properties view.
5. To modify the selection of objects that will be generated in the report from this book and how they will be sorted, click the command bar *Selection* tool and specify filter and sorting criteria or a manual selection of objects (see Selecting Objects to Display for a Report Item [page 329]).
6. To modify the default styles for a sub-item, click its icon to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.

1.8.4.3 List and Card Report Items

List and card report items are generated as tables. A list displays all objects, with one column per property and one row per object. A card displays a single object, with one row per property. You can specify which attributes will be displayed, in which order, and the width of the columns.

Procedure

1. To insert a list or card item, click in the Design or Outline view in the position where you want to add the item and then double click or drag and drop the appropriate Toolbox item.



List of entities

Name	Code	Parent	Generate	Number
%(Name)	%(Code)	%(Parent)	%(Generated)	%(Number)

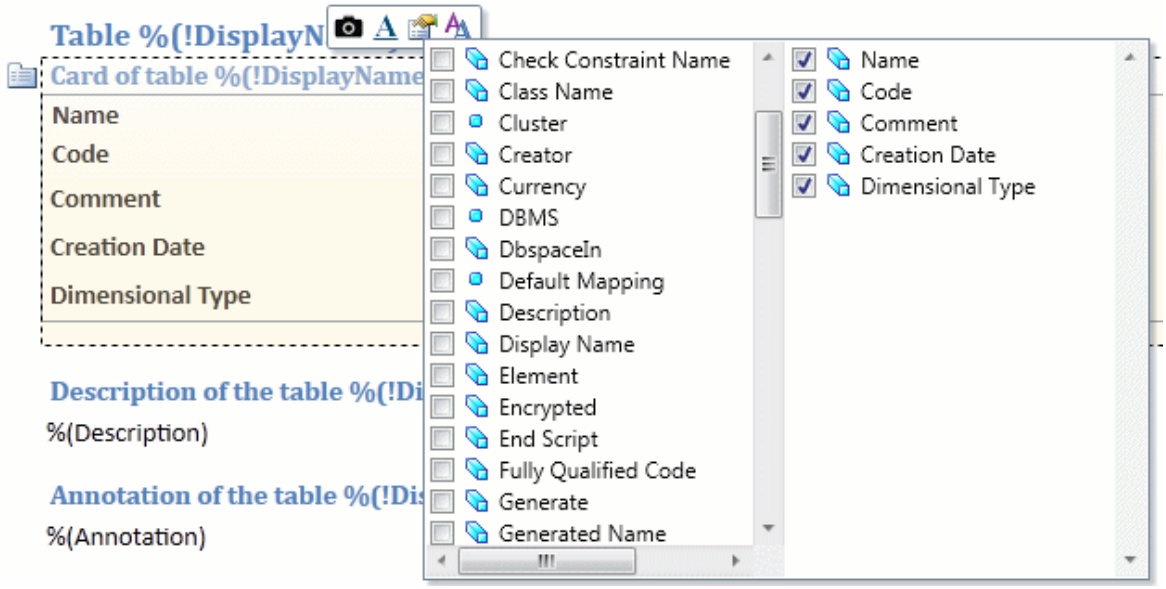
Entity %(Name)

Card of entity %(Name)

Name	%(Name)
Code	%(Code)
Parent	%(Parent)
Comment	%(Comment)
Generate	%(Generated)
Number	%(Number)

> **i** Note
>
> Cards and lists are commonly added to the report as part of a book (see Book Report Items [page 317]).

2. To modify the properties displayed, click the card or list icon in the Design view to display the command bar, and then click the *Properties* tool to display a list of available properties, with properties selected for display on the right, and unselected properties listed in alphabetical order on the left:



3. Select or deselect properties as appropriate and then click outside the list to close it. Properties selected will be added as columns in a list or rows in a card.

> **i** Note
>
> All properties, including extended attributes are available for selection in this list.

4. To reorder list columns or card rows click in a list column header or card row header to select it, and then click again and drag and drop it to the desired position.
5. To modify the width of list columns or the column separating the card row header from the property values, hover over a dividing line until the double-arrow cursor appears, and then click and drag to increase or decrease the width.
6. To modify the default styles for a list or card item, click its icon to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.

1.8.4.4 Text Property and Free Text Report Items

Multi-line text properties, such as the Comment or Code Preview, are provided by default as text items. You can also add free text items wherever necessary.

Procedure

1. To insert a text property report item, click in the Design or Outline view in the position where you want to add the item and then double click or drag and drop the appropriate Toolbox item.

> **Note**
>
> Text properties are commonly added to the report as part of a book (see Book Report Items [page 317]).

2. To edit a free text item, double-click the item in the Design view and enter text as appropriate.
3. To modify the property referenced by a text property item, enter the name of the property in the format `%(Property).`
4. To modify the default styles for a text item, click its icon to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.

1.8.4.5 Diagram Report Items

Diagram report items are generated as images. If a report is generated as HTML, diagram symbols are clickable targets, with links to the corresponding object card.

Procedure

1. To insert a diagram item, click in the Design or Outline view in the position where you want to add the item and then double click or drag and drop the appropriate Toolbox item.

> **Note**
>
> You can display objects organized by the diagram in which they appear by dropping the appropriate book, card, or list items and dropping them onto a diagram item.

2. To modify the default styles for a diagram, click its icon to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.

   The zoom option offers the following settings:
   - **`Fit to page`** - Shrinks the diagram to fit on a single report page.
   - **`Custom zoom`** - Zooms by the given percentage.
   - **`Keep diagram layout`** - Each diagram page appears full size on its own report page.

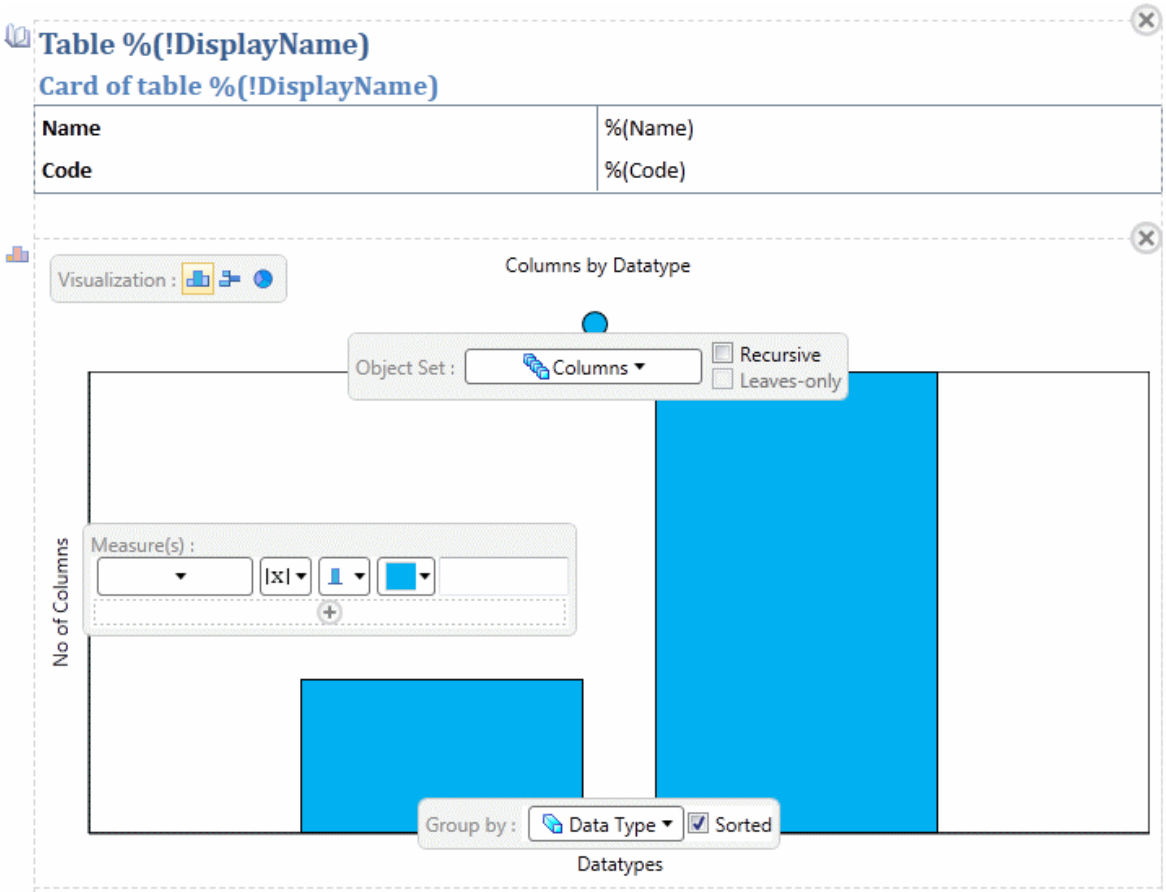
1.8.4.6 Chart Report Items

Chart report items provide a way to visualize statistics about a collection of objects as a column, bar, or pie chart.

Procedure

1. To insert a chart item, click in the Design or Outline view in the position where you want to add the item and then double-click or drag and drop the *Chart* tool.

   The position in which you drop the chart determines the statistics that you can visualize. In this example, the chart is a child of the Table book, which provides access to the Columns sub-object collection. To analyze tables or any other collection of standard objects, you should drop the chart at the top level of your report. You can base a chart on any of the standard collections in the PowerDesigner metamodel, or create you own extended or calculated collection to analyze (see *Customizing and Extending PowerDesigner > Extension Files*).

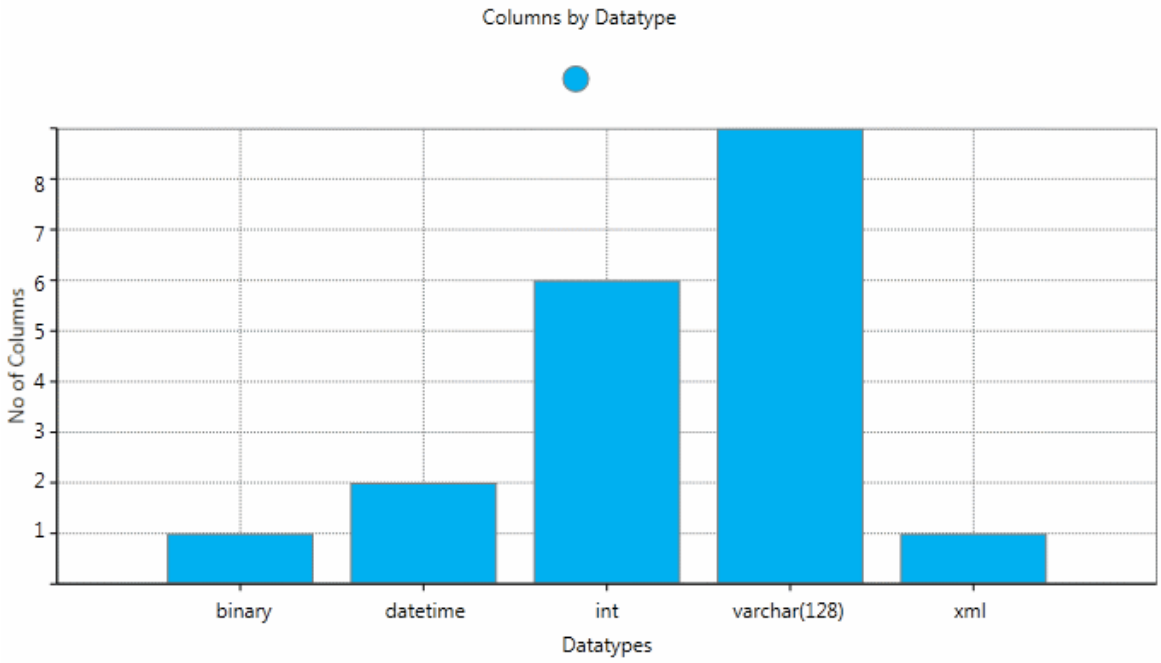


2. Select the type of visualization to use. You can choose between:
   - Column chart - vertical columns and/or line graphs.
   - Bar chart - horizontal bars or and/or line graphs.
   - Pie chart - a circle divided into sectors.

3. Specify the parameters of the chart:
   - *Object set* - Specifies the collection of objects that the chart will analyze.
     - *Recursive* - Specifies that the chart should include all sub-objects in the case of composite objects such as areas or BPM processes, and objects in packages. In the case of composite objects, selecting the *Leaves-only* sub-option restricts the data set to only those final objects in the tree that are not further decomposed.

     > **i** Note
   - *Group by* - Specifies the property that will be used to categorize objects for display. If you select nothing or the **Name** property, then one bar or segment will be displayed for each object. If you select a property of type date, you can choose to group by day, week, month, quarter, or year.
   - *Measure(s)* - Specifies how the size of bars or segments will be calculated. While a pie chart can only support one measure, column and bar charts can support multiple measures. For each measure, you can also specify:
     - The measure type - Count is always available, and sum and average are available for numeric properties.
     - The base color from which the color scheme is determined and the label to give to the measure. For column or bar charts, you can also specify whether the measure is represented as bars/columns or as a graph line.
4. Replace the placeholder text to the left of, below, and above the chart to give labels to the x and y axes and to provide a title for the chart.You can specify to print the column or bar names diagonally using *Primary* and *Secondary Axis Labels Rotation Angle* properties in the *Properties* view.

> **i** Note
>
> The graphic displayed in the Design view is for illustration purposes only. To preview your chart with actual data, click the chart background and then click the *Quick View* tool.



5. To modify the default styles for a chart item, click its icon to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.

1.8.4.6.1 Example: Creating Charts for Hardware Servers

In this example, we will create three charts to visualize information about a set of hardware servers modeled in an enterprise architecture model. We will analyze the distribution of servers by geographic site, the operating system split, and the quarterly spend on servers (using extended attributes).

Context

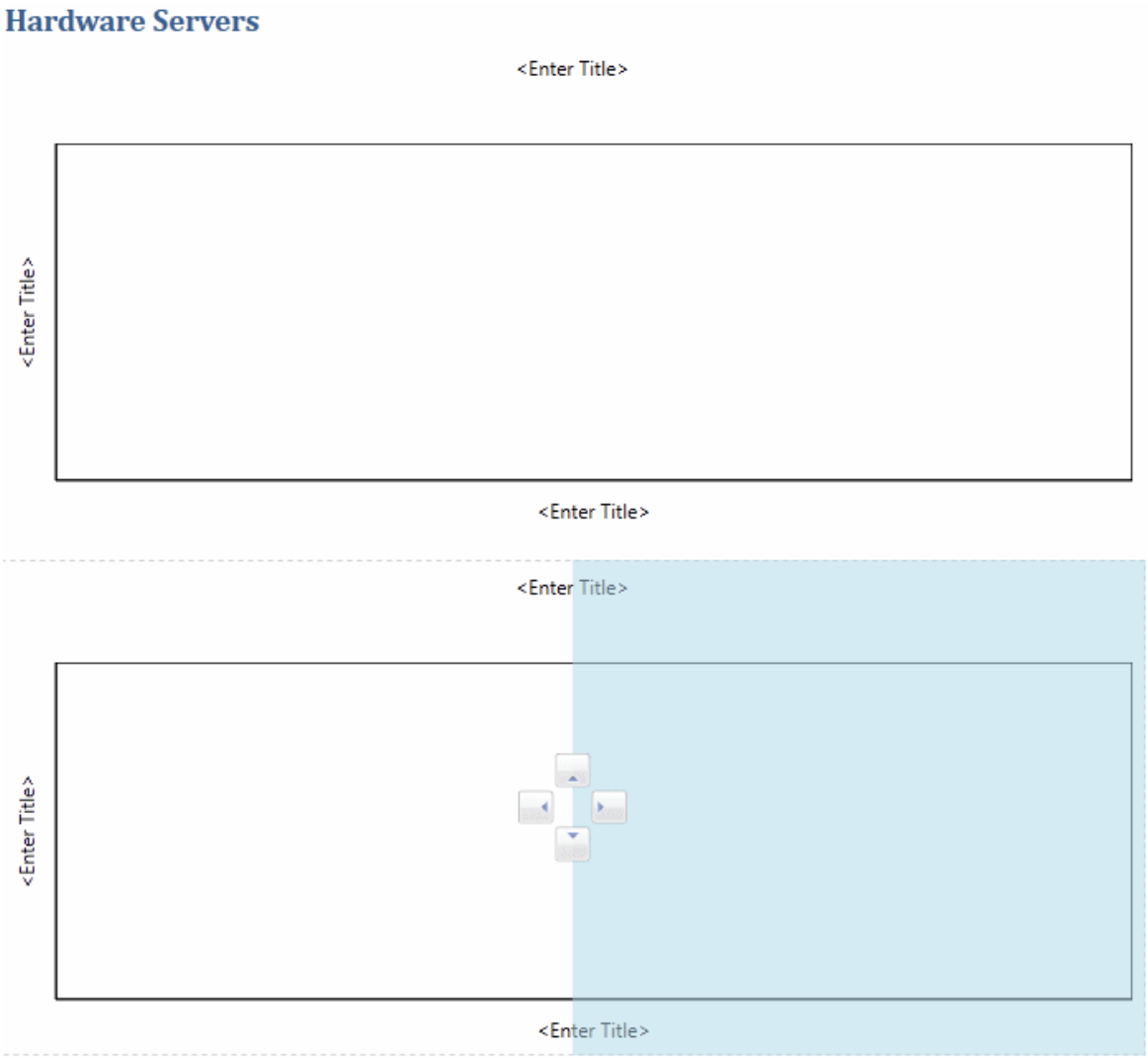
Before creating the report, the EAM model was created and populated with hardware servers assigned to various sites, and two new properties were added to the Hardware Server property sheet (see Adding New Properties to an Object [page 168]) as extended attributes:

Name	Data Type
Purchase Date	Date
Purchase Cost	Integer

Procedure

1. Create a report, expand the General Components category in the report Toolbox, and double-click the *Title* tool to add a title. Click the default title text and enter `Hardware Servers`.
2. Click the *Chart* tool and drag and drop it onto the center of the guide arrows on the title item to create a chart in the Hardware Servers item:

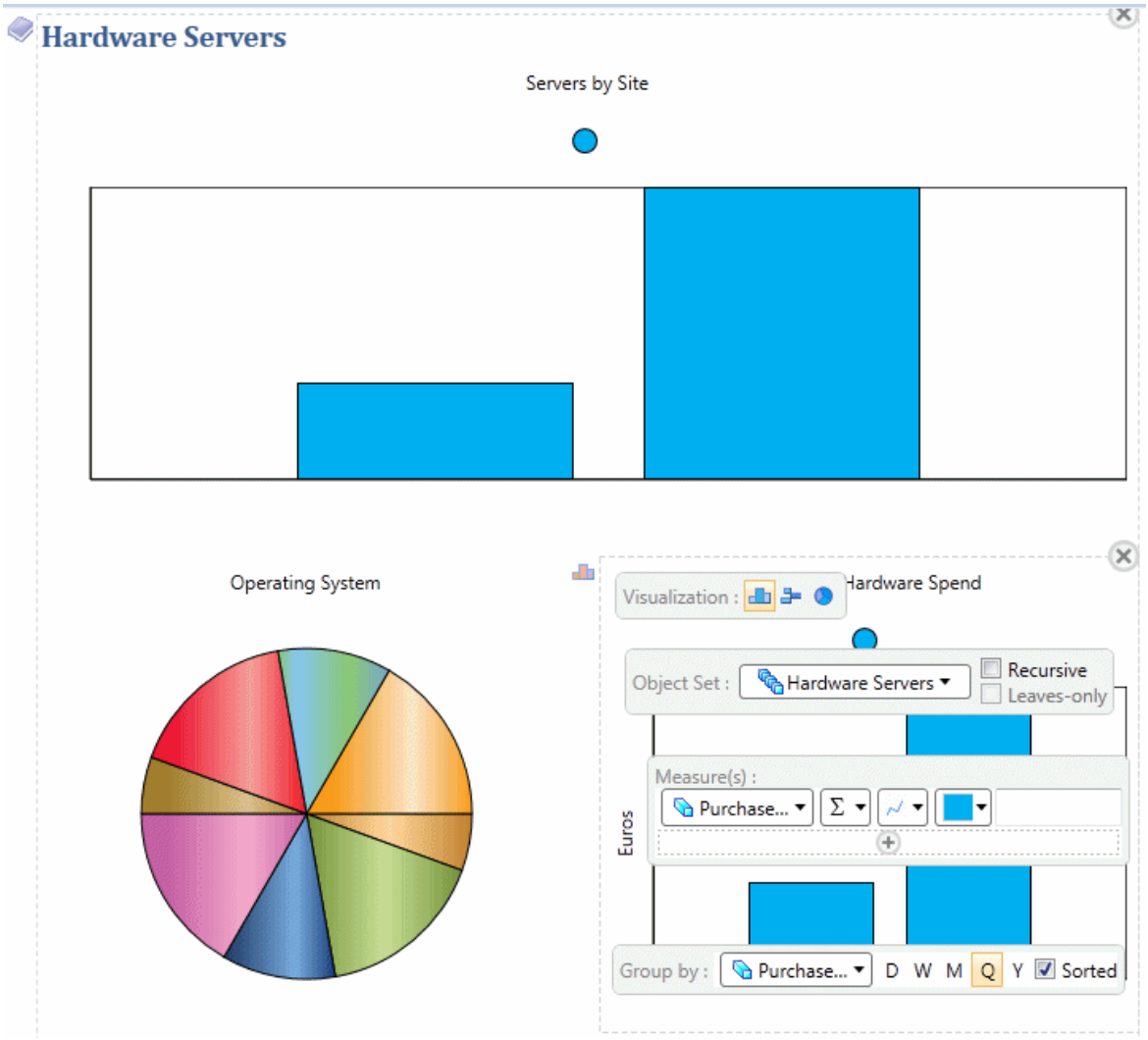
3. Drag the *Chart* tool again and drop it onto the bottom guide arrow on the first chart item to create a new chart underneath it. Then drag the *Chart* tool a third time and drop it on the right guide arrow on the second chart to create a new chart next to it:



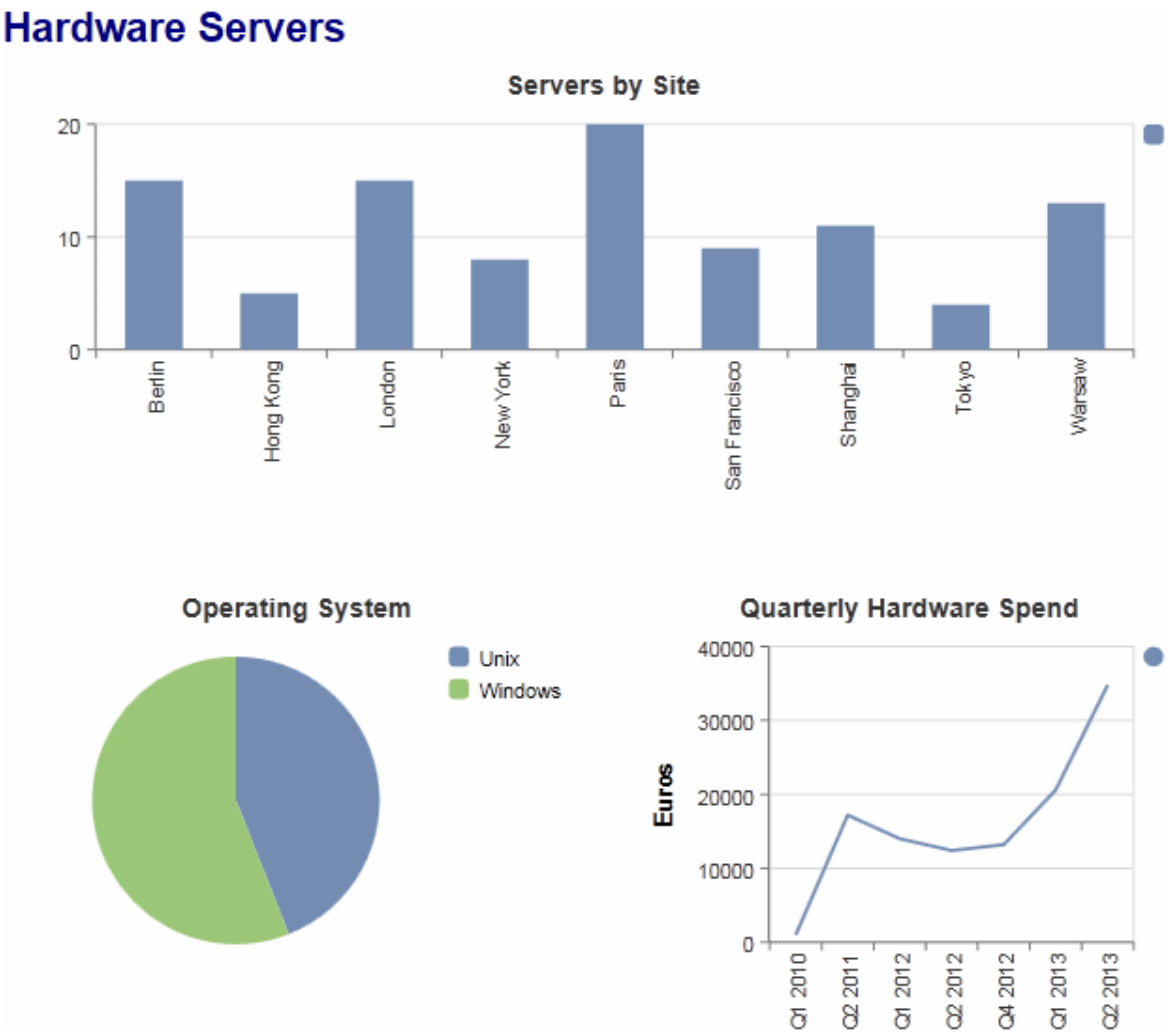
4. Enter the following chart parameters:

Parameter	First Chart	Second Chart	Third Chart
Visualization	Column	Pie	Column
Object Set	Hardware Servers	Hardware Servers	Hardware Servers
Group by	Site	Operating System	Purchase Date/ ([Q]uarter)
Measure	[None]/Count/Bar	[None]/Count	Purchase Cost/Sum/Line
Top Title	Servers by Site	Operating System	Quarterly Hardware Spend
Left Title	[Space]	N/A	Euros
Bottom Title	[Space]	N/A	[Space]

> **Note**
>
> The representation of the chart in the Design view does not change as you complete the parameters. You can, at any time, click the background of a chart and then click the *Quick View* tool to preview it with the data from your model.



5. Click the *Generate Web Report (HTML)* tool in the report tool bar and select to open it in your browser:

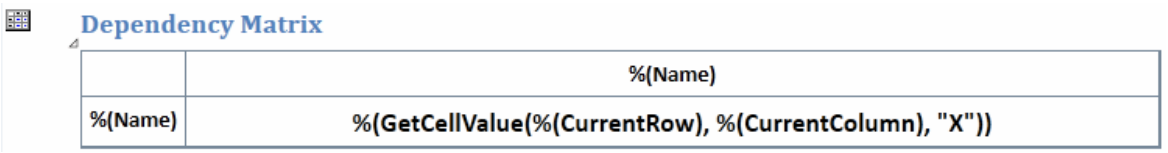


1.8.4.7 Dependency Matrix Report Items

Dependency matrix report items are generated as tables with one row per object and one column per dependent object with an X or other value in matrix cells to indicate a dependency.

Procedure

1. To insert a dependency matrix item, click in the Design or Outline view in the position where you want to add the item and then double click or drag and drop the appropriate Toolbox item.



Dependency Matrix

	%(Name)
%(Name)	%(GetCellValue(%(CurrentRow), %(CurrentColumn), "X"))

> **i Note**
>
> Dependency matrices are commonly added to reports as part of the Dependency Matrix book item, which also describes the properties of the matrix. To add only the matrix itself, expand the matrix book and then double-click or drag and drop the matrix item.

2. To modify the default styles for the matrix cells, column header, or row header, click in the appropriate field to display the command bar, and then click the *Styles* tool (see Specifying Global Styles and Local Formats for Report Items [page 327]). To override the default styles for this instance, click the *Format* tool.
3. To modify or override the default styles for row and column header dividers, the horizontal and vertical cell dividers and the overall format of the matrix, click in the top left field and then click the *Styles* or *Format* tool.

1.8.4.8 Specifying Global Styles and Local Formats for Report Items

The formatting of your report is highly customizable. You can set global default styles and override them for individual report items by specifying local formats.

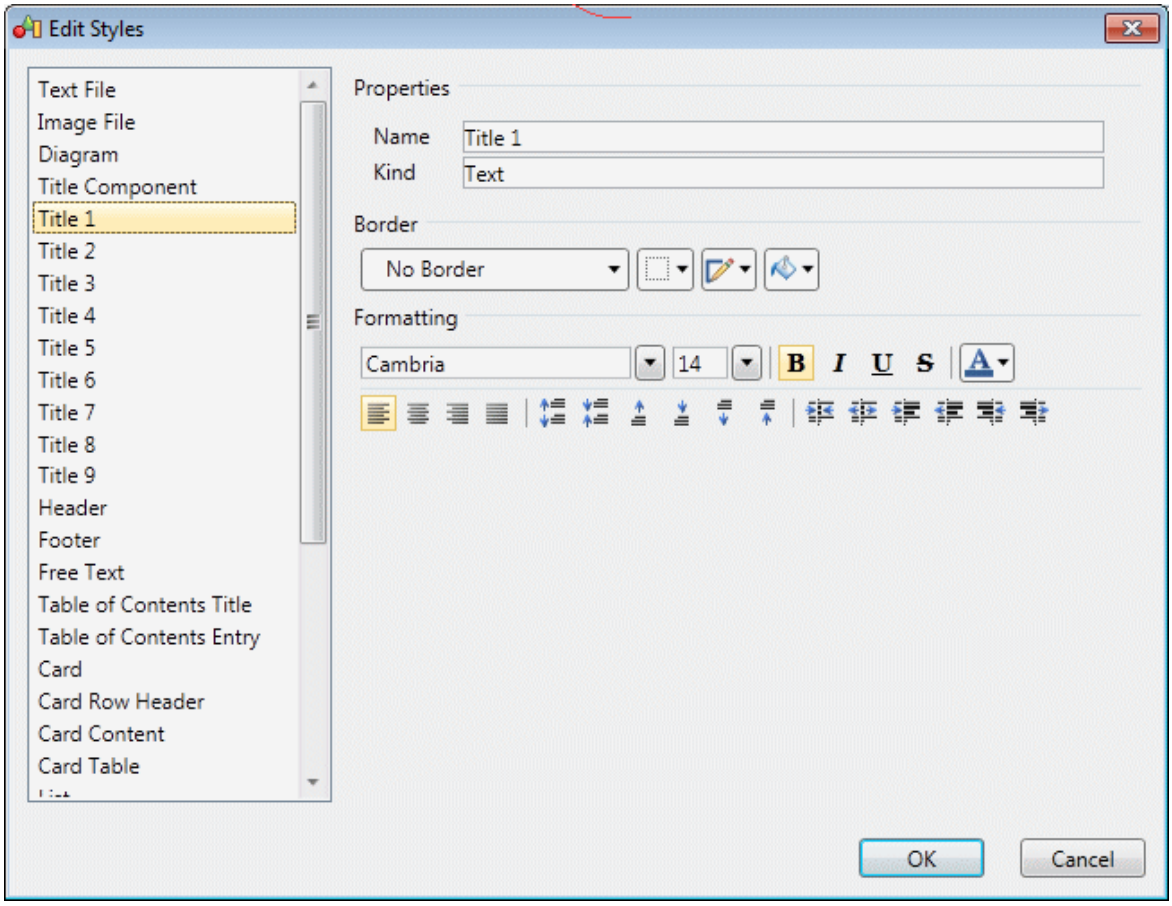
Context

Use the following tools on the Design view command bar to access global default styles and individual formats:

Tool	Description
	Format - Specifies the format for the selected report item
	Styles - Specifies the default global style for all report items of this type.

Procedure

1. Select ▶ *Report* ▶ *Styles* ▶ from the Report Editor window to open the Styles dialog:



2. Select the report item element for which you want to specify a default style in the left pane. The following elements are available:
   - Text / Image File
   - Title Component - Control the borders, fill, and spacing around the container headed by a title report item.
   - Title 1-9
   - Header / Footer
   - Free Text
   - Table of Contents Title / Entry
   - Card / List / Matrix - Control the borders, fill, and spacing around the item (including its title).
   - Card Row Header / List Column Header / Matrix Row / Column Header - Control the font and spacing of the header cells.
   - Card Content / List Content / Matrix Content. - Control the font and spacing of the content cells.
   - Card Table / List Table / Matrix Table - Control the borders and spacing of the table and the internal dividers between the header and the cells and between the cells.
3. Specify the default styles for the selected element using the following tools. Changes are visible immediately in the Design window:

Tool	Description
	Font
	Font Size
	Bold / Italic / Underline / Strikethrough
	Font Color
	Align Left / Center / Right / Justify
	Borders / Border Color / Fill
	Increase / Decrease Vertical Spacing - Increase or decrease the spacing above and below the element equally.
	Increase / Decrease Top / Bottom Spacing - Increase or decrease the vertical spacing above or below the element.
	Increase / Decrease Horizontal Spacing - Increase or decrease the spacing to the left and right of the element equally.
	Increase / Decrease Left / Right Spacing - Increase or decrease the horizontal spacing to the left or right of the element.
	Row / Column Header Divider - Toggle the display of a dividing line between the table row or column header and the content cells.
	Horizontal / Vertical Cell Dividers - Toggle the display of dividing lines between the table content cells.

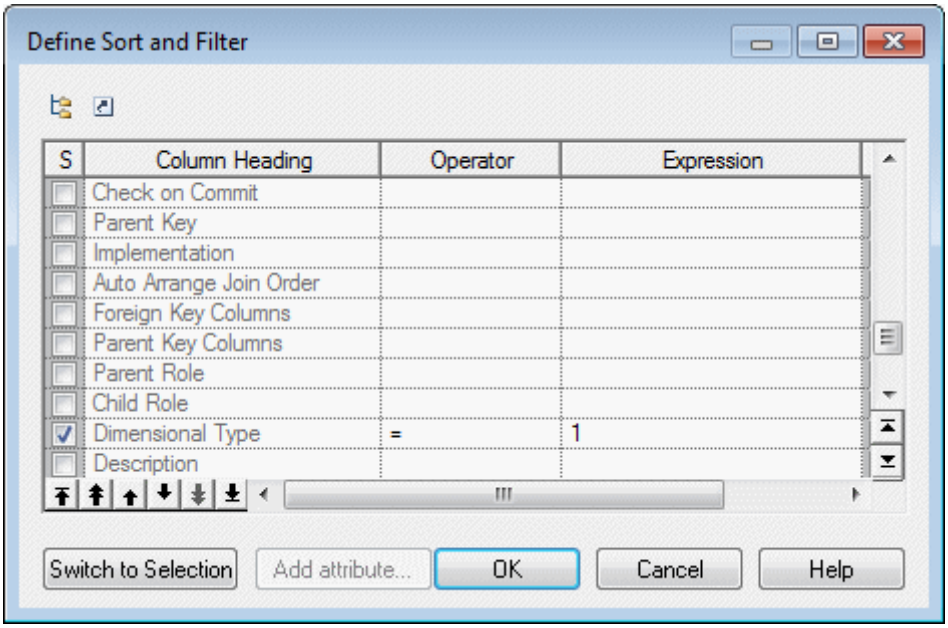
1.8.4.9 Selecting Objects to Display for a Report Item

You can refine the selection of objects for individual report items and control their order by criteria or by selecting individual objects from the list of objects included in the report. For example you can have a list of all the tables in your PDM, but you need detailed information about only some of them, and so specify a selection for the Table Book.

Context

Procedure

1. Right-click a book or list item and select *Edit Selection* to open the Define Sort and Filter dialog.
2. To define a filter and sorting criteria:
   - Select columns to sort on - by checking the [*S*]ort column checkbox.
   - Reorder the columns in the list - by using the arrows at the bottom left of the dialog. To move all the selected rows to the top of the list, press `Shift` + `Enter`.
   - Define one or more filter expressions - by selecting an operator in the *Operator* column and entering an expression to filter by in the *Expression* column. For information about the operators and expression syntax, see Defining a Filter Expression [page 127].

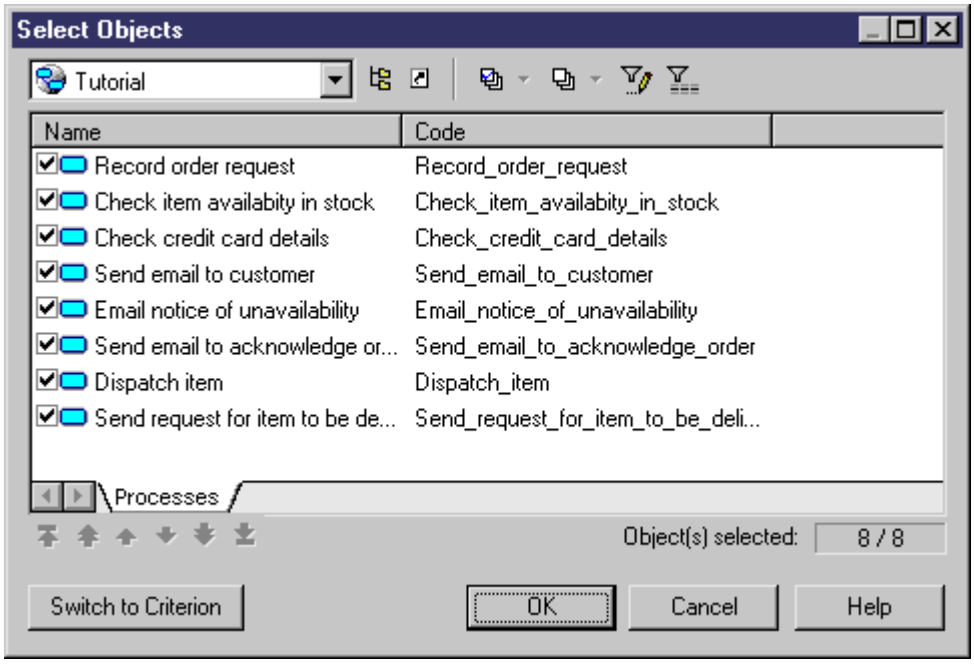


   The [*U*]sed checkbox is automatically selected when you enter a filter expression. To disable the expression but keep it in memory, clear the checkbox.

> **i Note**
>
> For lists containing multiple types of objects, to display properties other than those that are common to all the object types, click the *Add Attribute* button (see Customizing Columns in Lists Containing Multiple Types of Objects [page 124]).

3. To select individual objects from the list of objects included in the report, click the *Switch to Selection* button:

> **⚠ Caution**
>
> You can specify either a filter or a selection, but not both. When you click this button, any information entered in the Define Sort and Filter dialog will be lost.



> **i Note**
>
> If you have made a selection of objects at the report level (see Selecting Objects to Display in the Report [page 341]), objects excluded there will not be available for selection at the report item level.

4. By default, all the objects are selected. Select the objects that you want to display using this report item. You can reorder the objects in the list by using the arrow buttons at the bottom of the dialog.

> **i Note**
>
> If you define a selection of ordered objects on a package for which you have selected the *Hierarchical Display* option (see Adding Items to a Report [page 314]), any reordering you do here is ignored.

5. When you are satisfied with your filtering or selection, click *OK* to return to the Report Editor. Note that only the last of the filter or selection from which you clicked *OK* will be applied.

1.8.4.10 Modifying or Suppressing the Titles of Report Items

Most report items are preceded by a title item. To modify the title, select the item and then click the title text to enter edit mode. To suppress an item title from the body of the report, right-click the item and deselect the *Show Title* option.

You can reference object properties using the syntax `%(<Property>)`. For example, you could modify the title of the table card and use reference the Code property instead of the Name property:

Before	After
`Card of Table %(Name)`	`Table %(Code) Card`

> **i Note**
>
> You can edit the default titles of all report items in the report language resource file that is attached to your report. For more information, see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files* .

1.8.4.11 Modifying Report Section Headers and Footers

You can review and modify the report header and footer in the Report Editor Design view. In multi-model reports, you can specify a different header and footer for each section (model).

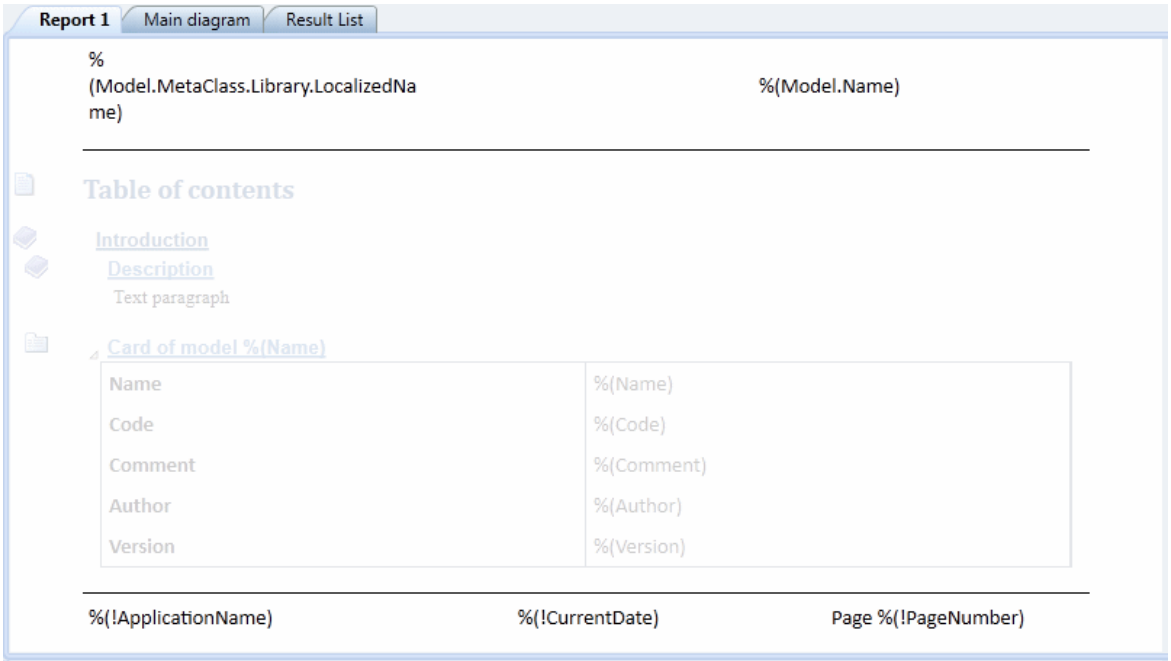
Context

For more information about the default header and footer, including specifying separate headers and footers for each section of your RTF report, see Report Properties [page 358].

Procedure

1. Click the *Header & Footer* tool to display the header and footer at the top and bottom of the Design view:

   Both the header and footer contain left, center, and right fields, which can contain a mixture of text and variables.



2. Click once in a field to display the command bar, from which you can launch a preview of the field or modify its format. Click again to edit the contents of the field, including modifying the variables referenced (see Referencing Object and Model Properties [page 333]).
3. [optional] Click the Preview tool in the command bar to review your changes.
4. Click the *Header and Footer* tool to return to your report.

1.8.4.12 Referencing Object and Model Properties

You can reference object and model properties in your reports using their public names as defined in the PowerDesigner Public Metamodel.

To reference object properties, use the following syntax:

```
%(Property)
```

For example, to reference the object code, use:

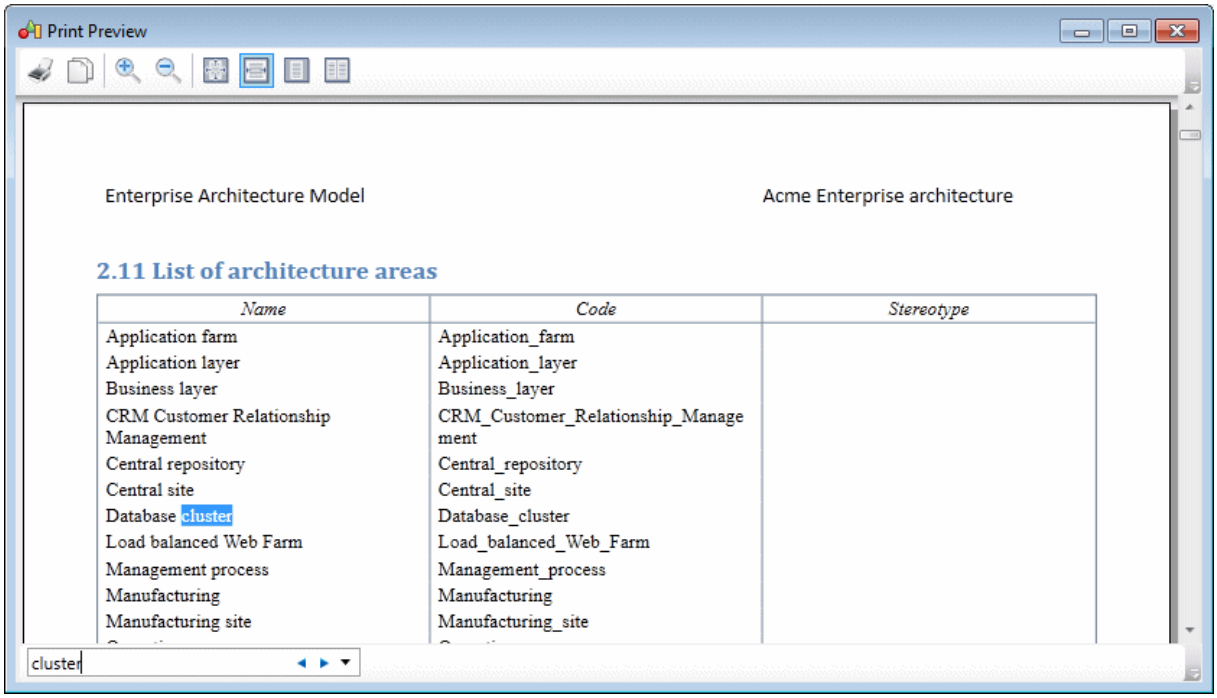
```
%(Code)
```

The following additional report variables are available:

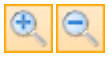
Variable	Description
%(!CurrentDate)	Report date defined in report property sheet.
%(!CurrentTime)	Report generation time.
%(Model.Name) %(Model.Code)	Model name or code of a report section.
%(Model.Metaclass.Library.Localized-Name)	Module name of a report section.
%(!ApplicationName)	Application name (PowerDesigner).
%(!PageNumber)	[RTF] Current page number.

1.8.4.13 Previewing a Report or Report Item

You can preview your report by selecting *Report* > *Print Preview* or by clicking the *Preview* tool in the End page of the Report Creation Wizard. You can preview a report item by selecting it in the Design or Outline view and clicking the *Quick View* tool.



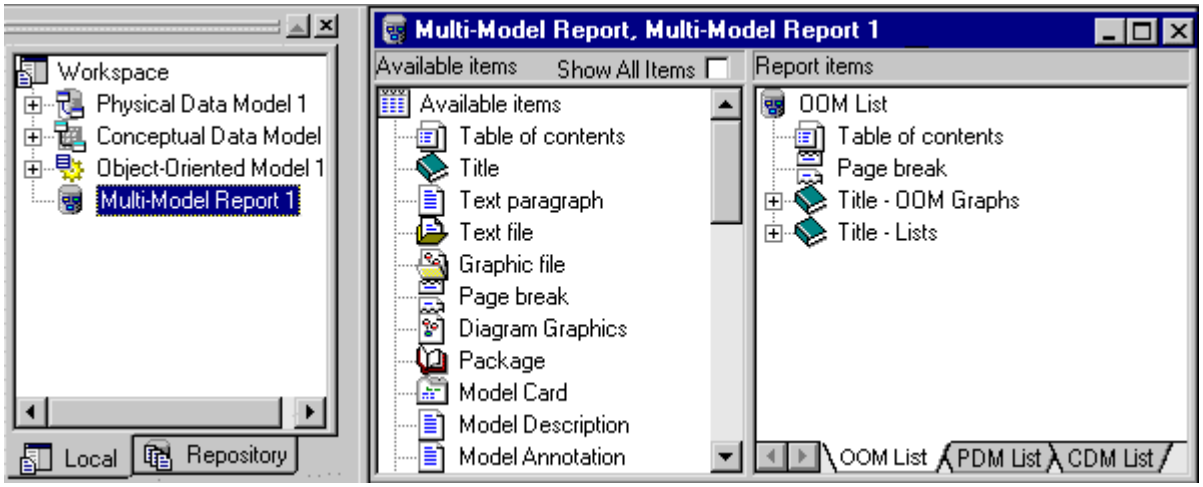
The following tools are available in this window:

Tool	Goes to preview
	Print - Prints the report.
	Copy - Copies the selected text.
	*Zoom In*/*Zoom Out*.
	Zoom to: • *100%* • *Page Width* • *One Page* • *Two Pages*

To search for text in your preview, enter the string in the search box at the bottom of the window and click the right or left arrow to search forward or backward.

1.8.5 The Legacy Report Editor

You can edit model reports, multi-model reports, and report templates in the Report Editor. To open a report in the Report Editor, double-click it in the Browser or select *Report* > *Reports* to open the List of Reports, select your report and click the *Edit Report* tool.



The left pane, *Available Items*, lists the items you can include in the report, while the right pane, *Report Items*, displays the structure of your report.

The *Available Items* pane lists only the report items that correspond to objects that exist in the current model. For example, if you have not created any interfaces in your OOM, then the List of Interfaces and Interface book will not be listed. Select *Show All Items* at the top of the pane to display all the report items available for this kind of model. By default, in the Multi-Model Report Editor, all available report items are displayed if the associated model is not open in the workspace.

The following tools are available from the Report toolbar:

Tool	Description
	Report Wizard - Opens the Report Wizard (see Creating a Report with the Report Wizard [page 296]).
	Print Preview - Displays a report print preview (see Previewing Your Report (Legacy Reports) [page 353])
	Print - Prints the report.
	Generate RTF - Generates the report as an RTF file.
	Generate HTML - Generates the report as an HTML file.

Tool	Description
	Add Item - Adds an item to the Report Items pane.
	Up One Level - Moves item up one level.
	Down One Level - Moves item down one level.
	Raise Level - Moves item at the same level as the book item (item that contains other items) that precedes.
	Lower Level - Moves item within the book item that follows.

1.8.5.1 Adding Items to a Report (Legacy Reports)

The Available Items pane of the Report Editor lists the items that you can add to your report.

You can add an item to the Report Items pane in any of the following ways:

- Double-click the item in the Available Items pane to place it below the currently selected item in the Report Items pane.
- Drag the item from the Available Items pane and drop it at the desired position in the Report Items pane. Use the `Ctrl` or `Shift` key to select multiple items. If the pointer becomes a barred circle, that means the drop target is unauthorized. Note that you can drag and drop an item only when the Report Items pane already contains at least one item.
- Right-click the item in the Available Items pane and select Add from the contextual menu

The following kinds of report items are available:

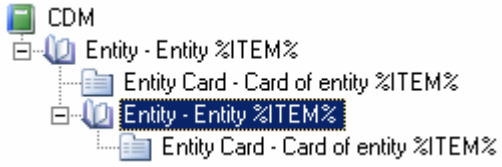
Icon	Description
	Table of contents – Inserts a table of contents. PowerDesigner automatically inserts a following page break.
	Title - Free text to introduce what is next.

Icon	Description
	Paragraph – the following types are available, depending on the context: • Text paragraph – free text that can be inserted anywhere • Description – contents of the *Description* sub-tab of the *Definition* tab of the model or object property sheet • Annotation – contents of the *Annotation* sub-tab of the *Description* tab of the model or object property sheet • Other – text paragraphs are available as necessary to display code preview, scripts, validation rules, etc
	Text file – Inserts the contents of a text file, selected via a file chooser.
	Graphical file - Inserts the contents of a graphics file, selected via a file chooser.
	Page break – Inserts a page break. Right-click the inserted item and deselect *Generate in HTML* to disable it when generating HTML reports.
	Card – Inserts a table listing the properties of the model or package. Right-click the item after insertion and select Layout to control the properties listed (see Controlling the Layout of Card Report Items (Legacy Reports) [page 350])
	List - Inserts a table listing the objects of a given type belonging to the model or another object. Right-click the item after insertion and select Layout to control the properties listed (see Controlling the Layout of List Report Items (Legacy Reports) [page 348])
	Graphics – Inserts an image of a diagram.
	Book – Inserts a set of sub-items to report on a particular type of object. May contain cards, lists, text items, graphics, and other books as necessary. When you add a book item to a report, you automatically add its dependent items. You can delete any dependent item that you do not need.

When you add an item to the Report Items pane, the item remains in the Available Items pane. You can insert the same item several times in the same report contents.

Notes:

- Composite objects - BPM processes can contain other processes, OOM activities can contain other activities, and CDM entities can contain other entities. Such composite object hierarchies are not included by default in the report, but you can display them by adding a second instance of the appropriate report item inside the first. In the following example, the second Entity book is added inside the first Entity book to display child entities inside their parent entities:



- Package and composite object hierarchy - When you add a package or composite object book item to a report, you can select the Hierarchical Display command in the item contextual menu to display the package or composite object hierarchy in the generated report. You do not need to use this command when you generate for an HTML report, as HTML reports always provide a hierarchical display.
- Drag and drop between reports - You can drag and drop items between reports of the same type. If you simply drag and drop an item from one report to another without pressing any key, you move the item from one report to another. You can copy an item by pressing the `Ctrl` key while dragging and dropping it. The report in which you want to drop the item must already contain at least one item.

> **i** Note
>
> When you generate an HTML report that contains a diagram graphic and the object cards of the symbols, hyperlinks are created between the diagram symbols and the corresponding object cards. You can click a symbol in the diagram to access the object card that corresponds to it in the HTML page.

1.8.5.2 Modifying Items in the Report Items Pane (Legacy Reports)

You can move items in the Report Items pane by dragging and dropping them.

To copy an item, hold down the `Ctrl` key while dragging it.You can copy any item within the Report Items pane except for the root folder. Any formatting applied to the copied item is preserved. When you copy a book item, you also copy its dependent items.

Contextual Menu

The following actions are available by right-clicking items that you have added to the report in the Report Items pane:

Command	Description
Show Title	Controls the display of the item title in generated reports.
Show Book Title in HTML TOC	[book items] Controls the display of the book item title in the table of contents of HTML reports.
Show Contents in HTML TOC	[book items] Controls the display of the sub-item titles in the table of contents of HTML reports.

Command	Description
Up	Moves the item up one line in the tree.
Down	Moves the item down one line in the tree.
Raise Level	Moves the item up one level in the tree.
Lower Level	Moves the item down one level in the tree.
Format	Opens the dialog allowing you to modify the formatting of the item (see Formatting Textual Report Items (Legacy Reports) [page 343] and Formatting Graphical Report Items (Legacy Reports) [page 347]).
Layout	[list and card items] Opens a dialog allowing you to modify the layout of the item (see Controlling the Layout of List Report Items (Legacy Reports) [page 348] and Controlling the Layout of Card Report Items (Legacy Reports) [page 350]).
Collection	[list and book items] Opens a dialog allowing you to define the collection to which the report item will be applied (see Modifying the Collection of an Item (Legacy Reports) [page 340]).
Selection	[list and book items] Opens a dialog allowing you to refine the selection of objects to which the report item will be applied (see Selecting Objects to Display for a Report Item [page 329]).
Edit Title	Opens the item in a dialog allowing you to edit its title (see Modifying the Title of an Item (Legacy Reports) [page 339]).
Delete	Removes the item from the Report Items pane. You can delete any item except the root.
Quick View	Generates a preview of the item (see Previewing Your Report (Legacy Reports) [page 353]).

1.8.5.3 Modifying the Title of an Item (Legacy Reports)

Package, object, and object-dependent items all have default titles, and you can insert Title items anywhere in the Report Items pane and as often as you want. You can edit the default text of these report items.

Procedure

1. Right-click an item in the Report Items pane and select Edit Title to open it in the Editor dialog box.
2. Enter any appropriate changes to the title text. The user-defined check box will be automatically selected. To revert to the default value, clear this check box.
3. [optional] Clear the Show Title check box if you do not want to show the title of the item in the previewed or generated report.
4. [optional] Insert GTL variables that represent the name of the object, its parent, or model, etc at the cursor, by selecting them with the Insert button.

5. [optional] Click the Format button to open the Format dialog box, and select the appropriate formatting.
6. Click OK to return to the Report Editor. The first line of the title is displayed next to the item in the Report Items pane.

> **i** Note
>
> You can edit the default titles of all report items in the report language resource file that is attached to your report. For more information, see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files* .

1.8.5.4 Modifying the Collection of an Item (Legacy Reports)

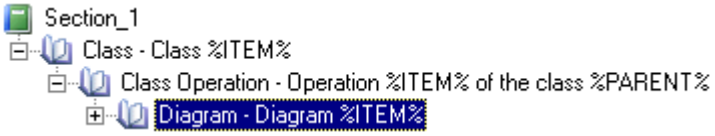
You can extend the PowerDesigner metamodel to, for example, add new properties to an existing object (extended attributes), to create new objects (extended objects), and to create new connections between object types (extended and calculated collections and extended compositions). These latter collection extensions are not included, by default, in the object book report items in the Available Items pane of the Report Editor and must be added by hand.

Context

> **i** Note
>
> If you create extended or calculated collections in your model and use the Report Wizard (see Creating a Report with the Report Wizard [page 296]) to create your report, the generated report automatically creates a list for each type of extended and calculated collection inside the book item for each affected metaclass.

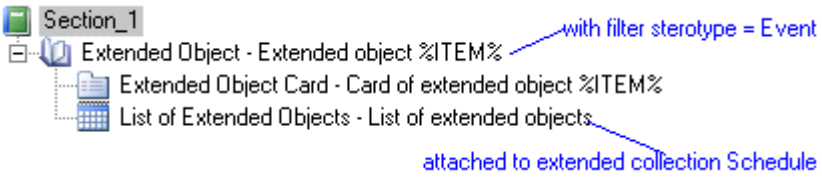
You can add any object list or book inside any other object book to allow you to represent any extended or calculated collection. Since there is no control over the location where you drop report items, and you are responsible for the global consistency of items and collections in your report. Once you have dropped the appropriate item, you can modify its collection and select a calculated or extended collection.

In the following example, a calculated collection is created on the operations metaclass that retrieves the diagrams where messages using a given operation are displayed. To display these diagrams in the report, a diagram item must be inserted inside the operation book item and the appropriate calculated collection selected for its collection:



In the following example, in order to manage the "schedule" and "event" concepts, the extended object <<event>> has been created, along with an extended collection of extended objects <<schedule>>. The

Extended Object report item is added to the report and filtered on the stereotype Event (see Selecting Objects to Display for a Report Item [page 329]) and the List of Extended Objects is added to the Extended Object book, with its collection set to Schedule:



Procedure

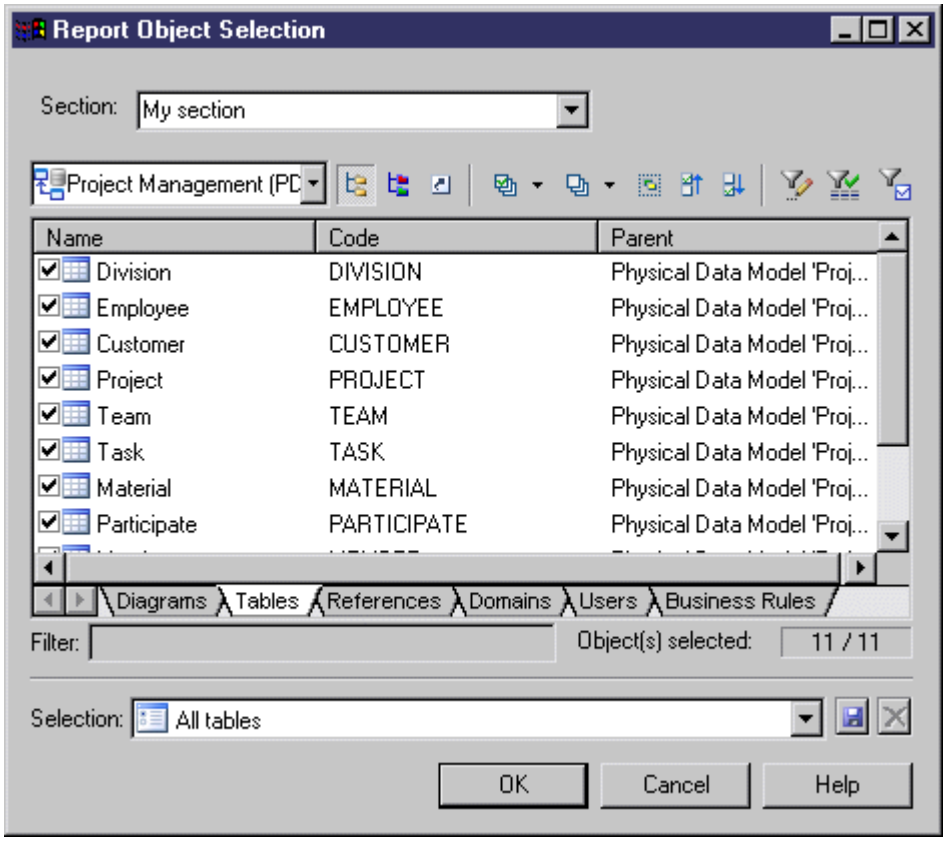
1. Drag and drop the appropriate book or list item under the selected book item in the Report Items pane.
2. Right-click the book or the list item and select *Collection* to open its Collection dialog box.
3. Select a collection in the Collection list and click *OK* to return to the Report Editor.

1.8.5.5 Selecting Objects to Display in the Report

By default, the report includes all the objects in your model. You can restrict the objects included by making a selection, which is saved with the report.

Procedure

1. Select *Report* > *Select Objects* from the Report Editor to open the Report Object Selection dialog box. All the objects (except any external shortcuts) are selected by default.



2. Select a report section from the Section list (see Report Properties [page 358]).
3. [optional] Select a package from the package list.
4. Select the objects that you want to include in the report from each of the object-specific sub-tabs.
5. [optional] Specify a name for your selection in the Selection list and click the Save tool beside the list. The selection is saved as part of the model file (or in the .mmr file for a multi-model report).
6. Click OK to confirm your selection and return to the Report Editor.

Results

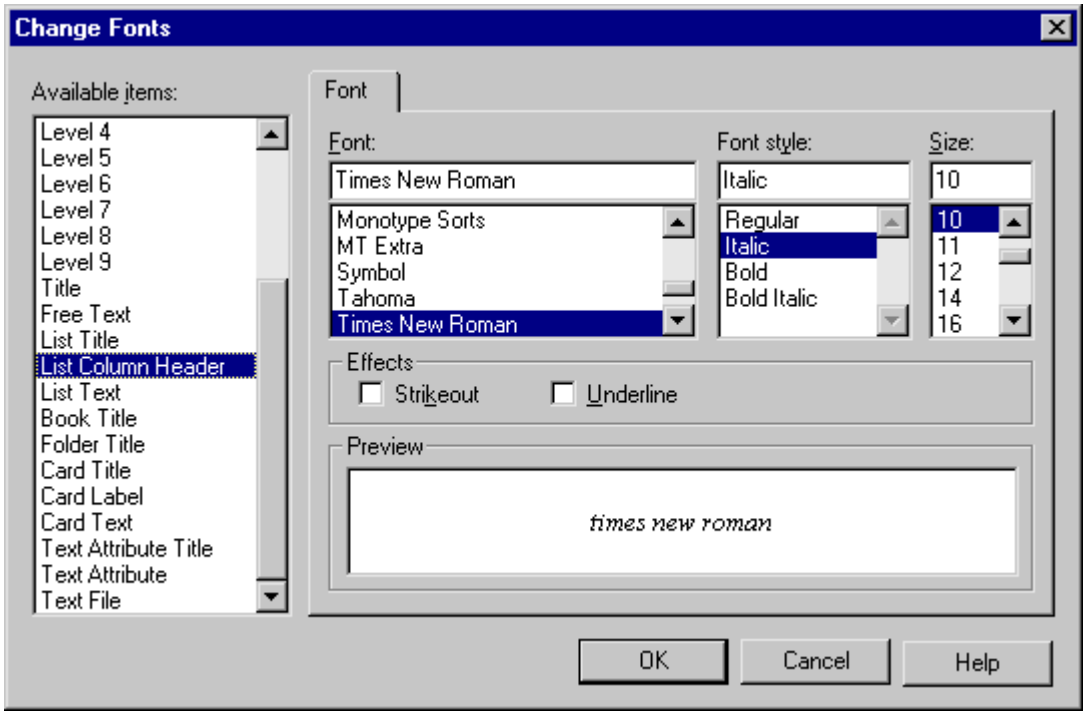
For information about the tools in this dialog, see Adding an Item from a Selection List [page 129].

1.8.5.6 Specifying Default Fonts for Report Items (Legacy Reports)

You can specify default fonts for report items, report templates and reports. The default fonts are saved in the registry.

Procedure

1. Select *Report* > *Change Fonts* from the Report Editor window to open the Change Fonts Properties dialog box:



2. Select one or more available items in the left-hand list, and specify a font, style, size, and effects as appropriate.
3. Click OK to return to the Report Editor.

1.8.5.7 Formatting Textual Report Items (Legacy Reports)

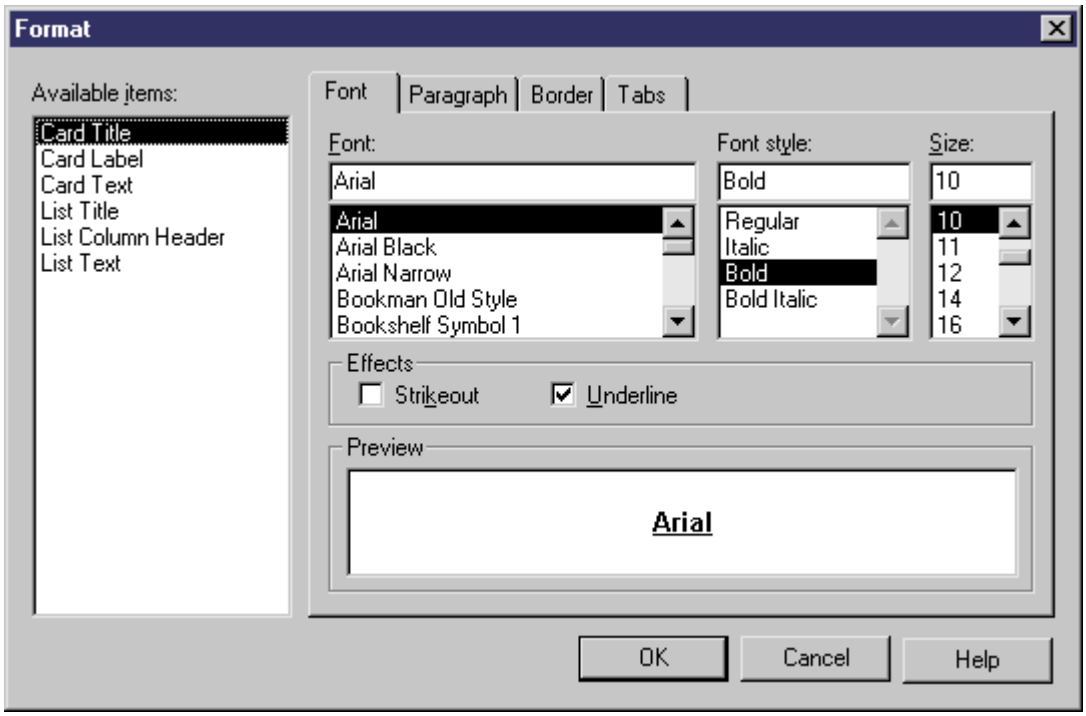
The *Format* dialog allows you to specify formatting for the selected report items.

You can access this dialog in any of the following ways:

- Select one or more text or list report items in the left-hand *Available Items* pane, and select *Report* > *Format* in order to set default formatting for them.

Each instance of the item that you subsequently add to the right-hand *Report Items* pane will have this same formatting, but your changes will not affect any report items already in the *Report Items* pane.

- Select one or more text or list report items in the right-hand *Report Items* pane, and select *Report* > *Format* to change their formatting.



Available Items

Some report items contain more than one text element. You can format each separately or use the *Ctrl* or *Shift* keys to select several at once:

Report Item	Text Selections
Card items	The following selections are available for card items (lists of properties), which are output in table format: • Title - Title of the item, eg "Card of the Table" • Label - Property name on a list of properties, eg "Table Code" • Text – Value of a property, eg "EMPLOYEE"

Report Item	Text Selections
CRUD Matrix items	The following selections are available for CRUD Matrix items, which are output in table format: • Matrix title – Title of the item, eg "CRUD Matrix " • Matrix row and column – Resource and process name, eg " Check item (process name in rows)" and " Stock (resource name in columns)" • Matrix text – CRUD value, eg " RU "
Model- and object-dependent items	The following selections are available for model- and object-dependent items, which are output in text format: • Title – Title of the item, eg " Table Description " • Text – Text of the description, annotation, or script, eg "This table describes different employee characteristics "
List items	The following selections are available for list items, which are output in table format: • Title – Title of the item, eg " List of Indexes " • Column Header – Heading of column, eg " Code " • Text – Value in the column, eg " PK_EMPLOYEE "
Title page	The following selections are available for the title page (see Report Properties [page 358]), which is output in text format: • Title – Title of the report, eg "OOM Report" • Optional Fields – Text of the author, date, version or summary, eg " VB, 09.15.99, draft, report with lists only "
Table of contents	The following selections are available for the table of contents, which is output in text format: • TOC title – Title of the table of contents, eg " Table of contents " • Level <x>– Text of the different hierarchy levels in the table of contents, eg "2.1 Model information "

> **i** Note
>
> Select the levels to display from the *Show levels* list beneath the Available items pane of the *Format* dialog. To define the levels to be shown in the TOC of an HTML report, see Report Properties [page 358].

Font Tab

The Format dialog *Font* tab allows you to specify a font, style, size, and effects for the text elements that you select in the left-hand *Available items* list.

Paragraph Tab

The Format dialog *Paragraph* tab allows you to specify indentation, spacing, and alignment for the text elements that you select in the left-hand Available items list.

Border Tab

The Format dialog *Border* tab allows you to specify elements of a frame around the text elements that you select in the left-hand *Available items* list. The following border formatting is available:

Option	Description
Left / Right / Top / Bottom	Inserts a line to the left, right, top, or bottom of all the occurrences of the selected text elements.
Box	[card and list items only] Inserts a rectangle around all the occurrences of the selected text elements. For card items, inserts a rectangle that groups all the occurrences of Text Label (property names) including Text (corresponding property values). For list items, inserts a rectangle that groups all the occurrences of Column Header (heading of column, example Code), and also a rectangle that groups all the occurrences of Text (value in the column), if you have previously highlighted each of these text selections in the *Available Items* pane.
Width	Specifies the line width for the Left, Right, Top and Bottom frame options
Box width	Specifies the line width of the box frame
From text	Specifies the amount of space from text to top, bottom, left, and right border

Tab Tab

The Format dialog *Tab* tab allows you to specify tab positions and alignment for the text elements that you select in the left-hand *Available items* list, which is particularly useful for the layout of headers and footers (see Modifying Report Section Headers and Footers (Legacy Reports) [page 352]).

The following tab formatting is available:

Option	Description
Tab stop position	Specifies the measurement for a tab stop
Default tab stops	Specifies the default spacing between tab stops

Option	Description
Alignment	Specifies how text is to be aligned at the tab stop. To change the alignment for an existing tab stop, click it in the Tab Stop Position box, and then click the new alignment option

1.8.5.8 Editing Textual Report Items (Legacy Reports)

You can edit the text of a text paragraph or a text file in a text editor by double-clicking it or right-clicking it and selecting *Edit*.

Context

For text paragraphs, you can select the following kinds of content:

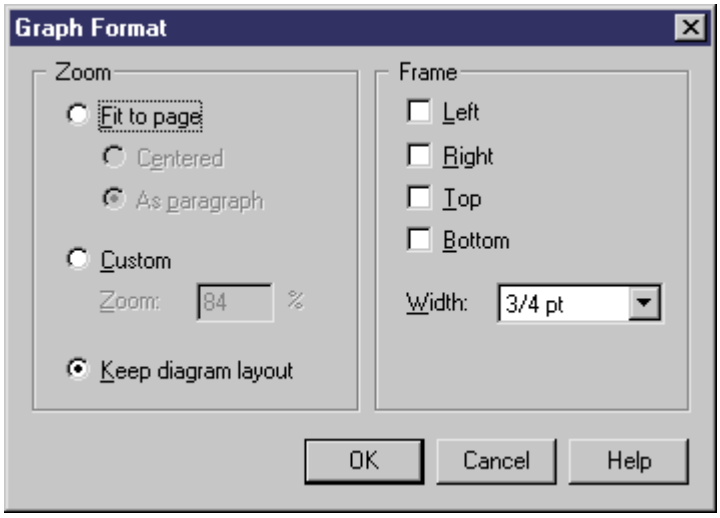
- User-defined - Enter text directly into the box. Click the Format button to apply formatting to selected text.
- Object attribute - Select an attribute from the list.

1.8.5.9 Formatting Graphical Report Items (Legacy Reports)

You can access the report items Graph Format dialog in any of the following ways:

Context

- Select one or more graphical report items in the left-hand Available Items pane, and select *Report* > *Format* in order to set default formatting for them.
  Each instance of the item that you subsequently add to the right-hand Report Items pane will have this same formatting, but your changes will not affect any report items already in the Report Items pane.
- Select one or more graphical report items in the right-hand Report Items pane, and select *Report* > *Format* to change their formatting.



The following options are available in the Graph Format dialog:

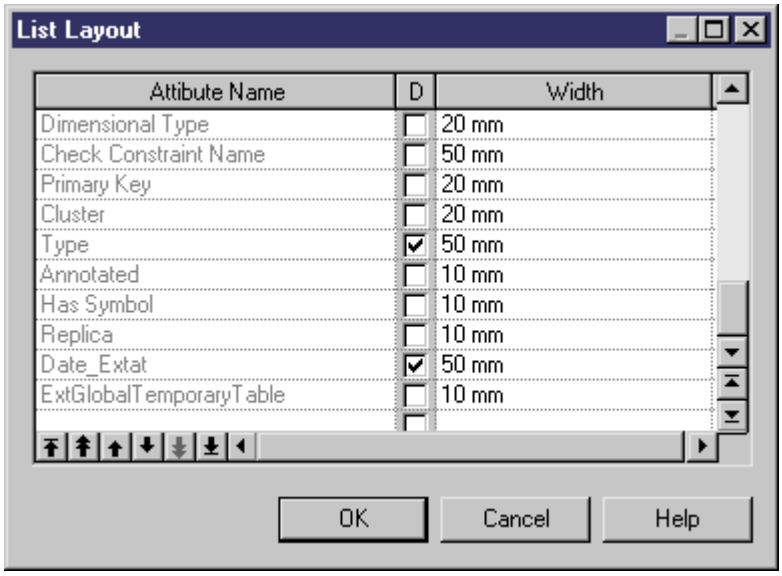
Option	Description
Zoom	Specifies the resizing of the graphic in the report. You can choose between: • Fit to page • Custom zoom • Keep diagram layout - one page of the diagram per report page
Frame	Specifies the borders to insert around the graphic.

1.8.5.10 Controlling the Layout of List Report Items (Legacy Reports)

List report items are published as tables, ordered by the name or code of the objects. You can specify which attributes will be displayed, in which order, and the width of their columns.

Procedure

1. Select one or more list report items in the *Report Items* pane and select *Report* > *Layout* to open the List Layout dialog, which lists all the attributes available for the object:



2. Enter any appropriate layout changes as follows:
   - Use the arrow buttons at the bottom of the dialog to reorder columns in the table.
   - Select the *D* column next to a property to display it in the table.
   - Enter a width in the width column. You can specify the width in millimeters (mm), inches (in) or as a percentage of the table width.
3. Click *OK* to return to the Report Editor.

Results

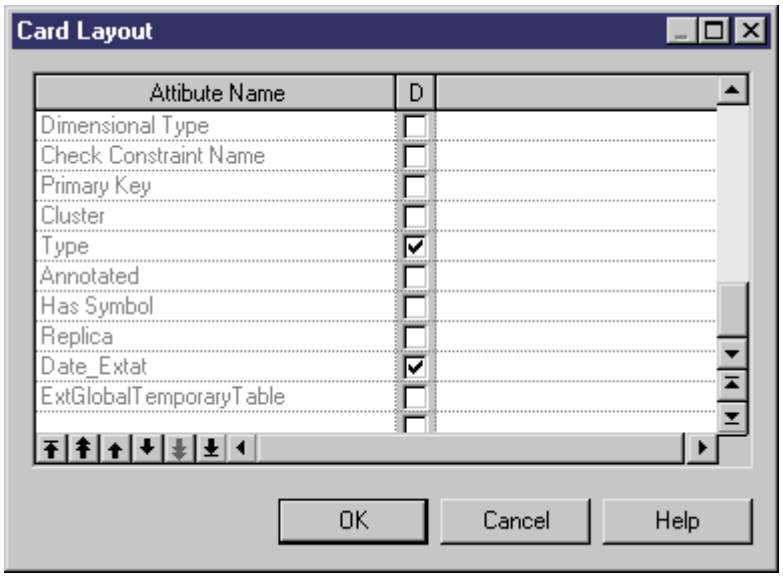
> **Note**
>
> If extended attributes are specified for the object, they will be available for selection in the List Layout dialog. You can choose to select all extended attributes or only those whose value has been modified using the *Display* > *All* and *Display* > *Only modified ones* commands from the list item contextual menu.

1.8.5.11 Controlling the Layout of Card Report Items (Legacy Reports)

Card report items represent the property sheets of individual objects and are published as two-column tables. You can specify which attributes will be displayed, and in which order.

Procedure

1. Select one or more card report items in the *Report Items* pane and select *Report* > *Layout* to open the Card Layout dialog, which lists all the attributes available for the object:



2. Enter any appropriate layout changes as follows:
   - Use the arrow buttons at the bottom of the dialog to reorder properties in the table.
   - Select the *D* column next to a property to display it in the table.
3. Click *OK* to return to the Report Editor.

Results

> **i** Note
>
> Select the *Displayed* check box for the *Class Name* attribute to distinguish shortcuts cards from the other object types cards in the generated report.

1.8.5.12 Suppressing the Titles of Report Items (Legacy Reports)

You can suppress the title of any item in your report. The following illustration shows a portion of a report, with the title of the process card displayed:

Context

1 Process Process_1

1.1 Card of process Process_1

Name	Process_1
Code	Process_1
Comment	

This illustration shows the same report with the title of the process card suppressed:

1 Process Process_1

Name	Process_1
Code	Process_1
Comment	

Procedure

Right-click an item in the *Report Items* pane and deselect the *Show Title* command.

or

Double-click an item to open the Editor dialog box, clear the *Show Title* check box, then click *OK*.

1.8.5.13 Modifying Report Section Headers and Footers (Legacy Reports)

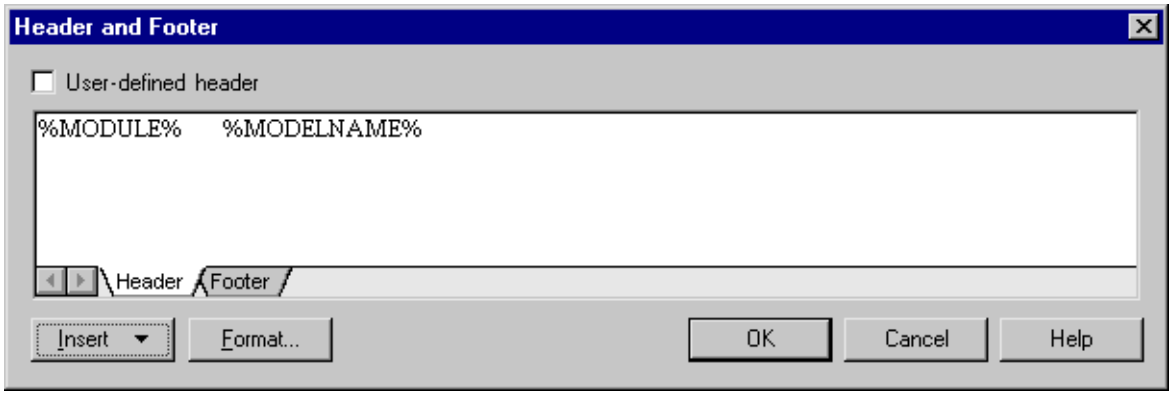
The Report includes a default header and footer that you can modify.

Context

For more information about the default header and footer, including specifying separate headers and footers for each section of your RTF report, see Report Properties [page 358].

Procedure

1. Right-click the root section item in the *Report Items* pane and select *Header/Footer* (or click the *Header & Footer* button in the *Presentation Options* page of the Report Creation Wizard) to open the Header and Footer dialog:



2. Enter any appropriate changes to the header text. Edit the Footer text by clicking on the *Footer* sub-tab. You can use any of the variables available to the RTF template (see RTF Presentation Templates [page 364]).

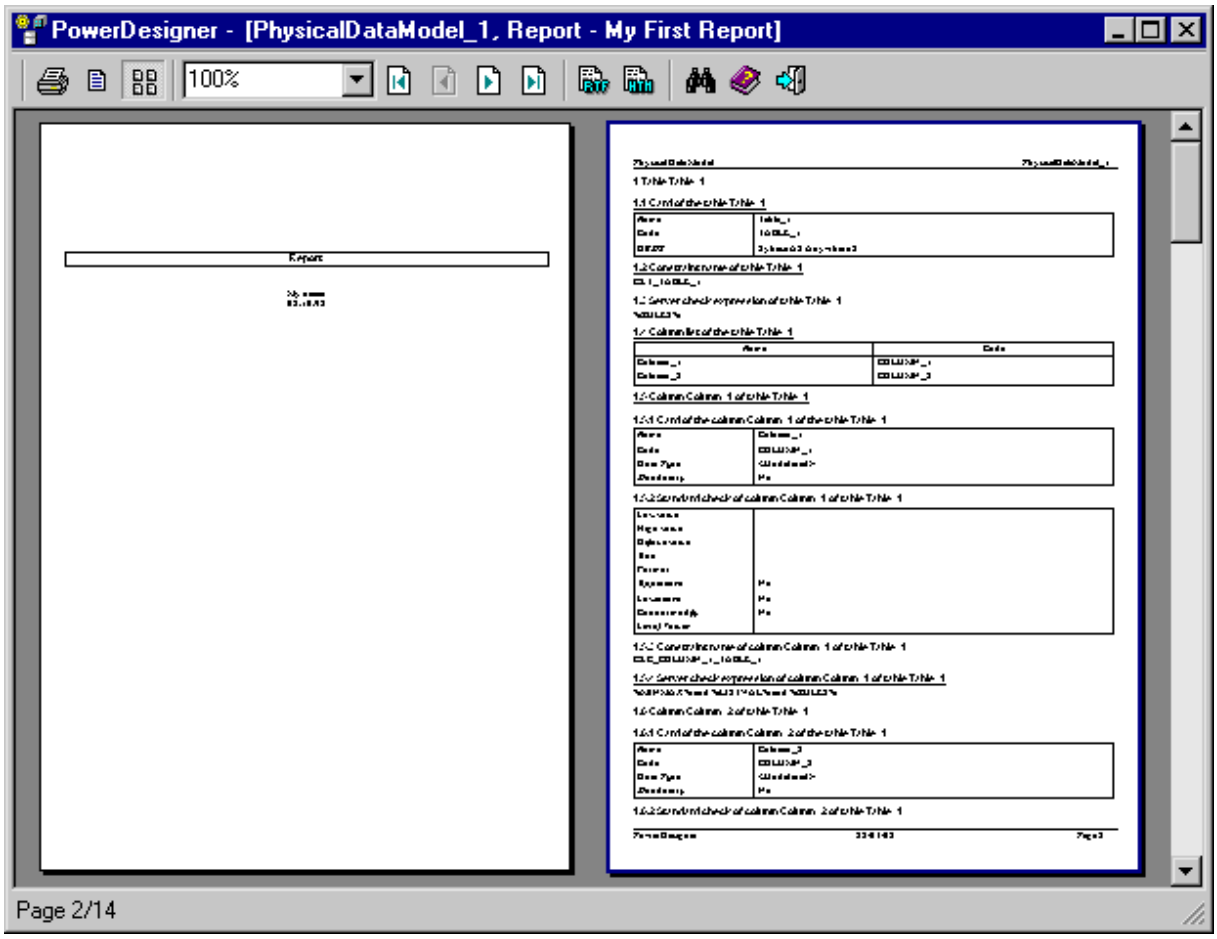
   If you change the default header or footer text, then the *User-Defined* check box is selected, to indicate that you have overridden the default values in the associated report language resource file. To revert to the default value, clear the check box. For more information about report language resource files, see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files* .

   > **i Note**
   >
   > Use the key combination *Ctrl+Tab* to insert tabulations in a header or open the *Tabs* tab in the Format dialog box (see Formatting Textual Report Items (Legacy Reports) [page 343]) to set tab stop values.

3. Click *OK* to return to your report.

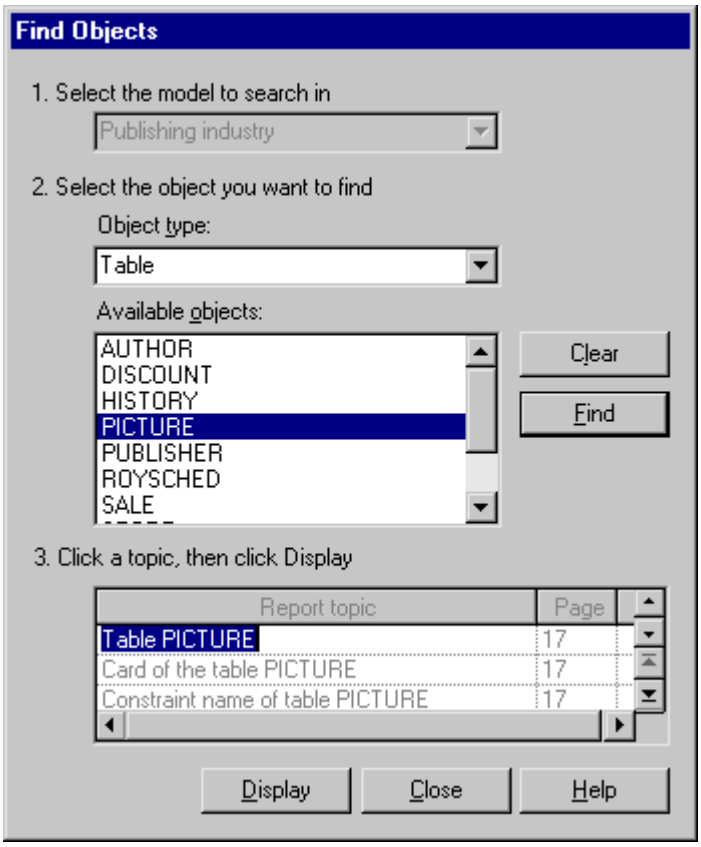
1.8.5.14 Previewing Your Report (Legacy Reports)

You can preview your report by selecting *Report* > *Print Preview* or by clicking the *Preview* tool in the End page of the Report Creation Wizard. You can preview a report item by right-clicking it in the Report Items pane and selecting *Quick View*.



The following tools are available in this window:

Tool	Goes to preview
	Print the report.
	Preview the report one or two pages at a time.
	Go to the first, previous, next, or last page of the report.
	Generate the report in HTML or RTF format.

Tool	Goes to preview
	Open the Find Objects dialog. Specify the model (for multi-model reports), object type, and object you want to find, and then click *Find* to display a list of topics that mention the object. Then select a topic in the list and click *Display* to go to that topic in the report preview: 
	Help
	Close the preview.

1.8.5.15 Upgrading Legacy Reports for Use with the Graphical Editor

You can upgrade a legacy report to allow it to be edited in the new graphical Report Editor at any time from the Browser or the List of Reports. You can upgrade a legacy report template from the List of Report Templates.

Context

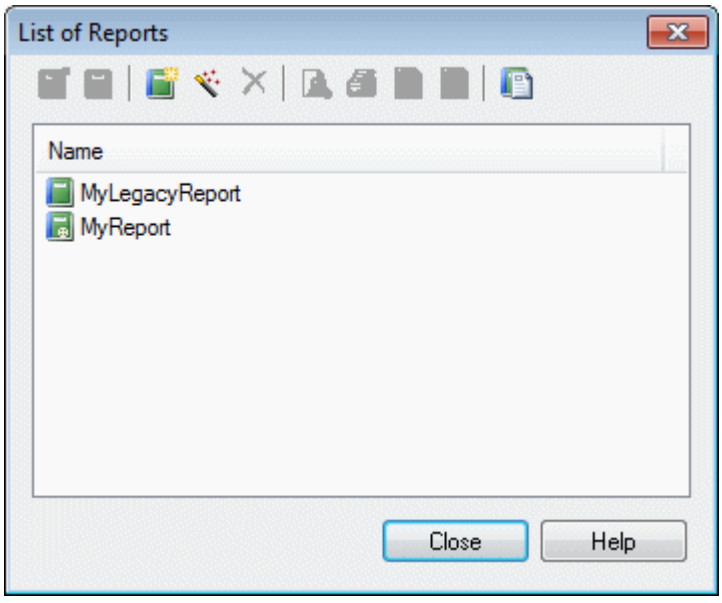
> **i Note**
>
> Upgrading a legacy report cannot be undone. You may want to copy your legacy report (see Moving and Copying Objects [page 103]) before upgrading it, to use it for reviewing the upgraded version

To upgrade a report from the Browser, right-click it and select *Upgrade to New-Style Report*.

To upgrade a report from the List of Reports, select *Report* > *Reports* to open the list, select the report to upgrade, and then click the *Upgrade to New-Style Report* tool.

To upgrade a report template from the List of Report Templates, select *Report* > *Report Templates* to open the list, select the template to upgrade, and then click the *Upgrade to New-Style Report* tool.

In both the Browser and list, new-style reports display a small overlay to identify them:

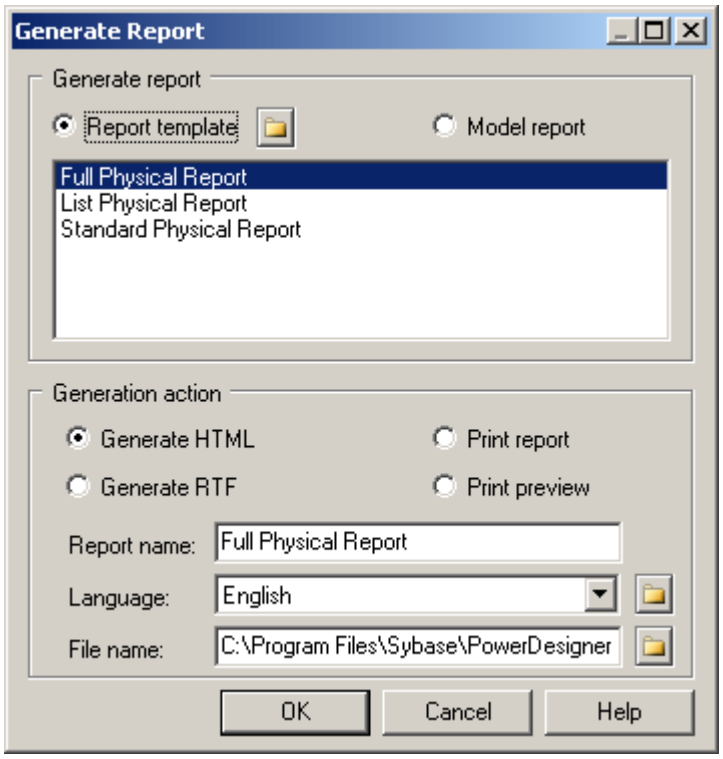


1.8.6 Generating a Model Report

You can generate your report directly from the Report Editor by clicking the *Generate Web Report* or *Generate Rich Text Document* tools in the toolbar or selecting the corresponding commands from the *Report* menu. You can generate any report in the model or a generic report from a template from the *Generate Report* dialog.

Procedure

1. Select *Report* > *Generate Report* to open the Generate Report dialog.



2. Select whether to generate your report from a:
   - *Report template* – Select a standard template or one of your own (see Report Templates [page 357]). Click the *Browse* button to change the folder where your templates are stored.
   - *Model report* – Select any report defined in the model.

> **i Note**
>
> To generate a report directly from its Browser entry, right-click it, and select *Print*, *Generate* > *RTF*, or *Generate* > *HTML*.

3. Select how you want to generate your report:
   - *Generate HTML* – Generates a set of files to display your report in a Web browser.
   - *Generate RTF* - Generates a Rich Text Format file for opening in Microsoft Word or another RTF editor.
   - *Print report* – Prints the report directly.
   - *Print preview* – Generates a preview of the report (see Previewing a Report or Report Item [page 333].
4. [for HTML or RTF only] Specify a *Report name* and *File name* for the generated files.
5. Specify the report *Language* to use for generating item titles and other report artifacts.
6. Click *OK* to begin the generation.

1.8.7 Report Templates

A report template specifies content and formatting for a report (or, for a multimodel report, a section). PowerDesigner provides a set of standard report templates, and you can create your own. You can create a report template from scratch, or based on an existing template, from the List of Report Templates.

Procedure

1. Select *Tools* > *Resources* > *Report Templates* to open the List of Report Templates.
2. Click the *New* tool and enter a name for your template.
3. Select a language and model type for the template, and then click *OK* to open the empty template in the Report Template Editor.
4. Edit the report template as necessary (see The Report Editor [page 312]).
5. Select *File* > *Save* to save the template and make it available for selection when you create a report.
6. [optional] Check the report template into the repository `Library/Resources/Report Templates` folder to make it available to other users (see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

1.8.7.1 Saving a Report or Report Section as a Report Template

You can save your report (or a section in a multimodel report) as a report template for use with other reports.

Procedure

1. In the Outline or Design view, click in the section you want to save as a template.

> **Note**
>
> For legacy reports, click the tab at the bottom of the Report items pane to select the section you want to save.

2. Select *Report* > *Create Template From Section*.
3. Select *File* > *Save*, enter a name for the template, and click *OK* to save it for future use.
4. [optional] Check the report template into the repository `Library/Resources/Report Templates` folder to make it available to other users (see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

1.8.7.2 Modifying and Saving a Template

You can modify a report template from the List of Report Templates.

Procedure

1. Select *Report* > *Report Templates* to open the List of Report Templates

> **Note**
>
> You can open the List of Report Templates from the List of Reports by clicking the *Manage Report Templates* tool.

2. Select a template type to display the list of available templates of that type.
3. [optional] Click the *Path* tool to select a different directory in which to search for templates.
4. Select a template and click the *Properties* tool to open it in the Report Editor.
5. Make any appropriate changes to the template and then click the X at the top right of the Design view to close it and be prompted to save your changes.

1.8.8 Report Properties

To open the report property sheet, select *Report* > *Report Properties* from the Report Editor window or right-click the report in the Browser and select *Properties*.

The report property sheet *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Language	Specifies the language of the report. Click the tools to the right of this field to edit the selected report language resource file or to change the path that populates the list. For more information see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files* .
No paragraph numbering	Suppresses section numbers in RTF and printed reports. Paragraph numbering is not used in HTML reports.
Generate empty paragraphs	Generates titles for all book items, even if they have no content.

Title Page Tab (Legacy Reports)

The report property sheet Title Page tab allows you to specify the contents of a title page for your report:

Property	Description
No title page/ Include title page	Select whether to include a title page. Selecting *Include title page* enables the other fields on this tab.
Title	Specifies the title of the report. Default: The name of the report.
Author	Specifies the report author. Default: User shown on the *Version Info* tab.
Date	Specifies the date of the report. Default: Today's date.
Version	Specifies the version number of the report. Default: Version in the model property sheet.
Summary	Specifies any additional text to be printed on the title page.

Click the *Format* button to customize the formatting of these fields.

> **i** Note
>
> For HTML reports, a home page is generated instead of a title page. Title pages are never generated for RTF reports generated with a report template.

RTF Format Tab

PowerDesigner provides a set of predefined RTF presentation templates, style sheets, and header and footer files for generating RTF reports, which are located in the `Resource Files\RTF Report Templates` directory. You can use these files as a basis to create your own templates.

The report property sheet RTF Format tab allows you to specify an RTF presentation template for your report, and contains the following properties:

Property	Description
Template	Specifies the RTF presentation template to be used to style the report. Click the tools to the right of this field to edit the selected template (see Referencing Object and Model Properties [page 333]) or browse to an alternate template directory.
Use sections header and footer	Specifies to use the header and footer defined in each report section instead of those defined in the RTF presentation template (see Modifying Report Section Headers and Footers [page 332]. If you want to preserve the header and footer of the title page and of the table of contents page, you should add a section break after the table of contents. This option is recommended when formatting a multi-model report, as otherwise all the pages of the report will have the same header and footer.

> **i** Note
>
> Click the *Set As Default* button to specify your current settings as the default for RTF reports. To revert to the previously saved values, click the *Default* button.

HTML Format Tab

PowerDesigner provides a set of predefined HTML presentation templates, style sheets and header and footer files for generating HTML reports, which are located in the `Resource Files\HTML Report Style Sheets` directory. You can use these files as a basis to create your own templates.

The report property sheet *HTML Format* tab allows you to specify HTML presentation options for your report, and contains the following properties:

Property	Description
Template	Specifies the HTML presentation template to be used to style the report. Click the tools to the right of this field to preview, save, or delete the selected template, or to browse to an alternate template directory.
Style sheet	Specifies the style sheet to be used to style the report as an absolute or relative path or a URL. Click the tools to the right of this field to edit the selected stylesheet (see Referencing Object and Model Properties [page 333]) or browse to an alternate style sheet.

Property	Description
Header / Footer	Specifies the HTML file to be used as the report header or footer as an absolute or relative path or a URL. Click the tools to the right of the field to edit the selected file (see Referencing Object and Model Properties [page 333]) or browse to an alternate file. You can also specify the height, in pixels, of the header and footer.
Home page	Specifies the HTML file to be used as the report homepage as an absolute or relative path or a URL. Click the tools to the right of this field to edit the selected file (see Referencing Object and Model Properties [page 333]) or browse to an alternate file.
Table of contents	These options control the format of the table of contents. You can specify: • *Level* - the maximum number of levels to display in the HTML table of contents. Default: 4. • *Width* - the width of the frame as a percentage of the total width of the report window. Default: 25%. • *TOC at right* - whether the TOC will be generated on the left (default) or right side of the screen. You can reduce the size of the HTML TOC by right-clicking book report items and selecting: • *Show contents in HTML TOC* - deselect this option to display only the title of the item without its contents in the TOC. • *Show book title in HTML TOC* - deselect this option to remove the title of the book item from the table of contents. Automatically deselects the Show contents in HTML TOC option.
Image Format	Specifies the file format for graphic files to be generated and embedded in the report: • SVG - XML-based language for Web graphics, which allows you to interact with the graphics. Note that SVG is not supported for graphics embedded in RTF in PowerDesigner property fields, and such graphics will be generated as PNG. • PNG – higher quality than JPEG. • JPEG – smaller file size than PNG.
List format	Specifies the format for lists. You can specify the number of rows you want to display per page in the list results and the number of links to additional results displayed below the list. Additional results are numbers with hyperlinks to other pages containing the next rows of the list.
Web browser	[legacy reports] Specifies the primary type of browser that will be used to view the HTML report in order to correctly generate SVG graphics. You can specify HTML5-compliant browsers or Internet Explorer 6,7, or 8 (which support SVG via the Adobe SVG plug-in).
Add page break for top level list and cards	[legacy reports] Inserts a page break before each top level list and card to avoid long HTML pages.

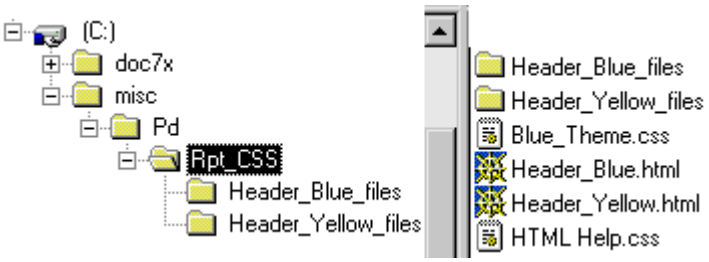
> **i Note**
>
> Click the *Set As Default* button to specify your current settings as the default for HTML reports. To revert to the previously saved values, click the *Default* button.

Depending on the type of location (local or UNC paths, or URL), style sheet files, header and footer files and home page files are used as follows:

Location	Use
Local path or UNC path	The file is copied into the report generation folder.
URL	The file is referenced in its location.

We recommend that you gather your style sheets and header and footer files in a subfolder with the same name as the presentation template file with which they are associated to ensure that the structure of links between them is simplified.

For example, the *Header_Blue.html* header uses files that are gathered in the *Header_Blue_files* folder:



Sections Tab (Legacy Reports)

You can use sections to structure your report. Reports must contain at least one section, and sections are displayed as tabs at the bottom of the Report Items pane of the Report Editor. Each section can only report on one type of model so, in a multi-model report, each different type of model (PDM, OOM, BPM, etc) must have its own section.

The report property sheet Sections tab allows you to create, configure, and delete report sections, and contains the following properties:

Property	Description
Name	Specifies the name of the section, which will be displayed on the sub-tab at the bottom of the Report Editor window. You can also change the name of a section by right-clicking the section node in the Report Editor and selecting *Rename*.
Model	[multi-model reports only] Specifies the model on which the section will report.

Property	Description
Template	Specifies the template upon which the section is based. You can choose between: • None [default] – creates an empty section. • Full `<Model Type>` Report – provides lists of all the types of objects in the model, together with detailed information on each object. • List `<Model Type>` Report – provides lists of all the types of objects in the model. • Standard `<Model Type>` Report – provides lists of all the types of objects in the model, together with detailed information on each of the main objects.
Apply Template	Re-applies the specified template to the section, deleting any modifications you may have made. **i Note** You cannot undo this action.

i Note

You can launch the Report Wizard (see Creating a Report with the Report Wizard [page 296]) from the Report Editor to configure the present section. The wizard configures only one section at a time, and will delete the current contents of the section. To configure a second section, select it using the section tabs at the bottom of the window, and relaunch the wizard.

1.8.8.1 Valid Codepage

By default, PowerDesigner generates in the HTML page a character set (charset) built from the current language of the report. If this is not correct for your needs, you can change the charset.

Procedure

1. Right-click the report in the Browser, and select *Properties* from the contextual menu to open its property sheet.
2. On the *General* tab, click the *Properties* button to the right of the *Language* field to open the Report Language resource file in the Resource Editor.
3. Expand the *Report Titles* > *Common Objects* category, and set the appropriate values for the *CharSet* and *CodePage* items.

Results

For information about working with charsets and coding pages, see: http://www.w3.org/International/tutorials/tutorial-char-enc/ .

> **Note**
>
> You can translate the navigation buttons (Previous, Next, Home) generated by default in your HTML report using HtmlNext, HtmlPrevious and HtmlHome items in the Report Titles\Common Objects category of the Report Language Editor. For more information, see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files* .

1.8.8.2 RTF Presentation Templates

An RTF presentation template file has an .RTF extension and allows you to define the header, footer, title page, table of contents, and general style of your RTF report. You can insert pictures, page borders and variables in an RTF presentation template.

The presentation template takes into account all the presentation options that you have defined in the Report Wizard or Report Editor, except the table of contents report item as this is defined in the RTF presentation template.

> **Note**
>
> If you open a report saved in a previous version, the RTF Template list is set to <None>.

1.8.8.3 HTML Presentation Templates

An HTML presentation template file (`*.htmltpl`) allows you to define the header, footer, home page, and style sheet (for font, color, background, size, margin, and alignment) for your HTML report. The formatting defined in your style sheet will override any defined in the Report Wizard or Report Editor.

> **Note**
>
> If you open a report saved in a previous version, the HTML presentation template has the values of the default HTML presentation template or the Presentation template box will be set to (<None>).

Style Sheet Structure

The following table describes the use of class names in the report style sheets:

Class names	Description
BODY	Background styles for all report pages, except the home page frame and the browser frame page.

Class names	Description
.BROWSERBODY	Background styles for the browser page.
.HOMEBODY	Background styles for the home page.
.TEXT	Font styles for text blocks, like description, annotation, etc.
.TITLE	Font styles for the title of the home page.
TABLE	Styles for tables.
TD	Styles for table cells.
Table.GRID TABLE.GRID TD.HEADER TABLE.GRID TD TABLE.GRID TD TD2	Styles for the tables, header cells, even row, and odd row cells of object lists.
Table.FORM TABLE.FORM TD.HEADER TABLE.FORM TD	Styles for the tables, row header cells, and value cells of object cards.
Table.TEXT TABLE.TEXT TD	Styles for the tables and the cells of tables around the text blocks.
Table.TITLE TABLE.TITLE TD	Styles for the tables and the cells of tables around the home page title.
Table.GRAPHICS TABLE.GRAPHICS TD	Styles for the tables and the cells of tables around graphics.
H1, H2, H3	Styles for level 1, 2, and 3 headings.
HR	Styles for separator lines.
A:LINK A: VISITED A: HOVER	Colors for non-visited, visited, and highlighted hyperlinks.
.BROWSER	Font styles for the browser.
A.BROWSER:LINK A.BROWSER:VISITED A.BROWSER:HOVER	Colors for non-visited, visited, and highlighted hyperlinks of the browser.

Class names	Description
TABLE.NAVGROUP TABLE. NAVGROUP TD	Styles for the tables and the cells of tables around the navigation buttons.
.NAVBUTTON A.NAVBUTTON:LINK A.NAVBUTTON:VISITED A.NAVBUTTON:HOVER	Font styles for navigation buttons and for non-visited, visited, and highlighted hyperlinks of navigation buttons.

1.9 Comparing and Merging Models

PowerDesigner provides powerful tools for comparing models and merging them.

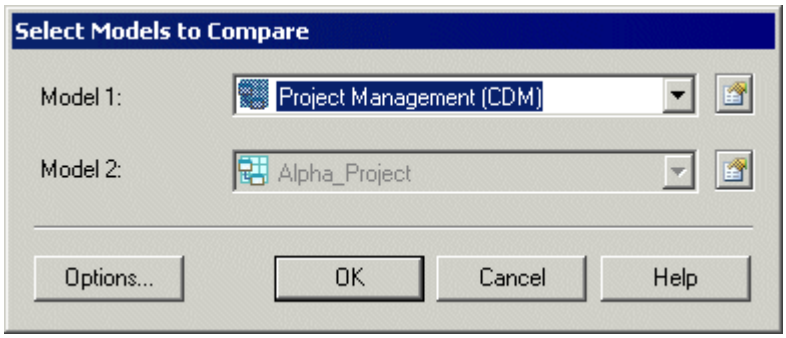
1.9.1 Comparing Models

You can compare the content of two PowerDesigner models or two resource files of the same type at any time. You may want to compare models or resource files to review evolutions in models or resources manipulated by different development teams or to evaluate the differences that exist between the models or resources before merging them.

Procedure

1. Select *Tools* > *Compare Models* to open the Select Models to Compare dialog box.

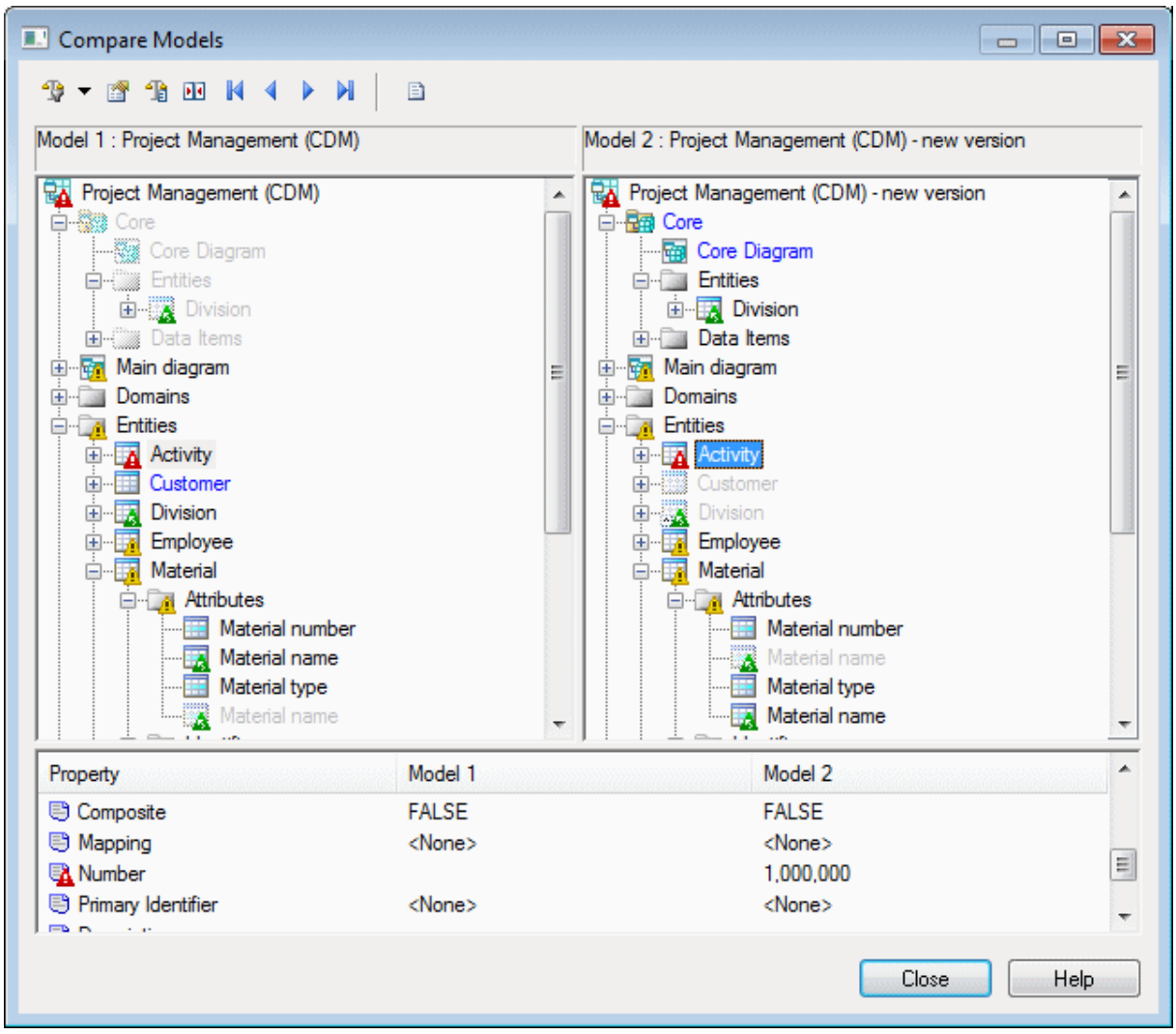
   The current model is displayed in the *Model 2* field.
2. Select a model to compare in the *Model 1* list. Only models of the same type in the workspace are available for comparison.



3. [optional] Click the *Options* button to open the Comparison Options window and specify which objects and properties you want to include in the comparison (see Excluding Objects and Properties from a Compare or Merge [page 373]).
4. Click *OK* to open the Compare Models window, which displays the objects contained within the models or resource files in a tree format, and highlights the differences between them (see Reviewing Differences Between Models [page 367]). You can apply a filter to simplify the list of differences (see Filtering Objects and Changes During a Compare or Merge [page 375]).

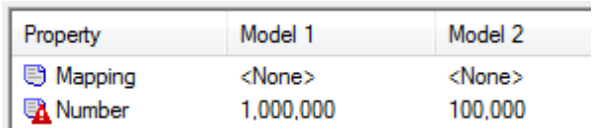
1.9.1.1 Reviewing Differences Between Models

The Compare Models window displays the models to compare as two trees of objects in left and right panes (with the model from which you launch the comparison on the right), and a third pane at the bottom with three columns, listing each of the properties of the selected object, and the values assigned to these properties in the two models.



Explore the tree by opening nodes to reveal categories, objects, and sub-objects. To expand all nodes, press the numpad star (*). To collapse all nodes, press the numpad minus sign (-).

The following types of differences can be detected during a comparison:

Symbol	Description
	Existence - The object is present in one model (blue text) and absent in the other (greyed text and icon). **i Note** The owner of PDM objects (such as tables) is included in their identifiers and objects with the same name but different owners are treated as separate objects. For example, a table **T1** has an owner **DEV** in **PDM1** and an owner **DBA** in **PDM2** and the tables are not linked. If an owner were specified in only one model, then the tables would be linked and show a change in the **Owner** property value.
 Properties pane: 	*Modification* (red warning triangle) - One or more attributes of the object has different values in each of the models. To review the differences in properties, look for the red warning triangles in the list of properties in the bottom pane. **i Note** To open the property sheet of any object in either model, right-click it in the tree and select *Properties*.
	Child Modification (yellow warning triangle) - One or more sub-objects beneath the object has differences.
	Movement (green triangle with arrow) - The object is in a different position in the collection or in a different package. The normal icon and text represent the object's position in the present model and the greyed text and icon represent its position in the other model. The green triangle becomes red if the object also has modifications. **i Note** To go to the other instance of the object, right-click it in the tree and select either *Next Occurrence* or *Previous Occurrence*.

The following tools are available in this window:

Tool	Description
	Filter - Apply one or more of the predefined filters to control the types of differences that are displayed (see Filtering Objects and Changes During a Compare or Merge [page 375]).

Tool	Description
	Properties - Open the property sheet of the selected object.
	Options - Open the Comparison Options dialog to control the types of metaclasses, attributes, and collections that are displayed (see Excluding Objects and Properties from a Compare or Merge [page 373]
	Manually Synchronize Two Objects - Select an object in either model and then click this tool to link it to an object in the other model or to break such a link (see Repairing or Breaking Links Between Objects [page 376]).
	First/Last Difference - Jump to the first or last difference in the tree.
	Previous/Next Difference - Jump to the previous or next difference in the tree.
	Compare Preview - Open the Compare Preview window, which gives a list of the differences in textual or list formats (see Previewing, Printing, and Saving Comparison Results [page 377]).

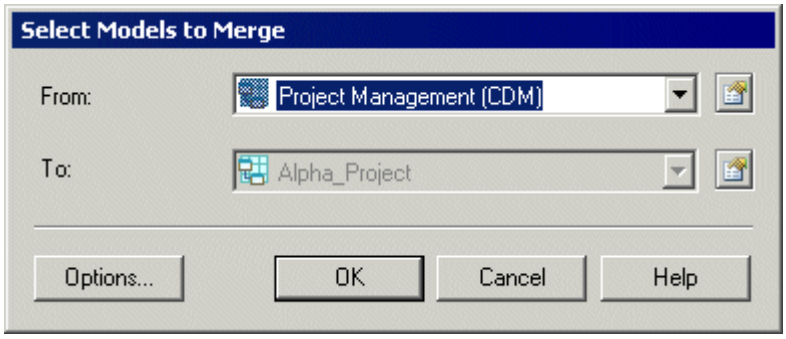
1.9.2 Merging Models

You can merge the content of two PowerDesigner models or two resource files of the same type at any time.

Procedure

1. Select *Tools* > *Merge Models* to open the Select Models to Merge dialog.

   The current model is displayed in the *To* field.
2. Select a model to merge in the *From* list. Only models of the same type in the workspace are available for merging.



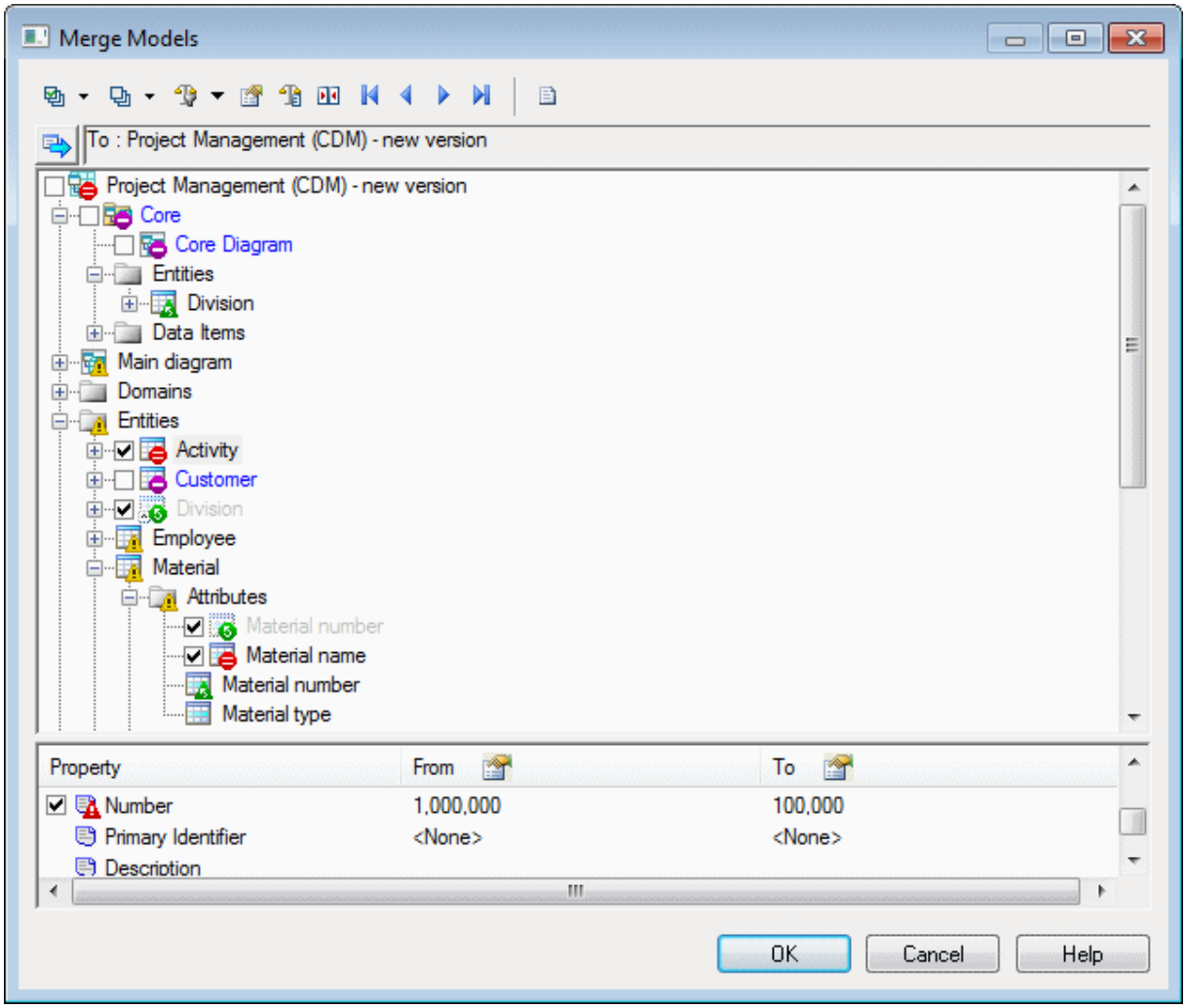
> **i Note**
>
> If you want to merge two PDMs, BPMs, OOMs or XSMs, you have to select two models with the same target DBMS or language.

3. [optional] Click the *Options* button to open the Comparison Options dialog and specify which objects and properties you want to include in the comparison and possible merging (see Excluding Objects and Properties from a Compare or Merge [page 373]).
4. Click OK to open the Merge Models window, which displays the objects contained within the models or resource files in a tree format, and highlights the differences between them (see Selecting Merge Actions [page 370]). You can apply a filter to simplify the list of differences (see Filtering Objects and Changes During a Compare or Merge [page 375]).
5. Selecting the merge actions that you want to perform, and then click *OK*.

   The Merge Models dialog box closes and the merge begins, with messages displayed in the *Output* pane. A final message informs you that the models were successfully merged.

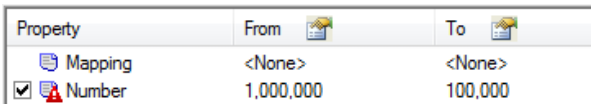
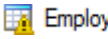
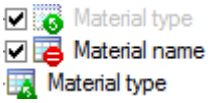
1.9.2.1 Selecting Merge Actions

The Merge Models window displays the anticipated result of your proposed merge as a tree of objects, with check boxes to the left of each potential change to allow you to approve or reject it. The properties pane lists each of the properties of the selected object, and allows you to select or reject the values proposed in the To column. Select or deselect actions as appropriate and then click *OK* to perform the merge.



Explore the tree by opening nodes to reveal categories, objects, and sub-objects. To expand all nodes, press the numpad star (*). To collapse all nodes, press the numpad minus sign (-). Use the *Show/Hide Source Model* toggle at the top left of the pane to split the pane and display the source model to the left of the target model.

The following types of merge actions can be approved by selecting or rejected by deselecting the check box:

Symbol		Description
	(blue plus sign and greyed text and icon)	*Creation* - The object is absent from the model to be modified and present in the model to apply. Selected: The object will be created.
	(purple minus sign and blue text)	*Deletion* - The object is present in the model to be modified and absent in the model to apply. Selected: The object will be deleted.
 Properties pane:  (red warning triangle)	(red equals sign)	*Modification* - One or more attributes of the object has different values in each of the models. To review the differences in properties, look for the red warning triangles in the list of properties in the bottom pane. **Note** Double-click a property to display differences in textual format between models. The value to be applied is in blue, preceded by >>, and the value to be deleted is in red preceded by <<. Selected (object): All values in the model to be applied are copied to the model to be modified. Selected (property): The value in the From column is applied to the model to be modified.
	(yellow warning triangle)	*Child Modification* - One or more sub-objects beneath the object has differences. Expand the tree beneath the object to review and approve these changes.
	(green circle with arrow and greyed text and icon and green triangle with arrow)	*Movement* - The object is in one position in the model to be modified (green circle with arrow and greyed text and icon) and in a different position in the collection or in a different package in the model to be applied (green triangle with arrow). The green triangle becomes red if the object also has modifications. Selected: The object is moved from the position with the green circle to the position with the green triangle. **Note** To go to the other instance of the object, right-click it in the tree and select either *Next Occurrence* or *Previous Occurrence*.

By default, merge actions are selected as follows:

- During merge of unrelated models - all creation, modification and move actions are selected, but deletion actions are deselected to preserve the superset of all objects in the two models as far as possible.
- During check in/check out from the repository - based on when the latest changes were saved in the model.
- When updating a generated model - based on whether changes have been made to the generated model (if the *Preserve modifications* generation option is selected).

The following tools are available in this window:

Tool	Description
	Select/Deselect All - Select or deselect all merge actions. Click the arrow to the right of the button to select or deselect all creation, deletion, modification, or move actions.
	Filter - Apply one or more of the predefined filters to control the types of merge actions that are displayed (see Filtering Objects and Changes During a Compare or Merge [page 375]). **i Note** The filters control only what is displayed in this window. Even if an action is not displayed, if it is selected, it will be applied when you click *OK*.
	Properties - Open the property sheet of the selected object.
	Options - Open the Comparison Options dialog to control the types of metaclasses, attributes, and collections that are displayed (see Excluding Objects and Properties from a Compare or Merge [page 373]. **i Note** The options control whether merge actions will be applied to objects. If you exclude a metaclass, attribute, or collection, then it will not appear in the window, and no merge actions will be applied to it.
	Manually Synchronize Two Objects - Select an object in either model and then click this tool to link it to an object in the other model or to break such a link (see Repairing or Breaking Links Between Objects [page 376]).
	First/Last Difference - Jump to the first or last difference in the tree.
	Previous/Next Difference - Jump to the previous or next difference in the tree.
	Merge Preview - Open the Merge Preview window, which gives a list of the differences in textual or list formats (see Previewing, Printing, and Saving Comparison Results [page 377]).

1.9.3 Excluding Objects and Properties from a Compare or Merge

You can control what types of objects will appear in the comparison or merge window along with the attributes and collections to compare for them through the Comparison Options dialog. By default, all objects and properties are selected.

Context

> **i** Note
>
> When merging models, differences between the two models that concern deselected metaclasses, attributes, or collections are disregarded, and will not be merged.

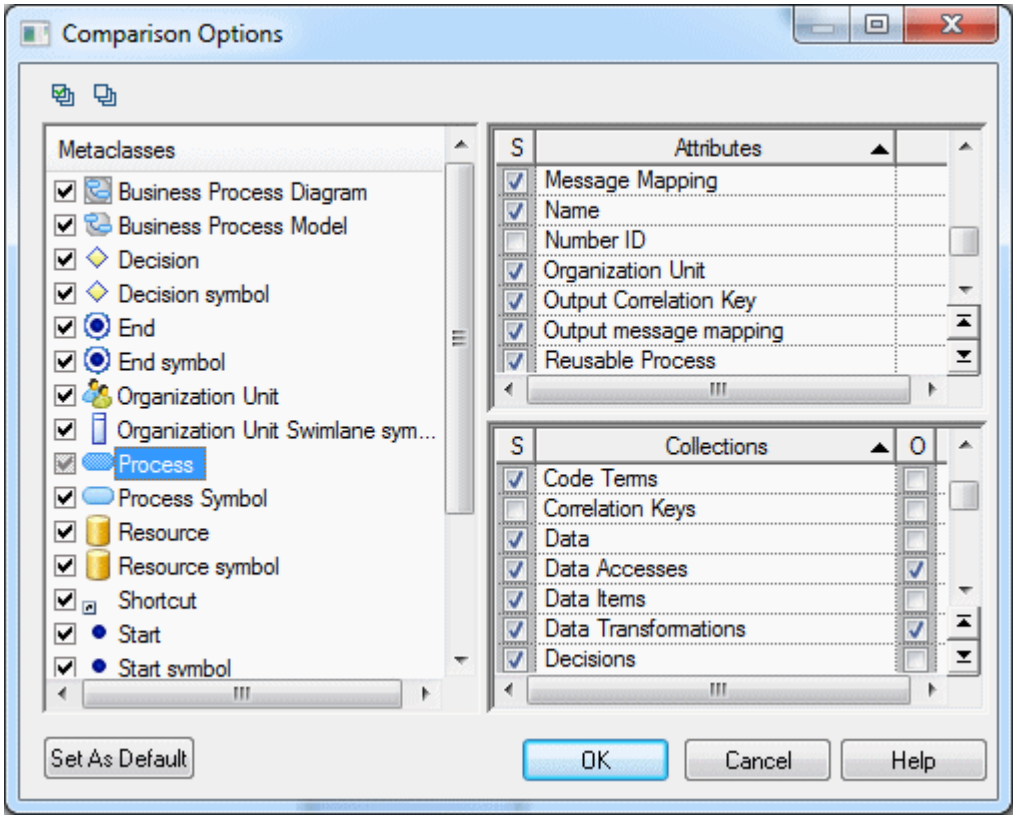
Procedure

1. Click the *Options* button in the *Select Models to Compare/Merge* or the *Compare/Merge Models* window.

   The *Comparison Options* dialog lists *Metaclasses* in the left pane, and the *Attributes* and *Collections* of the selected metaclass in the two right panes.
2. To exclude an attribute or collection from comparison, click on the name of the metaclass in the left pane and then deselect the appropriate check boxes in the right panes.

   > **i** Note
   >
   > For collections that can be reordered (for example, Associated Business Rules, or Columns in a PDM table), changes in the order of members of the collection are considered, by default, as differences. To ignore such changes, deselect the checkbox in the *[O]rdered* column.

   For example, to exclude the **`Number ID`** attribute and **`Correlation Keys`** collection under the **`Process`** metaclass from comparison, click on the **`Process`** metaclass name, and then deselect the relevant check boxes:



Metaclasses with one or more attributes or collections deselected have their check box greyed.

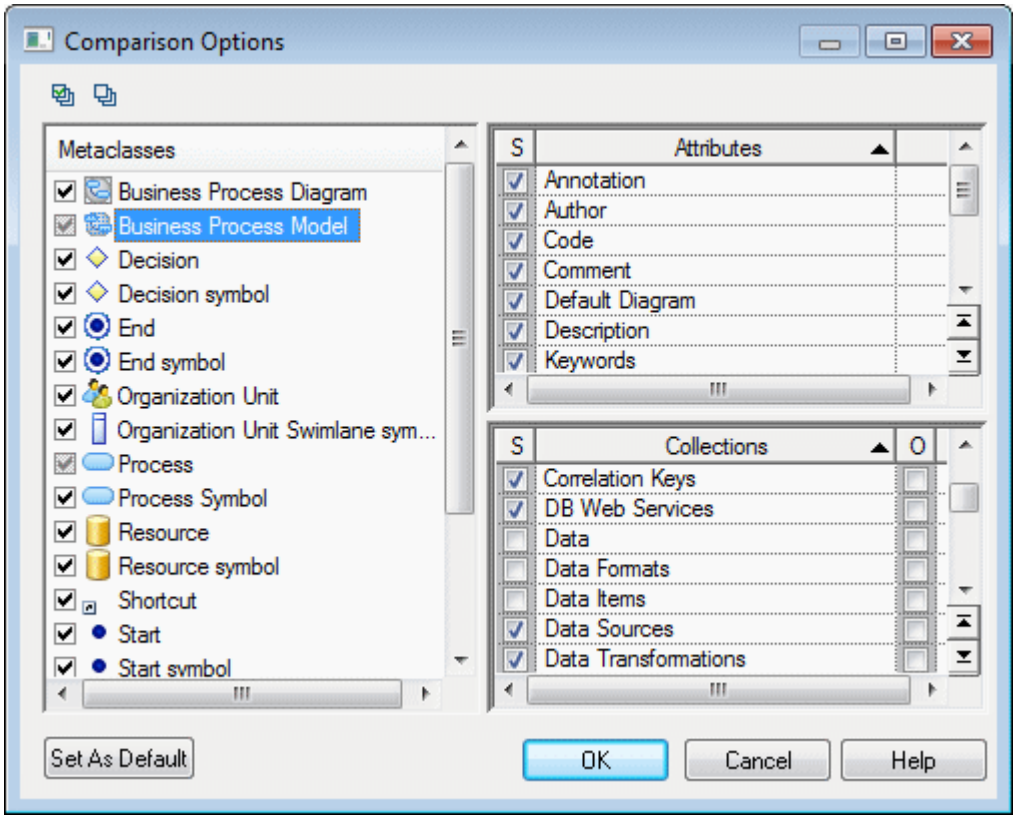
To exclude the same attribute or collection from every metaclass where it appears, right click it and select *Deselect for All Metaclasses*.

You can select multiple metaclasses at once using the CTRL or SHIFT keys, in which case only the attributes and collections available under all the selected metaclasses are displayed in the right panes.

> **i Note**
>
> To exclude all the properties (attributes and collections) of a metaclass from comparison, deselect its check box in the *Metaclasses* list. Doing so will only exclude any changes in properties from comparison, but will not exclude the metaclass itself and will leave any creations or deletions to be detected.

3. To exclude a metaclass from comparison (both modifications of properties and creations and deletions), click on the model metaclass name, and then deselect the check box of the appropriate collection.

   For example, in a BPM, to exclude the **`Data`**, **`Data Formats`**, and **`Data Items`** collections from comparison, click on the **`Business Process Model`** metaclass, and then deselect these collections:



To ensure that these metaclasses will not be considered, even if objects appear under a package, right-click each collection and select *Deselect for All Metaclasses*.

4. [optional] Click the *Set as Default* button to save your exclusions as the default for future comparisons or merges for this type of model.
5. Click *OK* to return to your comparison or merge dialog.

1.9.4 Filtering Objects and Changes During a Compare or Merge

Click the arrow next to the *Filter* tool in the Compare or Merge window to select a filter to apply to the object trees and display only certain types of changes, conflicts or, any combination thereof.

The following filters are available:

- *All Objects* - [default] Remove all filters.
- *All Changes* - Show only objects that have some difference.
- *Only Created Objects* - Show only objects that will be created during a merge (or that are absent from Model 2 in a comparison).
- *Only Deleted Objects* - Show only objects that will be deleted during a merge (or that are absent from Model 1 in a comparison).
- *Only Modified Objects* - Show only objects that have differences between attribute values
- *Only Moved Objects* - Show only objects that have moved in a collection or to another package or parent. In the Advanced Filters dialog, you can distinguish between:

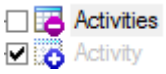
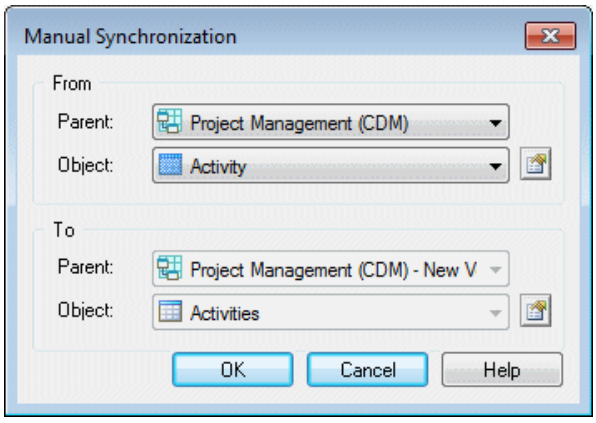
    - *Show Moved Objects (Parent Change)* - When moving to a new parent, package, or collection
    - *Show Changes of Position with Collection* - When remaining in the same parent, package, or collection. For example, when you change the order of attributes within the same entity.
- *Only Identical Objects* - Show only objects that have no changes.
- *Only Selected Objects* - [merge only] Show only objects that have merge actions selected.
- *Only Conflicts* - [repository and generation merge only] Show only objects that have conflicting modifications in both models A merge conflict occurs when the same object has been modified differently in the origin and generated model or the local and Repository model.
- *Hide All Preservations* - [generation merge only] Show only changes made in the origin model since the last generation.
- *Advanced Filters* – Opens the Comparison Filters dialog, which allows you to combine several filters

1.9.5 Repairing or Breaking Links Between Objects

Sometimes, when an object has been renamed or modified outside PowerDesigner, its link with the corresponding object in another model may be lost, and the two objects are shown independently, with a creation and a suppression in each model tree. You can manually repair the link between the objects or, in rare cases, break a link between objects that you do not want to compare or merge.

Select the item for which you want to repair or break the link, click the *Manually Synchronize Two Objects* tool, and select the object with which to pair it in the *From* or *To* box as appropriate.

For example, if the **`Activities`** entity in your model needs to be link to the **`Activity`** entity in the other model:

Problem	Solution
	

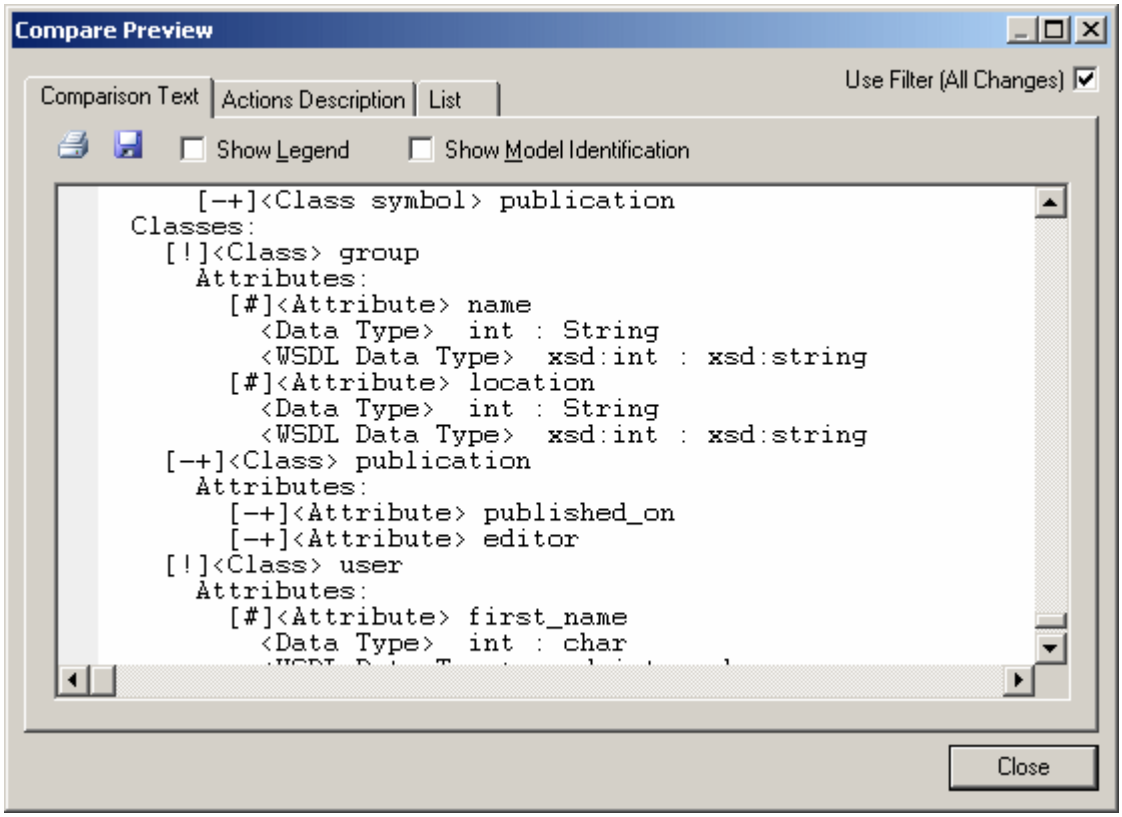
> **Note**
>
> To break a comparison link, select an object that is linked to an object in the other model, click the tool, and select **`None`** in the other field.

1.9.6 Previewing, Printing, and Saving Comparison Results

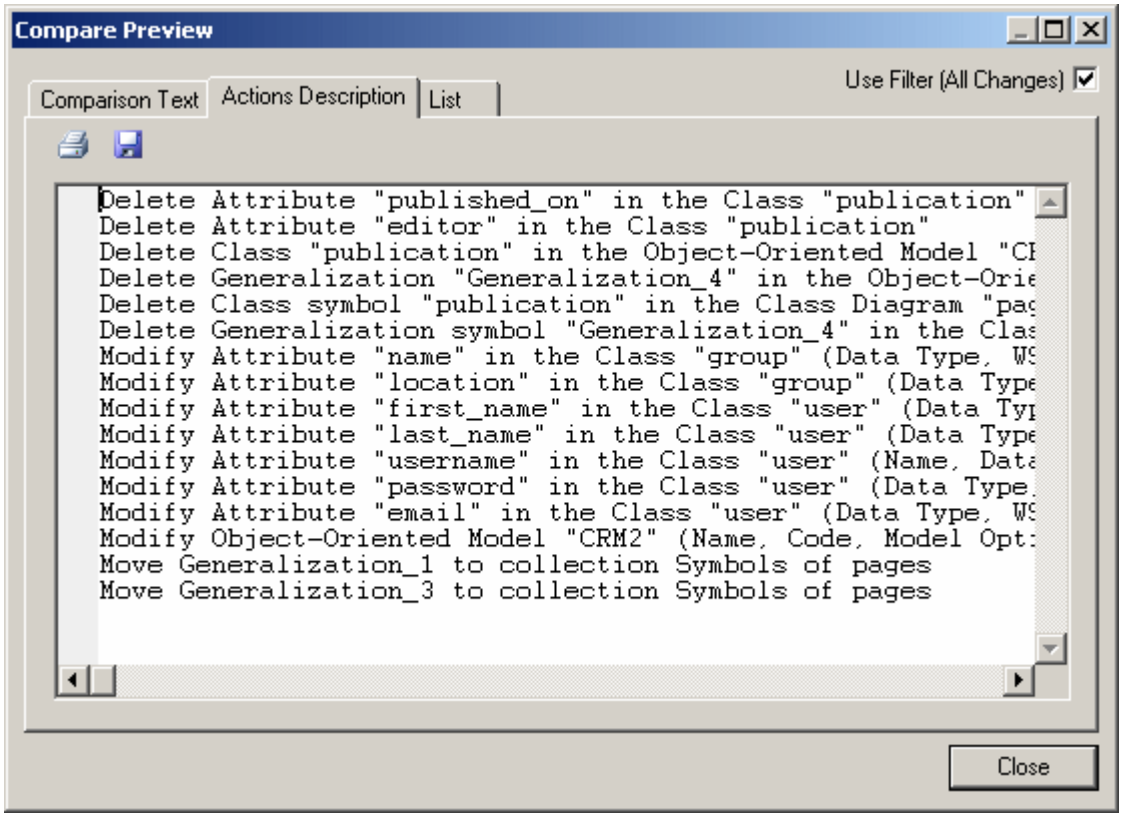
You can preview, save, and export the results of the comparison by clicking the Compare Preview tool to launch the Compare Preview window.

This window provides three formats viewing the comparison results on the following tabs:

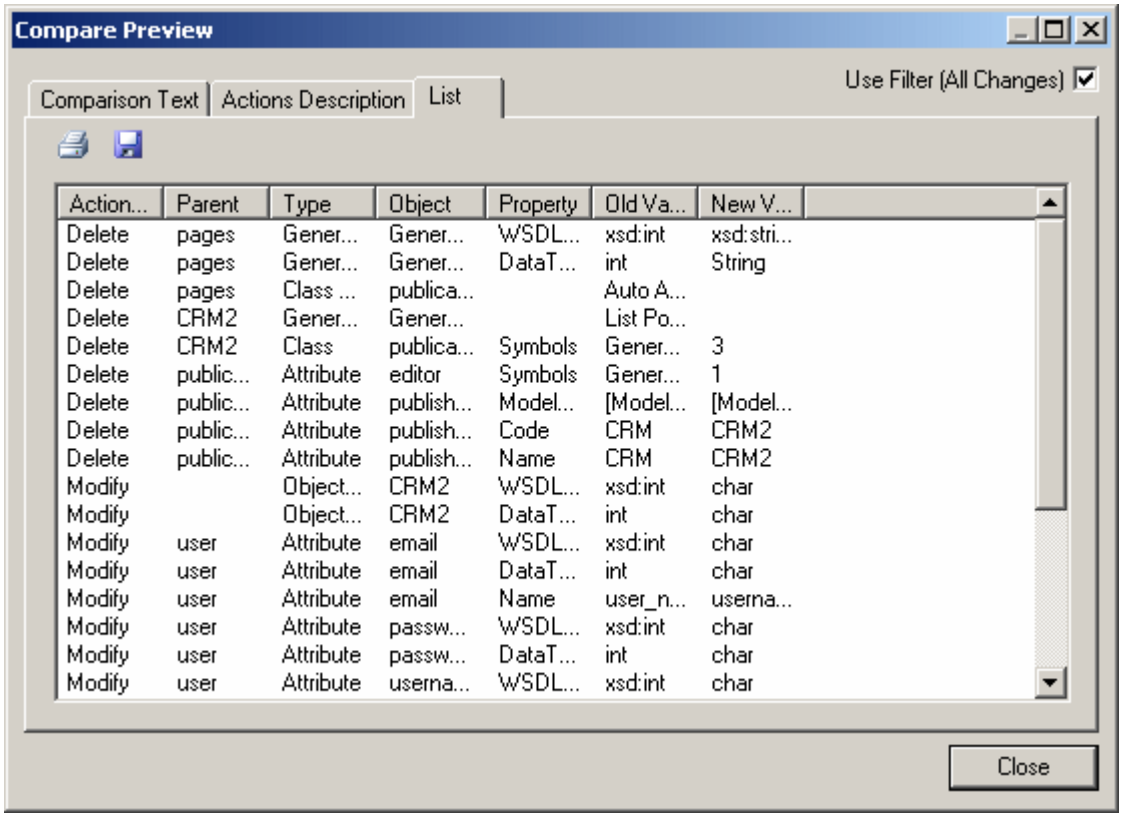
- Comparison Text – a textual representation of the differences between the models. This is the most exhaustive format, and provides complete information about the whole object tree. Select the Show Legend and/or Show Model Identification options to display additional information at the beginning of the text.



- Actions Description – a list of the actions that would need to be taken to render the models the same. This will often be the most readable format as it represents each change in the form of a sentence:



- List – a columnar representation of the required actions. This format is the most easily manipulated. You can sort by any of the columns, and export the list in a variety of formats to allow for various kinds of transformation:



The following tools are available on each of the tabs:

Tool	Description
Use Filter	Filters the changes using the filter specified in the main window (see Filtering Objects and Changes During a Compare or Merge [page 375]). If no filter has been specified before the preview is launched, then this option is not visible.
	Print
	Save - Saves the comparison in text format for Comparison Text and Actions Description , and in XML, CSV, RTF, or HTML format for List.

The following columns appear on the List tab:

Column	Description
Action Type	Specifies the type of action to perform. Can be any of the following: • Create – creates an object • Delete – deletes an object • Modify – modifies an object • Move Object – moves an object to a new parent • Col Insert – inserts an object into a collection • Col Remove – removes an object from a collection • Col Move – moves an object from one position to another within a collection
Parent	Specifies the parent of the object to be changed (the original parent, in the case of an object that will be moved). For example, a class for an operation, or a diagram for a symbol.
Type	Specifies the type of the object to be changed.
Object	Specifies the name of the object to be changed
Property	Specifies the object property or collection to be changed. Empty for a Create, Delete, or Move Object action.
Old Value	Specifies the original value of the property to be changed. Empty in the case of a Create or Col Insert
New Value	Specifies the updated value of the property. Empty in the case of a Delete or Col Remove.

For example, changing the datatype of attribute MyVariable from int to long would yield the following column entries:

```
Action Type  Parent Type   Object  Property  Old Value  New Value
Modify   MyClass  Attribute  MyVariable  DataType  int    long
```

1.10 Customizing Your Modeling Environment

PowerDesigner is highly customizable. You can modify its interface to suit your work habits, set default naming conventions, change the appearance of object symbols, add new properties to objects, and even create your own types of objects.

> **i** Note
>
> Administrators can prepare customizations to the PowerDesigner interface and push them to users (see Customizing the PowerDesigner Interface [page 554]).

1.10.1 User Profiles

User profiles help you to standardize the look and feel of your models and to support standards by making groups of options and preferences portable for sharing and reuse across your organization. Various user profiles are provided with PowerDesigner, and you can create your own.

Various kinds of PowerDesigner options and preferences are stored in your model files and/or in your Windows registry. User profiles can contain default values for:

- Display Preferences – [model or registry] to control the color, shape, size, etc. of your diagram symbols and the information that is displayed upon them (see Display Preferences [page 228]).
- Model Options – [model only] to control naming conventions (see Naming Conventions [page 155]), case sensitivity, notation, default values, etc (see the appropriate modeling guide).
- General Options – [registry only] to control dialog preferences, environment variables, fonts, etc (see General Options [page 388]).
- Other options – [registry only] such as the layout of toolbars and windows (see Organizing Views [page 402]), favorite property tabs (see Customizing a Property Sheet [page 109]), default columns for object lists (see Customizing Object List Columns and Filtering Lists [page 123]), etc.

User profiles are commonly used by an administrator to standardize and simplify the PowerDesigner interface, and are delivered automatically through the repository (see Controlling Defaults for Options and Preferences [page 567]).

> ⚠ Caution
>
> We discourage the delivery of user profiles via a shared network drive, which can cause PowerDesigner to become unresponsive if the connection to the shared drive is lost.

To view the list of user profiles, select *Tools* > *Resources* > *User Profiles*. For information about the tools available in resource file lists, see PowerDesigner Resource Files [page 408].

User profiles can contain any number of default preferences and options, the application of which is cumulative. Thus, if you apply a user profile that contains display preferences specifying that PDM table symbols are by default red, and view symbols blue, and then apply a second user profile that sets tables by default to green, then table symbols will be green, and view symbols blue. You can, of course, override these default preferences and options locally in the model.

1.10.1.1 Applying a User Profile

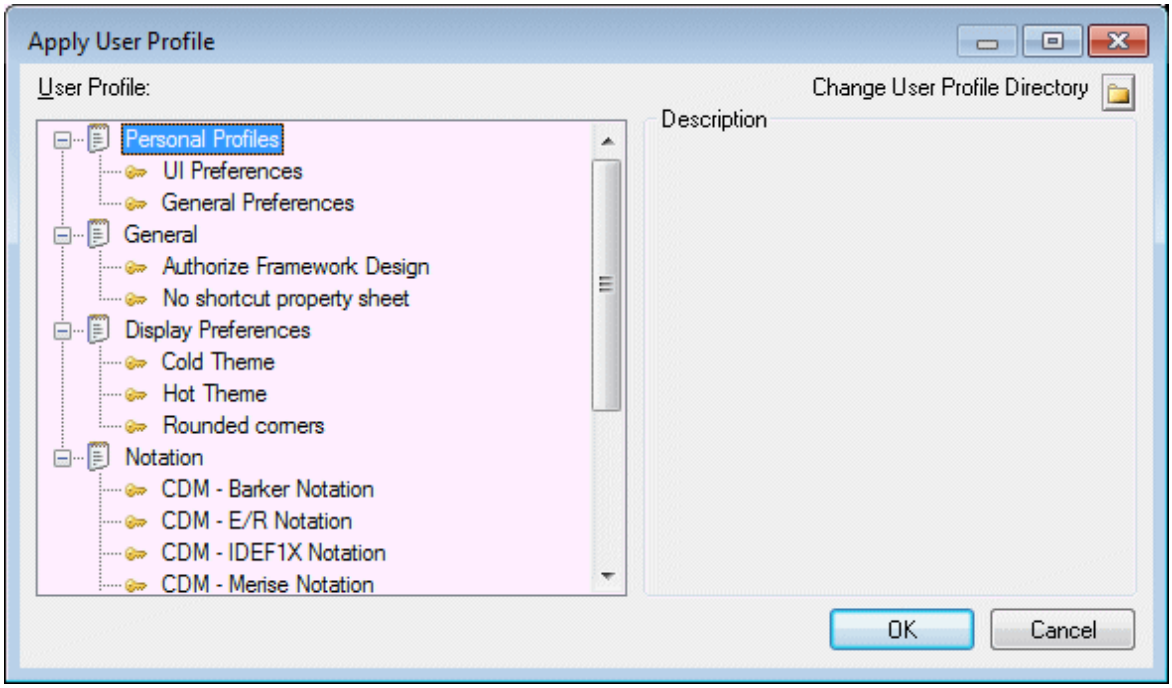
You can apply a user profile at any time. The defaults, preferences, and options contained within the profile will overwrite the existing values for those defaults, preferences and options, while leaving all others unchanged. Note that while general options take effect immediately, model options and display preferences only take effect when you create a new model.

Context

> **Note**
>
> To return to a default user profile prepared for you by an administrator(see Controlling Defaults for Options and Preferences [page 567]), select the **`General Preferences`** entry in the *Personal Profiles* category. To return to your default UI profile, select the **`UI Preferences`** entry.

Procedure

1. Select *Tools* > *Apply User Profile* to open the Apply User Profile dialog:



2. [optional] Click the *Change User Profile Directory* button, and browse to a folder where you have saved user profiles. By default they are saved in the `<install_dir>/Resource Files/Profiles` directory.
3. Select a profile from the list and click *OK*.

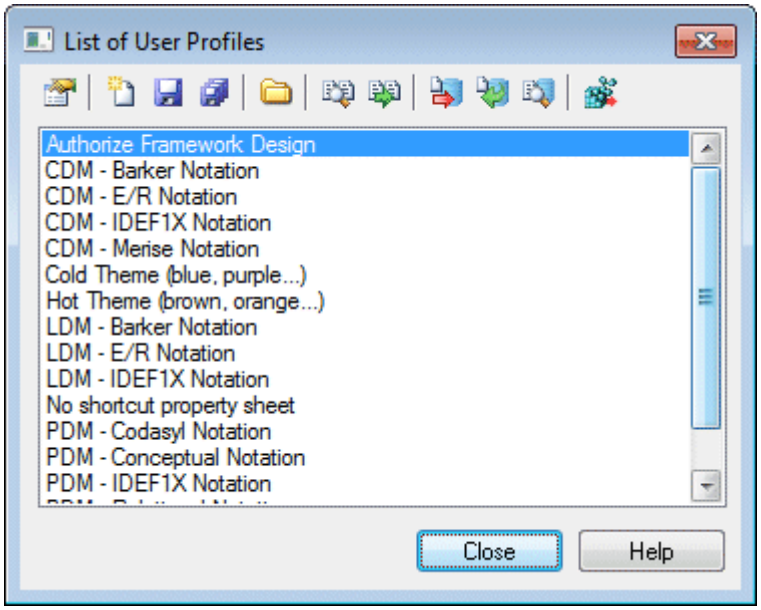
   The user profile is applied, and overwrites the defaults, preferences, and options contained within it with the values it defines.

1.10.1.2 Creating a User Profile

You can create a user profile by copying an existing profile, or by copying values from an open model or from your registry.

Procedure

1. Select *Tools* > *Resources* > *User Profiles* to open the List of User Profiles:

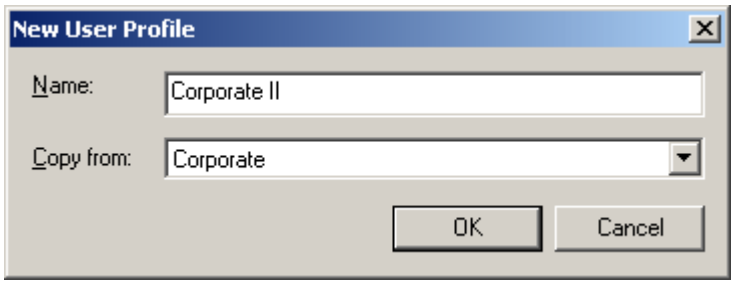


The following tools are available above the list:

Tool	Description
	Properties – opens the selected profile in the resource editor.
	New – creates a new profile.
	Save – saves the selected profile.
	Save All – saves all the profiles.
	Path – Specifies a directory from which to populate the list.
	Compare – Opens the Compare window to compare the structure of two profiles.

Tool	Description
	Merge – Updates the selected profile by merging preferences from another profile, a model, or the registry (see Updating a User Profile [page 385]).
	Check In - [if the repository is installed] Checks the selected resource file into the repository. For information about storing your resource files in the repository, see Sharing Resource Files via the Library [page 551].
	Update from Repository - [if the repository is installed] Checks out a version of the selected file from the repository to your local machine.
	Compare with Repository - [if the repository is installed] Compares the selected file with a resource file stored in the repository.
	Apply to Registry – Applies the selected profile to your registry, overwriting your previous defaults and preferences. This has the same effect as selecting *Tools* > *Apply User Profile* (see Applying a User Profile [page 381]).

2. Click *New* to open the New User Profile dialog:



3. Enter a name for your profile and select a source from which to copy from. You can choose between:
   - Another user profile – makes a copy of the existing profile.
   - <Registry> - copies all your PowerDesigner defaults from your Windows registry.
   - <Model> - opens the Model for User Profile dialog which allows you to specify the preferences to copy from an open model (see Copying Preferences from a Model [page 385]).
4. 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.

1.10.1.3 Updating a User Profile

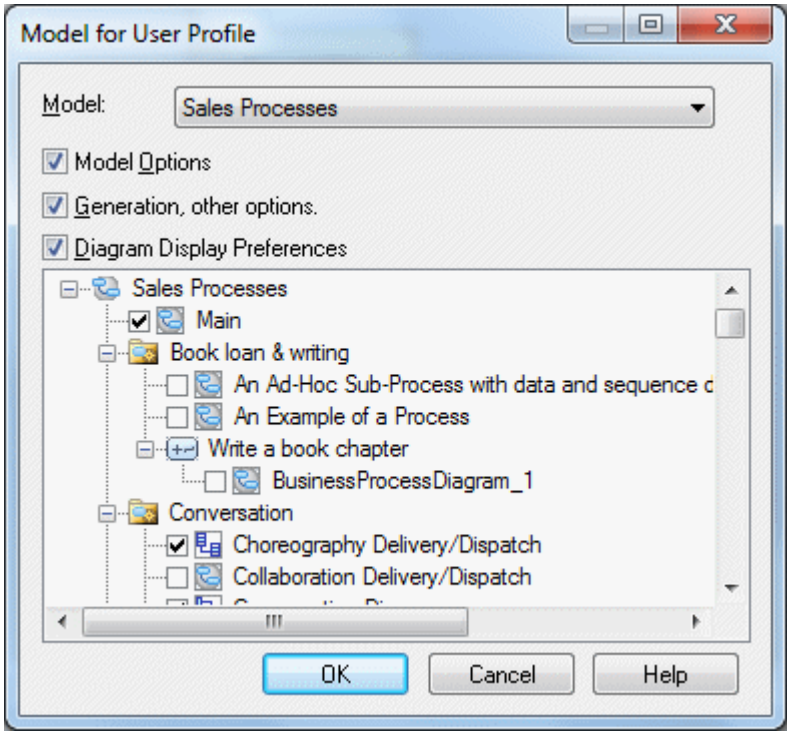
You can update a user profile by copying values from an existing profile, from an open model, or from the registry.

Procedure

1. Select *Tools* > *Resources* > *User Profiles* to open the List of User Profiles, select the profile to update and click the *Merge* tool to open the Select User profiles to Merge dialog.
2. In the *From* field, select a source from which to copy from. You can choose between:
   - Another user profile
   - <Registry> - copies all your PowerDesigner defaults from your Windows registry. Note that model check options and generation options are not stored in the registry and can only be extracted from a model or another user profile.
   - <Model> - opens the Model for User Profile dialog which allows you to specify the preferences to copy from an open model (see Copying Preferences from a Model [page 385]). Note that general options, and options relating to the PowerDesigner interface (such as toolbar and window layout, favorite property sheet pages and grid layouts) are not stored in models and can only be extracted from the registry or another user profile.
3. Click *OK* to open the Merge window, which lets you review and approve each proposed change before committing it. For detailed information about using this window, see Comparing and Merging Models [page 366].
4. When you are satisfied, click *OK* to update the profile and return to the list.

1.10.1.4 Copying Preferences from a Model

You can create or update a user profile by copying preferences from a model.



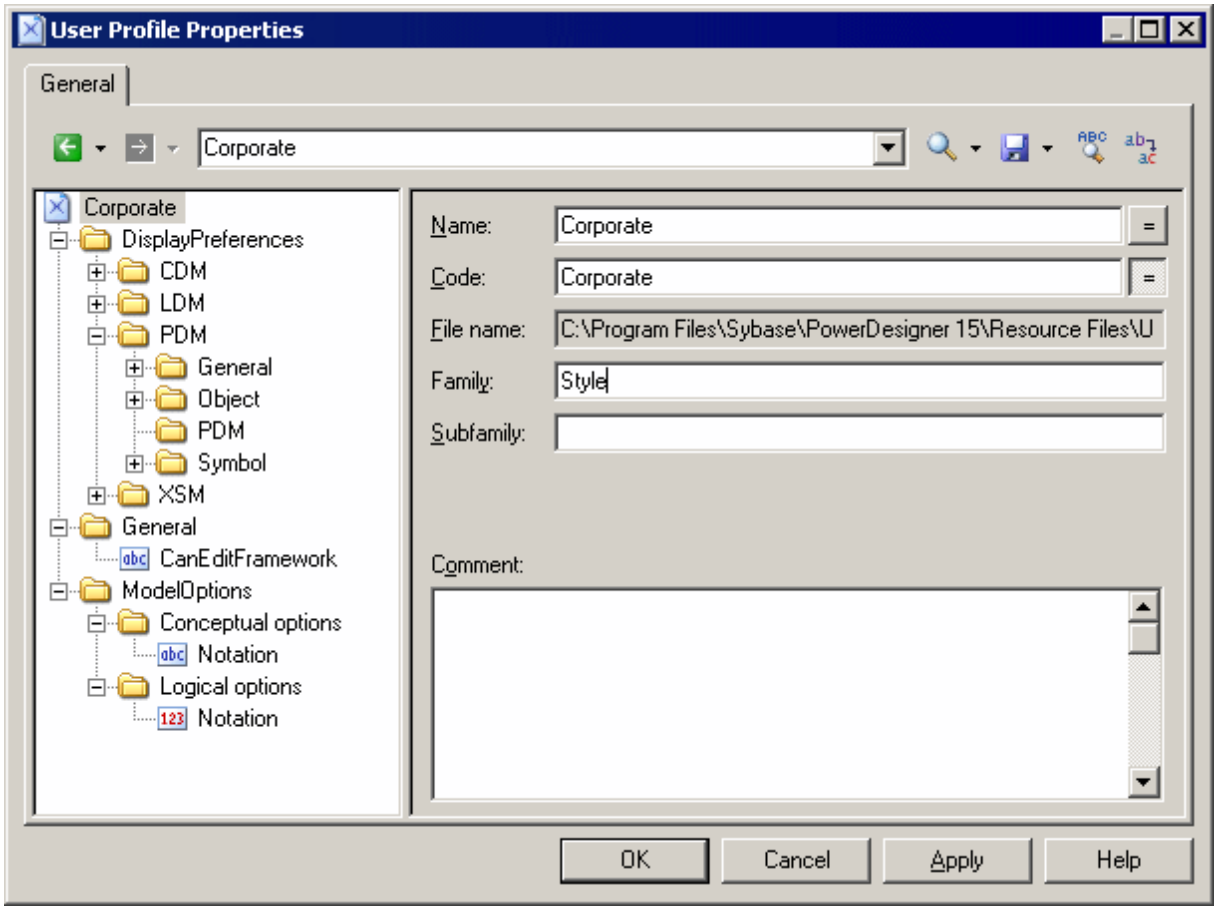
Select which model open in the Workspace to copy from and then select which of the following options and preferences to capture:

- Model Options
- Generation, and other options.
- Diagram Display Preferences - Select the diagrams to copy preferences from in the tree. To avoid conflicts, you can only select one instance of each diagram type.

1.10.1.5 Reviewing and Editing User Profiles in the Resource Editor

You can open any user profile in the Resource Editor to review and edit it. Since the registry key names and values are not necessarily easy to read, we recommend that you restrict your edits in this environment to deleting unwanted defaults. You can always make additions or adjustments to the values of user profiles by extracting them from models or other profiles.

In the following example, the Corporate user profile contains display preferences for CDMs, LDMs, and PDMs, along with a general option and some model options:



To delete an item, simply right-click it and select *Remove*.

The profile root node has the following properties:

Property	Description
Name/Code	Specifies the name and code (used for scripting) for the profile.
File name	[read-only] Specifies the path to the profile file.
Family	Specifies the family in which the profile will appear in the Apply User Profile window (see Applying a User Profile [page 381]) and if it is added to the PowerDesigner installer image. If no value is entered, the profile is added to the General family.
Subfamily	Specifies a subfamily within the profile family.

1.10.1.6 Making Custom User Profiles Available in the PowerDesigner Installer

Once you have developed your own user profiles, you may want to add them to your PowerDesigner install image in order to allow your users to install them when they install PowerDesigner.

The user profiles that are installed with PowerDesigner are stored in the following directory on the first install CD:

```
Setup/User Profiles
```

To include your profiles in the install, simply copy them to this directory in your install image.

For information about installing PowerDesigner, see *Installation Guide* > *Installing PowerDesigner* .

1.10.2 General Options

You can control the look and behavior of the PowerDesigner interface through the general options, the values for which are saved in the registry of your local machine.

To set general options, select *Tools* > *General Options* . The following options are available under the *General* category:

Option	Description
Autosave	*Save recovery backup file every <x> minutes* - Instructs PowerDesigner to save changes to all open models to a recovery backup file at the specified interval. Enabling this option provides you with a recovery option if PowerDesigner or your computer crash before you have had a chance to save your changes. The save will take place after the interval has passed, but only when PowerDesigner is idle for more than ten seconds. Note that saving large or multiple models may take several seconds, and that PowerDesigner will not be responsive while it is performing the save. In the event that your modeling session exits abnormally, when next you open PowerDesigner you will be invited to restore your unsaved models. You can choose: • *Yes* - to open the recovered models in the Browser allowing you to review and save or delete them as necessary. • *No* - to reject the recovered models and return your Browser to its last saved state.

Option	Description
Delete	*Confirm object deletion* - Controls the display of a confirmation dialog box when you delete an object. If you disable this option, the confirmation dialog will no longer be shown and the default behavior is as follows: • Deleting an object symbol - the symbol only is deleted. • Deleting an object in the browser - the object, any sub-objects it contains, and any associated diagram symbols are deleted. You can delete any object without confirmation at any time by selecting its symbol in a diagram and pressing `Shift` + `Delete`. For more information, see Deleting Objects [page 106].
Startup	*Auto-reload last workspace* - Opens the last-edited workspace when you launch PowerDesigner. If this option is unchecked, PowerDesigner will open with an empty workspace. *Show welcome page* - Displays the Welcome Page when starting PowerDesigner. If you select the Do not show this page again check box in the Welcome page, this option is deselected.
Browser	*Drag and drop: Default action* - Specify the default result of dragging and dropping an object (without any modifier key) in the Browser. You can choose from the following: • *Move* (`Shift`) – The object is displaced to the new location (parent object, package, model, etc). • *Copy* (`Ctrl`) – A copy of the object is created in the new location. • *Create Shortcut* (`Ctrl` + `Shift`) – A shortcut to the object (see Creating Shortcuts [page 446]) is created in the new location. • *Create replica* (`Alt` + `Shift`) – A replica of the object (see Creating Replicas [page 451]) is created in the new location. The modifier keys given after the option are valid no matter what the default behavior. *Show shortcuts* - Specifies that shortcuts are displayed in the Browser.
Output log	*Log path* - Specifies the path to the log file where PowerDesigner records all of its outputs.
Graphical tool behavior	*Edit in place after creation* - Allows you to directly modify the name of an object from the object symbol in the diagram without opening its property sheet whenever you create an object with the Toolbox.
Sort	*Sort object categories alphabetically* - Lists object categories in the browser and *Model* menu alphabetically. By default, object categories are ordered by importance with core objects (such as PDM tables, CDM entities, and OOM classes) listed first. *Sort object names naturally* - Treats numbers numerically when sorting objects in the Browser. For example, a naturally sorted list of tables would have the order: Table_1, Table_3, Table_12, Table_20.

1.10.2.1 Dialog Box Options

To review and edit options that control the look and behavior of dialog boxes, select *Tools* > *General Options*, and click the *Dialog* category in the left pane.

The following options are available:

Operating Modes

Option	Description
Auto Commit	Specifies that any change made in a property sheet is automatically committed in the object definition. You can cancel a change by pressing using the Undo feature. If this option is disabled, you must click *Apply* or *OK* to commit your changes.

Property Sheets

In PowerDesigner, property sheets appear by default with a certain size and a series of tabs in the upper part of the dialog box. The following options are available for property sheets:

Option	Description
Keep size	Preserves the customized size you have defined
Keep last tab	Opens the property sheet to the last selected tab
Open mode	Controls how property sheets are opened. You can choose between: • Open each object in the same property sheet • Open each object in its own property sheet
Tabs mode	Controls the display of tabs. You can choose between: • Tabs on one row - Aligns all tabs on a single line with arrow buttons < and > for browsing • Tabs on several rows - Displays tabs on two lines, their size corresponding to the length of their title.

Shortcut Property Sheets

The following options are available:

Option	Description
Internal Shortcut	Controls whether double-clicking an internal shortcut opens the property sheet for the shortcut or for the target object. You can switch from one mode to the other by pressing the `Shift` key while opening the property sheet. If the target object cannot be found (target model is closed for example), the shortcut property sheet automatically opens.
External Shortcut	Controls whether double-clicking an external shortcut opens the property sheet for the shortcut or for the target object.

For more information, see Creating Shortcuts [page 446].

Object Lists

The following options are available:

Option	Description
Auto-insert rows	Specifies that a new item is created automatically in a list when you click an empty row
Use default name	Specifies that new items are given default names upon creation, allowing you to create multiple new objects without having to supply names or other properties.

Timestamp Template

Specifies the text generated by the *Timestamp* tool in the *Definition* tab rich text editor (see Definition Tab [page 110]).

By default, the timestamp contains the following GTL code to generate the date, time, and user name:

```
*** %CurrentDate% - %CurrentUser% ***\n
```

For detailed information about working with GTL, see *Customizing and Extending PowerDesigner > Customizing Generation with GTL* .

1.10.2.2 Text Editors

By default, all files are edited using PowerDesigner's internal editor. To define external editors for editing various kinds of text, such as object descriptions and annotations or PDM SQL scripts or OOM Java files, select *Tools* > *General Options*, and click the *Editors* category in the left pane.

Context

By default, all files are edited using PowerDesigner's internal editor. You can define an external text editor to launch automatically for editing various kinds of objects. You can define as many text editors as you need, and the same file extension can have several text editors allocated, with the first listed acting as the default.

Procedure

1. Select *Tools* > *General Options* and click the Editors category in the left-hand pane.
2. Click the Insert a Row tool, and enter a file extension (such as .DOC, .RTF, .TXT, .XLS) in the Extension column.

   The Editor Name and Editor Command columns are both set to <internal> to indicate that the internal PowerDesigner editor will be used to edit files with this extension.
3. [optional] Enter an editor name (such as MS Word, Notepad, MS Excel) in the Editor Name column, and enter an editor command (such as winword.exe) in the Editor Command column. You can click the ellipsis button in this field to browse to the relevant directory.
4. Click OK to close the dialog box.

1.10.2.3 Environment Variables

Environment variables provide access to paths and executables on your workstation for use in generation commands and GTL templates. To review and edit variables, select *Tools* > *General Options*, and click the *Variables* category in the left pane.

Click in the row of an existing variable in order to edit its value. To create a new variable, click the *Add a Row* tool and specify a name and value. The following variables are defined by default:

Variable	Description
CMD	Windows command interpreter. Default: command.com or cmd.exe
HOME	Home directory.

Variable	Description
J2EEVERIF	Batch program for verifying if the deployment jar for an EJB is correct Default: verifier.bat
JAR	Command for archiving java files Default: jar.exe
JAVA	Command for running JAVA programs Default: java.exe
JAVAC	Command for compiling JAVA source files Default: javac.exe
JAVADOC	Command for defining JAVA doc comments Default: javadoc.exe

> **i Note**
>
> To use a variable in GTL (see *Customizing and Extending PowerDesigner > Customizing Generation with GTL*), prefix it with a dollar sign, and wrap it in percent signs. For example, `%$CMD%`.

1.10.2.4 Named Paths

Named paths allow you to resolve links between models and resource files in different environments. To review and edit named paths, select *Tools* > *General Options*, and click the *Named Paths* category in the left pane.

When you save a PowerDesigner model, project, or workspace, the name of a named path is saved in place of the actual path, and when you open a file, the named path is replace with the locally defined value . For example, if the `_DBMS` named path is defined as `C:\DBMS` and you have the path `C:\DBMS\syase1550.xdb` in your model, then the path will be saved as `%DBMS%\syase1550.xdb` to allow the model to be opened by another user who stores their DBMS definition files elsewhere.

To view the existing named paths, select *Tools* > *General Options*, and then click the *Named Paths* category.

To add a new named path, click the *Add a Row* tool and enter a name and a path for the named path. The multi-folder named paths are edited by clicking the ellipsis button in the Path column.

> **i Note**
>
> An administrator may push named paths to users in a general preferences profile (see Controlling Defaults for Options and Preferences [page 567]).

The following paths are initialized by default:

Name	Path
[XSL]	Multi-folder path for storing XML language definition files. Replaces `_XMLLANG`, which is deprecated.
[XRL]	Multi-folder path for storing report language files. Replaces `_RTPLANG`, which is deprecated.
[XPL]	Multi-folder path for storing process language definition files. Replaces `_PRCSLANG`, which is deprecated.
[XOL]	Multi-folder path for storing object language definition files. Replaces `_OBJLANG`, which is deprecated.
[XEM]	Multi-folder path for storing extension files. Replaces `_XEM`, which is deprecated.
[XDB]	Multi-folder path for storing DBMS definition files. Replaces `_DBMS`, which is deprecated.
[RUL]	Multi-folder path for storing impact and lineage rule set files (see Impact and Lineage Analysis [page 489]). Replaces `_RULESET`, which is deprecated.
_EXAMPLES	Folder where demo examples are stored.
_HOME	PowerDesigner installation folder. This path facilitates the installation of your add-ins without having to modify your XEM or your XML or ActiveX add-ins commands. For example: `%_HOME%\add-ins\SpellChecker\SpellCheckAddIn.dll`
_JSF	Folder where JSF libraries are stored.
_LIBRARY	Folder where object libraries are stored.
_MODELCATEGORY	Folder where model category files are stored (see Model Creation Options [page 401]).
_SHARED	Folder where the glossary and library model files are stored (see The Glossary [page 38] and The Library [page 42]).
_SMARTWINFORM	Folder where smart winform files are stored.
_WINFORM	Folder where winform files are stored.
_DBMS	[deprecated] Folder where DBMS definition files are stored. This path is now included in `[XDB]`.
_OBJLANG	[deprecated] Folder where object language definition files are stored. This path is now included in `[XOL]`.
_PRCSLANG	[deprecated] Folder where process language definition files are stored. This path is now included in `[XPL]`.
_RTPLANG	[deprecated] Folder where report language definition files are stored. This path is now included in `[XRL]`.

Name	Path
_RULESET	[deprecated] Folder where impact and lineage rule set files are stored. This path is now included in `[RUL]`.
_XEM	[deprecated] Folder where extension files are stored. This path is now included in `[XEM]`.
_XMLLANG	[deprecated] Folder where XML language definition files are stored. This path is now included in `[XSL]`.

> **i Note**
>
> You should not modify the name of a predefined named path. If you do so, the predefined named path is preserved in the list and a new path is created with the modified name and value. If you want to cancel a change to a predefined named path, simply delete it and close the General Options dialog. When you reopen the dialog the path is recreated with its default value.

If you open a file containing a named path not defined on your machine, a dialog will open, offering you the following options:

- Ignore the named path and keep the file name unresolved - The link between files is broken and you may have update problems.
- Define a new named path ... - Allows you to define the path on your machine.
- Browse another existing variable to replace with – Allows you to resolve the path using one of your existing names paths.
- Directly browse the target file – Allows you to browse for another target file. The selected file will replace the file you were trying to open

1.10.2.5 Interface Fonts

To review and modify the fonts used in the general user interface, code and rtf editors, and matrices, select *Tools* > *General Options*, and click the *Fonts* category in the left pane.

Select the interface element for which you want to modify the font in the *Item* list, and then select a *Font*, *Font style*, and *Size* for it.

1.10.2.6 Repository Options

To review and modify options for connecting to the repository, controlling the Repository Browser tab, and setting various defaults, select *Tools* > *General Options*, and click the *Repository* category in the left pane.

Connection

The following options are available in the *Connection* group box:

Option	Description
Auto-connect	Automatically connects to the last connected repository when you start a PowerDesigner session (see Connecting Automatically to the Default Repository [page 21].
Apply profiles	Re-applies all UI preferences profiles and general preferences profiles that are associated with the user (either directly, or by membership of groups) at each connection to the repository (see Using Profiles to Control the PowerDesigner Interface [page 561]). This option is selected by default and may cause local customizations to your interface, options, and preferences to be overwritten.
Default bytes per char	Specifies the default number of bytes per character used by the repository database (by default, 2). This value is used to initialize the Bytes per char parameter in the Direct Repository Definition dialog (see Setting up PowerDesigner for Direct Access [page 15]).

Browser

The following options are available in the *Browser* group box:

Option	Description
Auto-refresh	Refreshes the Repository Browser at the interval specified in the Refresh every box.
Display	Specifies whether the name or code of items is displayed in the Repository Browser.
Display model contents	Enables the display of model diagrams and objects in the Repository Browser.
Display version numbers	Enables the display of version numbers for items in the Repository Browser.
Display status icons in Local Browser	Enables the display of status icons (which track whether items are locked and if changes have been made locally and/or in the repository since check out) in the Local Browser (see The Browser [page 37]).

Check Out

The following options in the *Check Out* group box control the default selection of options in the Check Out dialogs (see Checkout Parameters [page 270]):

Option	Description
Merge document	Specifies that the *Merge document* option is selected by default.
Add to workspace	Specifies that the *Add to workspace* option is selected by default.
Check out dependencies	Specifies that the *Check out dependencies* option is selected by default.
Automatic resources update	Enables the automatic checking out and updating of shared resource files. **i Note** This option controls a legacy feature, which has been superceded by the library (see Sharing Resource Files via the Library [page 551]).

Check In

The following options in the *Check In* group box control the default selection of options in the Check In dialogs (see Check-In Parameters [page 252]):

Option	Description
Batch update	Specifies that the *Batch update* option is selected by default.
Freeze after check in	Specifies that the *Freeze after check in* option is selected by default.
Save diagrams for PowerDesigner Portal	Specifies that the *Save diagrams for PowerDesigner Portal* option is selected by default.

Local Cache

The following options in the *Local cache* group box control the caching of models checked into and out of the repository on your local workstation. The cache, which is enabled by default, saves a local copy of models you

check out in the cache directory, and can speed up check ins and check outs when you are primarily working with models that are rarely modified by other users. The following options are available:

Option	Description
Path	Specifies the location of the cache.
Maximum size (MB)	By default, the maximum size is set to 50MB. Click the *Clear Cache* button to empty the contents of the cache.

1.10.2.7 Security Options

Certain PowerDesigner features depend on executing scripts, which can be embedded in model files or located in resource files attached to models. To reduce the risk posed by malicious scripts, you can instruct PowerDesigner to notify you when an untrusted script is run, to prompt you for permission to run an untrusted script, or to block such scripts from running altogether.

To enable the monitoring of scripts, select one of the following settings in the *Security level* option:

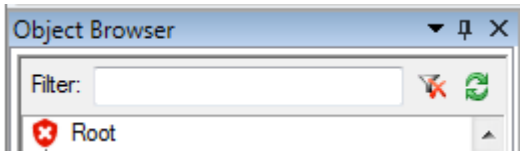
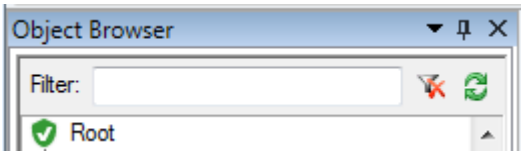
- *No Restrictions* - Allows scripts in untrusted files to run without notification or warning.
- *Run and Notify* - Allows scripts in untrusted files to run and displays a notification of their execution in the *Output* window.
- *Prompt to Run* - Requires your confirmation before scripts in untrusted files can run. When you approve running the script, its file is added to your whitelist of trusted files, and you will not be prompted again for this file.

  > **i Note**
  >
  > To remove all files that have been trusted to date from your whitelist, click the *Reset Trusted File Whitelist* button.

- *Block* - Prevents any scripts in untrusted files from running.

PowerDesigner's security policy is based on two concepts:

- Secure locations - Only folders for which write access is restricted to your own user account and administrator user accounts are considered secure.
  When you enable the *Security level* option, PowerDesigner will review its installation. If any of your named paths (see Named Paths [page 393]) point to folders with more permissive write permissions, then your PowerDesigner installation is deemed to be insecure, your local workspace root displays a red shield, and errors and warnings are written to the *Output* window.
  To secure your installation, review the errors and warnings and change the location of each of the insecure named path folders (or constrain their permissions) appropriately. If all of your named paths point to secure folders, your workspace root displays a green shield:

Insecure Environment	Secure Environment
	

- Trusted files - Files can be trusted:
  - By default -If they are located in folders for which write access is restricted to administrator user accounts only. Thus, for example, if you install PowerDesigner, as recommended, under the Program Files folder then the resource files delivered with it will be trusted by default.
  - By adding them to your trusted files whitelist when you are prompted before running them using the *Prompt to Run* setting.

> **i Note**
>
> Folders to which your own user account has write access cannot be trusted by default, but files contained in them can be added to your trusted file whitelist.

1.10.2.8 Add-Ins

To review and enable add-ins to provide additional features and services in PowerDesigner, select *Tools* > *General Options*, and click the *Add-Ins* category in the left pane.

Context

Add-ins installed with PowerDesigner are called system add-ins, and are declared in a `Local Machine` registry key. The following system add-ins are installed by default:

- *Microsoft SQL Server 2005 Analysis Services* - Generates and reverse engineers multidimensional objects for Microsoft SQL Server 2005 (see *Data Architecture* > *DBMS Definition Reference* > *Microsoft SQL Server*). Require the client for the appropriate version of SQL Server.
- *Microsoft Word Import* - Imports and exports MS Word documents to and from requirements models (see *Requirements Management* > *Working with Word Documents and RQMs*).
- *SAP BusinessObjects* - Generates universes from physical data models (see *Data Architecture* > *Building Data Models* > *Generating and Reverse-Engineering Databases* > *Generating an SAP BusinessObjects Universe*). Requires BusinessObjects 4.0 or higher.
- *SAP ERP* - Imports SAP Business Suite Data Dictionary and allows its generation it to HANA (see *Data Architecture* > *DBMS Definition Reference* > *SAP Business Suite*)
- *SAP HANA* - Generates database schemas using the HANA API (see *Data Architecture* > *DBMS Definition Reference* > *SAP HANA*).
- *SAP Solution Manager* - Generates and reverse engineers objects to a Solution Manager server (see *Business Process Architecture* > *SAP Solution Manager*).
- *Spell Checker* - Uses the MS Word spell checker to check PowerDesigner object names, codes, comments, descriptions, and annotations (see Spell Checking [page 167]). Requires Word 2000 or higher.

- *SAP IQ Reference Architecture* - Analyzes the architecture required to deploy a SAP® IQ data warehouse solution suitable for your workload (see *Enterprise Architecture* > *SAP IQ Reference Architecture Model*).
- *XML Validation* - Checks that an XML document conforms to the schema in the XML model (see *XML Architecture*). Requires MSXML 4.0.

> **i** Note
>
> If you modify system add-in properties, you can click the *Reset Values For System Add-In* tool to recover the parameters registered in the registry.

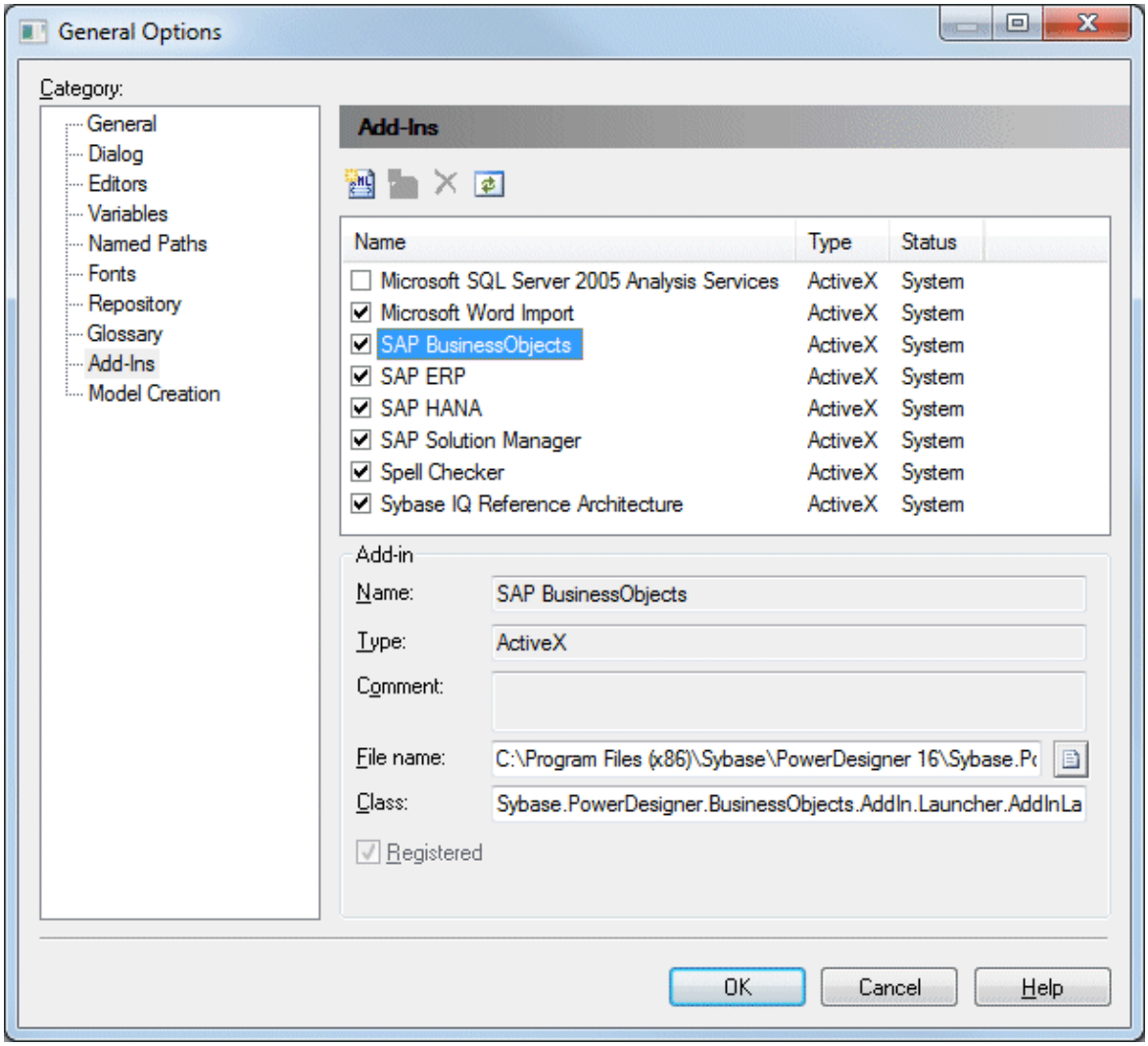
You enable an add-in by selecting its checkbox.

You can create your own XML or ActiveX user add-ins, which are declared in a `Current User` registry key:

Procedure

1. Click one of the following tools:
   - *Add New ActiveX Add-In* - An ActiveX add-in implements an interface that defines methods, which are invoked by PowerDesigner in order to dialog with menus and execute commands that are defined by the ActiveX.
   - *Add New XML Add-In* - An XML add-in defines a menu structure for calling scripts or executables from within PowerDesigner.

   For information about developing these add-ins, see *Customizing and Extending PowerDesigner* > *Scripting PowerDesigner* > *OLE Automation and Add-Ins*
2. Enter the name of your add-in and click the *Select File* tool to browse to the `.xml`, `.dll`, or `.exe` file.
3. [ActiveX only] Enter the name of the ActiveX implementation class in the *Class* field.



4. Click *OK*.

   If your add-in has been correctly defined, it will immediately be available under the appropriate PowerDesigner menu.

1.10.2.9 Model Creation Options

To review and modify options for creating models in the New Model dialog, select *Tools* > *General Options*, and click the *Model Creation* category in the left pane.

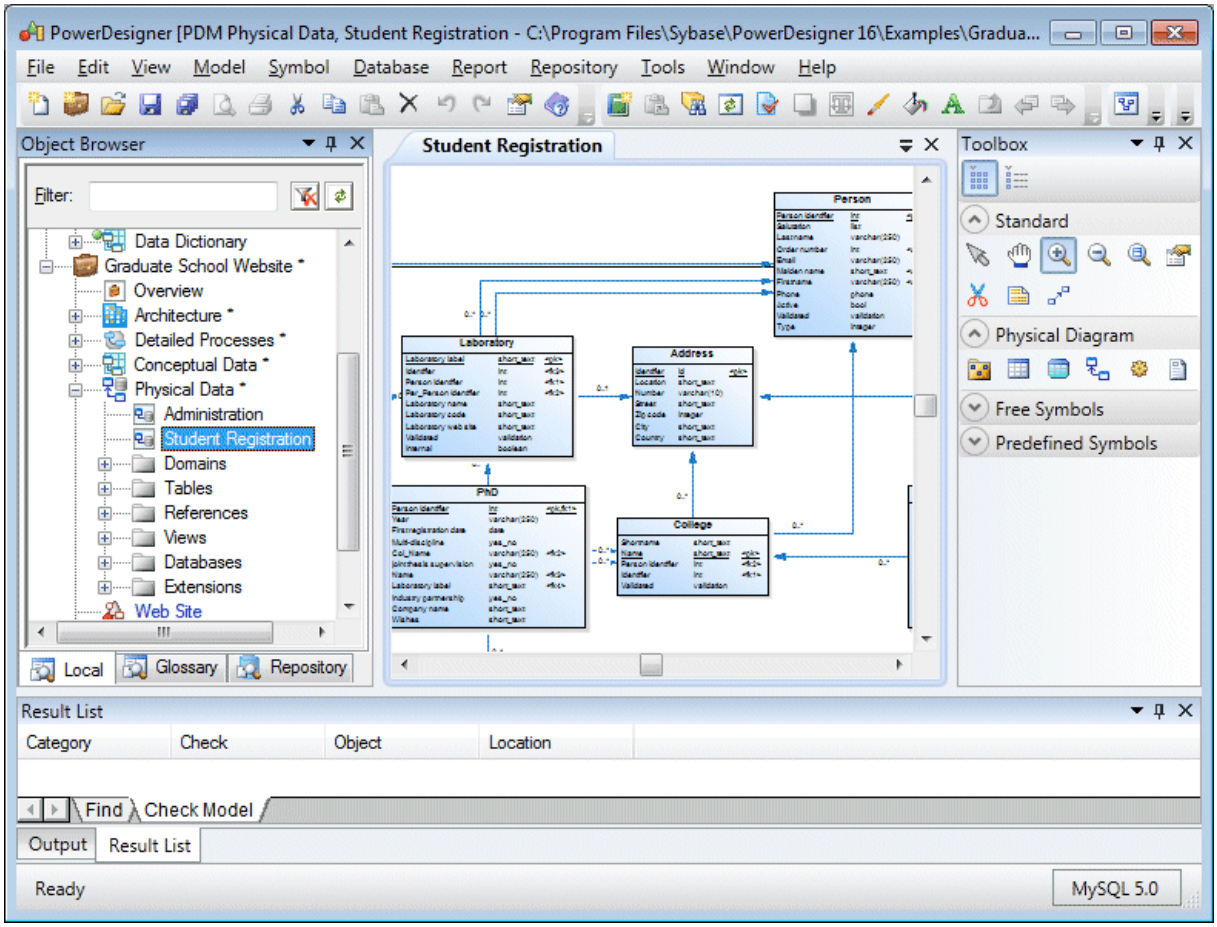
The following options are available:

Option	Description
Enable Categories	Enables the display of the Categories button in the New Model dialog, which lets you create models from predefined templates. Select the model category set to use in the Default category set list. If no valid model category set is selected then the Categories button will not be available. Click the Select Path tool to the right of this field to browse to another folder containing model category sets or click the Properties tool to open the currently-selected set in the Resource Editor. For more information about category sets, which are often developed by an administrator, see Customizing the New Model Dialog [page 554].
Enable Model Types	Enables the display of the Model types button in the New Model dialog, which lets you create models from the classic list of model types.
Enable model template files	Enables the display of the Template files button in the New Model dialog, which lets you create models from model templates. If no valid template directory is selected then the Templates button will not be available. Click the Select Path tool to the right of this field to browse to another folder containing model templates. For more information about model templates, see Model Templates [page 14].

For information about using the New Model dialog, see Creating a Model [page 12].

1.10.3 Organizing Views

The PowerDesigner window contains the Browser, Toolbox, Canvas, Result List, and Output sub-windows or views, which you can move, group, split, hide, or auto-hide to suit your workflow.



To move a view, click on its title bar or upper or lower tab and drag and drop it onto the docking control that appears as you hover over a view. You can perform the following kinds of movement in this way:

- To group the view with another view - drop it on the central dock selector:

  The dropped view is stacked on the other view, and each has a tab at the bottom of the group to select it.
- To place the view next to, above, or below a view - drop it on the left, right, top, or bottom dock selector. You can use this to divide the canvas and show multiple diagrams together.
- To place the view next to, above, or below a view, and have it extend the full height or breadth of the PowerDesigner window - drop it on the separate left, right, top, or bottom dock selectors that appear near the extremities of the window:

- To promote a view to an independent window - drop it anywhere except on a dock selector.

> **i Note**
>
> The Canvas is a special view that displays diagrams as tabbed documents.You can drag other views onto the Canvas, but you cannot drag Canvas diagrams elsewhere in the PowerDesigner window (though you can promote them to independent windows). Right-click a diagram tab to access a menu that lets you close the diagram, close every diagram except this one, or split the canvas horizontally or vertically.

All views except the canvas have a menu available on their title bar to:

- *Float* - promote the view to an independent window.
- *Show* - disable auto-hiding (or click the sideways pin to pin the view open).
- *Dock as Tabbed Document* - add the view as a tab to the Canvas.
- *Auto Hide* - hide the view as a small tab on the side of the screen, that re-expands when you hover over it (or click the vertical pin to unpin the view).
- *Hide* - remove the view from your window. To show a hidden view, select *View* > `<View Name>`.

> **i Note**
>
> The Browser is a single view with up to three sub-views (Local, Glossary, and Repository). These sub-views cannot be moved independently.

1.10.4 Organizing Toolbars

The PowerDesigner window provides various toolbars, which you can show, hide, organize, and customize to suit your workflow.

- To move a toolbar in the toolbar tray, click and hold on the four dots on the left edge of the toolbar and drag it to the desired position.
- To show a hidden toolbar or hide a visible toolbar - right-click the toolbar tray and select *Toolbars* and check or uncheck the toolbar.
- To access tools on a toolbar that is not fully visible due to lack of space - click on the down arrow on the right edge of the toolbar.
- To lock all toolbars to protect them from accidental movements - right-click the toolbar tray and select *Toolbars* > *Lock Toolbars*.
- To add tools to or remove them from a toolbar, or to create new toolbars - right-click the toolbar tray and select *Toolbars* > *Customize Menus and Tools* (see Customizing Menus, Toolbars, and Toolboxes [page 405]).

> **Note**
>
> The Toolbox (see Creating Objects from the Toolbox [page 92]) is not a toolbar, and you cannot add Toolbox tools (such as those for creating a PDM table or OOM class) to a toolbar, nor add toolbar tools to the Toolbox.

1.10.5 Customizing Menus, Toolbars, and Toolboxes

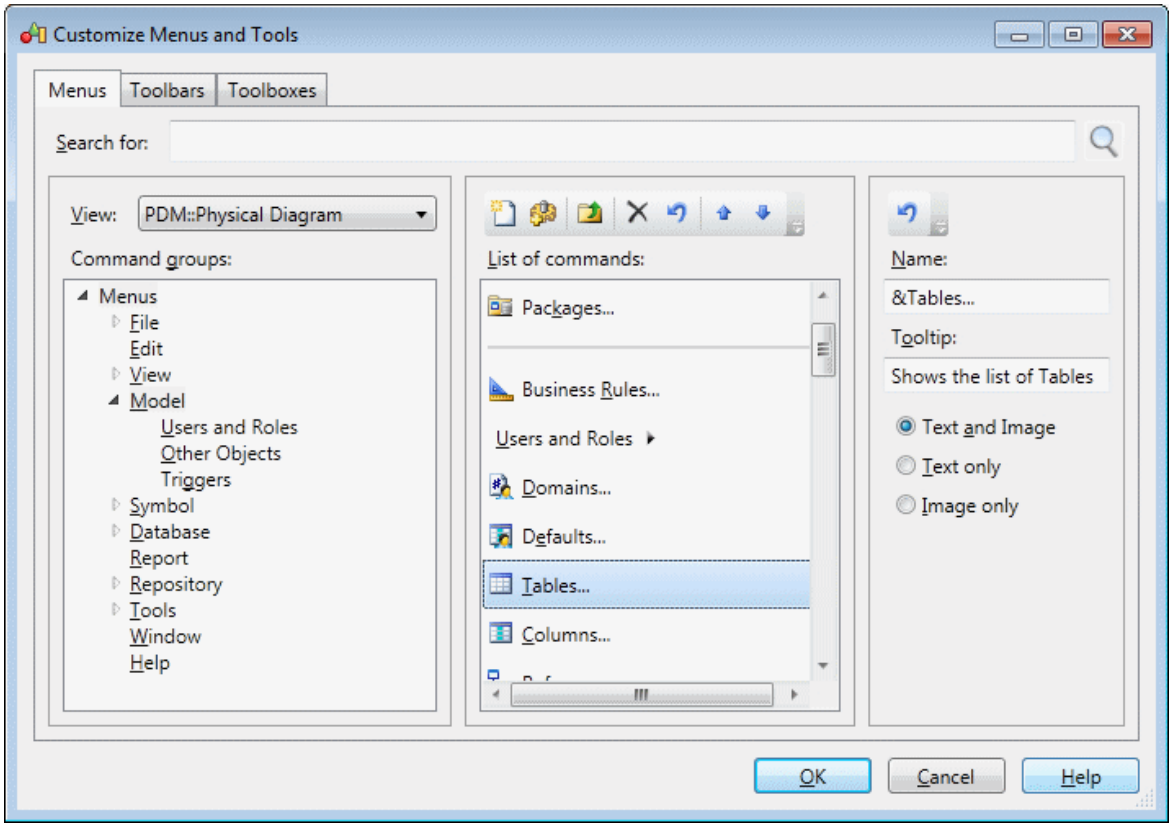
To customize PowerDesigner menus, toolbars, and toolboxes, select *Tools* > *Customize Menus and Tools*.

Context

> **Note**
>
> An administrator can prepare customized UI profiles, which are made available to users on their first connection to the repository (see Controlling Defaults for Menus, Toolbars, and Toolboxes [page 566]).

Procedure

1. Select *Tools* > *Customize Menus and Tools*. The dialog opens to the *Menus* tab, initialized to your current view (here, the PDM Physical Diagram):



> **i Note**
>
> Search for any command available in the present context by entering it in the *Search for* field and clicking *Find*. Methods and menus defined in resource files and custom commands are not available for selection and cannot be moved in this dialog (see *Customizing and Extending PowerDesigner > Extension Files > Menus (Profile)*).

2. [optional] Select a different view to customize. The following views are available:
   - **`No View`** - when PowerDesigner is running but no model is open and selected.
   - **`<Type> Diagram`** - each diagram is listed separately.
   - **`All Views`** - provides a union of all the menus available in all possible views.

> **i Note**
>
> You can make changes in multiple different views. All your UI preferences are saved in a single resource file.

3. Navigate to and select a command group (menu or submenu) to modify in the tree in the left panel. The commands that are available on the selected menu, submenu, toolbar, or toolbox are listed in the middle panel.

   The middle panel shows the contents of only one level at a time. To see the contents of a submenu, double-click it to descend into its commands. Click the *Go Up One Level* tool to return to the parent menu. For example, to see the contents of the *Repository* > *Administration* submenu, expand the *Repository* node in the left panel, and select the *Administration* node, or click the *Repository* node in the left panel, and double-click the *Administration* node in the middle panel.

> **Note**
>
> The special commands in the format **<<Command>>** such as **<Recent Files>** and **<New Diagrams>** are placeholders for points where the menu dynamically calculates commands to insert. The **<Stereotypes used as Metaclass>** placeholder, which is present by default in the *Model* menu, instructs PowerDesigner to insert a command to open a list of objects for each type of new object defined in a model definition or extension file (see PowerDesigner Resource Files [page 408]). If you remove this command from this menu, then all such object lists will be hidden. If you want to display such object lists under another menu, then you must add both the parent metaclass from which they are derived and the **<Stereotypes used as Metaclass>** placeholder.

4. Use the tools in the middle panel to add, remove, reorder, or otherwise modify the commands of the selected command group:

Tool	Description
	Create New Menu - Creates a new sub-menu to add to the command group. The sub-menu acts as a container for commands.
	Add Commands - Opens a selection dialog to add a command to the selected command group. The Add Commands dialog has a *Pick Command* tab to let you select available commands from a tree and a *Filter Commands* tab to let you filter the list of available commands by entering a string. When you have located the command to add, select it and click *OK*.
	Go Up One Level - Displays the commands of the parent menu of the currently selected menu.
	Remove - Deletes the selected command from the command group.
	Reset List - Undoes any changes to the command group.
	Move Up - Moves the selected command up one place in the list.
	Move Up - Moves the selected command down one place in the list.

5. [optional] Select a command in the middle pane and use the fields in the right pane to change its name, tooltip, and display method (text and/or image).

> **Note**
>
> Inserting an ampersand sign in the *Name* field before a letter enables its use as an `Alt` key menu shortcut. For example, the *File* menu has, by default, its name defined as **`&File`** so that users can press `Alt` + `F` to activate this menu.

6. [optional] Select the *Toolbars* (see Organizing Toolbars [page 404]) or *Toolboxes* (see Creating Objects from the Toolbox [page 92]) tab to customize these UI elements.
7. Click *OK* to save your changes and apply them to your environment.

1.10.6 PowerDesigner Resource Files

The SAP® PowerDesigner® modeling environment is powered by XML-format resource files, which define the objects available in each model along with the methods for generating and reverse-engineering them. You can view, copy, and edit the provided resource files and create your own in order to customize and extend the behavior of the environment.

The following types of resource files, based on or extending the PowerDesigner public metamodel are provided:

- Definition file: customize the metamodel to define the objects available for a specific DBMS or language:
  - DBMS definition files (.xdb) - define a specific DBMS in the PDM (see *Customizing and Extending PowerDesigner > DBMS Definition Files*).
  - Process, object, and XML language definition files (.xpl, .xol, and .xsl) – define a specific language in the BPM, OOM, or XSM (see *Customizing and Extending PowerDesigner > Object, Process, and XML Language Definition Files*).
- Extension files (.xem) – extend the standard definitions of target languages to, for example, specify a persistence framework or server in an OOM. You can create or attach one or more XEMs to a model (see Extension Files [page 411]).
- Report templates (.rtp) - specify the structure of a report. Editable within the Report Template Editor (see Report Templates [page 357]).
- Report language files (.xrl) – translate the headings and other standard text in a report (see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Files*).
- Impact and lineage analysis rule sets (.rul) - specify the rules defined for generating impact and lineage analyses (see Analysis Rule Sets [page 503]).
- Object permission profiles (.ppf) - customize the PowerDesigner interface to hide models, objects, and properties (see Controlling the Availability of Models, Objects, and Properties [page 562]).
- User profiles (.upf) - store preferences for model options, general options, display preferences, etc (see User Profiles [page 381]).
- Model category sets (.mcc) - customize the New Model dialog to guide model creation (see Customizing the New Model Dialog [page 554]).
- Conversion tables (.csv) - define conversions between the name and code of an object (see Conversion Tables [page 161]).

You can review all the available resource files from the lists of resource files, available by selecting *Tools* > *Resources* > `<type>`.

The following tools are available on each resource file list:

Tool	Description
	Properties - Opens the resource file in the Resource Editor.
	New - Creates a new resource file using an existing file as a model (see Creating and Copying Resource Files [page 410]).
	Save - Saves the selected resource file.
	Save All - Saves all the resource files in the list.
	Path - Specifies the directories that PowerDesigenr should search to populate the list (see Specifying Directories to Search for Resource Files [page 410]).
	Compare - Selects two resource files for comparison.
	Merge - Selects two resource files for merging.
	Check In - [if the repository is installed] Checks the selected resource file into the repository. For information about storing your resource files in the repository, see Sharing Resource Files via the Library [page 551].
	Update from Repository - [if the repository is installed] Checks out a version of the selected file from the repository to your local machine.
	Compare with Repository - [if the repository is installed] Compares the selected file with a resource file stored in the repository.

1.10.6.1 Sharing and Embedding Resource Files

Resource files can be shared and referenced by multiple models or copied to and embedded in a single model. Any modifications that you make to a shared resource are available to all models using the resource, while modifications to an embedded resource are available only to the model in which it is embedded. Embedded resource files are saved as part of their model and not as a separate file.

> **i** Note
>
> You should never modify the original extensions shipped with PowerDesigner. To create a copy of the file to modify, open the List of Extensions, click the *New* tool, specify a name for the new file, and then select the .xem that you want to modify in the *Copy from* field.

The *File Name* field displays the location of the resource file you are modifying is defined. This field is empty if the resource file is embedded.

1.10.6.2 Creating and Copying Resource Files

You can create a new resource file in the appropriate resource file list. To create a copy of an existing resource file, select it in the *Copy from* field of the *New...* dialog.

Context

> ⚠ Caution
>
> Since each resource file has a unique id, you should only copy resource files within PowerDesigner, and not in Windows Explorer.

Procedure

1. Select *Tools* > *Resources* > `<Type>` to open the appropriate resource file list.
2. Click the *New* tool, enter a name for the new file and select an existing file to copy. Select the **<Default template>** item to create a minimally completed resource file.
3. Click *OK* to create the new resource file, and then specify a filename and click *Save* to open it in the Resource Editor.

   > i Note
   >
   > You can create an extension file directly in your model from the List of Extensions. For more information, see Creating an Extension File [page 412].

1.10.6.3 Specifying Directories to Search for Resource Files

Use the *Path* tool in the resource list toolbar to specify directories to search to populate the list. If you plan to modify the standard resource files or create your own, you must store these files in a directory outside the PowerDesigner installation directory.

Context

By default, only the directory inside the `Program Files` folder containing the standard resource files appears in the list, but PowerDesigner does not allow you to save modifications there, and will propose an alternative location if you try to do so, adding the selected directory to the list. You can add additional directories as necessary.

> **Note**
>
> If you have created or modified resource files inside `Program Files` before 16.5, when this rule was introduced, your files may no longer be available as Windows Vista or Windows 7 actually store them in a virtual mirror at, for example, `C:\Users\<username>\AppData\Local\VirtualStore\Program Files\Sybase\PowerDesigner 16\Resource Files\DBMS`. To restore these files to your lists, optionally move them to a more convenient path, and add their location to your list using the *Path* tool.

The first directory in the list is the default location, which is proposed whenever you save a file. The root of the library belonging to your most recent repository connection is searched recursively before the directories in the list (see Sharing Resource Files via the Library [page 551]), .

> **Note**
>
> In rare cases, when seeking resource files to resolve broken references in models, the directories in the list are scanned in order, and the first matching instance of the required resource is used.

1.10.6.4 Extension Files

Extensions files (`*.xem`) allow you to customize and extend the PowerDesigner metamodel to support your exact modeling needs. You can define additional properties for existing objects or specify entirely new types of objects, modify the PowerDesigner interface (reorganizing and adding property sheet tabs, Toolbox tools and menu items), and define additional generation targets and options.

Extension files have an `.xem` extension and are located in `<install_dir>/Resource Files/Extended Model Definitions`.

Lists of extension files by model type are available by selecting *Tools* > *Resources* > *Extensions* > `<model type>`. For information about the tools available in resource file lists, see PowerDesigner Resource Files [page 408].

> **Note**
>
> Extensions, such as the Excel Import extension, which can be attached to any model type, are available in the list at *Tools* > *Resources* > *Extensions* > *All Model Types*.

Each extension file contains two first-level categories:

- Generation - used to develop or complement the default PowerDesigner object generation (for BPM, OOM, and XSM models) or for separate generation. For more information, see *Customizing and Extending PowerDesigner > Object, Process, and XML Language Definition Files > Generation Category* .
- Profile - used for extending the metaclasses in the PowerDesigner metamodel. You can:
  - Create or sub-classify new kinds of objects:
    - Metaclasses – drawn from the metamodel as a basis for extension.
    - Stereotypes [for metaclasses and stereotypes only] – sub-classify metaclasses by stereotype.
    - Criteria – sub-classify metaclasses by evaluating conditions.
  - Add new properties and collections to objects and display them:
    - Extended attributes – to add metadata.

        - Extended collections and compositions – to enable manual linking between objects.
        - Calculated collections – to automate linking between objects.
        - Dependency matrices – to show dependencies between two types of objects.
        - Forms – to modify property sheets and add custom dialogs.
        - Custom symbols – to change the appearance of objects in diagrams.
    - Add constraints and validation rules to objects:
        - Custom checks – to test the validity of your models on demand
        - Event handlers – to perform validation or invoke methods automatically.
    - Execute commands on objects:
        - Methods – VBScripts to be invoked by menus or form buttons.
        - Menus [for metaclasses and stereotypes only] – to add commands to PowerDesigner menus.
    - Generate objects in new ways:
        - Templates – to extract text from object properties.
        - Generated Files - to assemble templates for preview and generation of files
        - Transformations – to automate changes to objects at generation or on demand.
    - Map correspondences between different metamodels:
        - Object generations - to define mappings between different modules in the PowerDesigner metamodel for model-to-model generation.
        - XML imports - to define mappings between an XML schema and a PowerDesigner module to import XML files as models.

> **i Note**
>
> Since you can attach several resource files to a model (for example, a target language and one or more extension files) you can create conflicts, where multiple extensions with identical names (for example, two different stereotype definitions) are defined on the same metaclass in separate resource files. In case of such conflicts, the extension file extension usually prevails. When two XEMs are in conflict, priority is given to the one highest in the List of Extensions.

1.10.6.4.1 Creating an Extension File

You can create an extension file directly in your model or from the appropriate list of extension files.

Context

> **i Note**
>
> Extensions, such as the Excel Import extension, which can be attached to any model type, can only be created from the *All Model Types* extension list. For information about creating an extension file from a list of extension files, see Creating and Copying Resource Files [page 410].

Procedure

1. Open your model, and then select *Model* > *Extensions* to open the List of Extensions.
2. Click the *Add a Row* tool and enter a name for the new extension file.
3. Click the *Properties* tool to open the new extension file in the Resource Editor, and create any appropriate extensions.
4. When you have finished, click *OK* to save your changes and return to the List of Extensions.

   The new XEM is initially embedded in your model, and cannot be shared with any other model. For information about exporting your extensions and making them available for sharing, see Exporting an Embedded Extension File for Sharing [page 414].

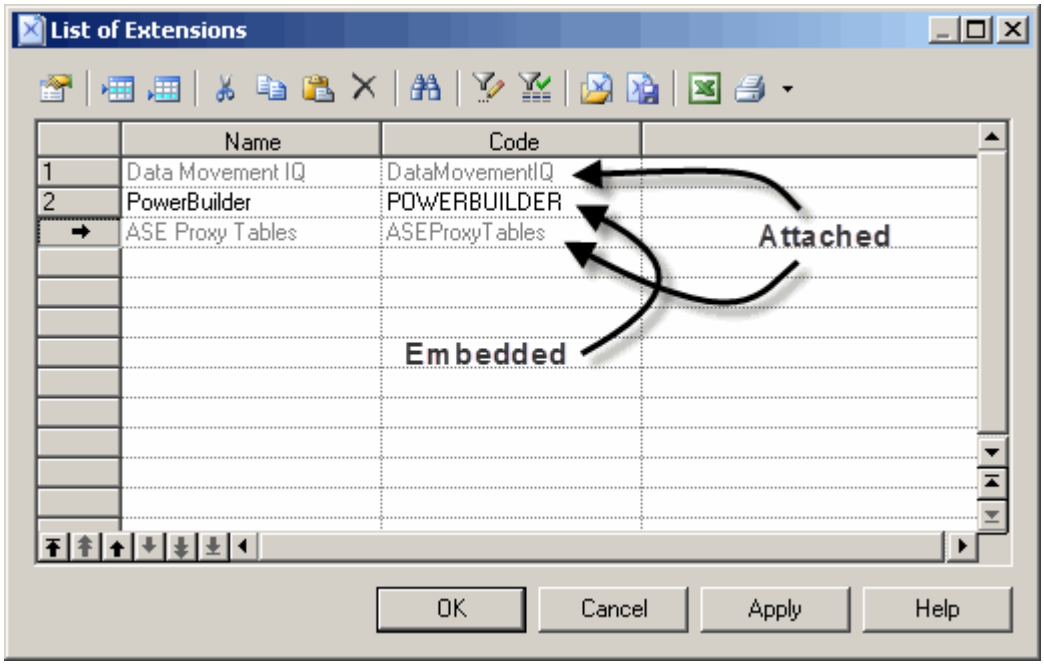
1.10.6.4.2 Attaching Extensions to a Model

Extensions can be stored in `*.xem` files that you can attach to one or more models. You can attach one or more extension files to a model at creation time by clicking the *Select Extensions* button on the New Model dialog. You can subsequently attach extension files to your model at any time from the List of Extensions.

Procedure

1. Select *Model* > *Extensions* to open the List of Extensions, which contains extensions attached to the model.
2. Click the *Attach an Extension* tool to open the Select Extensions dialog.
3. Review the different sorts of extensions available by clicking the sub-tabs and select one or more to attach to your model.

   By default, PowerDesigner creates a link in the model to the specified file. To copy the contents of the extension file and save it in your model file, click the *Embed Resource in Model* button in the toolbar. Embedding a file in this way enables you to make changes specific to your model without affecting any other models that reference the shared resource.
4. Click *OK* to return to the List of Extensions.



Extension files listed in grey are attached to the model, while those in black are embedded in the model.

> **i** Note
>
> If you embed an extension file in the model, the name and code of the extension may be modified in order to make it respect the naming conventions of the Other Objects category in the Model Options dialog.

1.10.6.4.3 Exporting an Embedded Extension File for Sharing

If you export an XEM created in a model, it becomes available in the List of Extensions, and can be shared with other models. When you export an XEM, the original remains embedded in the model.

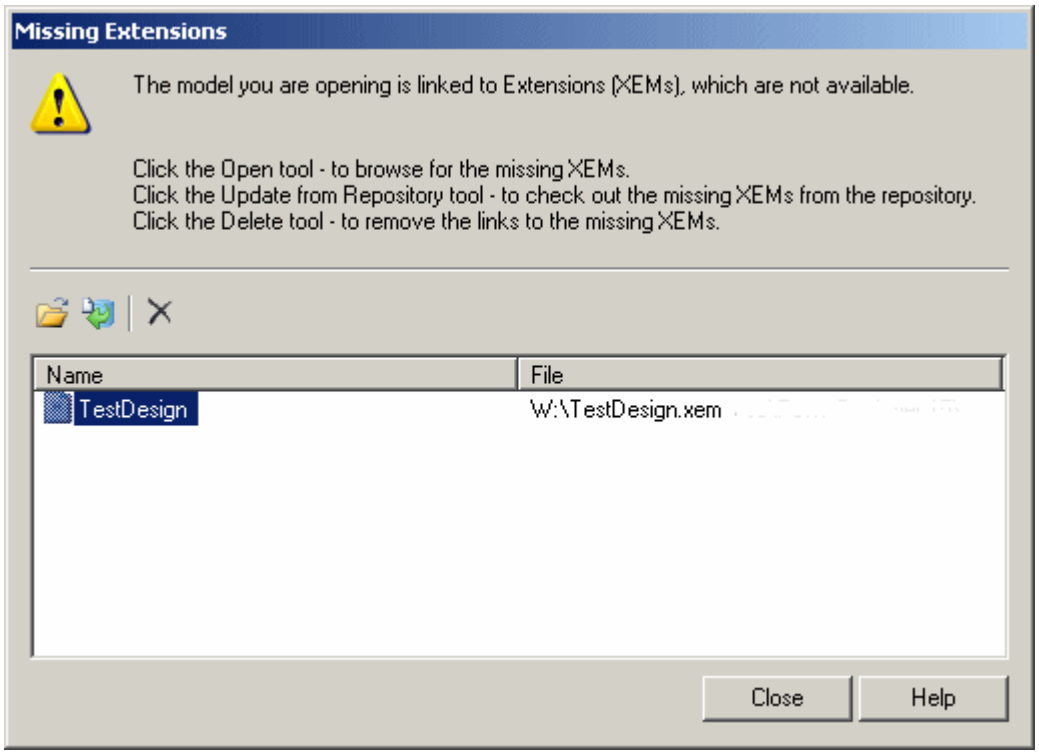
Procedure

1. Select *Model* > *Extensions* to open the List of Extensions, which contains extensions attached to the model.
2. Select an embedded extension in the list and click the *Export an Extension* tool.
3. Enter a name and select a directory to which to save the extension file and click *Save*.

> **i** Note
>
> For the extension to be available for attaching to other models, you must save it to a directory that is listed by the *Path* tool in the appropriate extension list (see Specifying Directories to Search for Resource Files [page 410]).

1.10.6.4.4 Reconnecting to Missing Extensions

If you rename, move or delete extension files that are attached to a model, you may lose the links to them in your model. When you next open a model sharing the affected extensions, PowerDesigner will try to help you reconstruct the broken links.



The following tools are available in this dialog:

Tool	Description
	Open – Lets you browse to, and re-attach, the missing XEMs.
	Update from Repository – Lets you check out the missing XEMs from the repository.
	Delete – Lets you remove the links between the model and the missing XEMs.

1.10.7 Shortcut Keys

PowerDesigner provides many shortcut keys to accelerate your workflow.

Toolbox Shortcuts

Shortcut	Action
Ctrl + A (or double-click *Pointer* tool)	Select all symbols in diagram.
Double-click *Delete* tool	Delete selected symbols (with confirmation).
Double-click *Grabber* tool	Zoom to whole diagram.
Double-click any object tool	Select all symbols of this type.
Shift+Double-click any object tool	Retain current selection and add all symbols of this type.
Right-click	Release tool and select pointer.

Browser Shortcuts

Shortcut	Action
* (number pad)	Expand all nodes.
+ (number pad)	Expand subnodes.
- (number pad)	Collapse subnodes.
F2	Rename selected item.
Ctrl + Shift + F	Find selected item in Diagram.

Diagram Shortcuts

Shortcut	Action
Del	Delete object with confirmation dialog.
Shift + Del	Delete object directly, without confirmation.
Alt + Enter	Open object property sheet.
Ctrl + K	Paste as shortcut.
Ctrl + H	[on selected link] Force link to horizontal.
Ctrl + L	[on selected link] Force link to vertical.

Shortcut	Action
Ctrl + Page Up	Bring symbol to front.
Ctrl + Page Down	Send symbol to back
Ctrl + Home	Align selection on vertical axis.
Ctrl + End	Align selection on horizontal axis.
Ctrl + T	Open Symbol Format dialog.
Ctrl + J	Auto-resize the symbol to display all the text it contains (up to the value specified for word wrapping in the diagram).
Ctrl while dragging a symbol handle	Resize object while keeping same center point.
Ctrl + Shift while dragging a symbol handle	Resize object while keeping same center point and proportions.
F2 (or Ctrl+Right-click on Name)	Rename object.
Ctrl Double-click (or Ctrl + + (plus sign)	[on package or other composite symbol] Open subdiagram.
Ctrl + U	Go to parent diagram.

Drag and Drop Shortcuts

Shortcut	Action
Shift+Drag and drop	Move object.
Ctrl+Drag and drop	Copy object.
Shift + Ctrl+Drag and drop	Create shortcut of object.
Alt+Drag and drop	Create replica of object.
Right-click+Drag and drop	Open menu to choose an action.

Window and General Shortcuts

Shortcut	Action
F4	Check model.
F5	Zoom to actual size.
F6	Zoom in.
F7	Zoom out.
F8	Zoom to whole diagram.
F9	Previous view.
F10	Display used pages.

Shortcut	Action
Shift + F5	Redisplay.
Shift + F9	Next view.
Shift +Click item	Select consecutive items.
Alt + 0	Show/hide Browser.
Alt + 1	Show/hide Output view.
Alt + 2	Show/hide Result List view.
Alt + 3	Show/hide Welcome page.
Alt + F6	Cycle between active windows.
Ctrl + Tab	Cycle between property sheet tabs.
Ctrl + Page Up / Ctrl + Page Down	Cycle between property sheet tab subtabs.
Ctrl + F6 / Ctrl + Shift + F6	Cycle between property sheets.
Ctrl + B	Find selected object in Browser.
Ctrl + D	Select diagram.
Ctrl + E	Open list of reports.
Ctrl + F	Find objects.
Ctrl + F4	Close diagram.
Ctrl + Alt + F4	Close model.
Shift + F12	Open workspace.
Shift + F3	Save workspace.
Shift + F4	Close workspace.

List Shortcuts

Shortcut	Action
F2	Select text for edit.
F4	Select cell to add text.
Ctrl + Shift + Spacebar	Select line.
Ctrl + A	Select all.
Ctrl + Shift + A	Deselect all.
Ctrl + N	Add entry.
Ctrl + I	Insert entry.
Ctrl + D	Delete entry.

Code Preview Tab Shortcuts

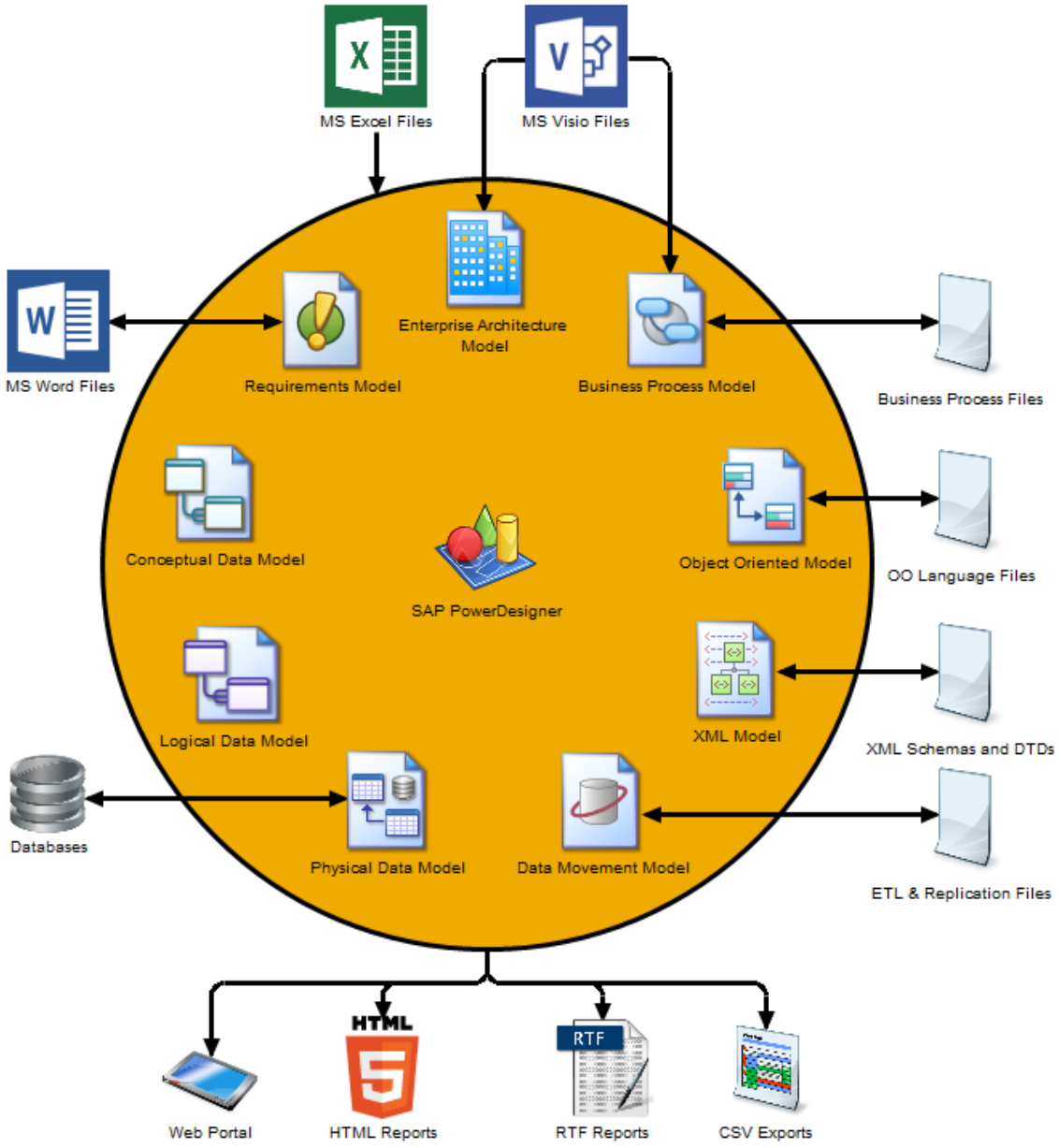
Shortcut	Action
F2	Next bookmark.
F3	Find next.
Shift + F11	Menu.
Ctrl + F2	Insert/Remove bookmark.
Ctrl + I	Insert file.

2 Linking and Synchronizing Models

The chapters in this part describe the PowerDesigner features that help you create links between models and trace the impact of changes between models.

2.1 Getting Started with Linking and Syncing

The PowerDesigner Enterprise Modeling environment allow you to create and exploit complex interdependencies between objects to model interconnected systems and view them in many different ways.



A single data item such as a product name can be implicated in multiple systems, which are modeled in a number of interdependent models. It may also be represented at various levels of abstraction in conceptual, logical, and physical data models, or in different contexts such as object-oriented, XML, business process, and enterprise architecture models.

Instead of defining this same object and all its accompanying metadata multiple times, PowerDesigner allows you to generate these instances or in other ways link them together, and provides powerful tools for tracking and analyzing the dependencies between them.

> **i Note**
>
> When working with multiple connected models, we recommend that you group your models together into a project (see Projects and Frameworks [page 46]).

2.1.1 Creating Links Between Models

PowerDesigner provides many ways to link the objects of your models together.

You can create the following kinds of links between models:

- Generation links - are created during model or object generation. Each generated object is linked with its origin object.
  - Model Generation - generates one model from another and allows subsequent synchronization on demand (see Generating Models [page 424]).
  - Object Generation - allows you to define the generation of objects from one model to another with subsequent synchronization on demand (see Generating Model Objects [page 435]).
- Shortcuts - create a reference to an object in another model. You either create shortcuts explicitly in order to share or reuse an object in another model (see Creating Shortcuts [page 446]) or implicitly when creating other kinds of links. To create any of the kinds of links in this list, the target model in which the referenced object resides must be open in your workspace:
  - Traceability Links - specify connections between any object and any other object in any model (see Traceability Links [page 146]).
  - Related Diagrams - specify connections between any object and any diagram in any model (see Specifying Diagrams as Related Diagrams [page 189]).
  - RQM Traceability Links - link requirements with design objects that are intended to satisfy them (see *Requirements Management > Requirements Views > Requirements (RQM) > Linking Requirements with Design Objects and External Files*).
  - Object Mappings - specify connections between certain predefined types of objects (see Object Mappings [page 463]).
  - EAM Object Export/Import - link enterprise architecture objects with the design objects that will implement them (see *Enterprise Architecture > Importing to and Generating and Exporting from an EAM > Exporting and Importing Objects to and from Other Models*).
  - BPM Data Export/Import - link data objects in the BPM with objects in other models (see *Business Process Architecture > Analysis BPM > Data (BPM)*).
  - Extended Attributes and Collections - allow you to extend the PowerDesigner metamodel to define new kinds of links between objects (see Extending Objects [page 167]).
- Object Replications - duplicate an object from one model to another and allow you to vary the properties you choose while keeping the rest synchronized with the original object (see Creating Replicas [page 451]).

2.1.2 Viewing Links Between Models

PowerDesigner provides an array of tools to allow you to view and analyze the links created between your models.

Some or all of the following tools are available to you to view, analyze and edit the links between your models:

- Project Diagram - provides a high-level read-only view of the links between the models in your project (see Project Diagrams [page 50]).
- Dependency Matrix - provides an editable view of all the links between the two object types that you specify in the matrix definition (see Dependency Matrices [page 199]).

- Generation Links Viewer - provides a read-only view of all the model or object generation links between the present model and any models it is associated through generation (see The Generation Links Viewer [page 439]).
- Mapping Editor - provides an editable view of all the mappings between the present model and any models it is associated with through mappings (see The Mapping Editor [page 468]).
- Multi-Model Report - provides a configurable report in HTML or RTF format documenting your models and the links between them (see Creating a MultiModel Report [page 311]).
- Impact and Lineage Analysis - provides a configurable analysis of the objects that impact and/or are impacted by the selected objects (see Impact and Lineage Analysis [page 489]).
- Object Lists - can include shortcuts to objects of the selected type if the *Include Shortcuts* tool is selected (see Object Lists [page 117]).
- Object property sheets (see Object Properties [page 107]) - provide information on various forms of links on the following tabs:
  - Related Diagrams (see Specifying Diagrams as Related Diagrams [page 189])
  - Traceability Links (see Traceability Links [page 146])
  - Mappings - where available (see Object Mappings [page 463])
  - Requirements - if enabled (see Requirements Tab [page 114])
  - Dependencies - lists the objects that depend on the object (see Dependencies Tab [page 115])

2.2 Generating Models and Model Objects

PowerDesigner provides a number of predefined model-to-model generation paths along with mechanisms for generating any model objects into any other objects in another type of model.

> **i** Note
>
> If you generate from more than one model to a single model, only the generation links from the last generated model are retained.

2.2.1 Generating Models

PowerDesigner provides powerful capabilities for generating one model from another, and keeping the two models synchronized. Thus, you could model your data at an abstract level in a CDM, generate an LDM to model implementation-neutral features, and then generate multiple PDMs for various DBMSs.

Context

The following table shows which model types you can generate from each model type. The left column lists the source model and the column headings the model types that can be generated from it:

	BPM	CDM	LDM	PDM	DMM	OOM	XSM
BPM	X						
CDM		X	X	X		X	
LDM		X	X	X			
PDM		X	X	X		X	X
DMM					X		
OOM		X		X		X	X
XSM				X			X

Procedure

1. Select *Tools*, and then one of the following to open the appropriate Model Generation Options Window:
   - Generate Business Process Model... Ctrl + Shift + B
   - Generate Conceptual Data Model... Ctrl + Shift + C
   - Generate Logical Data Model... Ctrl + Shift + L
   - Generate Physical Data Model... Ctrl + Shift + P
   - Generate Data Movement Model...
   - Generate Object-Oriented Model... Ctrl + Shift + O
   - Generate XML Model... Ctrl + Shift + M
2. On the *General* tab, select a radio button to generate a new or update an existing model, and complete the appropriate options. For more details, see Model Generation Options [page 425].
3. [optional – PDM-PDM generation only] Click the *DBMS Preserve Options* tab and set any appropriate options.
4. [optional] Click the *Detail* tab and set any appropriate options.

5. [optional] Click the *Target Models* tab and specify the target models for any generated shortcuts.
6. [optional] Click the *Selection* tab and select or deselect objects to generate. For information about using the tools on this tab, see Adding an Item from a Selection List [page 129].
7. Click *OK* to begin generation.

2.2.1.1 Model Generation Options

You can customize the generation of your models by selecting model generation options.

General Tab

The *General* tab lets you specify the new model to be generated or existing model to be updated.When generating a new model, you must specify the following options:

Option	Description
DBMS, Object, XML, or Process language	Specifies the definition file for the model to be generated. By default, PowerDesigner creates a link to this file. To copy the definition and save it in your model file, click the *Embed Resource in Model* button.
Name / Code	Specifies the name and code for the model to be generated. Click the *Configure Model Options* button to specify naming conventions (see Naming Conventions [page 155]) and other model options for the model to be generated.

When updating an existing model, you must specify the following options:

Option	Description
Select model	Specifies the target model to be updated. Select a model that has already been generated from the current model from the list, or click the ellipsis button to the right to open a list of other models of the same type open in the workspace.
DBMS, Object, XML, or Process language	Displays the definition file of the model to be updated.
Preserve modifications	Allows you to review the changes to be made to the model (see Comparing and Merging Models [page 366]). If this option is not selected, generated objects will replace objects with the same name in the model.

DBMS Preserve Options Tab

This tab is only available when you generate a PDM from another PDM, and allows you to choose to preserve, where possible, the values associated with:

- Physical options - for relevant objects.
- Triggers and stored procedures – If the target belongs to a different DBMS family, triggers are automatically rebuilt.
- Database objects – Only objects supported by the target DBMS will be generated.
- Extended attributes.

Detail Tab

The *Detail* tab contains options to control the generation of model objects.

Option	Description
Check Model	Invokes a model check before generation, and stops generation if errors are found. You can configure specific model checks before generation by selecting *Tools* > *Check Model* (see Checking Models [page 148]).
Save Generation Dependencies	Instructs PowerDesigner to retain links between each source object and the corresponding target object, which allow objects to be subsequently identified, even if they have been modified (see The Generation Links Viewer [page 439]).
Generate Mappings	Defines the source model as the data source of the generated model and creates mappings between all source and target objects (see Object Mappings [page 463]). This option is dependent on the *Save Generation Dependencies* option.
Rebuild Triggers	[CDM/OOM-PDM] Instructs PowerDesigner to rebuild triggers with unmodified templates after the PDM generation.
Convert Names into Codes	Instructs PowerDesigner to generate target model object codes using a conversion script. This option can be useful when generating models with very different naming conventions. If it is not selected, target object codes are generated from source object codes. For example, generate an OOM Java class attribute with a code of `customerName` to a PDM table column called `CUSTOMER_NAME` by clicking the *Configure Model Options* button on the *General* tab, selecting the appropriate conversion script (see Name and Code Conversion Scripts [page 159]), and selecting this option.
Preserve n-n relationships	[CDM-LDM only] Specifies that many-to-many relationships created in the CDM will be generated to the LDM if the destination model permits them.

Option	Description
Enable transformations	Displays the *Extensions* tab, the *Pre-generation* tab (if your model has transformations defined) and the *Post-generation* tab (if you are updating an existing model that has transformations defined) to select transformations to perform before and/or after generation (see Applying Pre- and Post- Generation Transformations [page 430]).
Skip single root element	[XSM-PDM only] Specifies that the single root element is ignored, and its immediate child elements are treated as multiple roots, which will be generated as tables.
Generate root element	[PDM/OOM-XSM only] Specifies that a single root element representing the PDM or OOM model is generated. Deselecting this option means that all the tables or classes are generated as multiple roots.
Generate tables/classes as	[PDM/OOM-XSM only] Specifies how PDM tables or OOM classes are generated. You can choose between: • Elements - [default] each table/class is generated as an untyped element directly linked to its columns/attributes generated as attributes or sub-elements. • Elements with complex types - each table/class is generated as an element typed by a complex type, generated in parallel, to contain the columns/attributes. You can override this option for individual tables or classes (see Customizing XSM Generation for Individual Objects [page 434]).
Generate columns/attributes as	[PDM/OOM-XSM only] Specifies how PDM columns or OOM attributes are generated. You can choose between: • Elements - [default] each column/attribute is generated as an sub-element of its table/class element or complex type. • Attributes - each column/attribute is generated as an attribute of its table/class element or complex type. You can override this option for individual columns or attributes (see Customizing XSM Generation for Individual Objects [page 434]).
Class Prefix	[OOM only] Specifies a prefix to apply to the codes for all classes generated.

Detail Tab - PDM Generation Options

The following additional options are available on the *Detail* tab when generating a PDM:

Option	Description
Naming Template	[Except XSM-PDM] Controls the generation of codes for tables, columns, domains, and business rules (default: `%Code%`). You can use any GTL or applicable variables in these fields.

Option	Description
Table	[XSM-PDM only] Controls table generation: • Table Prefix - Controls the generation of codes for tables. • ID column type - Specifies the type to use for ID columns, and which will be used to generate `xs:ID` types (default: `numeric`).
Index	Controls the generation of codes for keys: • PK index names - (default: `%TABLE%_PK`).You can use the following variables in all index name fields: ○ `%TABLE%` - Generated code of the table. ○ `%TNAME%`, `%TCODE%` - Table name and code. ○ `%TLABL%` - Table comment. • AK index names - (default: `%TABLE%_AK`).You can use the following variables in this field: ○ `%REFR%`, `%PARENT%`, `%CHILD%` - Generated code of the reference, parent table, and child table. ○ `%REFNAME%`, `%REFRCODE%` - Reference name and code. ○ `%PNAME%`, `%PCODE%` - Parent table name and code. ○ `%CNAME%`, `%CCODE%` - Child table name and code. ○ `%PQUALIFIER%`, `%CQUALIFIER%` - Parent and child table qualifier. • FK index names - (default `%REFR%_FK`). • FK threshold - Specifies the minimum number of estimated occurrences of an entity necessary to create an index on a foreign key. This value can be specified in the entity property sheet. If it is not set, foreign key indexes are generated by default. **i Note** The generated code of a variable is the code defined in the object property sheet, which may be truncated when generated if the code contains characters not supported by the DBMS. For more information about PowerDesigner variables, see *Customizing and Extending PowerDesigner > DBMS Definition Files* .

Option	Description
Reference	Controls the generation of references: • Update Rule - Specifies the default update constraint for referential integrity. • Delete Rule - Specifies the default delete constraint for referential integrity. • FK column name template - Specifies the naming convention for migrated foreign keys. You can select one of the default templates from the list or enter your own using the following variables: ○ `%REFR%` - Name / Code of the reference ○ `%PARENT%` - Name / Code of the parent table ○ `%COLUMN%` - Name / Code of the parent column ○ `%KEY%` or `%CONSTNAME%` - Name / Code of the key constraint attached to the reference ○ `%PROLE%` - Role of the CDM entity that generated the parent table. If no role is defined on the relationship or association, `%PROLE%` takes the content of `%PARENT%` to avoid generating a column with no name. In the following example, if `%PROLE%` is not equal to the parent name (and thus a parent role is present) then it is used. Otherwise, the role is absent and the template `%.3:PARENT%_%COLUMN%` is used: `[%PROLE%=%PARENT%?%.3:PARENT%_%COLUMN%:%PROLE%]` Customized naming templates reappear in the generation dialog box the next time you open it, but are not saved to the list of predefined templates. • Use template - Controls whether the FK column name template will always be used or only in case of conflict.

> **Note**
>
> If you modify a primary key in a PDM, then regenerate the PDM from a CDM, the modified primary key is not preserved. If you want to preserve a modified primary key, you need to modify the identifier in the CDM before regeneration.
>
> When you reverse engineer a database into a PDM and then generate an OOM, you can add an appropriate package hierarchy in the OOM by right-clicking the OOM in the Browser, selecting *Add Package Hierarchy*, and entering an appropriate hierarchy (for example, **`com.mycompany.myproduct.oom`**).

Target Models Tab

This tab contains a list of target models referenced by shortcuts or replications in the current model, and the appropriate generated models that the generated model will reference (see Generating Shortcuts and Replicas [page 461]). If any target model does not have a generated model listed, the generation of shortcuts or replications referencing it will fail. This tab will be empty if your model does not contain shortcuts to other models or if these shortcuts are set to generate as independent objects.

2.2.1.2 Applying Pre- and Post- Generation Transformations

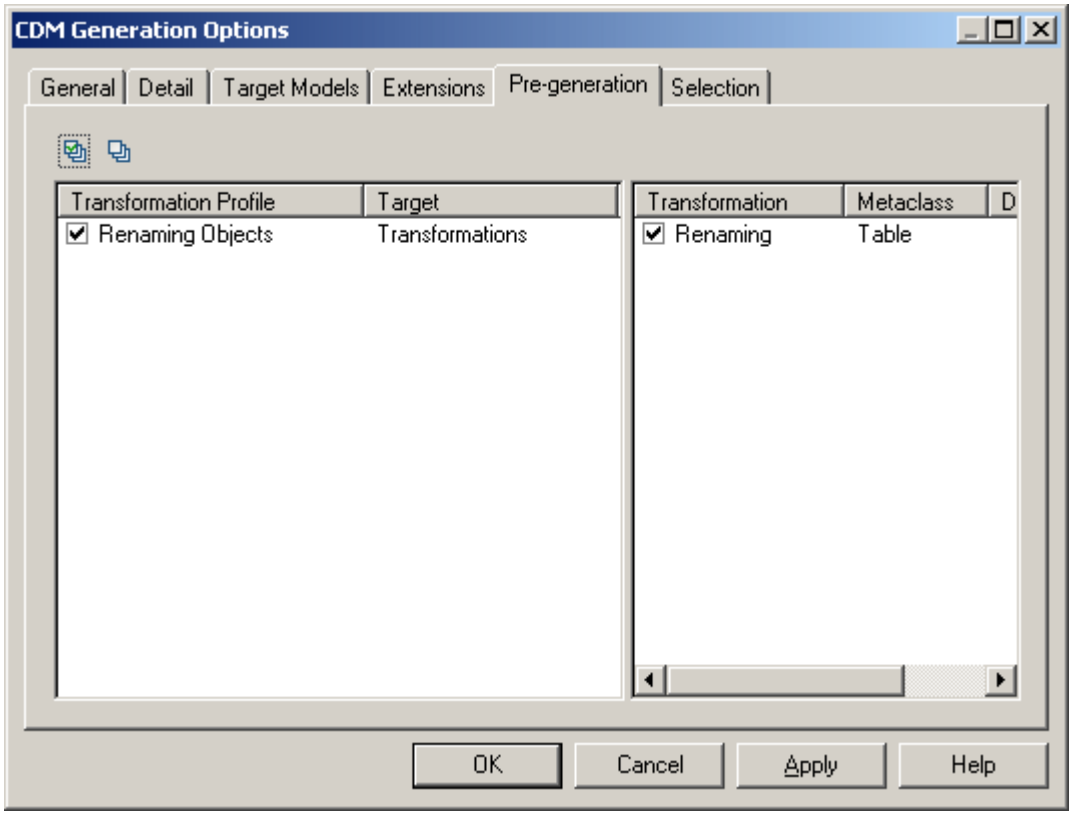
Click the *Enable transformations* button on the Model Generation Options *Detail* tab to display additional tabs for selecting transformations to apply to the source model objects before generation and to apply to generated objects after generation.

Context

In order to apply transformations, you must have either defined transformation profiles in your model's language definition file (for PDM, OOM, BPM, or XSM) or in an extension file (see *Customizing and Extending PowerDesigner > Extension Files > Transformations (Profile)*).

Procedure

1. Select *Tools* > *Generate Model* to open the Model Generation Options dialog.
2. Click the *Enable transformations* button on the *Detail* tab to display the *Extensions* tab, the *Pre-generation* tab (if your model has transformations defined) and the *Post-generation* tab (if you are updating an existing model that has transformations defined).
3. Click the *Extensions* tab to select extensions containing transformations.
4. Click the *Pre-generation* tab and select the profiles and transformations to be applied before generation. If you deselect a profile, none of its transformations will be executed. You can drag and drop profiles to modify the order of execution of transformations.



5. Click the *Post-generation* tab and select the profiles and transformations to be applied after generation.
6. Click *OK* to begin the generation. The following steps are performed:
    1. The object tree of the source model is traversed from the model object down and, for each type of object for which one or more pre-generation transformation profiles is defined, each profile is applied in the order in which they appear on the tab, with each transformation being applied in the order in which they are defined in the profile.
    2. The transformed source model is generated to the target model.
    3. The object tree of the generated target model is traversed and type of object and, for each type of object for which one or more post-generation transformation profiles is defined, each profile is applied in the order in which they appear on the tab, with each transformation being applied in the order in which they are defined in the profile.

> **i** Note
>
> You can apply post-generation transformation profiles to your model at any time by selecting *Tools* > *Apply Transformations* (see Applying Transformations to Objects [page 183]).

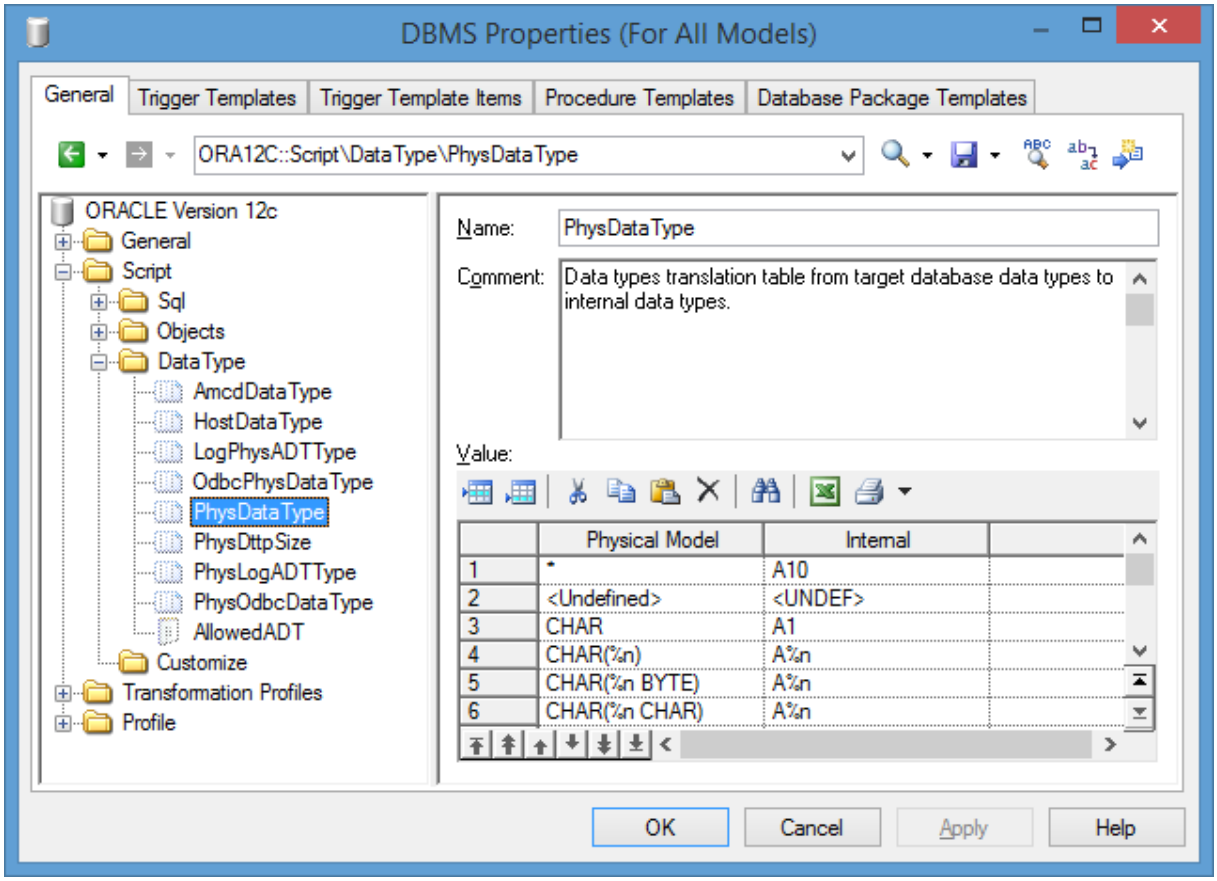
2.2.1.3 Customizing Data Type Mappings

When generating another PDM from your PDM, PowerDesigner maps the existing column datatypes to appropriate data types in the new model. If the standard mappings are not sufficient for you, you can use the

`Enhance Data Type Mapping` extension to specify alternative mappings, including on a column-by-column basis.

Context

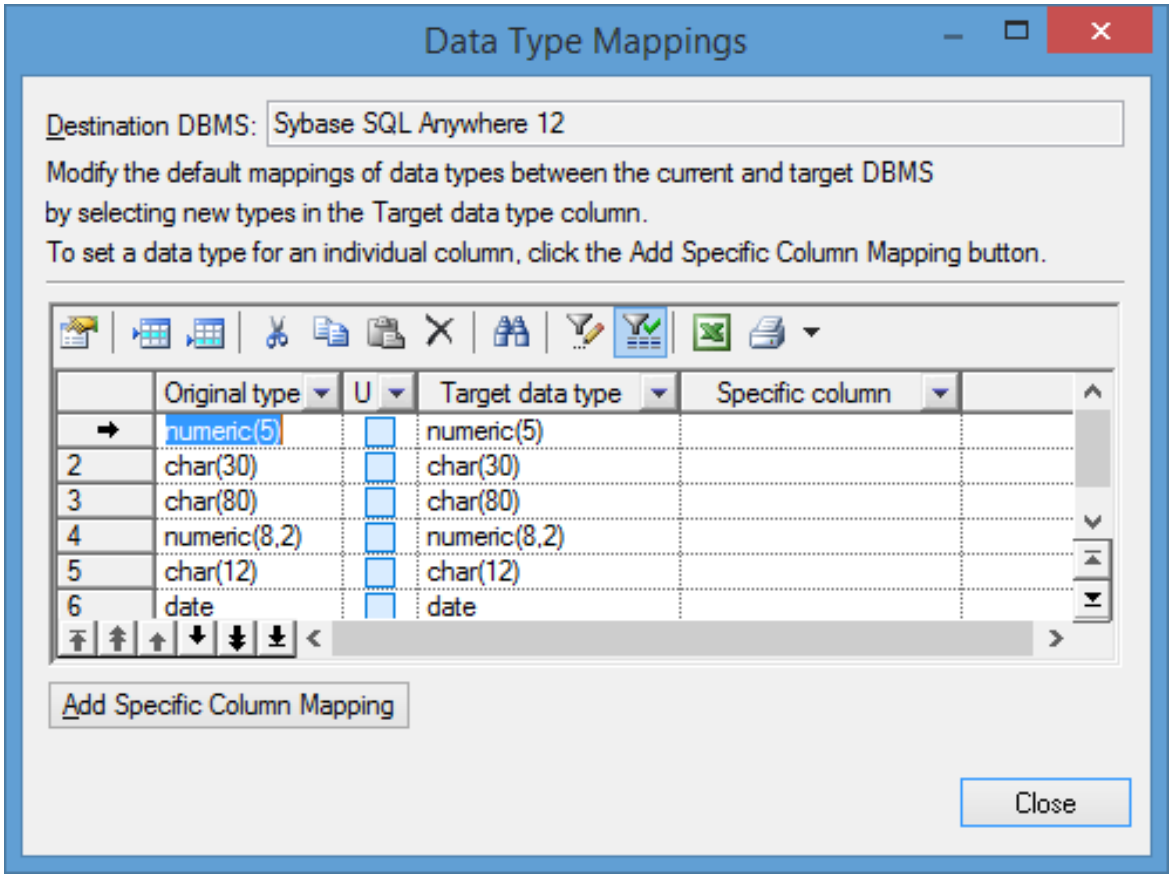
To review the conversions that PowerDesigner makes by default between the data types of a database or other modeling target and its standard conceptual types (which are also used in the CDM), select *Tools* > *Resources* > `<Type>`, select the appropriate file in the list and click the *Properties* tool. Expand the *Script* > *DataType* (for DBMSs) or *Settings* > *DataType* (for other resource files), and review each of the entries (which are described in their *Comment* field):



Procedure

1. Select *Tools* > *Generate Physical Data Model*, enter the appropriate generation options (see *Core Features Guide* > *Linking and Synchronizing Models* > *Generating Models and Model Objects*).
2. On the *Detail* tab, click the *Enable Transformations* button to display the *Extensions* tab, and select the `Enhance Data Type Mapping` extension.

3. Click *OK* to start the generation. The Data Type Mappings dialog appears, with the existing data types present in the model listed in the *Original type* column, and those that PowerDesigner proposes in the new DBMS in the *Target data type* column:



4. You can change data type mappings in two ways:
   - To change the mapping for all columns of a certain data type, select the desired new data type from the list in the *Target data type* column.
   - To change the mapping for one column only, click the *Add Specific Column Mapping* button, select the column from the tree, click *OK*, choose the new data type for the column, and click *OK* to add this mapping to the list.
5. When you have modified all the necessary data types, click *Close* and the generation will continue, using your custom mappings where appropriate.

> **i Note**
>
> You can also customize data type mappings when changing the DBMS of your model with the *Database* > *Change Current DBMS* command. To do so, you must first attach the `Enhance Data Type Mapping` extension, by selecting *Model* > *Extensions*, clicking the *Attach an Extension* tool, select the extension, and clicking *OK* to attach it to your model.

For more information about data types, see *Customizing and Extending PowerDesigner > DBMS Definition Files > Script/Data Type Category* and *Customizing and Extending PowerDesigner > Object, Process, and XML Language Definition Files > Settings Category: Object Language* .

2.2.1.4 Customizing XSM Generation for Individual Objects

When generating an XSM from a PDM or OOM, you can specify global generation options to generate tables/classes as elements with or without complex types and columns/attributes as elements or attributes. You can override these options for individual objects by attaching the **PDM XML Generation** or **OOM XML Generation** extension to your source model and selecting from their XML generation options.

Context

> **i** Note
>
> The extension provides new property sheet tabs for setting generation options for individual objects, but you can also set these options with or without the extension by selecting *Model* > `<objects>` to open the appropriate object list, clicking the *Customize Columns and Filter* tool, and selecting to display the **XML Generation Mode** column.

For example, if you want to generate the majority of your table columns to an XSM as XML attributes, but want to generate certain columns as elements, you should:

- Modify the XML generation options for those columns that you want to generate as elements.
- Select to generate columns as attributes on the Model Generation Options *Detail* tab.

Procedure

1. Select *Model* > *Extensions* to open the List of Extensions, and click the *Attach an Extension* tool.
2. On the *General Purpose* tab, select **PDM XML Generation** or **OOM XML Generation** and click *OK* to attach the extension to your model and *OK* to close the List of Extensions.

   These extension files enable the display of the *XML* tab in all table and column or class and attribute property sheets.
3. Open the property sheet of the table, column, class, or attribute whose generation you want to customize, and click the *XML* tab.
4. Use the radio buttons to specify how you want to generate the object in an XSM.
   - For tables and classes, you can specify to generate them as:
     - Elements - the table/class is generated as an untyped element directly linked to its columns/attributes generated as attributes or sub-elements.
     - Elements with complex types - the table/class is generated as an element typed by a complex type, generated in parallel, to contain the columns/attributes.
     - Default - generation of the table/class is controlled by the option selected in the *XML Generation* group box on the Model Generation Options *Detail* tab.
   - For tables, you can additionally specify to generate keys as:
     - Key - [default] The primary
       key columns are generated and also KEY and KEYREF wherever the table is referenced.

- ID attribute - The primary key columns are not generated and an ID attribute, `id`, is generated to replace them.
  Wherever the table is referenced, an IDREF attribute is generated to reference the appropriate ele ment. If the reference role name is assigned, this attribute is given this
  name. Otherwise, the referenced table name is used and the standard renaming mechanism is enf orced.
- Key and ID attribute - In many cases the primary
  key columns have significant data and you may want to generate them, as well as an ID attribute. In this case an ID attribute is generated for the element and IDREF is used systematically
  for any reference to the table:

The following rules apply to the generation of keys:

- If a Table generates an ID, all its child tables will generate an ID attribute.
- If a Table generates Key columns, all its child tables will generate Key columns.
- If a child table is flagged to generate PK only, ID Attribute will be automatically generated.
- If a table generates ID attribute, No Key nor KeyRef will be generated, and ALL references will gene rate IDREF attribute.. (Even if the table generates also Key Columns)
- If a table generates ID attribute ONLY, All Foreign Key Columns referencing its Key columns will be systematically removed and replaced by an IDREF attribute

- For columns and attributes, you can specify to generate them as:
  - Elements - [default] the column/attribute is generated as an sub-element of its table/class element or complex type.
  - Attributes - the column/attribute is generated as an attribute of its table/class element or complex type.
  - Default - generation of the column/attribute is controlled by the option selected in the *XML Generation* group box on the Model Generation Options *Detail* tab.

5. Modify the XML generation options for any other objects that you want to generate in a different manner.
6. Select *Tools* > *Generate XML Model*, ensure that the appropriate options are set in the *XML Generation* group box on the Model Generation Options *Detail* tab, and start your generation.

2.2.2 Generating Model Objects

You can define your own model-to-model object generation commands. You can define as many commands as you need, and generate any of your model's objects as any kinds of objects in any other model. The generated objects are linked to the original objects and can be resynchronized at any time.

Context

Before you can generate model objects, you must define the details of the generation (see Defining Simple Object Generations [page 436])

> **i Note**
>
> This procedure deals with generating one or more object types to another model. For information about generating your entire model to another model, see Generating Models [page 424].

Procedure

1. Select *Tools* > *Generate Objects* > `<Menu command name>` to open the Object Generation Options window for the generation that you have previously defined (see Defining Simple Object Generations [page 436] or Defining Advanced Object Generations [page 437]).
2. On the *General* tab, select a radio button to generate a new or update an existing model, and complete the appropriate options (see Model Generation Options [page 425]).
3. [optional] Click the *Detail* tab and set any appropriate options.
4. [optional] Click the *Target Models* tab and specify the target models for any generated shortcuts.
5. [optional] Click the *Selection* tab and select or deselect objects to generate (see Adding an Item from a Selection List [page 129]).
6. Click *OK* to begin generation.

2.2.2.1 Defining Simple Object Generations

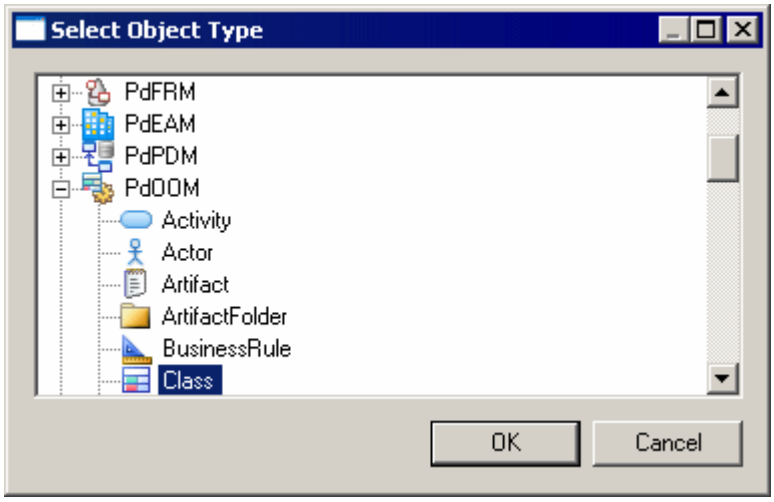
Simple object generations allow you to generate a single type of object (eg, Tables or Classes) to any other type of object. You can define multiple object generations. They are stored as objects in your model and are accessed by selecting *Tools* > *Generate Objects* > `<Generation Name>`.

Context

To generate more than a single type of object or to have greater control over how individual properties will be generated, use an advanced object generation (see Defining Advanced Object Generations [page 437]).

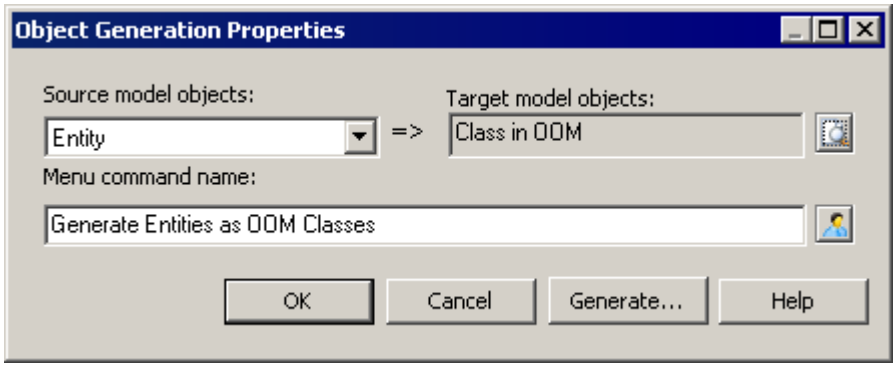
Procedure

1. Select *Tools* > *Generate Objects* > *Define New Object Generation* to open the Object Generation Properties dialog.
2. Select the type of model object you want to generate other model objects from in the *Source model objects* list.
3. Click the *Choose Metaclass* tool to the right of the *Target model objects* field to open the Select Object Type dialog, which allows you to select the type of objects you want to generate.



You can choose any object (including extensions) from any PowerDesigner model type, but some choices will be more appropriate than others.

4. [optional] Edit the *Menu command name* that PowerDesigner suggests, and which will appear under the *Tools* > *Generate Objects* menu to allow you to launch the generation.



5. Click *Generate* to launch the generation immediately or *OK* to save the generation definition for use later.

   Once you have defined an object generation, you can launch it at any time by selecting *Tools* > *Generate Objects* > `<Menu command name>`.

2.2.2.2 Defining Advanced Object Generations

Advanced object generations allow you to generate one or more types of object to any other types of object, with complete control over how all the attributes, compositions (sub-objects), and aggregations (related objects) will be mapped to the generated objects' properties. The generation mappings are defined in an extension file, can be shared with all users through the repository library, and are accessed by selecting *Tools* > *Generate Objects* > `<Generation Name>`.

Context

> **i Note**
>
> The following procedure gives an overview of the process for defining an advanced object generation. If an administrator has already defined a generation and provided the extension file to you, you can perform the generation directly by selecting *Tools* > *Generate Objects* > `<Generation Name>`. You can also define simple object generations for a single type of object directly in your model (see Defining Simple Object Generations [page 436]).

Procedure

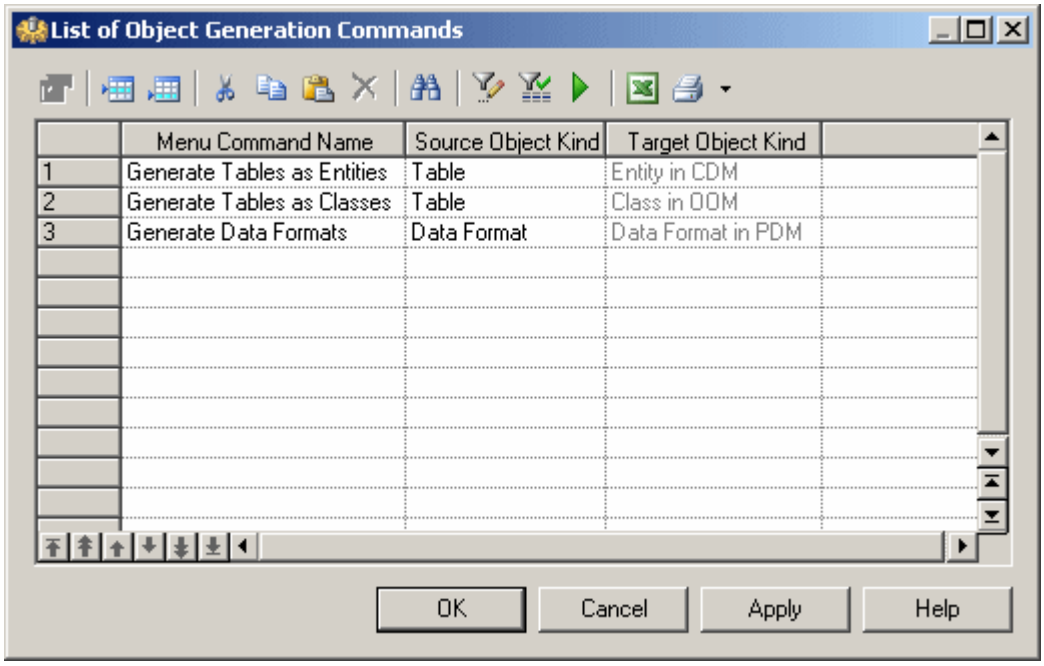
1. Select *Tools* > *Resources* > *Extensions* > `<Model Type>` to open the appropriate extension file list.

   > **i Note**
   >
   > As object generations are defined on the source objects, you must choose the correct type of extension file. For example, if you are generating EAM organization units and processes to a BPM, you should create the generation definition in an enterprise architecture model extension file.

2. Create a new extension file in the list to contain your object generation definition (see Creating an Extension File [page 412]).
3. Create an object generation definition in the extension file (see *Customizing and Extending PowerDesigner > Extension Files > Object Generations (Profile)*).
4. Save the extension file, attach it to a test model, and test the generation by selecting *Tools* > *Generate Objects* > `<Generation Name>`.
5. [optional] Check the extension file into the repository library to have it automatically deployed to all users (see Sharing Resource Files via the Library [page 551]).

2.2.2.3 Managing Object Generation Commands

You can review the object generations that you have defined by selecting *Tools* > *Generate Objects* > *Manage Object Generations*.



To review the definition of a generation, select it and click the *Properties* tool; to launch it, select it and click the *Run Object Generation* tool.

> **i** Note
>
> You can also create object generations directly in this list by clicking the *Add a Row* tool and specifying a menu command name and source and target object kinds.

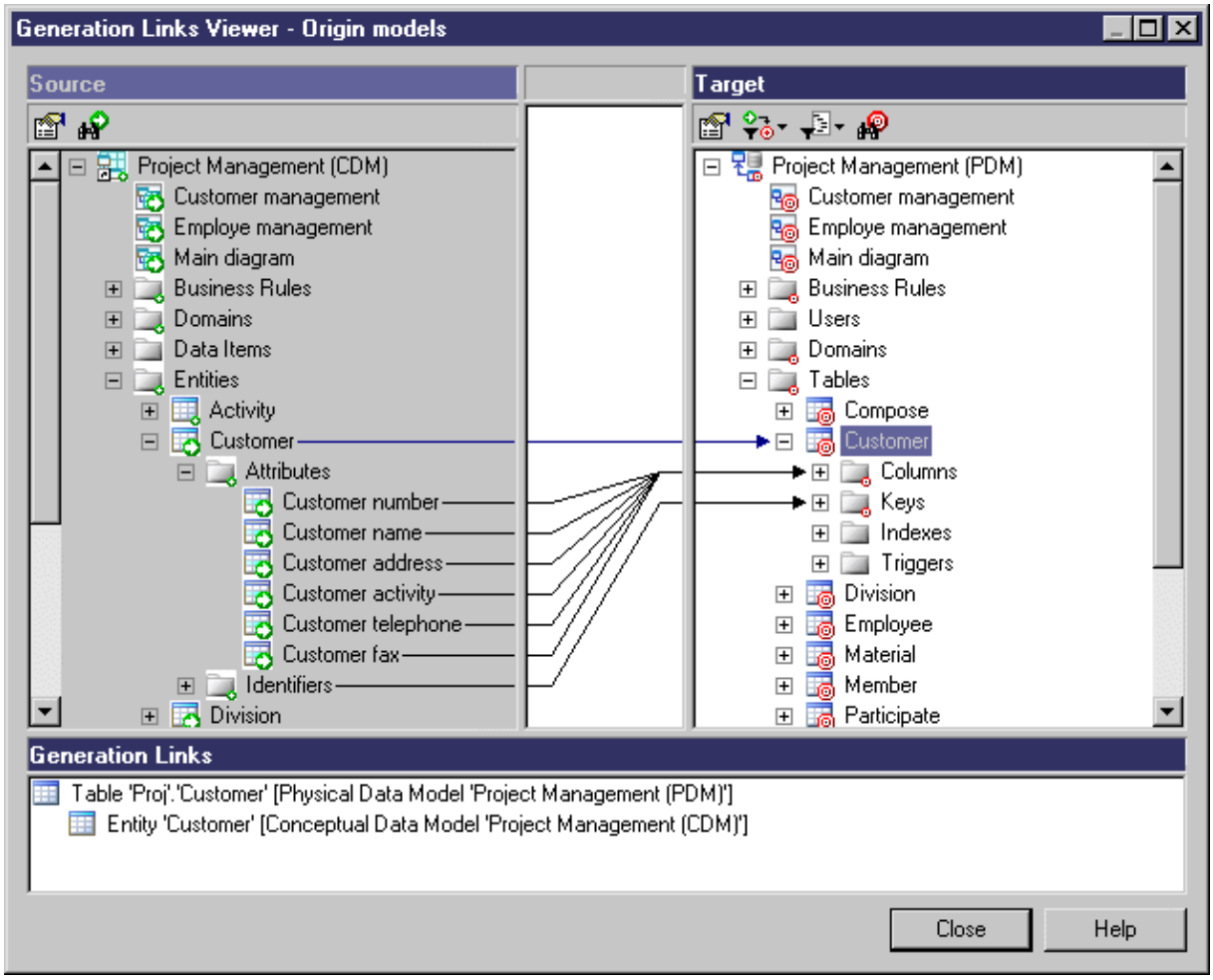
2.2.3 The Generation Links Viewer

The Generation Links Viewer allows you to view (but not edit) the generation links of a given model, which allow you to identify the origin of each generated or derived object of the model.

> **i** Note
>
> Links are created between models during model generation only if the Save Generation Dependencies option on the Detail tab of the generation dialog box is selected.

You can choose to view the:

- Links to the model(s) derived from the present model, by selecting *Tools* > *Generation Links* > *Derived Models* - the current model is shown in the Source pane and the derived models in the Target pane.
- Links to the model(s) from which the present model was derived, by selecting *Tools* > *Generation Links* > *Origin Model* - the current model is shown in the Target pane and the origin models in the Source pane.



The Generation Links Viewer dialog box is divided in three parts:

- The Source pane - which is on the left-hand side, and displays the structure of the source model
- The Target pane - which is on the right-hand side, and displays the structure of the target model(s)
- The Generation Links pane – shows you the sources or the targets of the selected object in the current model as a tree view. The root object is the source or target object from the current model, and the generation link name is composed of the object type and name (followed by, for link objects, the names of its extremities) followed by its parent model. You can double click a generation link in this pane to view the properties of the source or target object.

Source object icons have a small green arrow symbol and target object icons have a small red target symbol on their bottom right-hand corners.

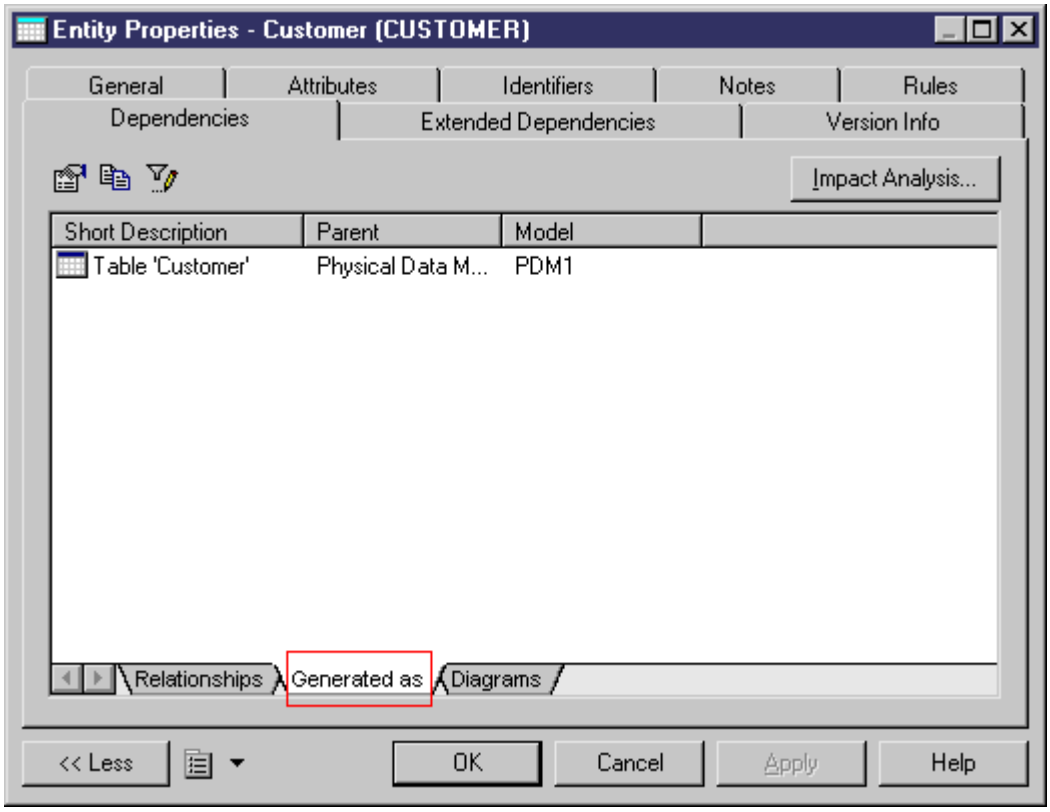
Non-editable links are drawn from the origin object in the Source pane to the derived object in the Target pane. Double-click the link in either pane to open the appropriate object's property sheet, or in the area between the two panes to display its details in the Generation Links pane.

The following tools are available in the Generation Links Viewer:

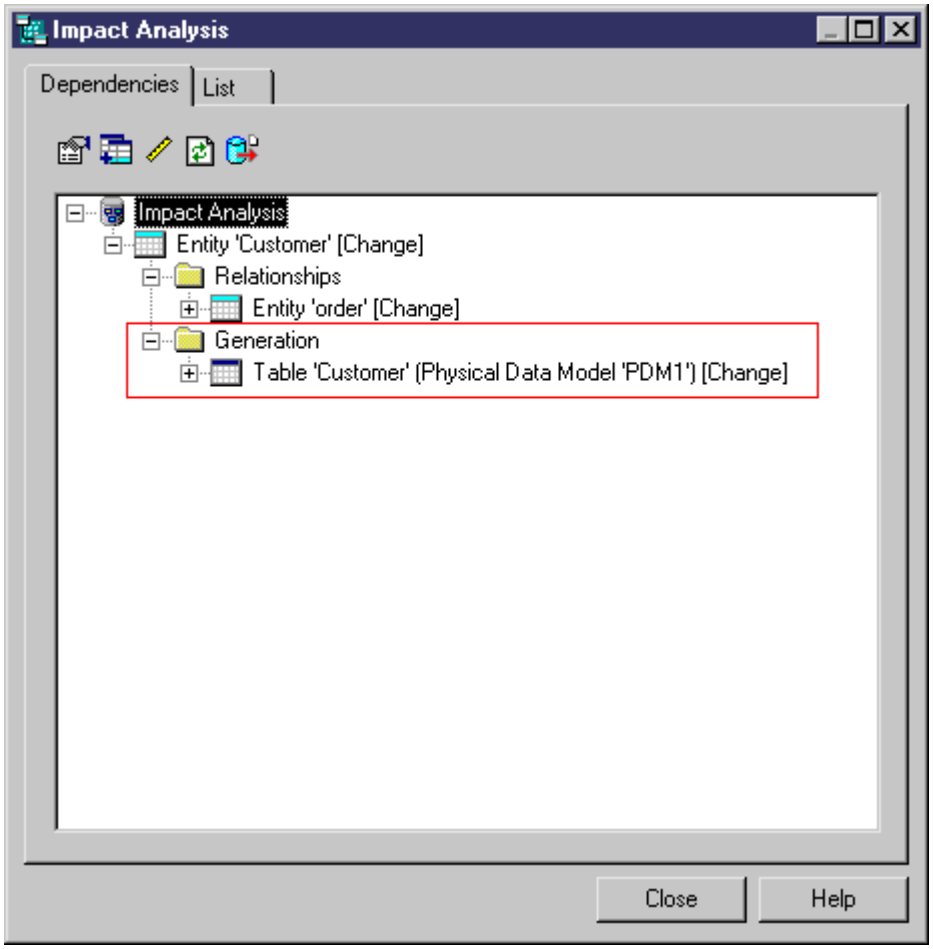
Tool	Description
	Properties - Opens the property sheet of the selected source or target object.
	Find Source Object - Finds an object in the Source pane and highlights it.
	Find Target Object - Finds an object in the Target pane and highlights it.
	Filter Generation Links – You can choose to filter by: • All generation links • Only generation links of the selected object • Only generation links of the selected object and its child objects
	Filter Objects - You can choose to filter by: • All objects • Only objects with generation links • Only objects without generation links

2.2.4 Using Impact Analysis with Generated Models

To use impact analysis on generated models, consolidate them in the Repository and extract them with the *Extract Dependencies* option selected to display the *Generated As* subtab in the *Dependencies* tab of source objects.



In the Impact Analysis dialog box, this link does not appear by default, you have to click the Extract Dependencies tool to display the generation links:



2.3 Shortcuts and Replicas

Shortcuts and object replications allow you to reuse objects defined in one model or package elsewhere. You should use a shortcut when you want to reference an object in another model or package, and a replica when you want to create a local copy of the object and make changes to one or more of its properties.

> **i** Note
>
> You cannot transform a shortcut into a replica, or vice versa. When choosing to create a shortcut or replica, you should identify how you want to use it. For example, if you want to reuse an existing table in your PDM, you should create a:
>
> - Shortcut - If you do not require the table to belong to your domain or namespace, and you just want to reference it without modifying its structure (adding, removing or otherwise changing columns or other sub-objects).
> - Replica - If you need a synchronized copy of the table in your namespace, or you want to modify its structure.

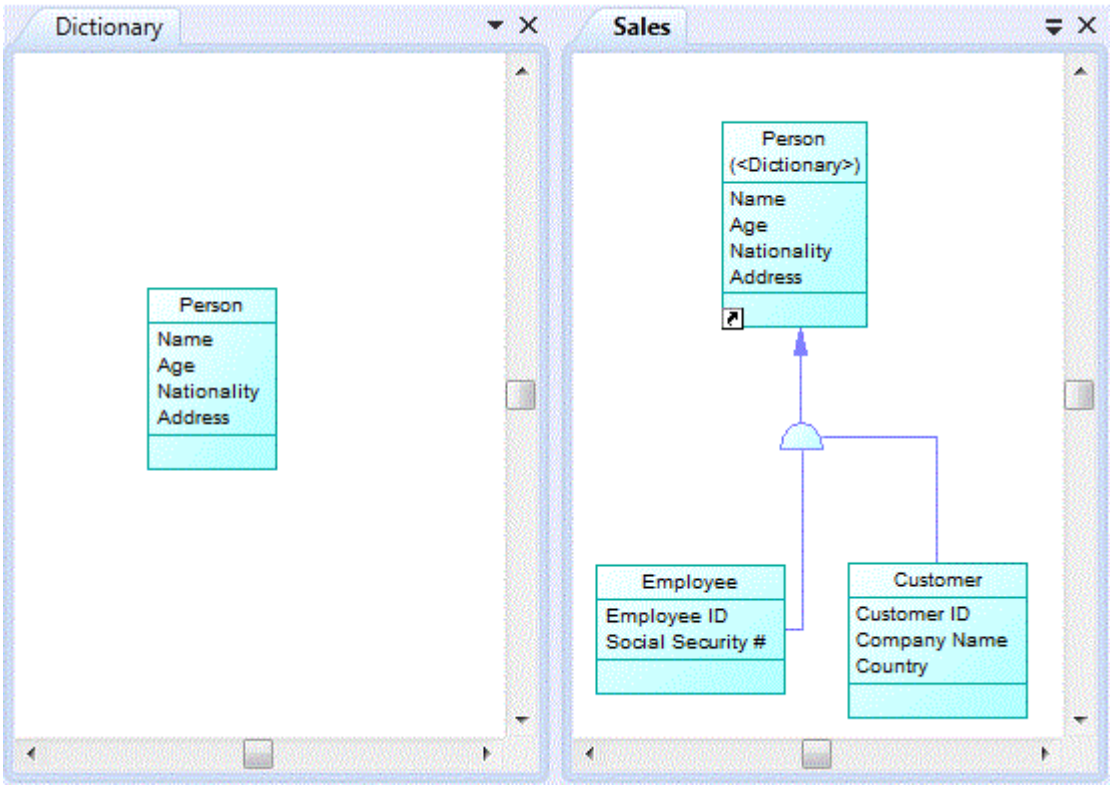
- Graphical synonym - If you merely want additional instances of the object symbol in your diagram to improve its readability and simplify linking (see Creating Graphical Synonyms for Object Symbols [page 220]).

A shortcut or replica is:

- Internal – if it points to an object in a different package in the same model.
- External – if it points to an object in a different model.

Shortcuts and replications allow you to benefit from:

- Reusability - You can create libraries of reusable objects whose properties are inherited by multiple objects in different models. For example, the **Person** entity and its four attributes (**Name**, **Age**, **Nationality** and **Address**) can be used via inheritance links in any model where you have need of a customer, employee, etc.
  In the following example, a **Person** entity is created in the Dictionary CDM, and a shortcut to it is created in the Sales CDM, where the **Employee** and **Customer** entities inherit from it:



- Automatic updates - When the target object changes, the updates cascade automatically (if both the referencing and target models are open in the workspace) to all the shortcuts and replicas.

You can create shortcuts or replicas for most PowerDesigner objects (including composite objects and other shortcuts or replicas). When you replicate an object, its sub-objects are also replicated. You cannot create shortcuts or replicas for the following types of objects:

Module	No Shortcuts For...	No Replicas For...
All	Diagrams	Diagrams, Links, Packages
RQM	Traceability link, user allocation	-
EAM	-	-
BPM	Starts, ends, decisions, synchronizations, variables	Starts, ends, synchronizations
MTM	-	-
CDM/LDM	Associations. External shortcuts not supported for associations or inheritances. **i Note** You can reuse data item shortcuts only if they are internal (same namespace). A duplicated data item shortcut has the same characteristics as the original data item shortcut.	Associations, Inheritances
PDM	Privileges	Privileges
DMM	-	Replications not supported
OOM	Decisions, synchronizations, interaction fragments, interaction references, messages	Starts, ends, synchronizations, interaction fragments, interaction references, messages
XSM	External shortcuts not supported for imports, includes, redefines, annotations	-

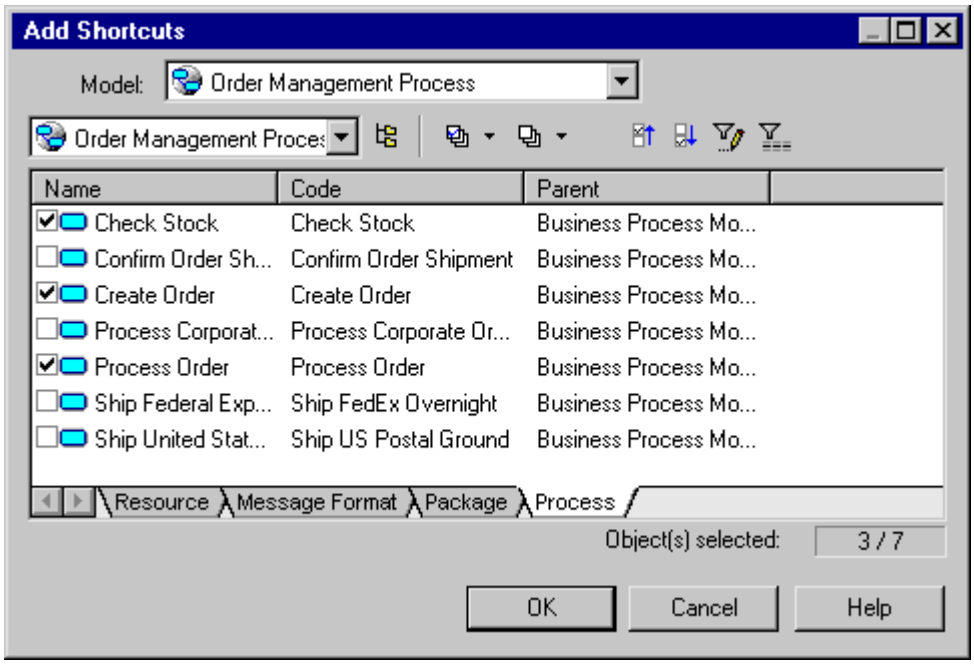
i Note

There is no need to create internal shortcuts or replicas to global objects, such as organization units or business rules, because they belong to the model and cannot be displaced into a sub-package.

2.3.1 Creating Shortcuts

You can create shortcuts to target objects from another package in the current model, or from another model open in the workspace through copy and paste, drag and drop, or from the List of Shortcuts.

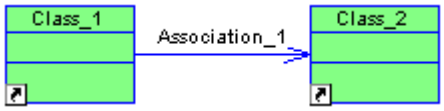
- Drag and Drop - Right-click on the target object in the Browser (or its symbol in a diagram) and drag and drop it into the diagram where you want to create the shortcut (or onto a model or package in the Browser). Release the right mouse button, and select *Create Shortcut(s) Here*.
  You can also create shortcuts of objects by left-clicking and dragging and dropping them while holding down `Shift` + `Ctrl`.
  For more information about dragging and dropping objects, including how to change the default behavior, see Moving and Copying Objects [page 103].
- Copy and Paste - Select the target object in the Browser (or its symbol in a diagram) and press `Ctrl` + `C` (or select *Edit* > *Copy*), then right-click in the diagram where you want to create the shortcut (or on a model or package in the Browser), and select *Edit* > *Paste as Shortcut*.
  For more information about copying and pasting objects, including how conflicts are managed, see Moving and Copying Objects [page 103].
- List of Shortcuts - Select *Model* > *Shortcuts* to open the List of Shortcuts and click the Add Shortcuts tool:



  Select the target model and package in the upper part of the dialog and objects in the sub-tabs in the lower part of the dialog, and then click *OK* to add them as shortcuts to the current model.
  For more information about the tools in this dialog, see Adding an Item from a Selection List [page 129].

If you select two linked objects in a model and create shortcuts from them, a shortcut of the link between them will also be created. If, after having created shortcuts from two objects, you subsequently create a link between them in the target model, you can update the referencing diagram to display the new link by opening it and selecting *Tools* > *Complete Links*.

Link shortcuts do not display the shortcut overlay symbol. In the following example, both classes and the association are shortcuts, but only the classes display the shortcut overlay:

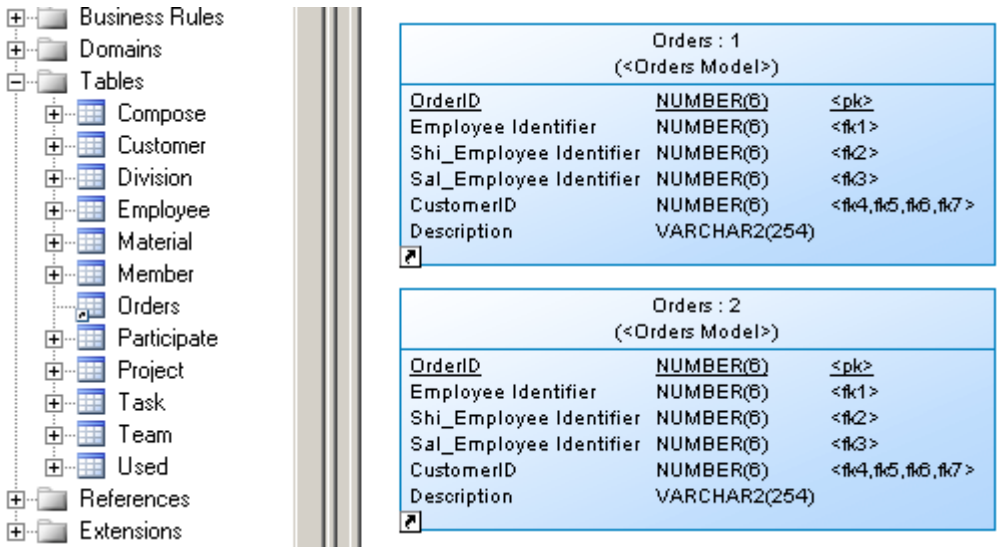


The following rules restrict the use of shortcuts:

- You cannot create more than one shortcut to the same target object in the same model or package.
- You cannot create a shortcut for a CDM data item outside the current namespace.
- A shortcut cannot serve as the child object for a parent/child hierarchy link (for example, a PDM reference link between two tables or a CDM inheritance link). In the case of internal shortcuts only, if you create such a link, PowerDesigner creates a shortcut to the parent object in the child object package, then creates the link in the child object package, and creates a shortcut to the link in the parent object package.

When you create a shortcut in a diagram, the following rules apply. If the model or package where you create the shortcut:

- Does not contain the shortcut - a shortcut and symbol are created
- Already contains the shortcut, but there is no symbol in the diagram - a symbol is created
- Already contains the shortcut, and there is a symbol in the diagram - a shortcut synonym symbol is created:



For more information about synonyms, see Creating Graphical Synonyms for Object Symbols [page 220].

2.3.1.1 Shortcut Properties

To open the property sheet of a shortcut and view or edit its properties, double-click its Browser entry or diagram symbol.

By default the shortcut property sheet contains read-only versions of all the tabs of the target object property sheet with the following exceptions:

- *Traceability Links* - The tab is not displayed and you cannot create traceability links with shortcuts as a source.
- *Dependencies* - Contains dependencies for the shortcut.
- *Version Info* - Contains creation and modification information for the shortcut.
- *Shortcut* - Contains the following properties describing the shortcut:

Property	Description
Target type / Name / Code	Specifies the type of the target model and of the target object, and the name and code of the target object. Click the *Change Target Object* at the bottom of the dialog to select a different target object (which must be of the same type, and cannot already be referenced by another shortcut in the same package).
Target model / package	Specify the name of the model and of the model package to which the target object belongs.
Shortcut type / Status	Specify whether the shortcut is external (from another model) or internal (from another package in the present model), and the status of the target model, which may affect what information is available for external shortcuts if the target model is closed or not found. For external shortcuts, (and you have not selected to display all properties), only the name of the shortcut and target model are available. Otherwise, all properties can appear in accordance with the display preferences.
Object1 and Object2	[link shortcut only] Specify the names of the source and target objects which the link connects
Generate / Generated as	[external shortcut only] Specifies that the shortcut should be generated when a model is generated from its parent model, and how it should be generated (see Generating Shortcuts and Replicas [page 461]). You can choose between: ○ Shortcut - Generate as a shortcut linked to the target object. ○ Object – Generate as an independent object that is not linked to the target object.

> **i Note**
>
> External shortcuts can display properties defined in the standard PowerDesigner metamodel and in extensions that are attached to the target model, but not any that are defined in extensions embedded in the target model. If you want to display such embedded extensions, you should export the extension file from the target model and then attach it as a shared extension.

To display a minimal shortcut property sheet (to reduce the size of your model), click *Tools* > *Model Options* > *Model Settings* and set *External Shortcut Properties* to *Name/Code*. The properties previously displayed on the *Shortcut* tab are now displayed on the *General* tab, and no other target object properties are shown.

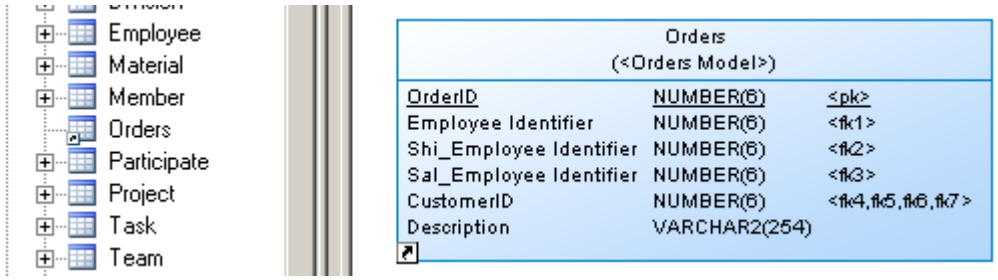
To display the editable target object property sheet directly instead of the shortcut properties (if the target model is open in the workspace), click *Tools* > *General Options* > *Dialog* and set *Internal Shortcut* and/or *External Shortcut* to *Target object*.

2.3.1.2 Viewing Shortcuts

Shortcuts can be identified in the Browser and diagrams by the shortcut arrow overlay on their bottom left corners.

> **i Note**
>
> In certain circumstances, when opening a model, PowerDesigner may display a message saying that it is unable to display complete properties for certain shortcuts, and offering to try to retrieve the missing information. If you click:
>
> - Yes - PowerDesigner will try to open the target models to retrieve the missing information. If it is unable to open one or more of the required models, it will report the failure. To resolve the problem, locate the missing target models, open them in your workspace, and then re-open the original model.
> - No - You can continue to work normally with the model and you will be prompted to retrieve the missing shortcut properties again next time you open it.

In the following example, the Orders table is a shortcut. It appears in the Browser with the other tables that belong to the model (to hide shortcuts in the Browser, deselect the *Show shortcuts* general option (see General Options [page 388]). Both the Browser entry and the diagram symbol display the shortcut arrow overlay:

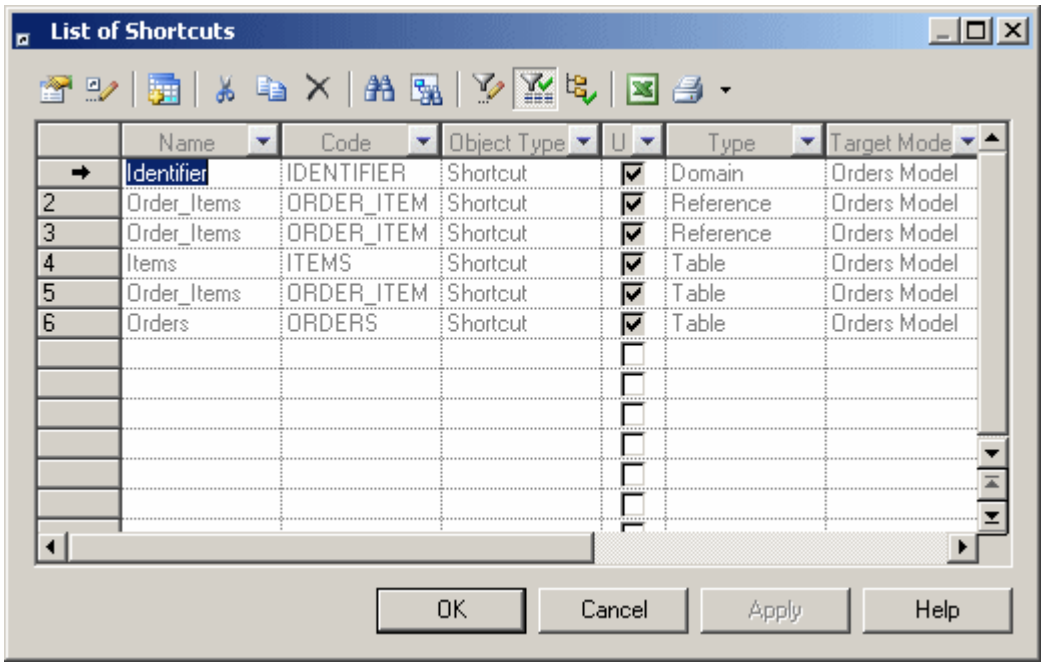


For external shortcuts, if you are displaying all properties (see Shortcut Properties [page 448]) or if the target model is open in the workspace, the properties that appear on its symbol are controlled by the standard display preferences for that type of object. Otherwise, only the name will appear, and you can modify the display

preferences specific to shortcuts by clicking *Tools* > *Display Preferences*, and selecting the Shortcut category. The following preferences are available:

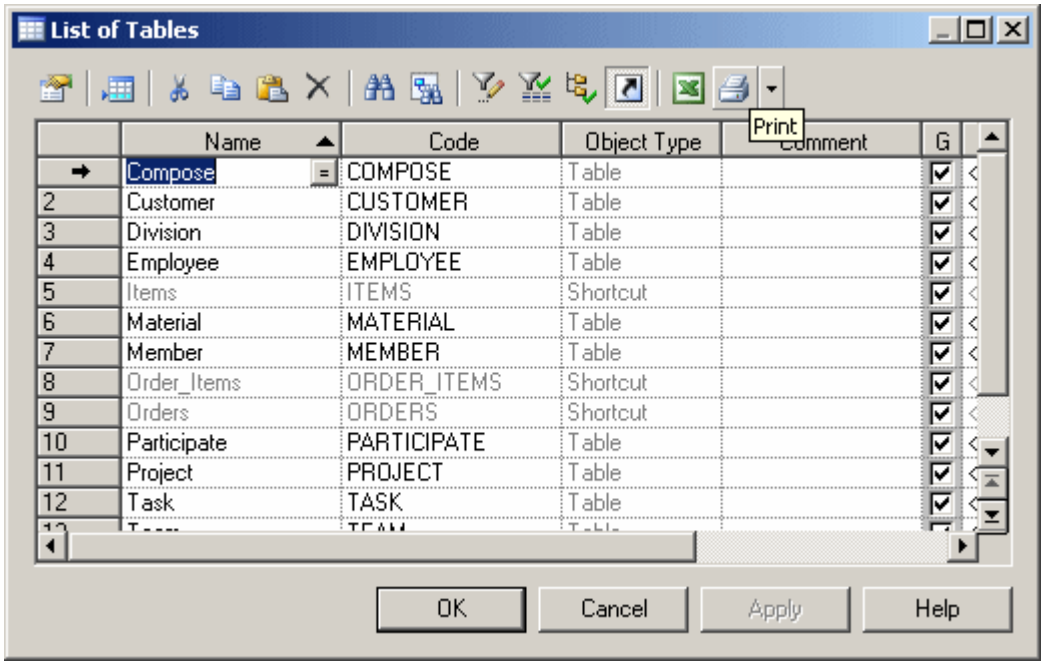
Preference	Description
Icon	Displays the shortcut icon on shortcut symbols
Model	[external shortcuts only] Displays the name of the target model on shortcut symbols
Package	Displays the package name on shortcut symbols. If you select this option, you must choose to display the: • Full path – displays the full path to the package • Last package only –displays the last package name only

To view all the shortcuts in a model, select *Model* > *Shortcuts* to open the *List of Shortcuts*. All the entries are greyed because you cannot modify a shortcut directly in this list:



The *U*[sed] column is checked when the shortcut has a symbol in a diagram or if it is referenced by at least one other object. If this column is not checked, the shortcut is not used in the model and you can safely delete it.

To view all the shortcuts for a particular type of object in a model, open the appropriate list by selecting *Model* > `<Objects>`, and click the *Include Shortcuts* tool:



For information about working with these lists, see Object Lists [page 117].

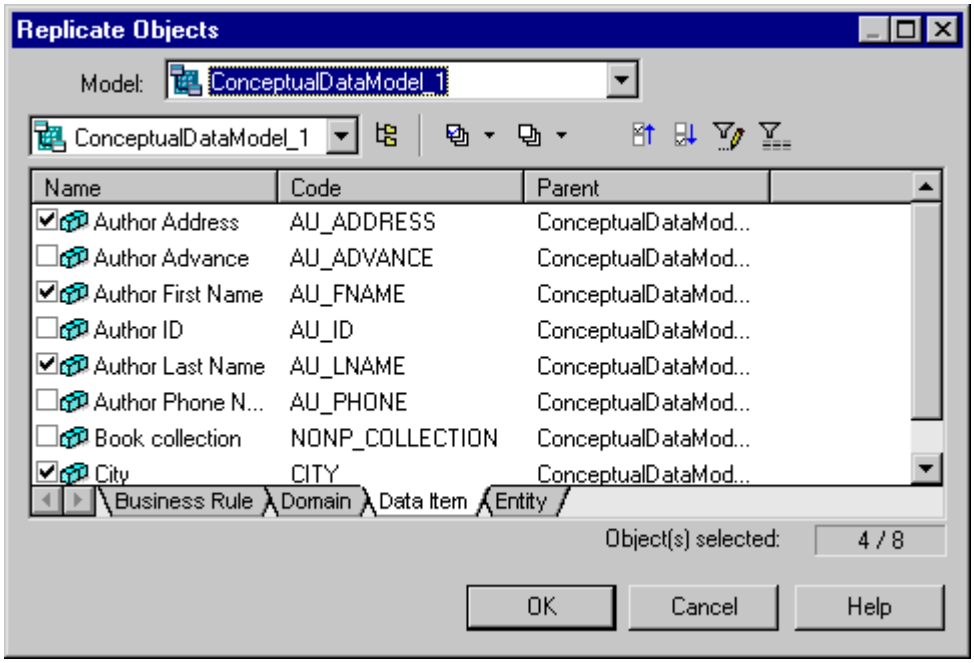
2.3.2 Creating Replicas

You can create a replica of an object from one model to another (or from one package to another) through drag and drop, or from the Replicate Objects dialog. The replica is a local copy of the object, all its properties (including replicas of all its sub-objects) are by default synchronized with its source object, and you can select to edit any or all of the properties and sub-objects locally, while keeping the others synchronized.

- Drag and Drop - Right-click on the object that you want to reuse in the Browser or a diagram, drag and drop it into the diagram (or onto a model or package in the Browser) where you want to create the replica, release the right mouse button, and select *Replicate Here*. Alternatively, left-click the object and drag and drop it while holding down `Shift` + `Alt` (see Moving and Copying Objects [page 103]).

  > **i Note**
  >
  > If you drag a symbol from one diagram to create a replica in another, PowerDesigner will copy the format of the initial symbol when creating the new symbol.

- Replicate Objects dialog - Select *Edit* > *Replicate Objects* in the model in which you want to create the replica to open the *Replicate Objects* dialog. Select the model and package from which to choose objects in the upper part of the dialog, select the objects to replicate in the sub-tabs in the lower part of the dialog, and then click *OK* to add them as replicas to the current model.



Sub-objects of the selected objects are replicated along with their parents. You can display sub-objects of composite objects (such as hierarchies of sub-functions inside an EAM function) for selection as replicas independent of their parents by clicking the *Include Sub-Objects* tool. Sub-objects such as table columns and indexes (which cannot exist outside their parent objects) cannot be selected for individual replication in this dialog, but you can replicate them by drag and drop from Browser to Browser.

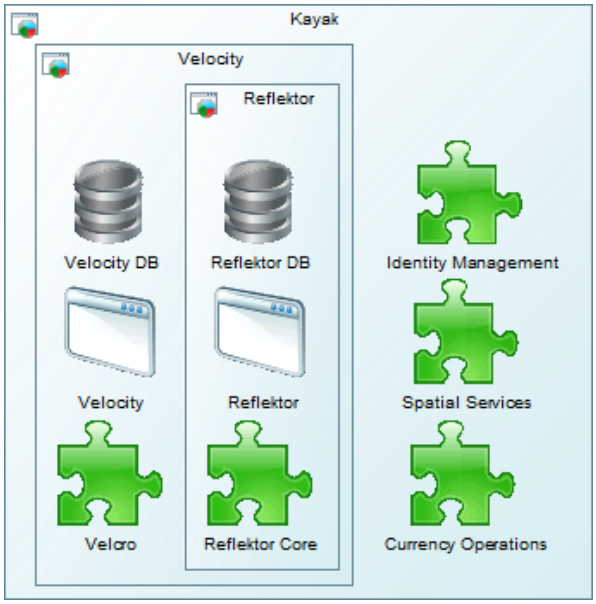
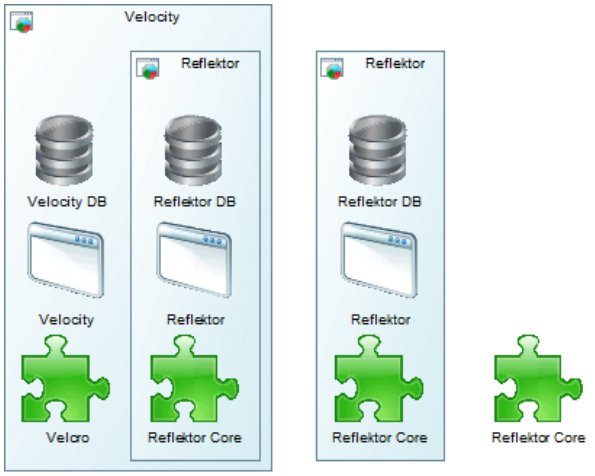
> **i** Note
>
> Objects, such as business rules, that are not sub-objects and are associated with an object, are not replicated but shortcuts to them are created in the replica if appropriate.

Name uniqueness rules are enforced when you replicate objects. If an object with the same name already exists, the replica will be renamed.

2.3.2.1 Replicating Composite EAM Objects

Composite objects in the enterprise architecture model, such as business capabilities can be decomposed into sub-objects, and these sub-objects can be decomposed into sub-objects and so on. You can replicate a composite object from any level of the hierarchy, and all the levels of sub-objects under it will be replicated as well.

When working with composite objects (such as hierarchies of sub-capabilities inside an EAM capability), you create a replica of any object at any level in the hierarchy, and the replica symbol will display all levels of sub-objects by default. In the following example, objects at each level of the **`Kayak > Velocity > Reflektor > Reflektor Core`** hierarchy of objects are replicated and display their sub-objects in the resulting replica symbols:

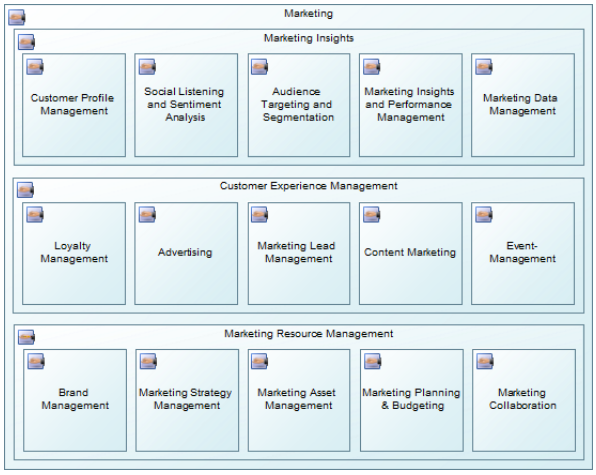
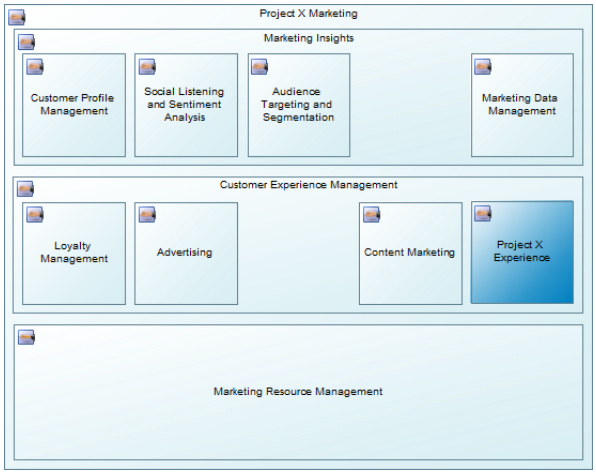
Source Object	Replicas
	

If you replicate an object by dragging and dropping a symbol from one diagram to another as a replica, the replicated object symbol will respect the formatting and content of the source symbol, including displaying only those sub-objects that are displayed on the source-symbol, and you can:

- Change the format of the replica symbol as if it were a local object, including controlling whether the composite view is shown at each level, and hiding individual sub-objects.
- Desynchronize collections of sub-objects to make them editable, and then delete sub-objects from the replica by selecting them and clicking `Delete` and create new sub-objects using the Toolbox or property sheet.
- Review and obtain any changes to desynchronized collections on the source object in your replica by right-clicking the replica in the Browser or diagram and selecting *Update Replica from Source Object*. Changes to any synchronized (non-editable) properties and collections continue to be pushed automatically to the replica.

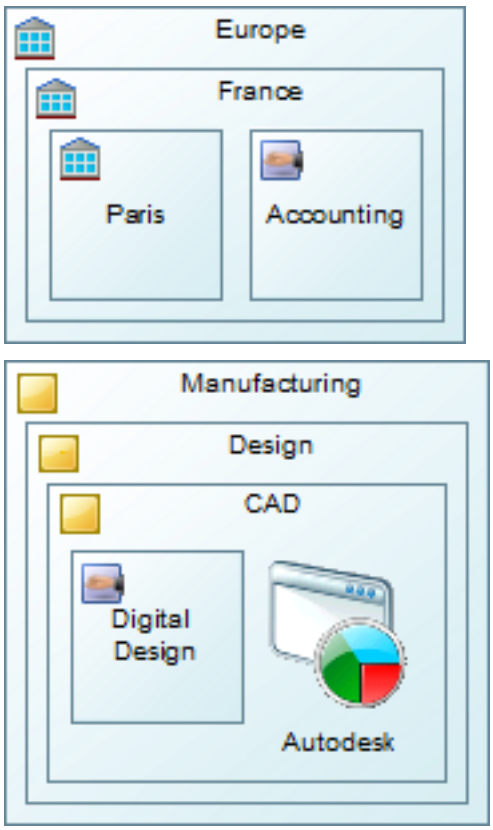
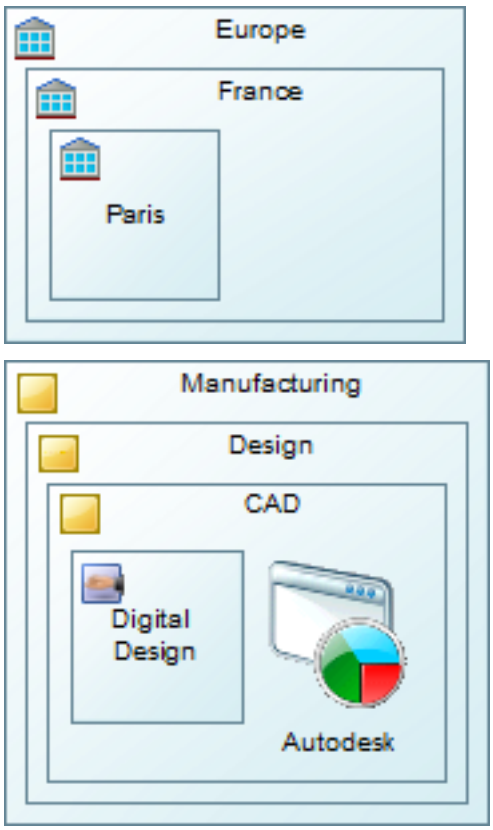
In the following example, the **Marketing** business capability is replicated and the following modifications made to the replica:

- The *Name* property has been desynchronized and changed.
- The **Marketing Insights and Performance Management** sub-sub-capability has been hidden.
- The **Customer Experience Management** *Sub-Capabilities* collection has been desynchronized, two sub-capabilities have been deleted, and **Project X Experience** has been added.
- The **Marketing Resource Management** symbol has been modified to hide its sub-capabilities by right-clicking it and selecting *Composite View* > *None*.

Source Object	Replicas
	

Only objects that are children of the composite object are replicated with it, and objects that are merely associated with it are not replicated. An exception is made for the objects associated with an area, which while not strictly sub-objects of the area, are nonetheless replicated with the area.

In the following example, the hierarchy of sites is replicated (though the business capability associated with **`France`** is not), while all the objects associated with the area are replicated with it:

Source Objects	Replicas
	

2.3.2.2 Editing the Properties of a Replica

By default, you cannot edit the properties of a replica. Their property sheets are read-only, and all their properties are synchronized with the properties of their source object.

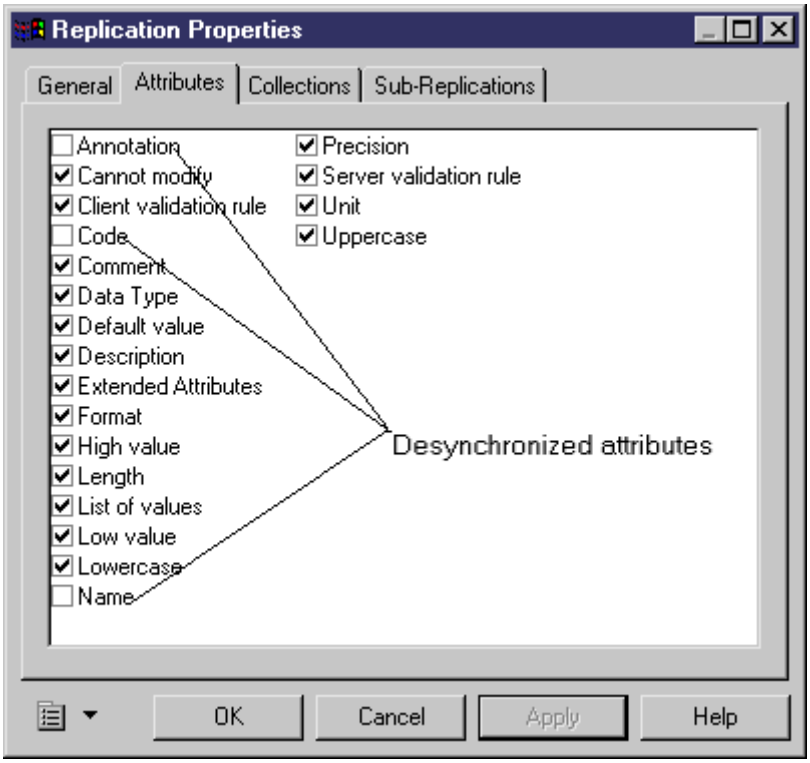
Procedure

1. Open the replica property sheet, select the *Version Info* tab, and click the *Replication Properties* button to open the replication property sheet.

   This property sheet manages the synchronization of the replica's properties with its source object. The *General* tab contains the following properties:

Property	Description
Original Object Model	Specifies the name of the model in which the source object is located. Click the *Properties* tool to the right of this field to view or edit its properties.
Full Name	Specifies the full path to the source object. Click the *Properties* tool to the right of this field to view or edit its properties.
Type	Specifies the type of the source object.
Status	Specifies the status of the source object model, which can be closed, open, or not found.
Replica Object Full Name	Specifies the full path to the replica object. Click the *Properties* tool to the right of this field to view or edit its properties.
Generate	Specifies to generate the replica along with standard models in a model-to-model generation (see Generating Shortcuts and Replicas [page 461]).

2. Select one of the following tabs and deselect the properties that you want to edit locally:
   - *Attributes* - Lists the attributes of the replica. Deselect the check box of an attribute to make it editable.



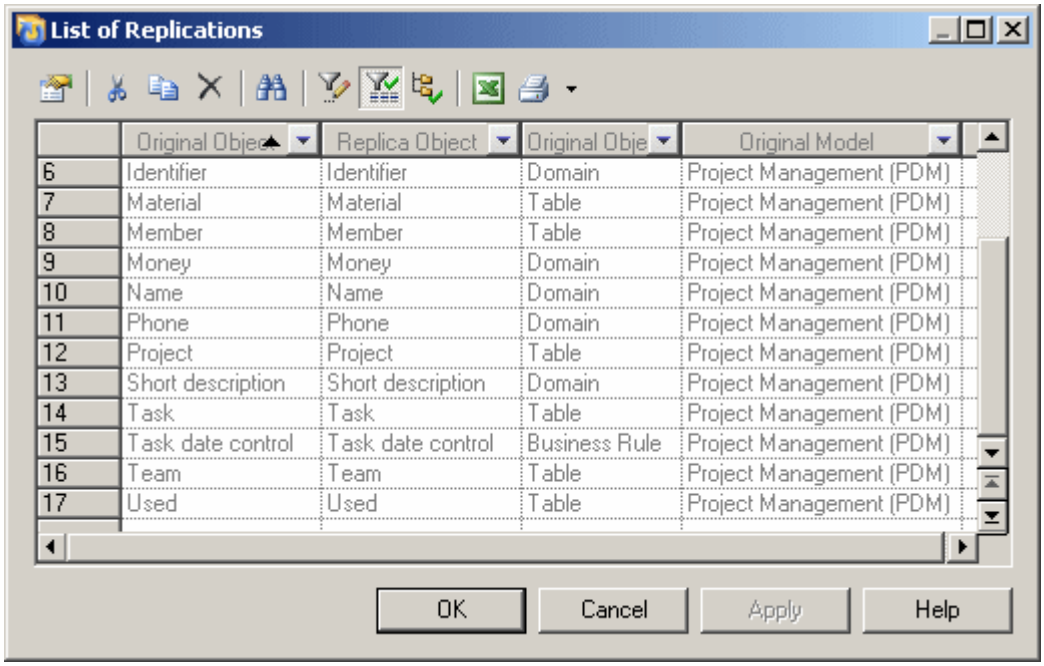
   - *Collections* - Lists the collections of sub-objects and related objects of the replica. Deselect the check box of a collection to make it editable.
   - *Sub-Replications* - Lists the sub-objects replicated as part of the object replication. Click the *Properties* tool to open the property sheet of the sub-replication in order to make one or more of its properties editable.

3. Click *OK* to close the replication property sheet and return to the property sheet of the replica. The properties that you have deselected are now editable.

> **i Note**
>
> To break the replica's link to its source object and turn it into an independent object, click the *Delete Replication* button on its *Version Info* tab. To make multiple replicas into independent objects, select *Model* > *Replications* to open the List of Replications, and delete the relevant replications.

2.3.2.3 Viewing Replicas

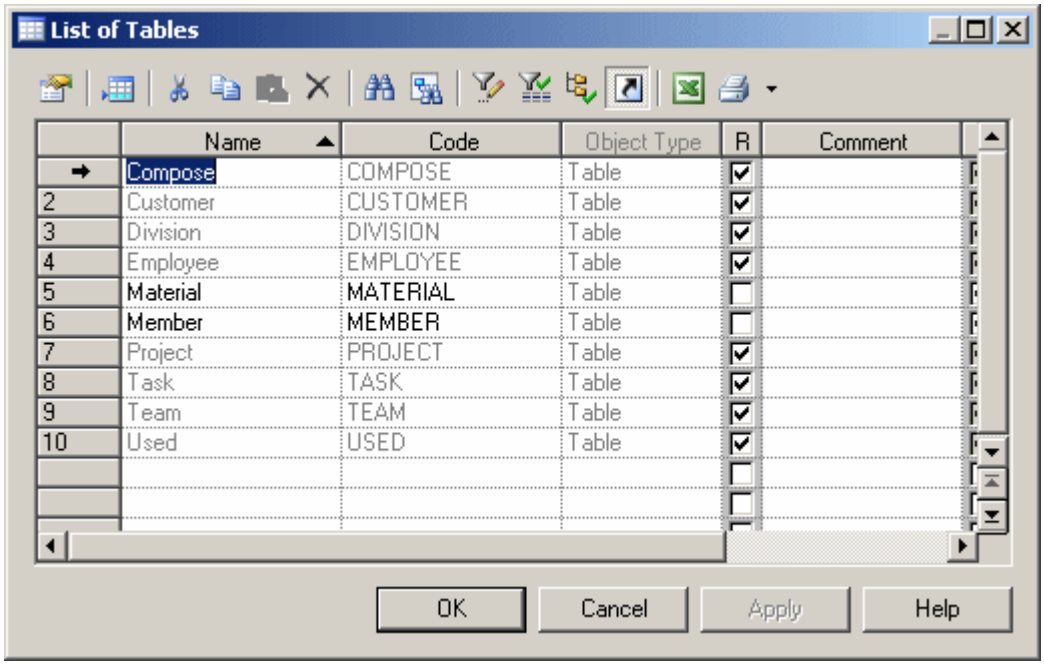
Replicas resemble normal objects, but all or part of their property sheets are read-only. The link between the replica and its target object is managed by an intermediate object, a replication, which can be opened by clicking the *Properties* tool on the replica's *Version Info* tab or by selecting *Model* > *Replications* to open the *List of Replications*.



As each replica requires a replication, you can view all the replicas present in the model from this list. You cannot create replicas here (see Creating Replicas [page 451]), but you can access the property sheets of replications and, from there, the property sheets of replicas and their target objects (see Editing the Properties of a Replica [page 455]).

If you delete a replication from the list, the associated replica will lose its link to its target object and become an independent object.

To view all the replicas for a particular type of object in a model, open the appropriate list by selecting *Model* > `<Objects>`, and look for checkmarks in the *R*[eplica] column:



If this column is not visible, click the *Customize Columns and Filter* tool, and select the **`Replica`** attribute to add it to the list.

Any property that is synchronized (and thus not editable) is grayed.

2.3.2.4 Comparing and Merging Replicas

You can compare and merge the properties of a replica with those of the original object. When merging models, to merge a replica as a replica, you must merge the replica and its associated replication. If you merge only the replica and not the replication (which links it to the original object), the replica will be transformed into an independent object.

For more information about comparison and merging see Comparing and Merging Models [page 366].

2.3.3 Synchronizing Shortcuts and Replicas

When you make changes to a target object, the shortcut or replica is synchronized instantly or, if the referencing model is closed, when it is next opened. Both the referencing and target models must be open for synchronization to occur.

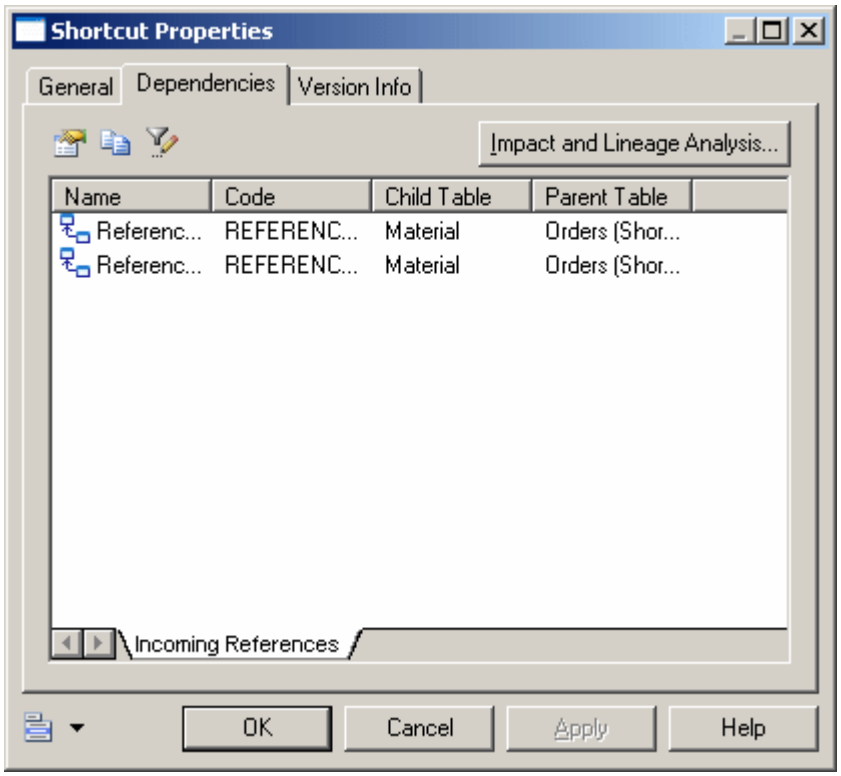
If you create an external shortcut or a replica from a target model that has never been saved, you should save the target model before the referencing model in order to allow synchronization between the models. If you attempt to save the referencing model first, you will be prompted to save the target model.

2.3.4 Viewing Shortcut and Replica Dependencies

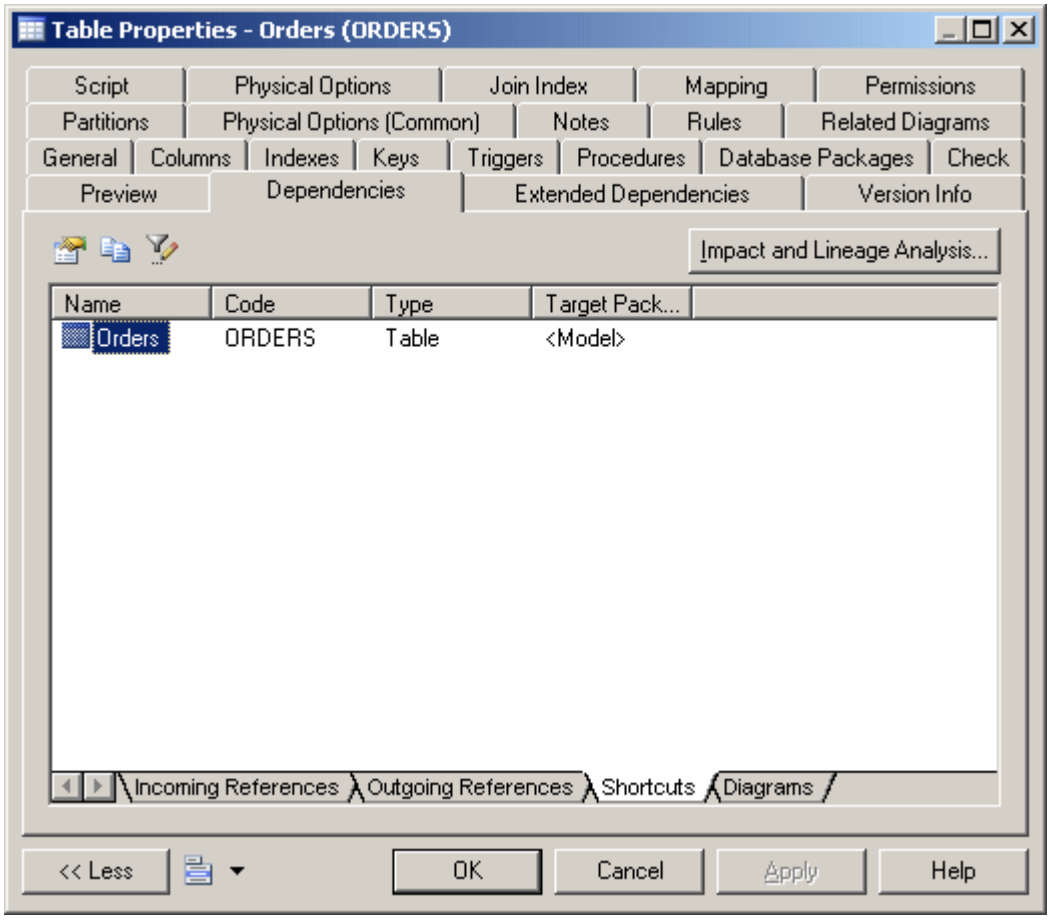
To view all the objects that are linked to or otherwise depend on a shortcut or replica, open the property sheet of the shortcut or replica and click the *Dependencies* tab. To view all the shortcuts or replicas that have been created from a target object, open the property sheet of the target object and click the *Dependencies* tab and the *Shortcuts* or *Replicas* subtab.

Context

The *Dependencies* tab on a shortcut or replica property sheet lists all the objects dependent on the shortcut or replica in all the models open in your workspace, and we recommend that you review this tab before deleting the object from your model. You can double-click an entry in the list to display the dependent object's property sheet, or click the *Impact and Lineage Analysis* button to determine the impact of deleting the shortcut or replica.



The *Shortcuts* and *Replicas* subtabs on an object property sheet *Dependencies* tab list all the shortcuts and replicas that have been created from the object in all the models open in your workspace:

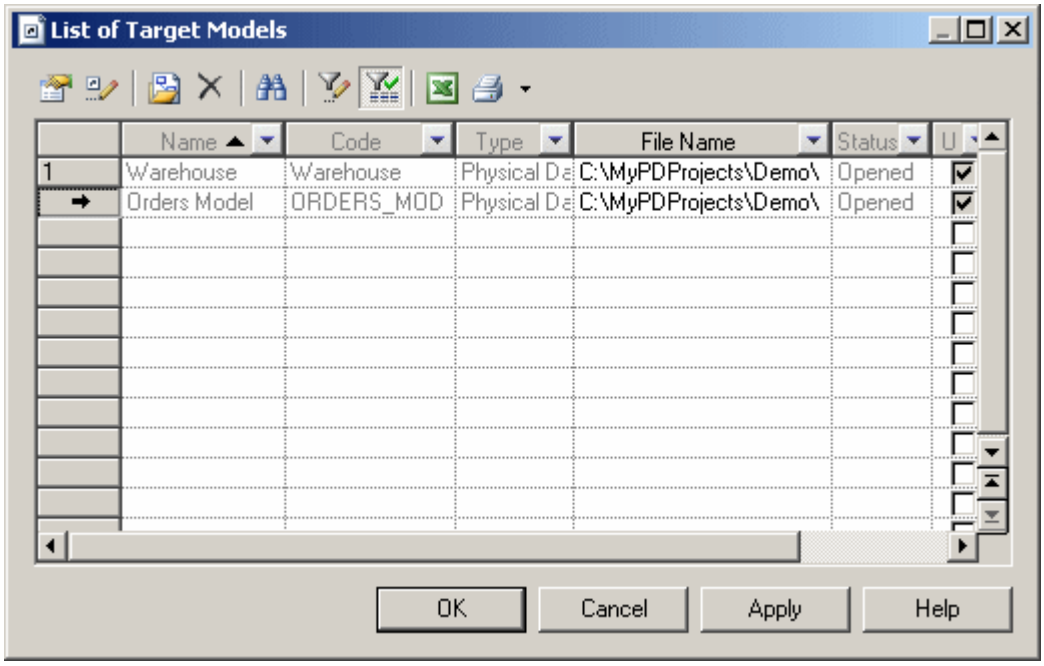


2.3.5 Working with Target Models

Shortcuts and replicas that you create yourself, as well as the shortcuts that PowerDesigner creates automatically when you link or map objects or perform other tasks, all point to objects in other models, which are called target models.

These models are listed in the Browser in the Target Models category with status icons to indicate whether your local copies are the latest available in the repository (see The Browser [page 37]).

This list of models is also available by selecting *Model* > *Target Models*.



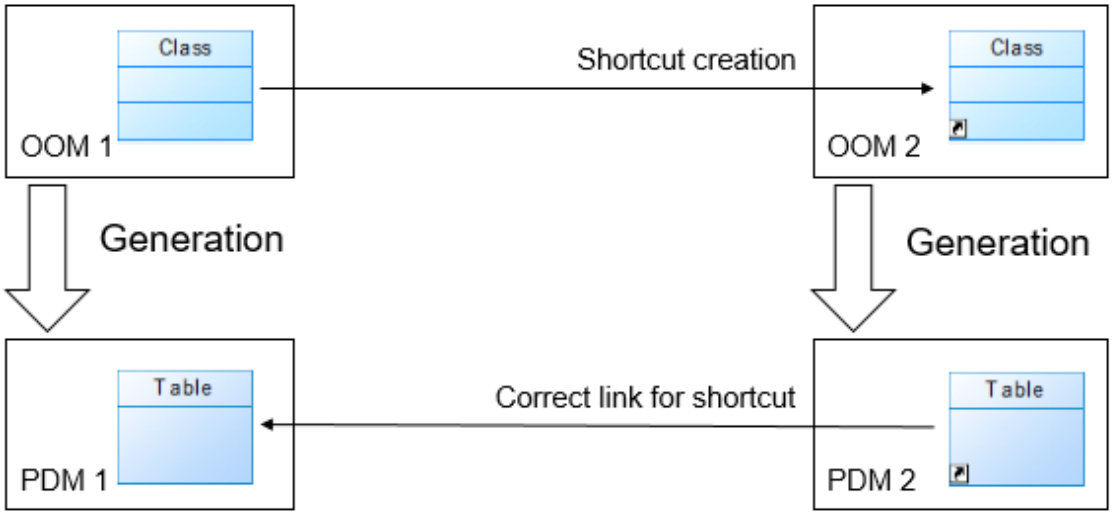
You can use the tools on this list to open a target model, view its properties, or change or delete it. For information about all the tools available on this list, see Object Lists [page 117].

> **i** Note
>
> If you delete a target model from the List of Target Models, you will also delete all the shortcuts that reference it in the present model and transform any replicas that reference it into independent objects. You cannot undo deleting or changing a target model.

2.3.6 Generating Shortcuts and Replicas

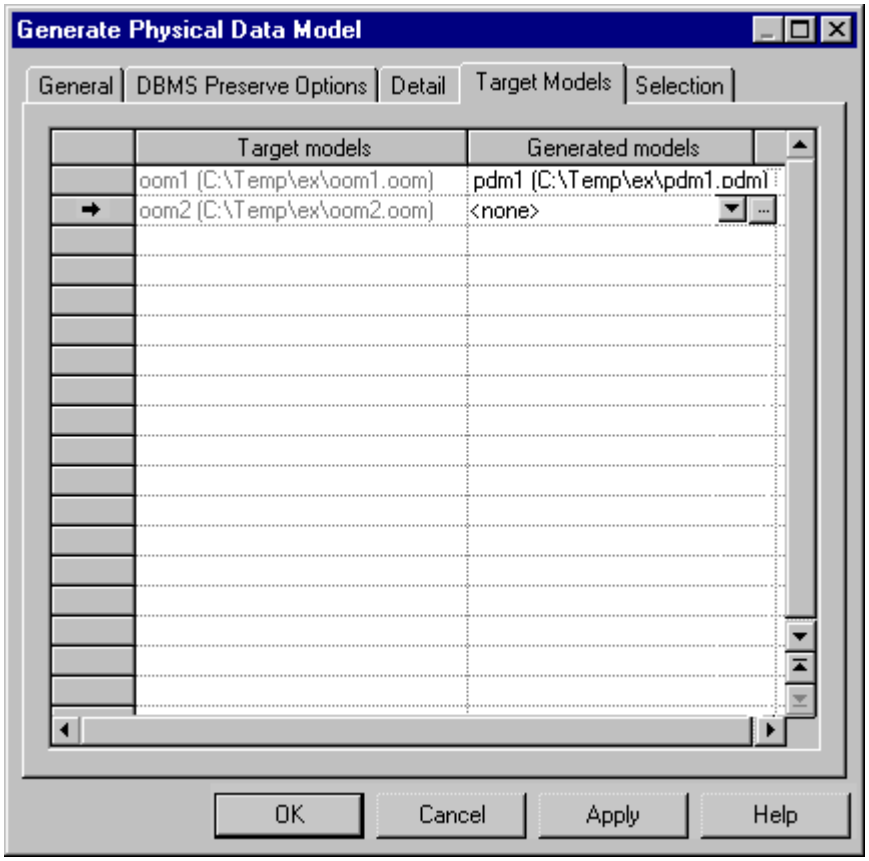
If you generate another model from your model, you can choose to generate your shortcuts as shortcuts, to transform them into independent objects, or not to generate your shortcuts and replicas. If you choose to generate your shortcuts and replicas when generating to another type of model (for instance, a PDM from an OOM), then you must first generate their target models so that they will have appropriate target objects to synchronize from.

For instance, if you have shortcuts of classes in your OOM, and want to generate a PDM from it, you should generate your target OOMs to PDMs, so that the table shortcuts generated from your class shortcuts will have target table objects to synchronize with. In the following example, the class shortcuts in OOM2 are generated to table shortcuts in a PDM2:



The necessary steps for correctly generating the class shortcuts and replicas as table shortcuts and replicas are:

1. Create the classes in OOM1, and create shortcuts or replicas to them in OOM2. Ensure that the *Generate* option on the shortcut properties sheet is selected and that the *Generated as* option is set to **`Shortcut`** (see Shortcut Properties [page 448]). For replicas, you must ensure that the *Generate* option on the associated replication property sheet is selected (see Editing the Properties of a Replica [page 455]). If this option is not selected, the replica will not be generated.
2. Generate OOM1 to PDM1 (see Generating Models [page 424]) to provide target table objects for the shortcuts and replicas to be generated from OOM2.
3. Before generating OOM2 to PDM2, review the *Target Models* tab of the model generation dialog (see Model Generation Options [page 425]) to ensure that PDM1 is listed in the *Generated Models* column. If OOM1 is closed in the workspace, you should click in the *Generated Models* column to open it, and you can then select PDM1 from the list in the cell:



4. Click *OK* to perform the generation. Your shortcuts and replicas will be retained in the generated model.

2.4 Object Mappings

Object mapping allows you to establish connections between objects belonging to heterogeneous models and diagrams. You create mappings between objects to model O/R (Object-Relational) mappings, which associate classes with tables to store OOM objects into a relational database, or simply to express a correspondence between objects in different models.

The following table lists the supported types of model-to-model mapping in PowerDesigner:

Source Model	CDM Target	LDM Target	PDM Target	OOM Target	XSM Target
CDM	X	X	X	X	
LDM	X	X	X	X	
PDM	X	X	X	X	X
OOM	X	X		X	X

Source Model	CDM Target	LDM Target	PDM Target	OOM Target	XSM Target
XSM					X

> **i Note**
>
> Mappings are also used to define data replications in the DMM. For more information, see *Data Movement Architecture > Replication Modeling > Data Movement Diagrams > Replication Processes (DMM)* .

You can create mappings:

- in the Mapping Editor - which gives you a global view of all the mappings to objects in the current (target) model (see The Mapping Editor [page 468])
- on the Mapping tab of the property sheet of the target object (see Creating Mappings from an Object's Property Sheet [page 482])
- During model generation, by selecting the *Generate Mappings* option in the Model Generation Options window (see Generating Models [page 424]).

Each mapping is recorded in a query, a textual expression or an XPATH expression defined in the target object, which permits the selection of data from the data source and its transfer to the target model. You can view and refine these mappings in the Mapping Editor.

The following tables list the objects that can be mapped for each kind of model.

CDM Mappings

CDM objects can be mapped to other model objects as follows:

CDM Source Objects	CDM Target Objects	LDM Target Objects	PDM Target Objects	OOM Target Objects
Entity	Entity, Association, Relationship	Entity, Relationship	Table	Class, Association
Entity attribute	Entity attribute, Association attribute	Entity attribute	Column	Class attribute
Relationship	Entity, Association, Relationship	Entity, Relationship	Table, Reference	Class, Association
Inheritance	Inheritance	Inheritance	-	-
Inheritance attribute	Inheritance attribute	Inheritance attribute	-	-
Association	Entity, Association	Entity, Relationship	Table, Reference	Class, Association
Association Attribute	Association attribute, Entity attribute	Entity attribute	Column	Attribute
Data item	Data item	-	-	-
Domain	Domain	Domain	Domain	Domain

> **i Note**
>
> You can map inheritances with CDM objects only when its child entities are not generated, i.e. when the "Generate children" option is deselected in the Generation tab of the inheritance property sheet.

LDM Mappings

LDM objects can be mapped to other model objects as follows:

LDM Source Objects	CDM Target Objects	LDM Target Objects	PDM Target Objects	OOM Target Objects
Entity	Entity, Relationship, Association	Entity, Relationship	Table	Class, Association
Entity attribute	Entity attribute	Entity attribute	Column	Attribute
Relationship	Entity, Relationship	Entity, Relationship	Table, Reference	Class, Association
Inheritance	Inheritance	Inheritance	Reference	Generalization
Inheritance attribute	Inheritance attribute	Inheritance attribute	-	-
Domain	Domain	Domain	Domain	Domain

PDM Mappings

PDM objects can be mapped to other model objects as follows:

PDM Source Objects	CDM Target Objects	LDM Target Objects	PDM Target Objects	OOM Target Objects	XSM Target Objects
Table	Entity, Relationship	Entity, Relationship	Table, Fact,Dimension	Class, Association	Element, Complex type
Table column	Entity attribute	Entity attribute	Table column, Measure, Dimension attribute	Attribute	Element, Complex type
View	-	-	Fact,Dimension	Class	-
View column	-	-	View column, Table column, Measure, Dimension attribute	Attribute	-
Reference	Relationship	Relationship	Reference	Class, Association	-
Domain	Domain	Domain	Domain	Domain	-

OOM Mappings

OOM objects can be mapped to other model objects as follows:

OOM Source Objects	CDM Target Objects	LDM Target Objects	OOM Target Objects	XSM Target Objects
Class	Entity, Association, Relationship	Entity, Relationship, Inheritance	Class	Element, complex type
Attribute	Entity attribute, Association attribute	Entity attribute, Inheritance attribute	Attribute	Element, Complex type
Generalization	-	Inheritance	Generalization	-
Association	Association	Entity, Relationship	-	-
Domain	Domain	Domain	Domain	-

XSM Mappings

XSM elements and complex types and their attributes can only serve as source objects for mapping to other XSM elements, complex types and attributes, but they can serve as the targets for mappings from OOM or PDM source objects.

2.4.1 Creating a Data Source

A data source provides a connection to one or more other models for mappings and (in the PDM) can also provide a connection to a database. You are prompted to create a data source with a wizard the first time you open the Mapping Editor or click the *Create Mapping* tool on an object property sheet *Mappings* tab.

Procedure

1. On the Data Source Identification page, enter an appropriate name for the data source, and then specify an access type, which, for PDM data sources, controls the types of queries available in the *Mappings* pane, and the availability of forward (Source > Target) and reverse (Target > Source) mappings. You can choose between:
    - Read/Write – Enables Criteria, Select, Insert, Update and Delete queries.
    - Read-Only - Enables Criteria and Select queries. You can only create forward mappings, and the data source is not displayed during reverse mapping (see Creating Reverse Mappings [page 481]).
    - Write-Only - Enables Criteria, Insert, Update and Delete queries. You can only create reverse mappings, and the data source is not displayed during forward mapping.
2. Select the type of the model to use as the data source, and click *Next* to go to the Source Model Selection page.

3. Select one or more models from the list of open models in the workspace, and then:
   - For data sources created from an object's *Mapping* tab - click *Finish* to create the data source and open a selection dialog to choose the external object to map to.
   - For data sources created from the Mapping Editor - click *Next* to go to the *Options* page. By default, the *Create default mappings* option, which creates mappings where possible based on shared names, is selected. Click *Finish* to create the data source and proceed to the Mapping Editor.

To view or edit a data source's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator. The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Model Type	Specifies the type of the models referenced in the data source.
Access type	For PDM data sources, controls the types of queries available in the *Mappings* pane, and the availability of forward and reverse mappings.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

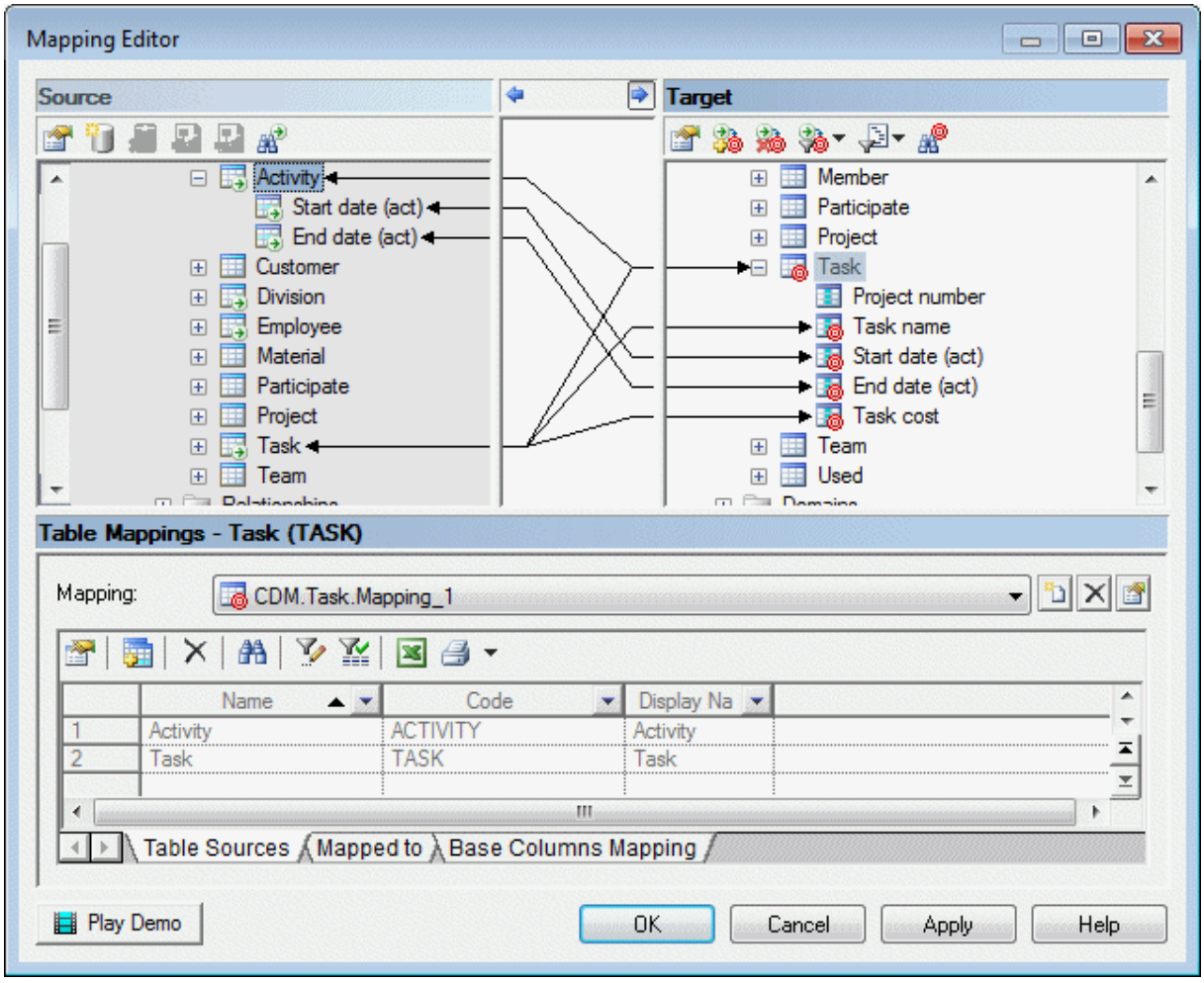
The following tabs are also available:

- *Models* - Lists the models associated with the data source. Use the *Add Models* tool to add models open in the current workspace.
- *Database Connection* - [for data sources created in a PDM] Contains the parameters needed to connect to the database associated with the data source. Click the *Select a Data Source tool* to specify an ODBC data source or connection profile (see Connecting to a Data Source [page 29]).
- *Cost* - [for data sources created in a PDM that supports lifecycle modeling] Lets you specify the cost per GB of storage for the data source along with the compression rate, which is used to estimate the relative size of the data once it has been loaded to the data warehouse (where 100% indicates no compression).
- *Data Movement (Lifecycle)* - [for data sources created in a PDM that supports lifecycle modeling] Lets you define parameters to connect to a remote database, whose data will be loaded into the warehouse.

> **i Note**
>
> For information about the *Create Data Source* and *Add Model to Data Source* tools available in the Mapping Editor, see Mapping Editor Tools [page 480].

2.4.2 The Mapping Editor

The Mapping Editor provides a graphical interface for creating and viewing mappings between models or between metamodels. It provides a global view of all the mappings to objects in your model, allowing you to quickly identify those which are mapped and unmapped.



To open the Mapping Editor, select *Tools* > *Mapping Editor*. If no data source is defined in your model, you will be prompted to create one (see Creating a Data Source [page 466]).

The Mapping Editor window is divided into three panes:

- *Source* – on the left side has a gray background and contains one or more data sources, which each contain one or more models. Expand the model nodes to display all their mappable objects and sub-objects.

  > **i Note**
  >
  > To expand all nodes, click in the pane and press the number pad * key.

- *Target* – on the right side has a white background and contains all the mappable objects and sub-objects in your model. The details of any mappings you create are saved in the objects in this model.

- *Mappings* - at the bottom contains either a list of mappings (when a model, package or folder is selected) or the selected object's property sheet *Mappings* tab (see Object and Sub-Object Mapping Properties [page 485]).

Mappings are represented as links drawn between a source and a target object, with an arrow at one or both ends to indicate in which directions data can flow. Double-clicking the link in the *Source* or *Target* pane opens the property sheet of the source or target object, while clicking it in the area between the panes, displays the properties of the mapping in the *Mappings* pane.

An overlay is displayed on the bottom right corner of objects bearing mappings (and, for sub-objects, their parent objects) and a smaller overlay on the containing model, package or folder (or, for XSM elements, ancestor elements):

Mapped Object		Containing Model, Package, or Folder	
Source	Target	Source	Target

> **Note**
>
> Click the Play Demo tool in the lower-left corner of the Mapping Editor window to launch a video that briefly illustrates its main features.

2.4.2.1 Model-to-Model Mappings

You can create mappings in the Mapping Editor by drag and drop, or by using the *Create Mapping* tool or contextual command. You can select multiple objects and sub-objects in the *Source* pane using the `Ctrl` or `Shift` keys, but only one object in the *Source* pane.

To create a mapping, perform one of the following actions:

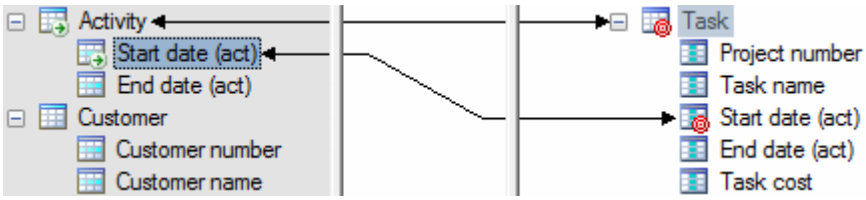
- Drag and drop - Select one or more objects and sub-objects in the *Source* pane and drag and drop them onto an object or sub-object in the *Target* pane. The pointer becomes a barred circle when it is over an object to which the selected objects cannot be mapped, and if one source object in a multi-selection would create an invalid mapping, the whole selection is rejected.
- *Target* pane - Select an object or sub-object in the *Target* pane and (optionally for objects), one or more objects and sub-objects in the *Source* pane and click the *Create Mapping* tool (or right-click one of the selected objects and select *Create Mapping*). If the selected source objects are ambiguous or would create an invalid mapping, a selection dialog opens, allowing you to select the source objects to map to.
- *Mappings* pane - Select an object or sub-object in the *Target* pane and click the *Create Mapping* tool in the *Mappings* pane. For objects, a selection dialog opens, allowing you to select the source objects to map to. For sub-objects, an empty mapping is created, which you should complete by selecting a source object on the *Sources* sub-tab.
- Default mappings - To create default mappings between source and target objects that have the same name, right-click a data source and select *Generate Default Mappings*.

> **Note**
>
> If a mapping already exists between a source and target object, then dragging and dropping will reuse and add additional information to the existing mapping object, while using the *Create Mapping* tool or contextual command will create a second mapping object. Each mapping object is available in the *Mapping* list at the top of the *Mappings* pane.

Mappings can be created for combinations of source and target objects and sub-objects as follows:

- Source object mapped to target object - The objects are mapped together, and any sub-objects that have the same name and code are also mapped together. A mapping overlay is displayed on each of the source and the target objects and sub-objects mapped, and a link is drawn between them:

  The mapping details are shown in the *Mappings* pane. The source objects are shown on the *Sources* sub-tab, and any sub-objects are shown on the `<Sub-object>` *Mappings* sub-tab.
- Source sub-object mapped to target sub-object – The sub-objects and their parent objects are mapped together and the mapping is added to the list on the `<Sub-object>` *Mappings* sub-tab of the parent object.

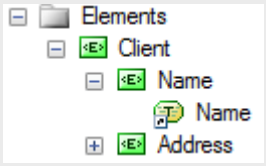
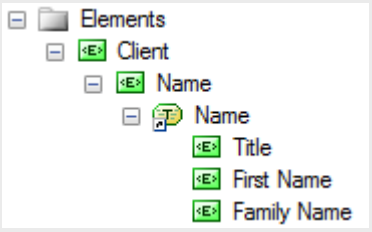


- Source sub-object mapped to target object – The parent source object is mapped to the target object.
- Source object mapped to target sub-object – Not permitted.

Mappings (including sub-object mappings) are always stored in target objects (for example, a mapping between a source CDM entity attribute and a target PDM table column will be stored in the table).

> **Note**
>
> In an XSM, elements can appear at any level in the hierarchy, and objects and sub-objects can be freely mapped to any element. Each element (whether parent, child, or grandchild) stores its own mappings, except in the case of elements bearing a complex type, where all mappings to any sub-elements are stored on the element bearing the type (though they are displayed on the sub-elements).

If you open the Mapping Editor with a source or target XSM containing shortcuts to complex types in a closed model, PowerDesigner will offer to open the model for you. If you choose not to open the model at this time, you can do so at any time by right-clicking a shortcut and selecting *Open Model*:

Shortcuts Closed	Shortcuts Open
	

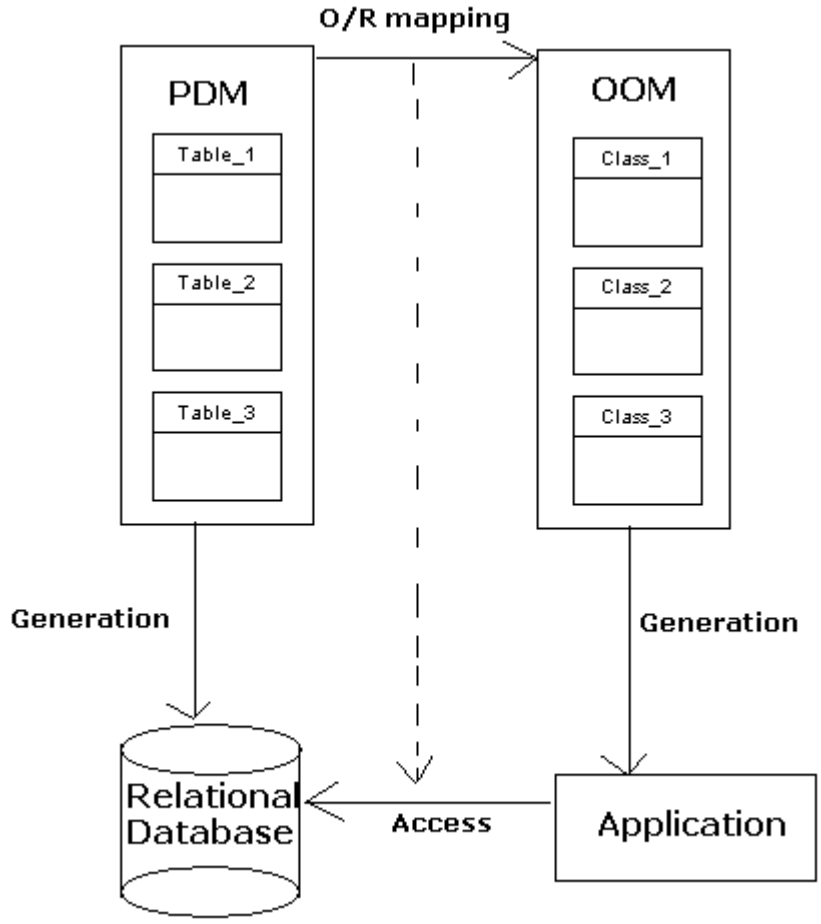
2.4.2.2 Object to Relational (O/R) Mappings

Developers tend to use object-oriented programming languages like Java, to develop business objects and components. These objects can be stored in a database. A problem arises when the user tries to store objects in a relational database because object modeling describes a system through objects that have identity, behavior and encapsulated state whereas relational modeling describes a system by its data.

In addition, object language object codes are often different than the codes used to identify tables and columns in a relational database, so that you will need to modify them when generating an OOM into a PDM or vice versa.

You can use object mapping to bypass this impedance-mismatch.

The following schema illustrates the link between classes and tables to store objects in a relational database:



Object persistence implies to store and extract objects in a relational database.

Mapping Classes with PDM Objects

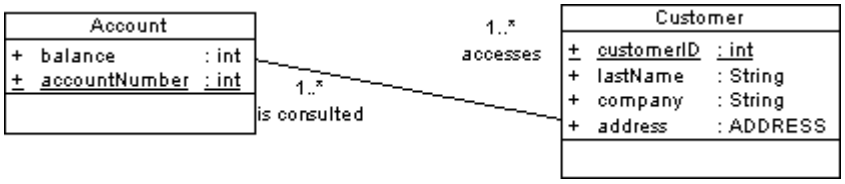
When a class inherits from a non-generated class via a generalization link, the attributes of the class appear in the Selection dialog box in order to let you create a mapping with these inherited attributes. The attributes of a non-generated derived class also appear in the Selection dialog box.

Mapping Associations with PDM Objects

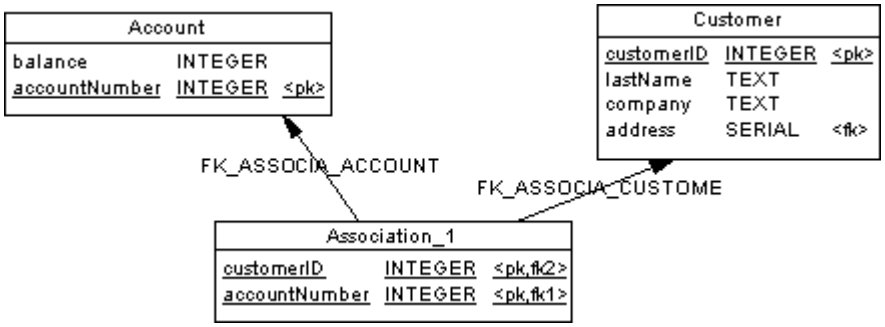
O/R mapping on an association allows you to define the role navigability of an association in the database. When an association role is not navigable, there is no need to design a mapping for the association since no information is transmitted between the classes. However, when an association role is navigable, you have to set up the structure for data transmission within the database. In relational databases, data transmission is implemented via foreign keys designed to relate a record in one table with a record in another.

The type of mapping for an association depends on the association multiplicity:

- One-to-one or one-to-many associations can be mapped to a reference in the source model. This reference is used in the database to transfer data and migrate key columns to the appropriate table
- Many-to-many associations have to be mapped to an associative table. This table is created to maintain a relationship between two or more tables in a relational database. The columns contained in the associative table are the combination of the keys in the tables involved in the reference. For example, the following classes have a many-to-many association:



In a relational database, this association is mapped to the following associative table:



SQL Queries

When the association roles are navigable, the following queries are automatically computed:

Query	Action
Select (Role A)	Retrieves related class A instances for the role A defined on class B
Insert (Role A)	Associates an instance of class A with class B. It is computed only when the source association is an associative table
Delete (Role A)	Deletes from class B the association with class A. This query is computed only when the source association is an associative table
Select (Role B)	Retrieves related class B instances for the role B defined on class A
Insert (Role B)	Associates an instance of class B with class A. This query is computed only when the source association is an associative table

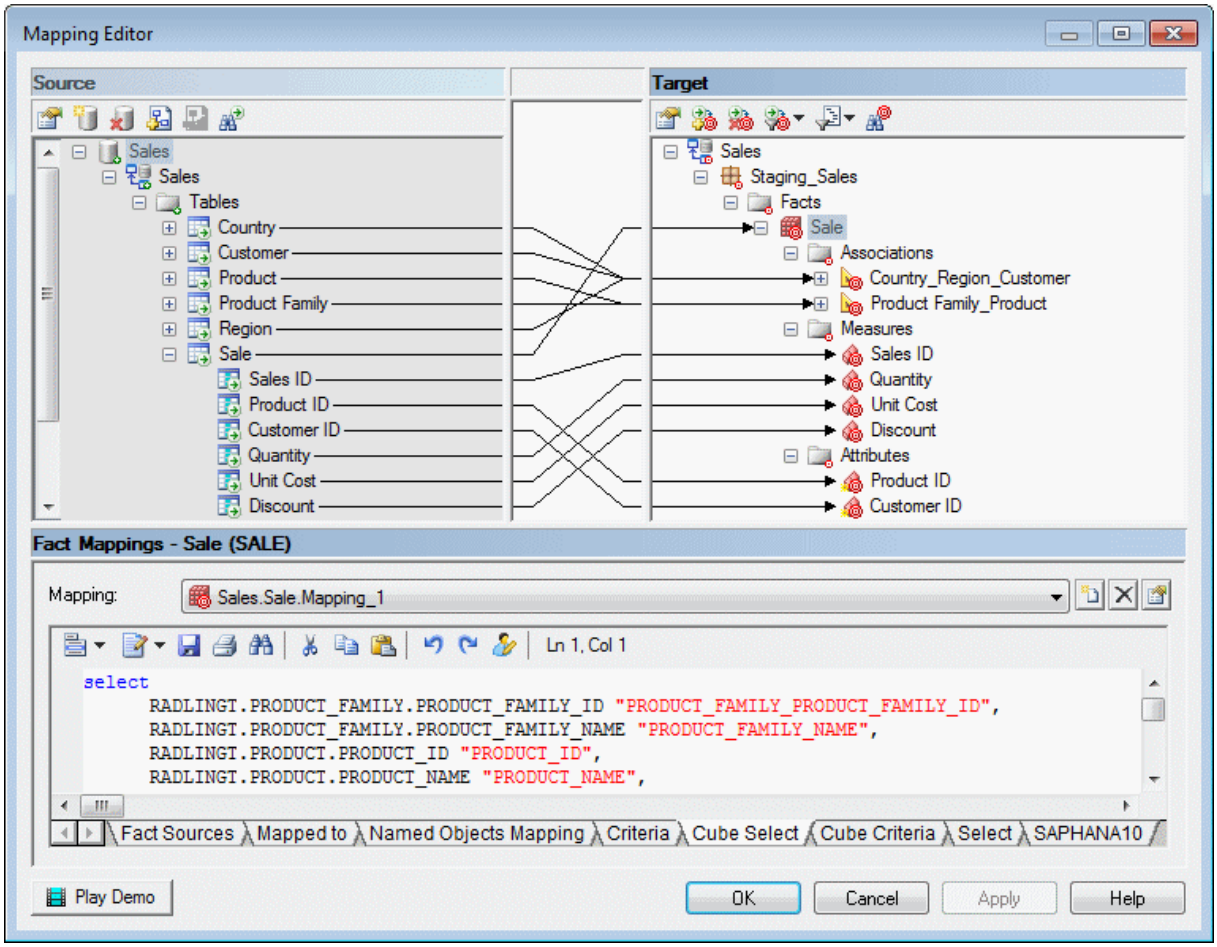
Query	Action
Delete (Role B)	Deletes from class A the association with class B. This query is computed only when the source association is an associative table

2.4.2.3 Operational to Warehouse Data Mappings

Data warehousing requires the extraction, transformation, and loading of data from operational systems to a data warehouse database. You can create mappings between operational and data warehouse data and from the data warehouse data and OLAP cubes. To review or edit these mappings, open your multidimensional diagram, and then select *Tools* > *Mapping Editor*.

You can model operational and data warehouse data structures in PDMs, and specify mappings between the operational data sources and the data warehouse to generate extraction scripts to populate the data warehouse with operational data. In this kind of relational-to-relational mapping, operational tables are mapped to data warehouse tables with a type of fact or dimension, and operational columns are mapped to warehouse columns.

The Generate Cube wizard automatically creates mappings between source tables and facts and dimensions and you can modify these or manually create mappings between these objects:



The *Select* sub-tab displays the SQL statement used to select data in the data source. The Generate Cube Data wizard uses this SQL statement to fill the text files used to populate cubes in an OLAP database.

2.4.2.4 XML Import Mappings

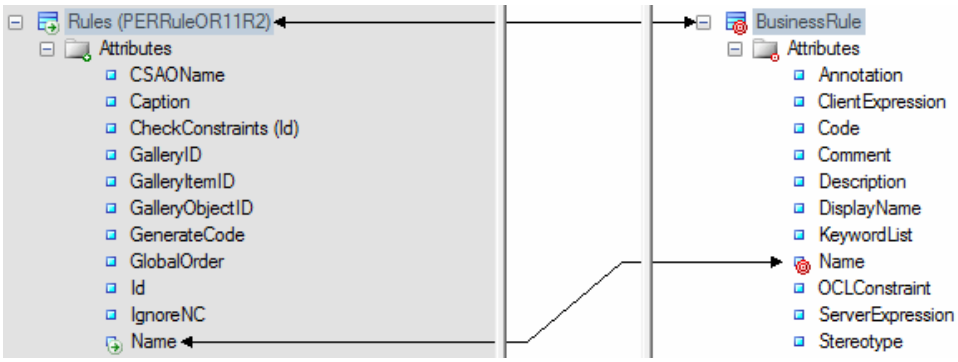
You control how elements defined in an XML schema are imported by mapping them and their attributes, compositions, and aggregations to objects in the PowerDesigner metamodel. The XML schema is analyzed and presented as a list of metaclasses on the left side of the Mapping Editor and the PowerDesigner metamodel (and any extensions) are displayed on the right side.

Context

> **i** Note
>
> It is not necessary to map all metaclasses (or all their contents), but only those with which you want to work. If the PowerDesigner metamodel does not contain appropriate metaclasses, attributes, compositions, or aggregations to map against, you can create them dynamically here or save any existing mappings, close the Mapping Editor, define or attach appropriate extensions, and then reopen the Mapping Editor to map to them.

Procedure

1. Drag and drop an external metaclass to a PowerDesigner metaclass to create an import mapping. Any external attributes and collections are automatically mapped to PowerDesigner attributes with which they share a name:

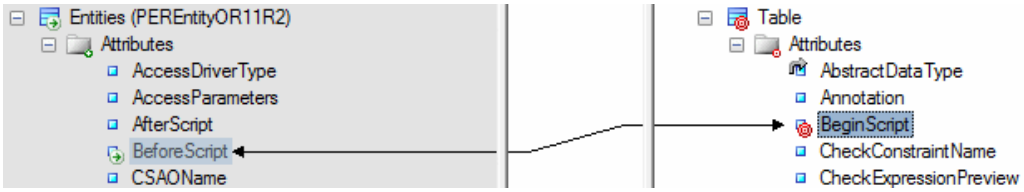


   By default, the Mapping Editor lists the standard attributes and collections of metaclasses, which are normally displayed in object property sheets. To display all available properties, click the *Filter Properties*

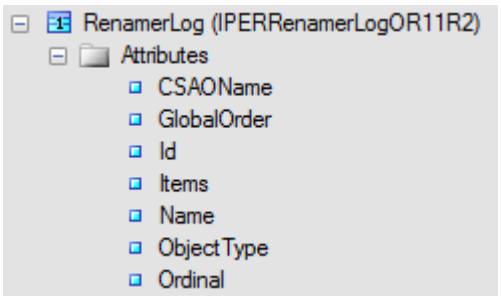
tool, and select **Show All Properties**. You can also filter the tree by using the *Filter Mappings* and *Filter Objects* tools.

> **i Note**
>
> If no suitable metaclass exists, to create and map to a new extended metaclass based on the **ExtendedObject** metaclass, drag and drop the external metaclass onto the PowerDesigner metamodel root.

2. Drag and drop additional attributes under the metaclass to PowerDesigner attributes with compatible data types to create mappings for them. Attributes are contained in a folder under the metaclass and represent individual properties such as Name, Size, DimensionalType, which have boolean, textual, numeric, or object ID values:



PowerDesigner identifies sub-object metaclasses in the schema that are limited to a single instance and displays a **1** overlay on their icons. Attributes under such metaclasses are treated as belonging to the parent metaclass and can be mapped to attributes under the PowerDesigner object with which the parent is mapped:



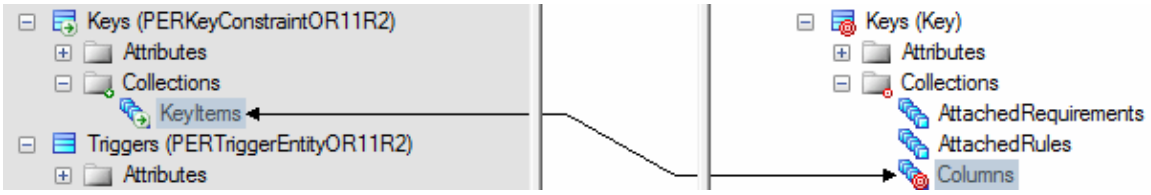
> **i Note**
>
> If no suitable attribute exists, to create and map to a new extended attribute, drag and drop the external attribute onto the PowerDesigner metaclass to which its parent is mapped.

3. Drag and drop external sub-object metaclasses (compositions) under the metaclass to PowerDesigner compositions to create mappings between them:

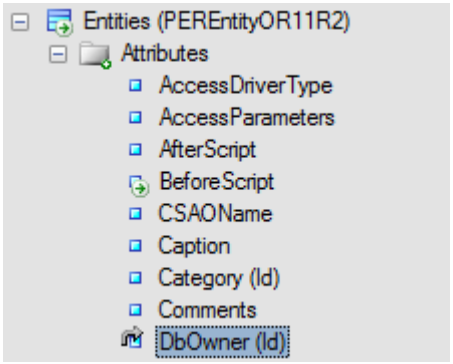
Any attributes under the sub-object metaclass are automatically mapped to PowerDesigner attributes with which they share a name. Map other sub-object attributes as necessary.

> **i Note**
>
> In certain circumstances, it may be appropriate to map an external sub-object metaclass to a PowerDesigner object metaclass, and so such mappings are also permitted.

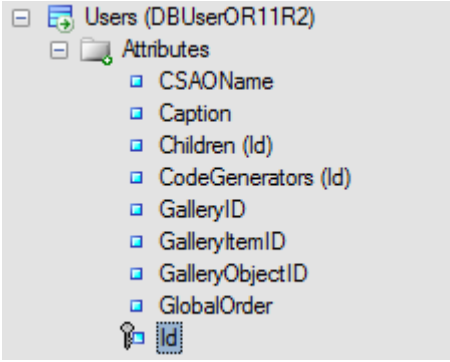
4. Drag and drop external collections (aggregations) under the metaclass to PowerDesigner collections to create mappings between them:



5. In certain schemas, it may be necessary to identify attributes as references and identifiers to link one metaclass to another through aggregation:
   a. Right-click an attribute and select *Declare as Object Reference* to specify that it acts as a pointer to another object. Such attributes often have a type of GUID, Token, or NCName (PowerDesigner automatically identifies attributes of type IDRef as references). A rounded arrow overlay is added to the attribute icon:



   b. Open the metaclass that the object reference points to, select its identifying attribute, right-click it, and select *Declare as Unique Identifier*. A key overlay is added to the attribute icon:



c. The object reference attribute can now be mapped to a PowerDesigner attribute of type object (which also bears a rounded arrow overlay):

6. [optional] Select a metaclass and enter an initialization or post-processing script to modify the objects at or after creation (see Metamodel Mapping Properties [page 487]).
7. [optional] Click the target model (root node) to display the global list of mappings in the *Mappings* pane at the bottom of the dialog and use the arrows at the bottom of the list to change the order in which objects are imported to ensure that dependencies are respected.

> **i** Note
>
> To control the order in which attributes, compositions, and aggregations are imported within objects, select the target metaclass to display its mappings in the *Mappings* pane, and use the arrows at the bottom of the lists on the *Attribute Mappings*, *Collection Mappings*, and *Sub-Object Mappings* sub-tabs.

8. Click *Apply* to save your changes.

2.4.2.5 Model-to-Model Generation Mappings

You control how metaclasses from one PowerDesigner model type will be generated to metaclasses in another model type by mapping them and their attributes and collections in the Mapping Editor. Any extensions defined for the source or target metamodels are displayed and available for mapping.

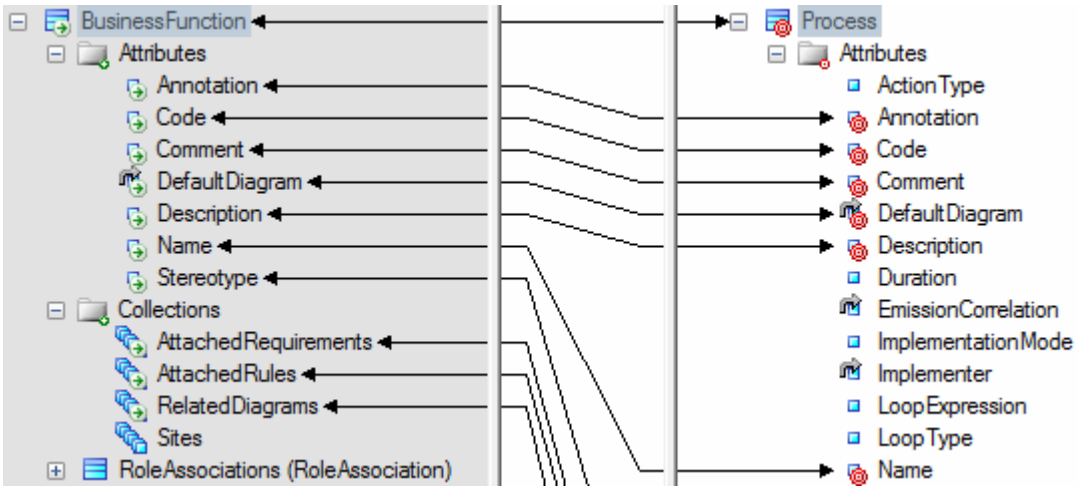
Context

> **i** Note
>
> It is not necessary to map all metaclasses (or all their contents), but only those with which you want to work. If the PowerDesigner metamodel does not contain appropriate metaclasses, attributes,

compositions, or aggregations to map against, you should save any existing mappings, close the Mapping Editor, define or attach appropriate extensions, and then reopen the Mapping Editor to map to them.

Procedure

1. Drag and drop a metaclass from the source pane on the left to a metaclass in the Target pane on the right. Any source attributes are automatically mapped to target attributes with which they share a name:



> **i Note**
>
> By default, the Mapping Editor lists the standard attributes and collections of metaclasses, which are displayed, by default, in object property sheets. To display all available properties, click the *Filter Properties* tool, and select **`Show All Properties`**. You can also filter the tree by using the *Filter Mappings* and *Filter Objects* tools.

2. Drag and drop additional source attributes under the metaclass to target attributes with compatible data types to map them. Attributes are contained in a folder under the metaclass and represent individual properties such as Name, Size, DimensionalType, containing boolean, textual, numeric, or object ID values:
3. Drag and drop source sub-object metaclasses (compositions) under the metaclass to target compositions to create mappings between them:

   Any attributes under the source sub-object metaclass are automatically mapped to target attributes with which they share a name. Map other sub-object attributes as necessary.

   > **i Note**
   >
   > In certain circumstances, it may be appropriate to map a source sub-object metaclass to a target object metaclass, and so such mappings are also permitted.

4. Drag and drop source collections (aggregations) under the metaclass to target collections to create mappings between them:
5. [optional] Select a metaclass and enter an initialization or post-processing script to modify the objects at or after creation (see Metamodel Mapping Properties [page 487]).

6. [optional] Click the target model (root node) to display the global list of mappings in the *Mappings* pane at the bottom of the dialog and use the arrows at the bottom of the list to change the order in which objects are generated to ensure that dependencies are respected.

   > **i** Note
   >
   > To control the order in which attributes, compositions, and aggregations are generated, select the target metaclass to display its mappings in the *Mappings* pane, and use the arrows at the bottom of the lists on the *Attribute Mappings*, *Collection Mappings*, and *Sub-Object Mappings* sub-tabs.

7. Click *Apply* to save your changes.

2.4.2.6 Mapping Editor Tools

The Mapping Editor *Source* and *Target* panes contain tools to help you create and review your mappings.

The following tools are available in the *Source* pane:

Tool	Description
	Properties - Opens the property sheet of the selected source object.
	Create Data Source - Launches the Data Source Creation Wizard (see Creating a Data Source [page 466]).
	Delete Data Source - Deletes the selected data source, along with any associated mappings. You can delete all the mappings from a source object by selecting it and pressing *Delete* or right-clicking it and selecting *Delete Mapping*. You can delete a source object from the source model by selecting it and pressing *Shift+Del* or right-clicking it and selecting *Delete*).
	Add Model to Data Source - Adds one or more source models of the same type to the selected data source.
	Remove Model from Data Source - Deletes the selected model from the data source, along with any associated mappings.
	Find Source Object - Opens a Find dialog to search for an object in the *Source* pane.

The following tools are available in the *Target* pane:

Tool	Description
	Properties - Opens the property sheet of the selected target object.
	Create Mapping - Creates a mapping between the selected source and target objects (see Model-to-Model Mappings [page 469]).
	Delete Mappings - Deletes all the mappings for the selected target object. Alternately, press *Delete* or right-click the object and select Delete Mapping. You can delete a target object from its model by selecting it and pressing *Shift+Del* or right-clicking it and selecting *Delete*.
	Filter Mappings - Filters the display of mapping lines to show all mappings or only the mappings on the selected object or those on the selected object and its sub-objects.
	Filter Objects - Filters the display of both the Source and Target trees to show all objects or to show only those objects with or without mappings.
	Find Target Object - Opens a Find dialog to search for an object in the *Target* pane.

2.4.2.7 Creating Reverse Mappings

The default forward mappings define how to populate the objects of the current model with those of the data source model. Reverse mappings are typically used in PDM-OOM mapping when, for example, data has several sources and you want to define how the data in the OOM is inserted, updated or deleted in the PDM.

To switch to reverse mapping mode, click the left arrow at the top of the area between the *Source* and *Target* panes. The left pane is renamed to *Target* and the right pane to *Source*, and the mapping tools which appear at the top of each pane are switched:

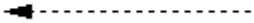
> **i Note**
>
> Mappings are always saved in the model on the right-hand side (the model from which you launched the Mapping Editor), whether the pane is called *Target* or *Source*.

If the access type of your data source is Read/Write (see Creating a Data Source [page 466]), then each mapping that you create is, by default, bi-directional (with an arrow at each end). To create a:

- Read-only (ie forward-only mapping), create the mapping in forward mode and then switch to reverse mode and select and delete it. The mapping link now becomes dotted and has an arrow only on its right end.

- Write-only (ie reverse-only mapping), create the mapping in reverse mode and then switch to forward mode and select and delete it. The mapping link now becomes dotted and has an arrow only on its left end.

When you create uni-directional mappings in this way, the following types of link line can be visible:

Forward Mode	Mapping Type	Reverse Mode
	Forward and reverse (read-write) mapping	
	Forward only (read-only) mapping	
	Reverse only (write-only) mapping	

Solid lines indicate that you can edit the details of the mapping in the *Mappings* pane (which, in reverse mode, is renamed to *Reverse Mappings*). In reverse mode, you can modify the default reverse mapping by editing the *Insert*, *Update* and *Delete* sub-tabs.

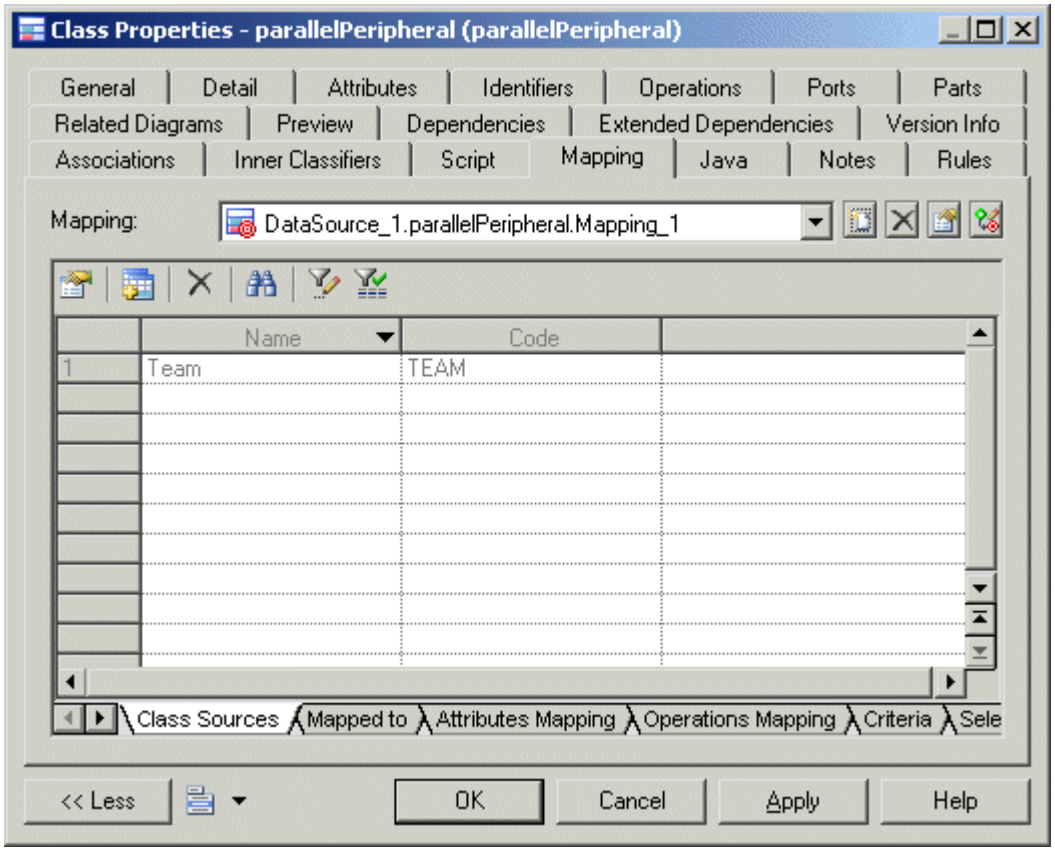
2.4.3 Creating Mappings from an Object's Property Sheet

You can create mappings to an individual object on the *Mapping* tab of its property sheet. Mappings created in this way will be visible the next time you open the Mapping Editor.

Procedure

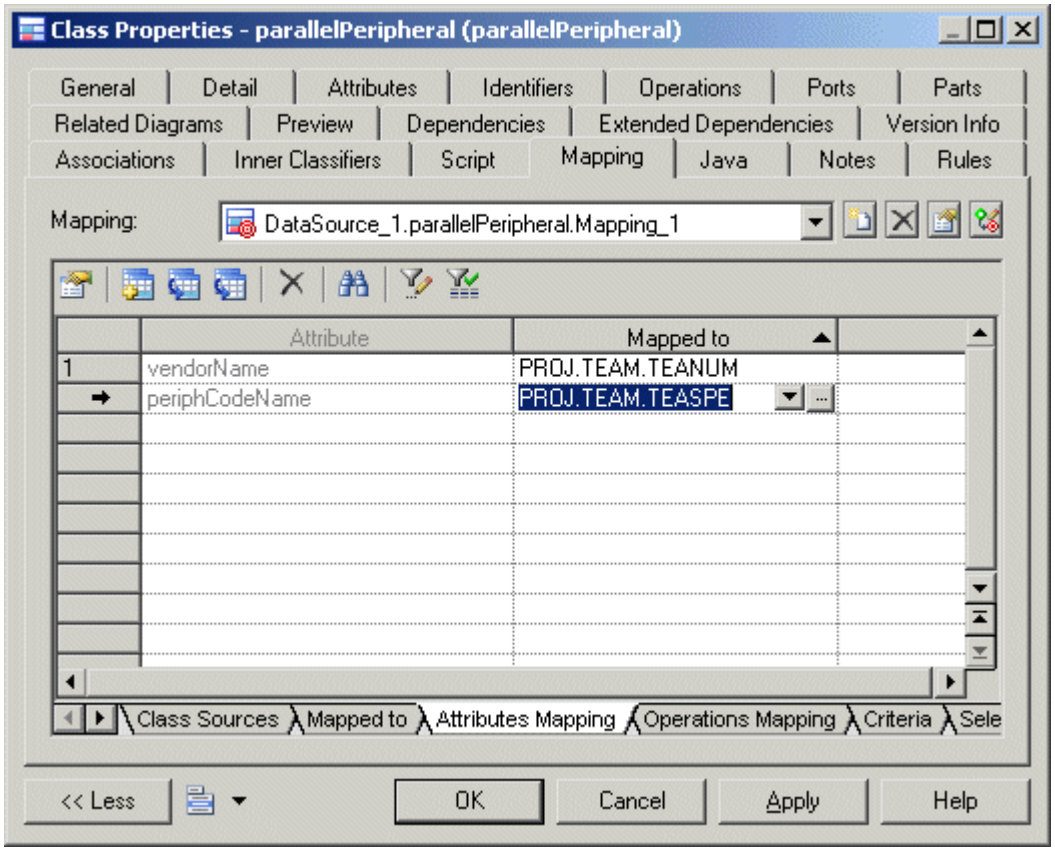
1. Open the property sheet of the object you want to create mappings to and click the *Mapping* tab.
2. Click the *Create Mapping* tool to the right of the *Mapping* list to create a new mapping for the target object. If:
   - No data source has yet been created in the model you will be prompted to create one (see Creating a Data Source [page 466]).
   - Several data sources are available, you will be prompted to select one and click *OK*.
3. Select the object from the data source to map to the present object and click *OK* to create the mapping.

   The new mapping is added in the *Mapping* list and the mapped object is displayed in the *Object Sources* sub-tab:



4. [optional] Click the *Add Objects* tool to select more source objects to map with the object.
5. Click the *Sub-Object Mapping* sub-tab and add or remove mappings between sub-objects (see Object and Sub-Object Mapping Properties [page 485])

   To automatically generate mappings for sub-objects that share a name, click the *Generate Mapping* tool.



6. [optional] Review the other mapping sub-tabs (which are the same as those available from the mapping's own property sheet (see Object and Sub-Object Mapping Properties [page 485]) and edit them as appropriate.
7. [optional] Use the *Mapping* list and the tools to its right to review and edit the other mappings defined on the object:

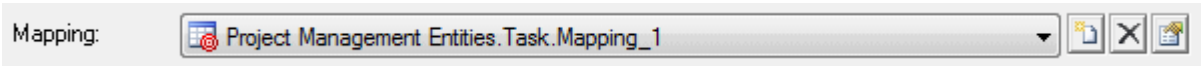
Tool	Description
	Create Mapping - Creates a new mapping with the object as a target.
	Delete Current Mapping - Deletes the currently selected mapping.
	Properties - Opens the property sheet of the currently selected mapping (see Object and Sub-Object Mapping Properties [page 485]).

Tool	Description
	Launch Mapping Editor - Opens the Mapping Editor (see The Mapping Editor [page 468]).

2.4.4 Object and Sub-Object Mapping Properties

Mappings are sub-objects of the target object on which they are defined. Mappings do not have diagram symbols or Browser entries, and are only accessible from the Mapping Editor or from the *Mapping* tab of their parent object property sheet.

To open a mapping property sheet, select the mapping from the list at the top of the Mapping Editor *Mappings* pane or parent object property sheet *Mapping* tab and click the *Properties* tool.



The tabs available on a particular mapping property sheet depend on the objects being mapped. The *General* tab contains the following properties:

Property	Description
Data Source	Specifies the name of the data source in which the source object is defined. Click the *Properties* tool to open the property sheet of the data source.
Target	Specifies the name of the target object that owns the mapping.
Parent	[Sub-object mappings] Specifies the name of the parent of the target object.
Name/Code/ Comment	Identify the mapping. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the mapping. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Mapped to	Specifies the mapping expression. You can edit the expression directly in this field, or click the *Edit* tool to open it in a dedicated SQL or Object Expression editor. In the editor, select an object type from the list in the upper left corner, to display a list of available objects of that type in the upper right corner. Double-click an available object to add it at the insertion point in the expression displayed in the lower half of the editor. To undo your changes and recover the default expression click to release the *User-Defined* button.

The following tabs are also available:

- *Sources* - Lists the source objects mapped to the object. Use the *Add Sources* tool to add additional source objects to the list.

- *Sub-Object Mappings* - Lists the sub-objects of the mapped object that are mapped with sub-objects of the object. The following tools are available on this tab.

Tool	Description
	Properties - Opens the property sheet of the selected sub-object mapping.
	Add Mapping - Select additional source sub-objects to map. After adding sub-objects, select the sub-objects to map them against in the *Mapped to* column.
	Create from Sources - Copy sub-objects from the source object to the object. The name, code , description, annotation, and comment are copied and the data types are converted in order to match the current model.
	Generate Mapping - Automatically generate mappings between sub-objects with the same name or code in the source and target models.

Click the Ellipsis button in the *Mapped to* column to edit the source expression for the sub-objects.

- *Operation Mappings* - [OOM classes only] Lists the operations and queries associated with the mapping. Allows you to associate a SQL query to the operation. When an operation implies action on the database, like data retrieval or data update, you can associate this operation with a SQL query that will execute an action in the database. Usually, this query is a SELECT statement used to retrieve one or several instances of the current class according to the operation parameter. It can also be an UPDATE query. In the context of an EJB, the SQL query mapped to an operation is used to implement finder or select methods.
- *Criteria* - Allows you to specify join criteria between source objects. For example: **`EMPLOYEE.ID < 100`**
- *Query* - Computes a SQL query from the information in the other tabs to retrieve or update data in the database. If you modify the SQL query, it can no longer be automatically calculated, even when you update the mapping. You can recover the computed expression by clicking to release the *User-Defined* tool.
  A Select query is available for a forward mapping, and Insert, Update and Delete queries for a reverse mapping:
  - The *Select* tab retrieves attribute values of class instances from the database using the class identifying attributes. For example:

```
select
  USER.USERNAME "USERNAME",
  USER.PASSWORD "PASSWORD"
from  USER
```

  - The *Insert* tab creates an instance of the class and saves it with its attributes. For example:

```
insert into USER(
   USER.USERNAME,
   USER.PASSWORD)
values (
   %USERNAME%,
   %PASSWORD%)
```

- The *Update* tab displays a statement that allows you to update attribute values of class instances apart from identifying attributes. For example:

```
update USER
set USER.USERNAME = %USERNAME%,
  USER.PASSWORD = %PASSWORD%
```

- The *Delete* tab displays a statement that allows to delete a class instance from the database using its identifying attributes. For example:

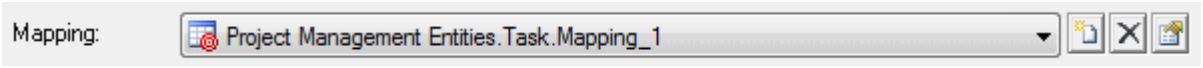
```
delete line* from USER
```

To modify the default syntax of the queries, edit the **SelectStatement**, **InsertStatement**, **UpdateStatement**, and **DeleteStatement** entries in the Profile category of the object language or extension (see *Customizing and Extending PowerDesigner* > *Extension Files*).

2.4.5 Metamodel Mapping Properties

Metamodel mappings are mappings between metamodel objects, which control how objects are imported or generated. Metamodel mappings are sub-objects of the PowerDesigner metamodel object on which they are defined.

To open a metamodel mapping property sheet, select the mapping from the list at the top of the Mapping Editor *Mappings* pane or parent object property sheet *Mapping* tab and click the *Properties* tool.



The tabs available on a particular mapping property sheet depend on the objects being mapped. The *General* tab contains the following properties:

Property	Description
Source object	Specifies the metamodel object being mapped to the target object.
Target object	Specifies the metamodel object being mapped from the source object. This object is the parent of the mapping itself.

Property	Description

Transformation script	[metaattribute mappings] Specifies a script to set the value of the attribute. In the following example, from an XML import, the `notnullable` attribute is imported to the `Mandatory` attribute and, because the sense of the attributes is reversed, the boolean value imported is set to the opposite of the source value:
	`Sub %Set%(obj, sourceValue)`
	`  obj.SetAttribute "Mandatory", not sourceValue`
	`End Sub`
	In the following example, from an object generation, the `NumberID` attribute is generated to the `Comment` attribute and a text string is prepended to make clear the origin of the value:
	`Function %AdjustValue%(sourceValue, sourceObject, targetObject)`
	`  Dim targetValue`
	`  targetValue = "The original process NumberID is "`
	`+cstr(sourceValue)`
	`  %AdjustValue% = targetValue`
	`End Function`

The following tabs are also available for metaclass mappings:

- *Initialization* - Specifies a script to initialize the metaclass to be created. In the following example, the value of the Stereotype attribute is set to SimpleType:

```
Sub %Initialize%(obj)
   obj.Stereotype = "SimpleType"
End Sub
```

- *Attribute Mappings* - Lists the mappings of attributes under the metaclass. Select a mapping and click the *Properties* tool to open its property sheet. To control the order in which attributes are created, in order to respect dependencies between them, use the arrows at the bottom of the list.
- *Collection Mappings* - Lists the mappings of collections under the metaclass.
- *Post-Process* - Specifies a script to modify the metaclass after creation and execution of mappings. In the following example, the value of the *Code* attribute is copied to the *Name* attribute:

```
Sub %PostProcess%(obj)
   ' Copy code into name
   obj.Name = obj.Code
End Sub
```

2.4.6 Metamodel Object Properties

To view the properties of metaclasses, metaattributes, and metacollections displayed in the Mapping Editor, double-click the object node in the Mapping Editor or right-click the node and selecting *Properties*.

The *General* tab contains the following properties:

Property	Description
Parent	[metaattributes and metacollections] Specifies the metaclass to which the metaobject belongs.

Property	Description
Parent collection	[sub-objects/compositions] Specifies the name of the composition collection that contains the sub-objects under the parent object.
Name	Specifies the name of the metaclass in the PowerDesigner metamodel or XML schema.
Data type	[metaattributes] Specifies the data type of the attribute.
Identifier	[metaattributes] Specifies that the attribute is used to identify the metaclass for referencing by another metaclass.
Reference / Reference path	[metaattributes and metacollections] Specifies that the attribute or collection is used to point to another metaclass to form an aggregation.
Singleton	[metaclasses] Specifies that only one instance of the metaclass is possible under each parent object.
Comment	Provides additional information about the metaobject.

The following tabs are also available for metaclasses:

- *Attributes* - Lists the metaattributes belonging to the metaclass. Select an attribute in the list and click the *Properties* tool to open its property sheet.
- *Collections* - Lists the metacollections belonging to the metaclass. Select a collection in the list and click the *Properties* tool to open its property sheet.

2.5 Impact and Lineage Analysis

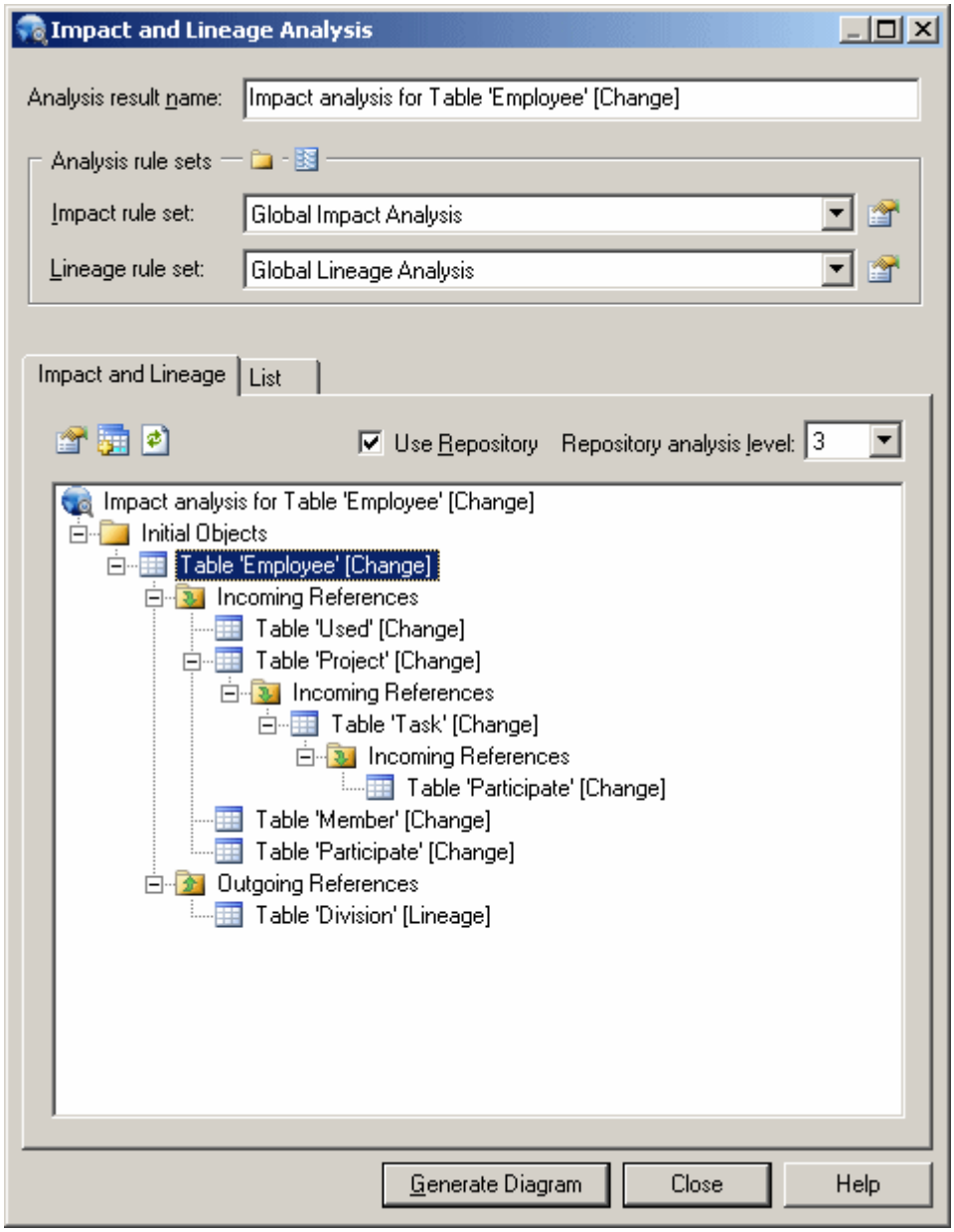
Before you perform an action on a model object, such as deleting a table or changing the datatype of a primary key column, you can preview which other objects will be impacted (impact analysis) and identify the objects that take part in the definition of the object (lineage analysis). You can generate the analysis to an impact analysis model (IAM) to review it in a diagram and save it for future reference.

Procedure

1. Open an impact and lineage analysis in any of the following ways:
   - Select one or more objects in the diagram and select *Tools* > *Impact and Lineage Analysis* (or press *Ctrl+F11*).
   - Right-click an object in the Browser *Local* tab or diagram and select *Impact and Lineage Analysis*.
   - [when deleting an object] Click the *Impact* button on the Confirm Deletion dialog box.
   - In an object's property sheet, select the *Dependencies* tab, and click the *Impact and Lineage Analysis* button.

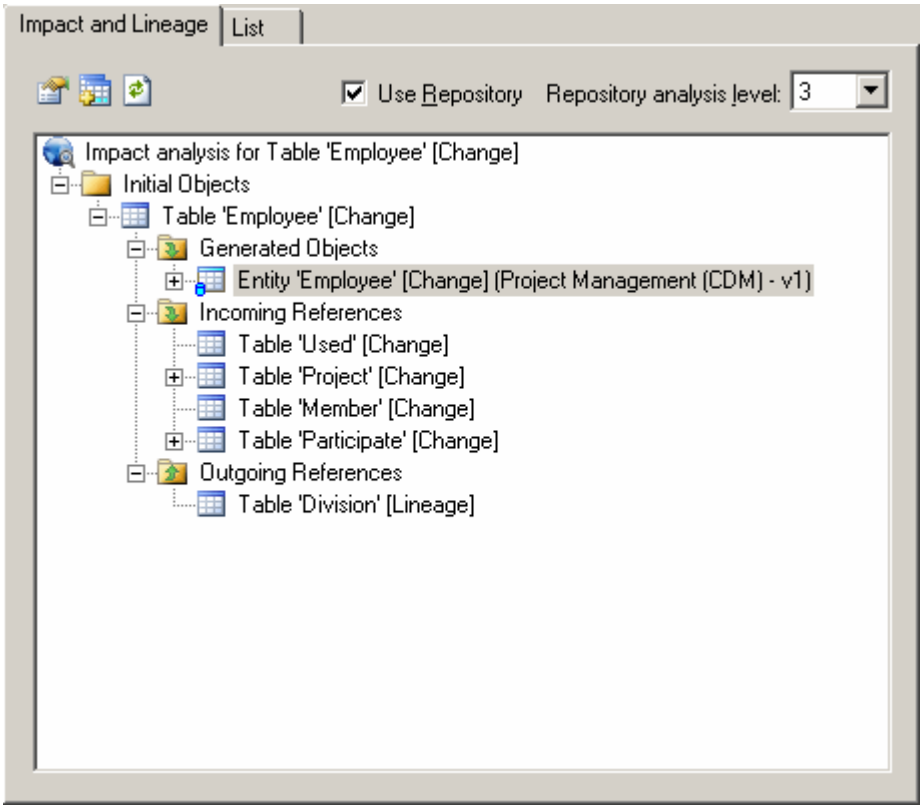
> **i Note**
>
> This feature is also available in the repository (see Impact and Lineage Analysis in the Repository [page 290]) and in PowerDesigner Web (see *PowerDesigner Web*)

The analysis applies rules to the initial objects to generate a tree view of influencing (lineage) and dependent (impacted) objects, which appears instantly in the dialog. The following example shows all the objects that will be impacted by a change to the **`Employee`** table, and all the objects that influence it (lineage analysis):



2. [optional] Select *Use Repository* and specify a *Repository analysis level* to search for additional impacted or lineage objects in the repository. If you are not currently connected to the repository, the *Connect* dialog opens to allow you to do so.

   In the example, you see an additional impacted object, an entity that was generated from the table to the `Project Management` CDM. The blue overlay in the bottom-left corner of the object indicates that it is located in the repository.

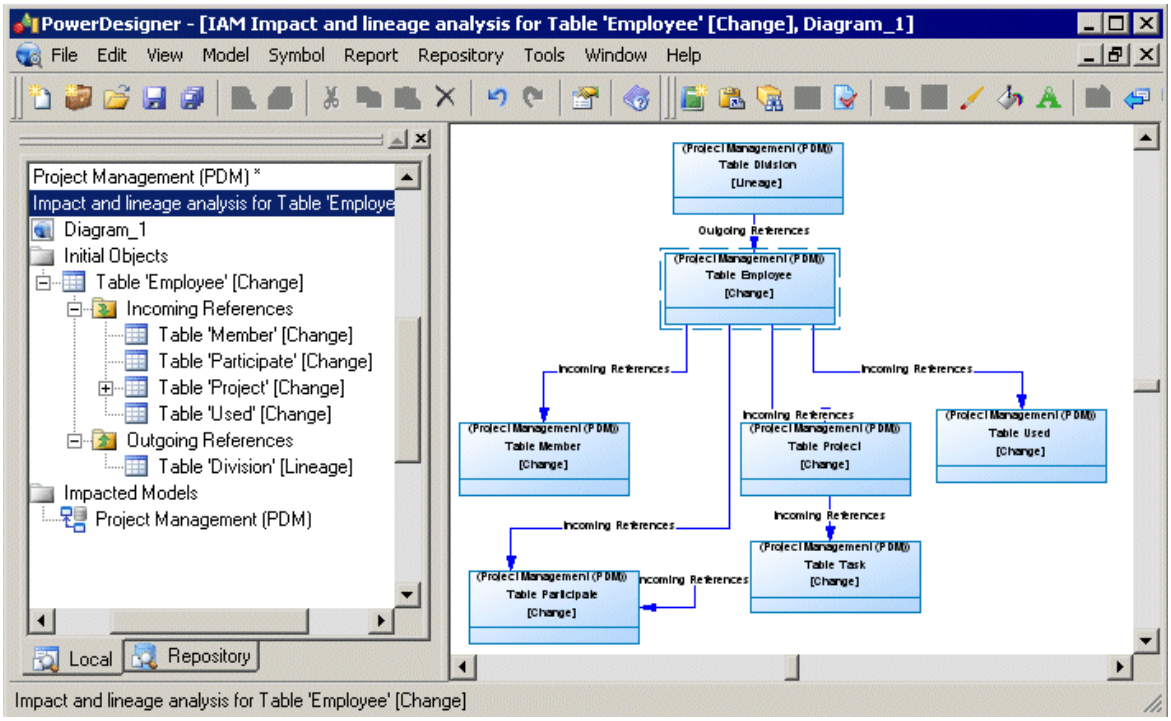


3. [optional] Enter a name for your analysis result, which is then used as the name of any generated model.
4. [optional] Change the impact rule set to apply to your analysis:
   - *Conceptual Impact Analysis* – identifies the objects impacted by changes to the object.
   - *Data Impact Analysis* – identifies uses of values contained in the object.
   - *Delete Impact Analysis* – [default when deleting] identifies the objects impacted by the deletion of the object.
   - *Global Impact Analysis* – [default when not deleting] identifies all the objects that depend on the object.
   - *None* – no impact rule set is selected.

   > **i Note**
   >
   > Click the *Properties* tool to the right of the field to review the selected rule set. Click the *Select Path* tool above the list to change the folder containing rule sets, or the *List of Rule Sets* tool to open any rule set (see Analysis Rule Sets [page 503]).

5. [optional] Change the lineage rule set to apply to your analysis:
   - *Conceptual Lineage Analysis* – identifies the objects that justify the existence of the object, and ensure it fulfills a well-identified need.
   - *Data Lineage Analysis* – identifies the origin of values contained in the object.
   - *Global Lineage Analysis* – [default when not deleting] identifies all the objects that influence the initial object.
   - *None* – [default when deleting] no lineage rule set is selected.

6. [optional] Click the *Generate Diagram* button to save the analysis to a diagram in an impact analysis model (IAM). The following example shows the 'Employee' table in the center of the diagram with its influencing objects above it and its dependent objects beneath it:



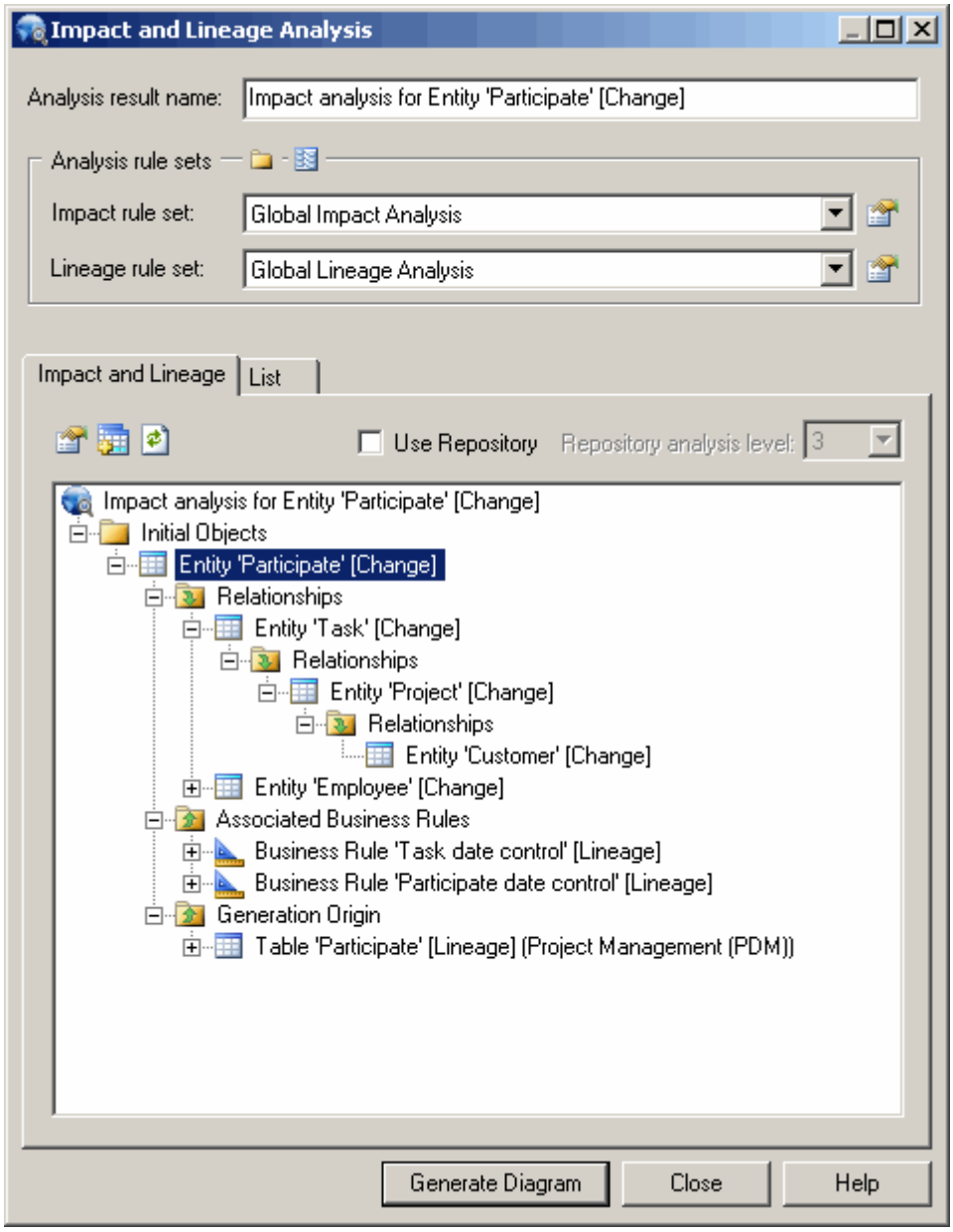
2.5.1 Reviewing and Modifying an Analysis

The Impact and Lineage tab displays the result of the analysis in a tree format, showing all the impacted and lineage objects. Each item in the tree displays either an [action] suffix to indicate the effect on an impacted object or a [lineage] suffix to designate lineage items

In this example, the analysis shows that:

- A change on the **`Participate`** entity will result in changes to the **`Task`** and **`Employee`** entities with which it has relationships (and that the change will cascade down from the **`Task`** entity to the **`Project`** and **`Customer`** entities).
- The business rules **`Task date control`** and **`Participate date control`** form part of the definition of the **`Participate`** entity, which was generated from the **`Participate`** table in `Project Management (PDM)`.

> **i Note**
>
> If an object is dependent multiple times then, for performance reasons, only the first entry appears in detail in the tree. Right-click subsequent undeveloped entries and select *Go to First Occurrence* to jump to the detail.



The following tools are available on this tab:

Tool	Description
	Properties – Opens the property sheet of the selected item.
	Add Initial Objects – Opens a selection dialog to add objects to analyze.
	Re-Analyze – Recalculates the analysis from the selected point in the tree downwards. Select the root object to re-analyze the entire tree.

You can modify your analysis in the following ways:

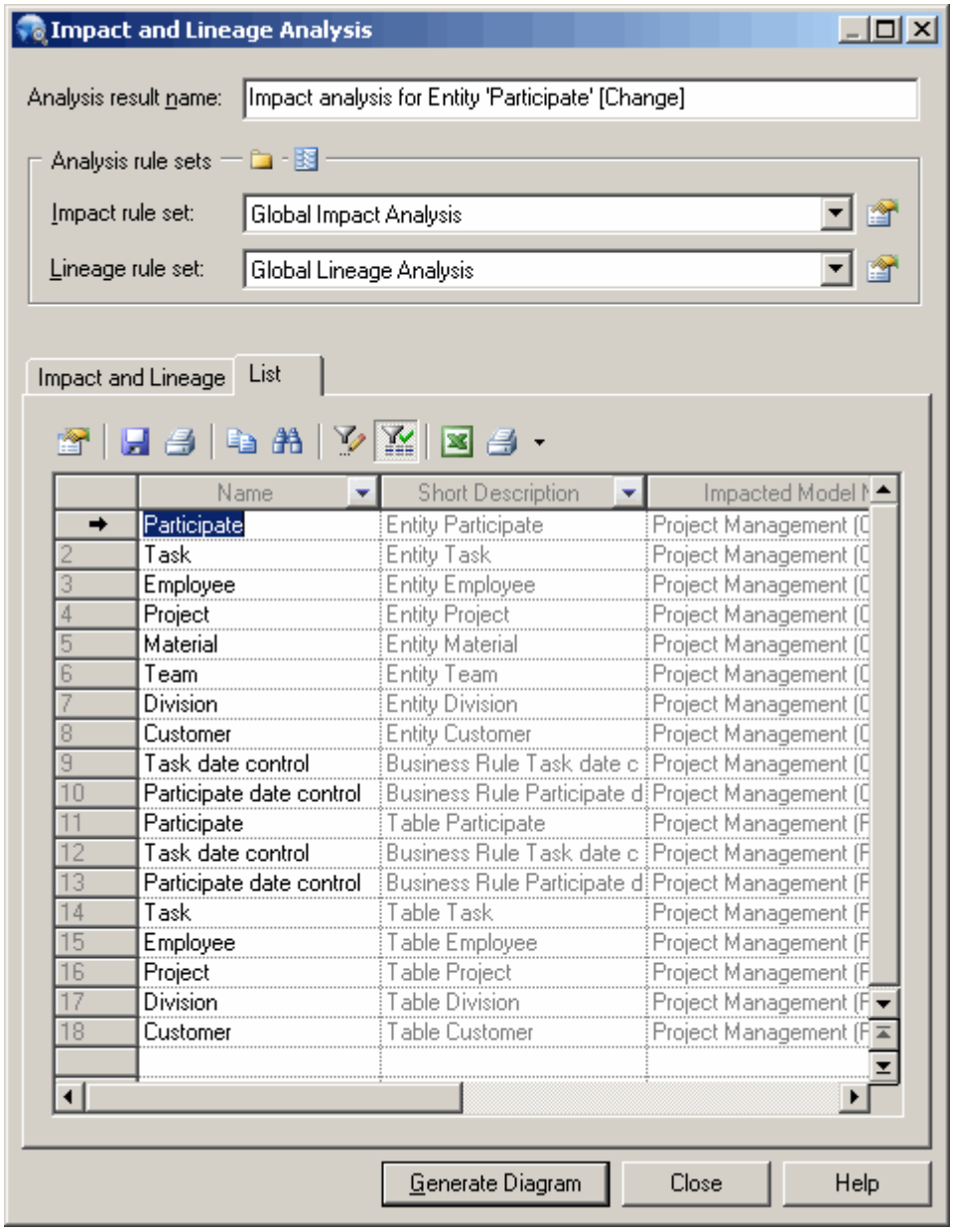
- To extend the analysis to include objects in the repository, select *Use Repository* and specify a *Repository analysis level* to control the number of levels of dependencies to search for.
- To add objects to analyze, click the *Add Objects* tool to open a selection dialog, select one or more objects, and click *OK*.
- To simplify the tree, right-click any object or collection and select *Remove* to remove it and its descendants.

> **i** Note
>
> Right-click any node in the tree and select *Re-analyze* to replace any objects or collections that have been removed in this way. To view the available collections and modify those that appear for an object, right-click it and select *Change Impact Analysis Objects* or *Change Lineage Analysis Objects* (see Controlling the Display of Object Collections [page 496]).

- To change the action being analyzed for any object, right-click it and select one of *Change Action to [Change]*, *Change Action to [Delete]*, or *Change Action Description* to open the Action Description dialog (and select one or more of *Delete*, *Change*, and any *User-defined actions* separated by semicolons).
- To change the rule set used in the analysis, select a new rule set from the lists at the top of the dialog.

2.5.1.1 Viewing the Preview as a List

The List tab displays a tabular report of all the objects contained in the impact and lineage analysis. You can customize the report by filtering which attributes you want to view, save the report in RTF or CSV format, and print it.



The following tools are available on the preview *List* tab:

Tool	Description
	Properties – Displays the property sheet of the selected object in the list.
	Save – Saves the list in RTF (for MS Word) or CSV (for MS Excel) format.
	Print – Prints the list.
	Copy – Copies the list in RTF or CSV format.

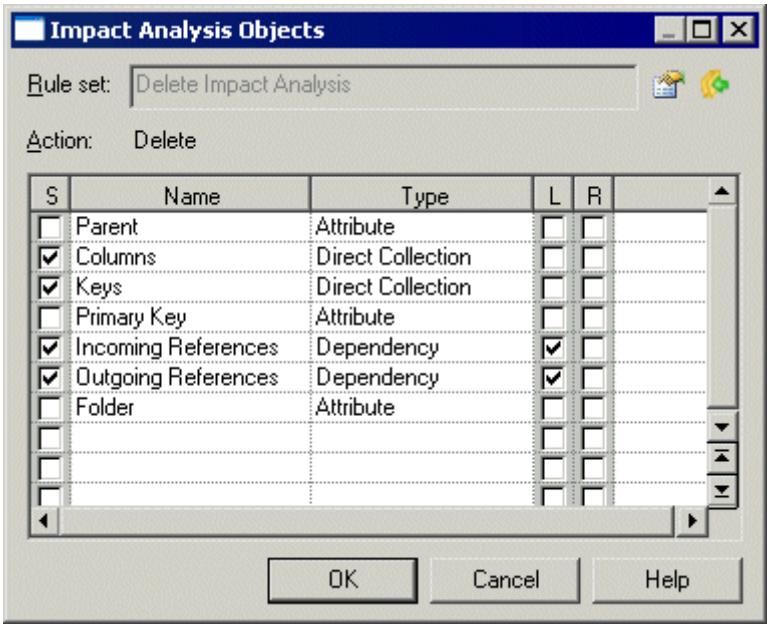
Tool	Description
	Find a Row – Opens the Find dialog box to let you search the list for a specific object.
	Customize Columns and Filter – Opens the Customize Columns and Filter dialog to select attributes to display in the list and specify expressions to filter by (see Customizing Object List Columns and Filtering Lists [page 123]).
	Enable/Disable Filter - Enables or disables the filter on the list.

2.5.1.2 Controlling the Display of Object Collections

You cannarrow or widen the analysis by changing the collections of impact and lineage objects that appear.

Procedure

1. Open the Impact (or Lineage) Analysis Objects dialog box in any of the following ways:
   - [from preview] Right-click an object in the preview tree, or in the IAM Browser or diagram, and select *Change Impact Analysis Objects* or *Change Lineage Analysis Objects* to open a list of available collections of objects. A collection is listed here only if one or more impact (or lineage) objects belonging to the collection are present in the current analysis:



2. Select or unselect the following check boxes to control the display of collections:

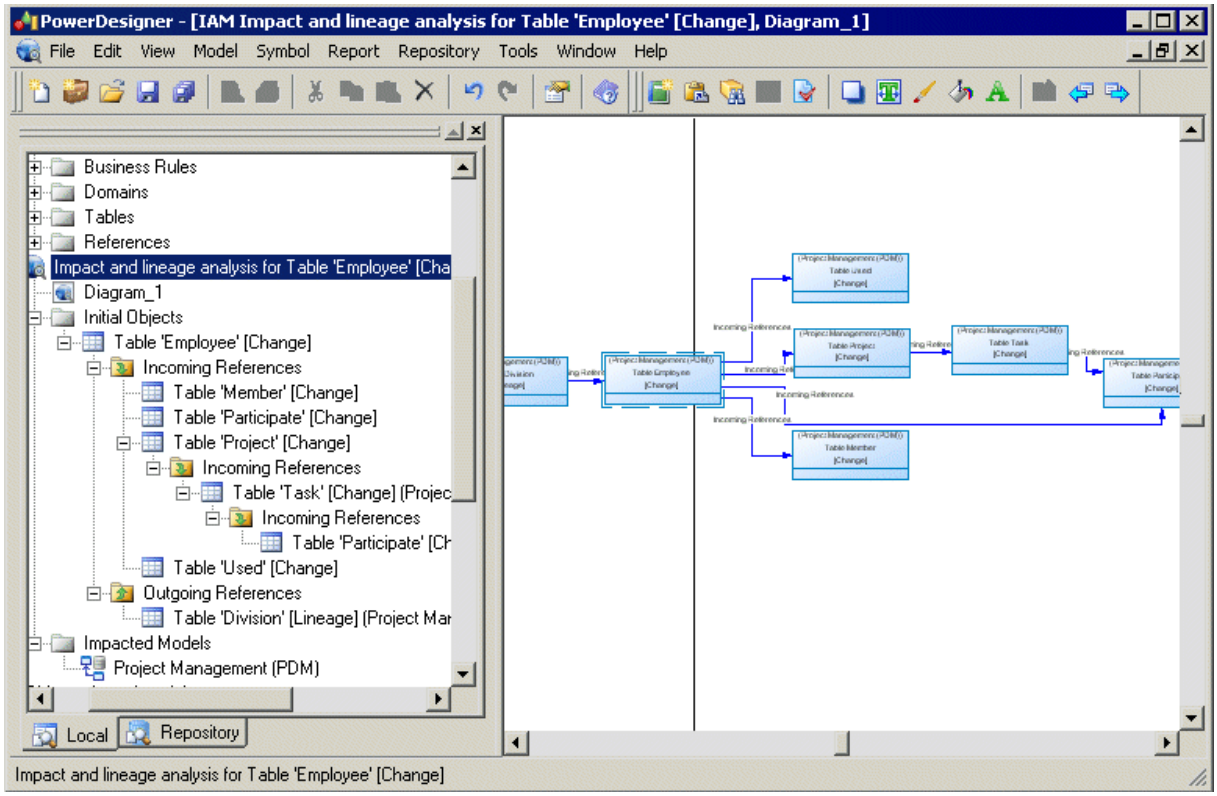
- *S* [Selected] - enables the display of the collection in your analysis.
- *L* [Replace link object by extremity] - If the object is a link, this option hides it to simplify the display and view only the object at the other end of the link.
- *R* [Recursive] - Displays recursive collections in a single list.

3. [optional] Click the *Properties* tool to open and review or edit the analysis rule set in the Resource Editor (see Analysis Rule Sets [page 503]).
4. [optional] Click the *Apply Changes to Rule Set* tool to apply your changes to the rule set so that they are automatically applied next time.
5. Click *OK* to close the dialog box.

   The display of the selected object collections is updated in the preview tree or IAM. If you re-analyze your model, then the collection rules specified in the rule set are reapplied.

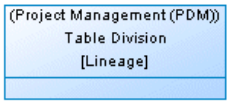
2.5.2 Reviewing and Modifying an IAM Model

You generate an impact analysis model (IAM) by clicking the *Generate Model* button on the Impact and Lineage Analysis dialog box. An IAM provides a graphical view of your analysis and allows you to save the analysis as a snapshot for later reference.



In the above example, the **Employee** table shows influencing (lineage) objects on the left and dependent (impacted) objects on the right. Links between the objects represent the analysis rules that propagate actions on objects.

The following types of objects are available in an IAM:

Symbol	Description
	Initial objects - The model objects being analyzed have symbols surrounded by dotted lines. In the Browser, click to expand the initial object to view the tree of its incoming references (impacted objects) and outgoing references (lineage objects). You can also see the list of initial objects by selecting *Model* > *Analysis Objects*.
	Impacted and lineage objects - The model objects that are impacted by or influencing the initial objects. In the Browser, these collections of objects are inside folders bearing a downward arrow () for impacted objects or an upward arrow () for lineage objects.
	Link - Analysis rule between initial objects and impacted objects.
[none]	Impacted Models - The models that are impacted by the change. You can see the list of impacted models in the Browser or by selecting *Model* > *Analysis Models*.

> **i Note**
>
> You can add graphical objects, such as notes or decorative symbols, to your diagram.

You can modify your analysis in the following ways:

- To extend the analysis to include objects in the repository, select *Tools* > *Change Analysis Options* to open the *Impact and Lineage Analysis Options* dialog, select *Use Repository* and specify a *Repository analysis level* to control the number of levels of dependencies to search for.
- To add or remove initial objects to analyze, click *Tools* > *Change Analysis Options*, select the *Initial Objects* tab, and use the *Add Objects* or *Delete* tool.
- To remove any object or collection and its descendants to simplify the tree, select it in the Browser or diagram and press *Delete*.

  > **i Note**
  >
  > Right-click any node in the Browser or diagram and select *Re-analyze* to replace any objects or collections that have been removed in this way. To view the available collections and modify those that appear for an object, right-click it and select *Change Impact Analysis Objects* or *Change Lineage Analysis Objects* (see Controlling the Display of Object Collections [page 496]).

- To change the action being analyzed for any object, right-click it and select one of *Change Action to [Change]*, *Change Action to [Delete]*, or *Change Action Description* to open the Action Description dialog (and select one or more of *Delete*, *Change*, and any *User-defined actions* separated by semicolons).
- To change the rule set used in the analysis, click *Tools* > *Change Analysis Options* to open the *Impact and Lineage Analysis Options* dialog, select a new rule set and click *OK*.

2.5.2.1 IAM Model Properties

The model property sheet displays the definition of the current IAM. You can check the last analysis date of your model or add a comment, for example.

To open an IAM property sheet, double-click its Browser entry. The *General* tab contains the following properties:

Property	Description
Name / Code / Comment	Identify the model. The name should clearly convey the model's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the model. By default the code is auto-generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Filename	Specifies the location of the model file. This field is empty if the model has never been saved.
Author	Specifies the author of the model. If you enter nothing, the Author field in diagram title boxes displays the user name from the model property sheet Version Info tab. If you enter a space, the Author field displays nothing.
Version	Specifies the version of the model. You can use this box to display the repository version or a user-defined version of the model. This parameter is defined in the Title page of the model display preferences.
Default diagram	Specifies the diagram that appears by default when you open the model.
Last analysis date	[Read-only] Computed date of the last model analysis, which is updated every time you re-analyze the model.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

2.5.2.2 Setting IAM Display Preferences

PowerDesigner display preferences let you customize the format of object symbols and the information that appears on them. To set IAM display preferences, select *Tools* > *Display Preferences*, or right-click the diagram background, and select *Display Preferences*.

These display preferences are available for analysis objects:

Preference	Description
Model name	Displays the model name of the analysis object.
Object type	Displays the type of the analysis object.
Action	Displays the action name of the analysis object.
Highlight initial object	Displays a dotted line around the symbol of the initial object.

The following display preferences are available for analysis links:

Preference	Description
Name	Displays the name of the link.

For information about changing the format of symbols, see Format Display Preferences [page 234].

2.5.2.3 Comparing Two IAM Models

Select *Tool* > *Compare Models* to view the differences between two IAMs. The comparison window displays the objects contained within the models in a tree format, and highlights the differences between them.

Context

For detailed information about comparing models, see Comparing and Merging Models [page 366].

2.5.3 Printing an Analysis

You can print the preview list or the IAM model diagram, or create a list report in the IAM to document and discuss proposed changes.

Context

You can print the analysis in any of the following ways:

- [from Preview] Click the *List* tab (see Viewing the Preview as a List [page 494]) and select the *Print* tool.
- [from the IAM] Select *File* > *Print* to print the diagram (see Printing Diagrams [page 191]).
- [from the IAM] Select *Reports* > *List Report Wizard* to create a list report for an object type and print it (see Creating a List Report [page 303]).

2.5.4 IAM Custom Checks

PowerDesigner does not provide any predefined checks for the data testing of an IAM. However, you can create custom checks in an extension.

For information about developing custom checks, see *Customizing and Extending PowerDesigner > Extension Files* .

You can check an IAM in any of the following ways:

- Press F4.
- Select *Tools* > *Check Model* .
- Right-click the diagram background and select *Check Model*.

The Check Model Parameters window opens, which allows you to specify the kinds of checks to perform, and the objects to apply them to. For detailed information about this window and correcting problems reported, see Checking Models [page 148].

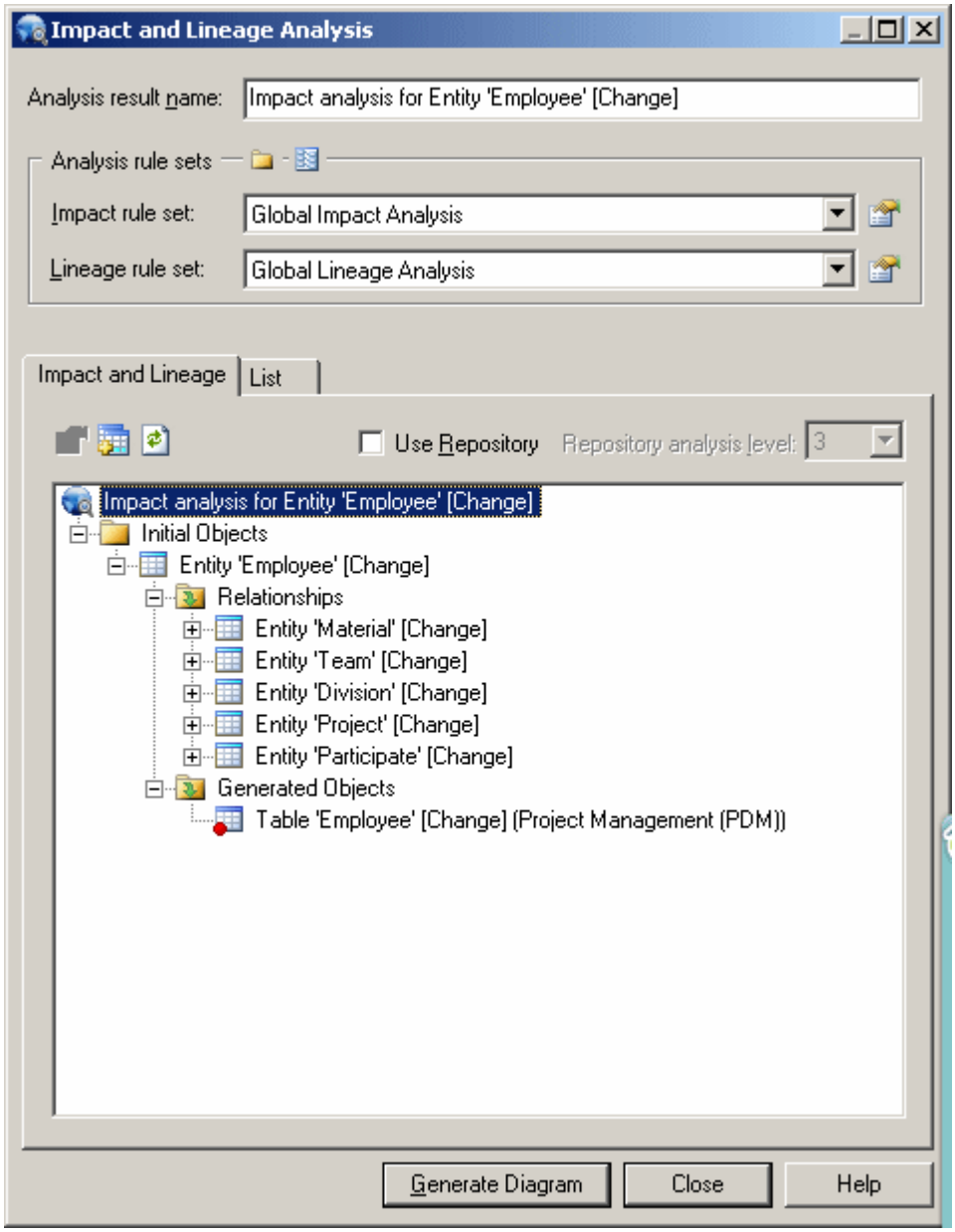
2.5.5 Obtaining Cross-Model Dependencies Without a Repository Connection

If you anticipate needing to obtain information about cross-model dependencies with models not in your workspace in a situation where you will not have a repository connection, you can obtain this information in advance by checking your model out with the *Check out dependencies* option.

For models checked out with the *Check out dependencies* option (see Checkout Parameters [page 270]) objects that are implicated in an impact or lineage analysis but that are not currently available display a red overlay in their bottom-left corner to indicate that the analysis cannot proceed beyond this point without connecting to the repository or otherwise obtaining the parent model. This option also ensures the cross-

model dependencies appear on the *Dependencies* tab of objects even when the other models are not open in the workspace.

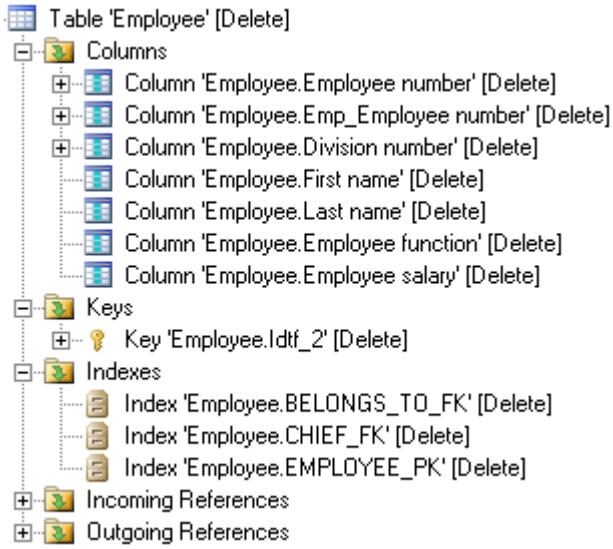
In the following example, the **`Employee`** table, which is generated from the **`Employee`** entity, is present in the analysis even though its model is not available, because the model from which the analysis was launched was checked out with dependencies:



2.5.6 Analysis Rule Sets

Impact and lineage analysis uses sets of analysis rules derived from relationships between metaclasses in the metamodel to construct the trees of impacted and lineage objects. PowerDesigner provides standard rule sets (XML resource files with a `.rul` extension), and you can create your own.

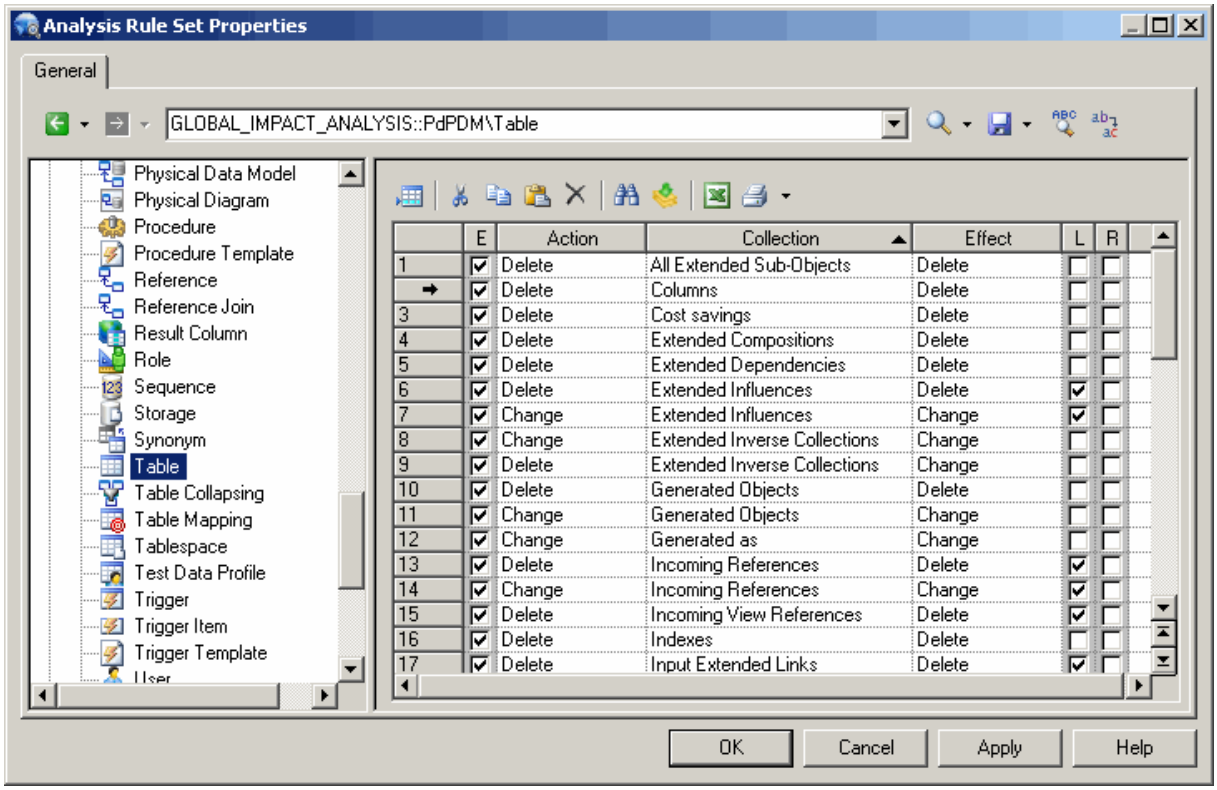
Each rule represents a potential collection of objects, and is represented as a folder in the analysis tree. In the following example, deleting the **`Employee`** table also deletes all the members of the **`Columns`**, **`Keys`**, **`Indexes`**, and **`Incoming`** and **`Outgoing References`** collections:



You can open an analysis rule set in any of the following ways:

- From an analysis preview, click the *Properties* tool next to the rule set.
- From an IAM, select *Tools* > *Change Analysis Options*, and click the *Properties* tool next to the rule set.
- From any model, select *Tools* > *Resources* > *Impact and Lineage Analysis Rule Sets*, select an available rule set, and click the *Properties* tool. For information about the tools available in resource file lists, see PowerDesigner Resource Files [page 408].

Analysis rule sets contain a list of analysis rules for each metaclass in the PowerDesigner metamodel. All analysis rule set files have the same basic category structure, with a library for each model type containing the metaclasses of that module. Clicking a metaclass displays the rules associated with that metaclass:



The following properties are available for each analysis rule:

Property	Description
E	Enables the rule for use in the analysis.
Action	Specifies the action performed on the metaclass that triggers the application of the rule. Select an action from the list or enter your own (see Creating a User-Defined Action [page 505]). For lineage analysis rules, **`Lineage`** is the only possible value.
Collection	Specifies the metaclass collection whose members the rule is applied to. Collections defined in extensions (see *Customizing and Extending PowerDesigner > Extension Files*) are included in one of the **`Extended...`** collections.
Effect	Specifies the effect on the objects in the collection caused by the action on the metaclass. Select an action from the list or enter your own (see Creating a User-Defined Action [page 505]). For lineage analysis rules, **`Lineage`** is the only possible value.
L	Replace Link Object by Extremity - Simplifies the preview tree and the IAM Browser and diagram by hiding the link and showing only the object at the other end of the link.
R	Recursive - Propagates the effect recursively to the dependent objects of the objects in the collection.

The root node of each file contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
File name / Type	[read-only] Specify the path to the `.rul` file and the type of the rule set (impact or lineage).

2.5.6.1 Creating an Analysis Rule Set

You can create your own rule sets, which are then available from the List of Impact and Lineage Analysis Rule Sets.

Procedure

1. Select *Tools* > *Resources* > *Impact and Lineage Analysis Rule Sets* to open the List of Impact and Lineage Analysis Rule Sets, which shows all the available rule sets.
2. Click the *New* tool, enter an appropriate name for the rule set, and select the appropriate type of rule set.
3. Select a rule set from the *Copy From* list to create a new rule set that is identical to the original set apart from the name.
4. Click *OK* to open a Save As dialog, enter a file name and then click *OK* to open the new rule set in the Resource Editor.

2.5.6.2 Creating a User-Defined Action

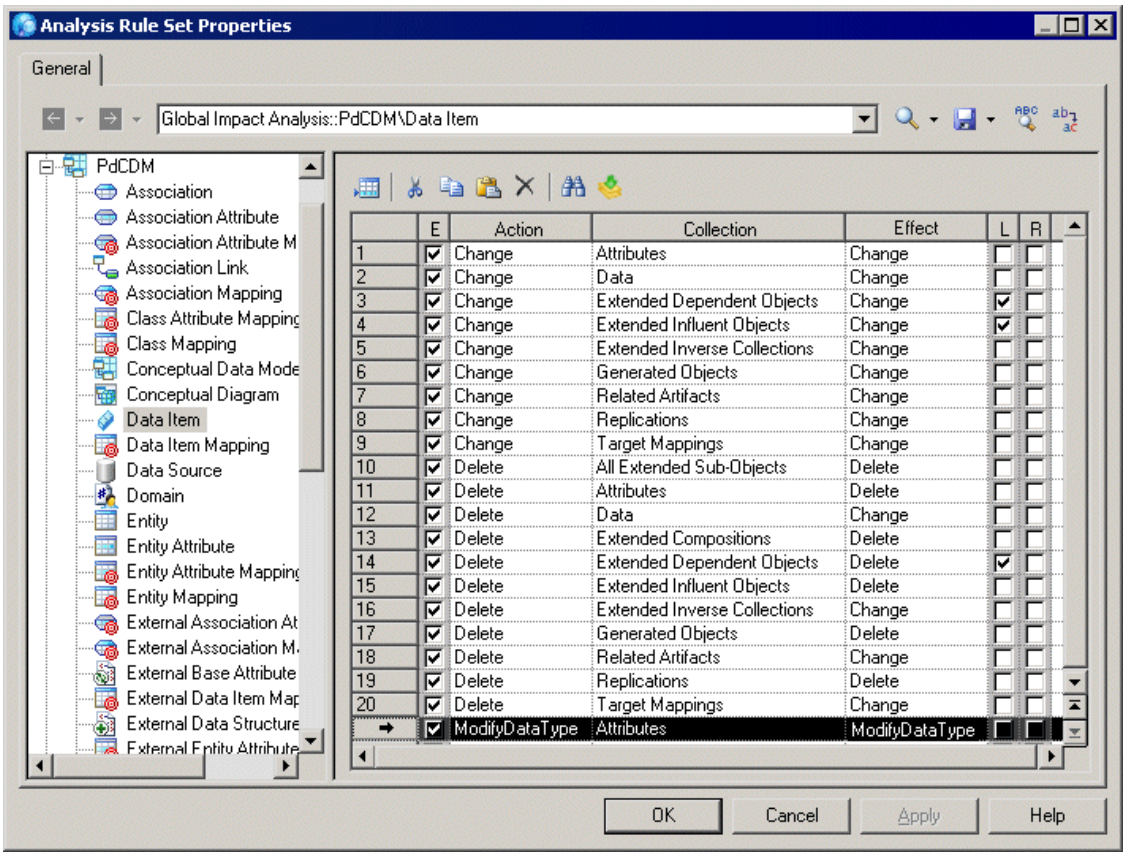
You can specify a user-defined action on an object to analyze its consequences. You must already have created the appropriate analysis rule in the Resource Editor.

Context

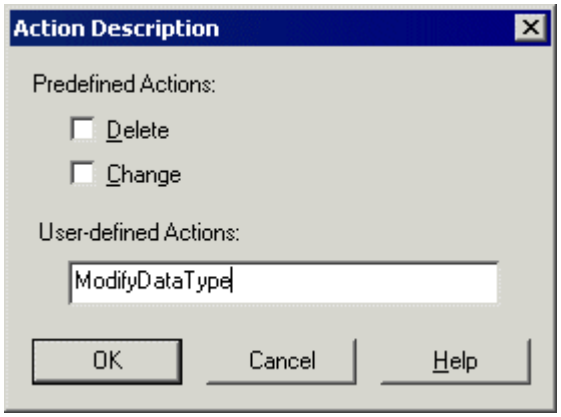
In the following example, create an impact analysis rule to specify that a ModifyDataType action on the Data Item metaclass will also change its attributes.

Procedure

1. In the Resource Editor, open an impact analysis rule set (see Analysis Rule Sets [page 503]).
2. Click the **PdCDM** library to expand its available metaclasses tree.
3. From the tree, select **Data Item** and create the appropriate rule by specifying the following options:
   - Action – enter **ModifyDataType**, which specifies the action on the data item, and triggers the impact analysis rule. The *Enable* check box is automatically selected.
   - Collection – select **Attributes**, which specifies the metaclass collection for which you define the impact rule.
   - Effect – enter **ModifyDataType**, which specifies the action that is propagated to attributes.



4. Click *OK* to save the rule set and close the Resource Editor.
5. Open the Action Description dialog:
   - [from preview] Right-click a data item in the tree, and select *Change Action Description*.
   - [from the IAM] Right-click a data item in the Browser or in the diagram, and select *Change Action Description*.
6. Enter **ModifyDataType** in the *User-defined Actions* field:

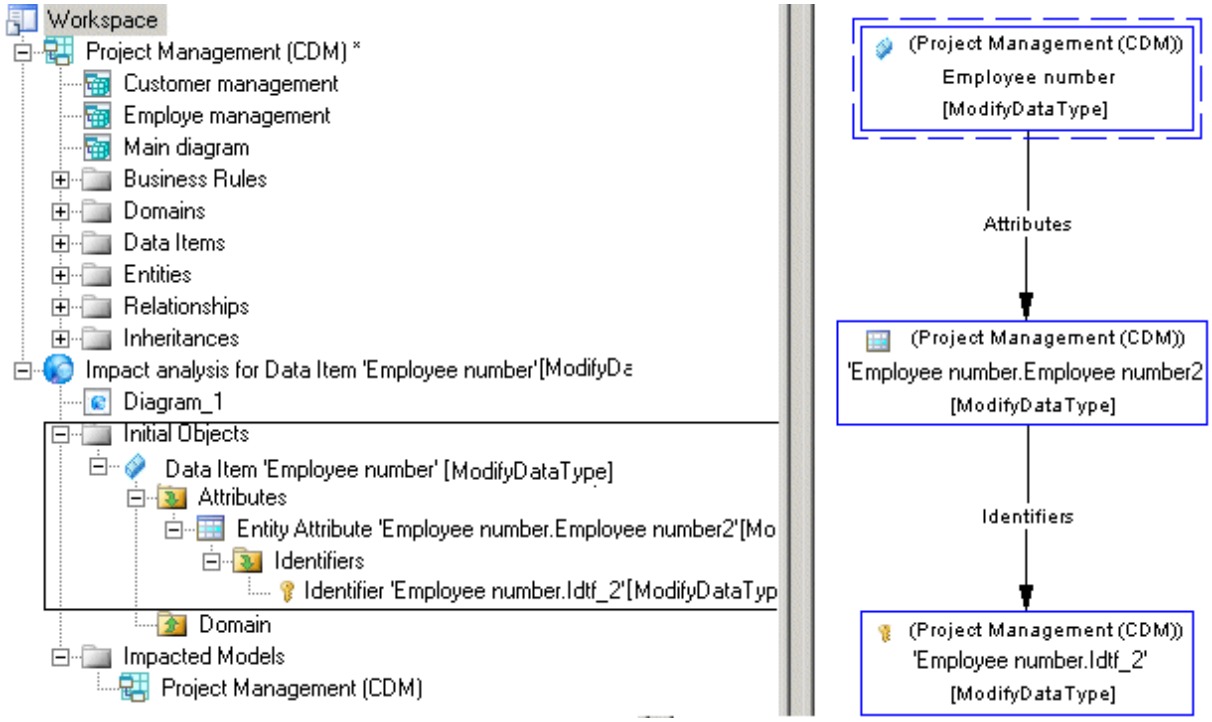


You can enter several actions in a single word using semicolons as separators.

7. Click *OK* to close the dialog box.

Results

The following example shows how the [ModifyDataType] action on the Employee number data item affects its Employee number2 attribute, which in turns affects its Idtf_2 identifier:



2.5.6.3 Checking Rule Sets into the Repository

When an administrator creates or upgrades the repository, default rules sets are checked into the Library folder at the root of the repository to enable impact and lineage analyses in the *Repository* tab of the Browser. To use other rule sets to perform analyses in the repository, you must check them in manually.

Context

> **i** Note
>
> If you have stored customized rule sets delivered with a previous version of PowerDesigner in the repository, they cannot be used for repository analyses until you have upgraded them by re-checking them into the latest version of the repository. For general information about updating resource files, see *Installation Guide > Planning Your PowerDesigner Installation > Upgrading PowerDesigner* .

Procedure

1. Select *Tools* > *Resources* > *Impact and Lineage Analysis Rule Sets* and create your custom rule set (see Creating an Analysis Rule Set [page 505]).
2. Select the rule set in the list and click the *Check In* tool.
3. In the Check In Document dialog (see Check-In Parameters [page 252]), select the *Folder* to which you want to check in the rule set.

   You can check rule sets into any folder for which you have write permission. For example, you might want to create custom rule sets for a particular project and check them into that project's folder.
4. Click *OK* to complete the check in.

   The rule set is now available for use in the repository. If you have checked your rule set into a folder that did not previously contain rule sets, you must use the *Select Path* tool to make your rule set available in the analysis dialog (see Impact and Lineage Analysis in the Repository [page 290]).

   > **i** Note
   >
   > All impact and lineage analyses launched from your local workspace, even those that use the repository, are performed with your local rule sets.

2.5.7 Repository Rule Set Limitations

While the majority of rules are supported in analyses launched or extended into the repository, many calculated collections and some other collections are not accessible to the repository analysis engine. In addition, shortcuts which point to a different branched folder may not be immediately available.

The following list indicates the main categories of collections that are not supported in the repository:

- All collections involving symbols. For example, Symbols, Objects in Related Diagrams.
- In the EAM, calculated collections based on application link types. For example, Implement Objects, Implemented By Objects, Produce Objects, Produced By Objects, Use Objects, Used By Objects.
- In the EAM, the Responsible For collection, which lists the objects for which people or organization units are responsible.
- In the OOM, calculated collections based on generalization or realization links or navigable roles. For example, Implemented Operations, Inherited Associations, Inherited Attributes, Inherited Operations, Navigable Associations.
- In the OOM, the Operations collection for the component metaclass.
- In the CDM, Barker notation, the SubEntities and SuperEntities collections.
- In the BPM for BPMN 2.0 the Participants collection for the choreography task metaclass.

Where one model in a branched folder contains a shortcut to a model in a different branched folder in another part of the repository tree, the shortcut may not be followed during an analysis if the shortcut is the first link between the models and the referencing model is checked in before the target model. In such a situation, checking out both the model containing the shortcut and the target model and checking them back in will enable PowerDesigner to follow the shortcut correctly.

3 Administering PowerDesigner

The chapters in this part explain the tools available for administering the PowerDesigner environment through the repository.

For information about creating, upgrading, and troubleshooting a repository, see *Installation Guide* > *Installing the Repository* .

3.1 Controlling Repository Access

The repository administrator is responsible for controlling access to the documents stored in the repository by creating users and groups and assigning them privileges, permissions, and profiles.

Context

Repository privileges give users access to general repository features, while permissions give them access to particular locations in the repository. The following privileges and permissions are available:

Privileges (Entire Repository)	Permissions (Per Folder or Item)
• **Connect** - Connect to the repository. • **Edit on Web** - Create and edit diagrams in PowerDesigner Web. • **Edit Extensions on Web** - Create and edit custom properties in PowerDesigner Web. • **Freeze Versions** - (see Freezing and Unfreezing Document Versions [page 275]). • **Lock Documents** - (see Locking and Unlocking Documents [page 276]). • **Manage Branches** - (see Branching Version Trees [page 281]). • **Manage Configurations** - (see Grouping Document Versions in a Configuration [page 278]). • **Manage All Documents** - Perform any action on any document version. Implicitly includes **Full** permission on all repository documents. • **Manage Users & Permissions** - Create, modify, and delete repository users and groups, grant them privileges, and add them to groups. Users with this privilege can list all repository documents and set permissions on them without needing explicit **Full** permission. • **Manage Repository** - Create, upgrade, and delete the repository database.	• **List** - View the document or folder in the repository browser and in search results. Without this permission, the folder or document is hidden from the user. • **Read** - Also check documents out from the repository and compare documents. • **Submit** - Also check the document into a changelist for review by a user with **Write** permission. • **Write** - Also check in (with or without a changelist), freeze, and lock document versions. • **Full** - Also move and delete documents, manage permissions granted to users and groups, and remove locks on documents. **i Note** Administrators, who have implicit **Full** permission on all repository objects, will only receive models for review if they have been granted explicit **Write** permission on them.

Procedure

1. [recommended] Connect the repository to an SMTP server to enable the automatic sending of emails for passwords, changelist submissions, and other notifications (see Connecting to an SMTP Server for Notifications [page 517]).
2. Determine how you will manage user authentication. You can choose one or more of:
   - Repository-managed authentication - Specify an appropriate password policy (see Defining a Password Policy [page 518]).
   - LDAP authentication - Connect the repository to an LDAP server to allow users to connect with their corporate credentials (see Connecting to an LDAP Server for User Authentication [page 513]).
   - X.509 certificate-based single sign-on - [PowerDesigner Web-only] Configure the server appropriately (see Enabling Single Sign-On for PowerDesigner Web [page 527]).

   i Note

   LDAP or X.509 certificates are only used for authentication. Privileges and permissions on repository folders and documents are controlled in the repository.
3. [optional] Create high-level functional groups (see Creating Repository Groups [page 531]) to organize users by type and assign appropriate privileges to them to govern general actions that they can perform in the repository (see Granting Privileges to Users and Groups [page 533]).

For example:

Groups	Privileges
Administrators	Connect, Manage All Documents, Manage Users & Permissions, Manage Repository
Senior Modelers	[use the PowerDesigner desktop client] Connect, Edit on Web, Edit Extensions on Web, Freeze Versions, Lock Versions, Manage Branches, Manage Configurations
Modelers	Connect, Edit on Web
Stakeholders	Connect (to provide read-only access to PowerDesigner Web).

> **i Note**
>
> There is no requirement to create groups - you can assign privileges and permissions to individual users - but we recommend that in all but the smallest deployments, you do create groups to simplify the process.

4. [optional] Apply profiles to your groups as necessary to filter the PowerDesigner interface to hide or render read-only types of models, objects, and properties, and to specify defaults for interface elements, options and preferences for different kinds of users (see Using Profiles to Control the PowerDesigner Interface [page 561]).

   In our example, business analysts need only contribute to requirement models, enterprise architecture models, and conceptual data models and all other types of models are hidden from them. We could imagine dividing the **`Modelers`** group into subgroups such as enterprise architect and information architect and hiding or rendering read-only models that do not concern them.

5. Create an appropriate folder structure in the repository to enable you to group documents by project or in any other appropriate way, and to simplify the granting of permissions.

   In this example, the team will use the library folder to share reference models and other documents, and to push resource files to users (see Deploying an Enterprise Library [page 549]). In addition, two modeling projects are proposed, and all the documents related to them will be kept in the two top-level folders:

   - `Library`
   - `Project A`
   - `Project B`

6. Determine your review policy either at a global or project by project level. PowerDesigner supports the following kinds of policy:
   - Simple review - Change lists submitted by users with the **`Submit`** permission are reviewed by users with the **`Write`** or **`Full`** permission.
   - Peer review - Users with the **`Write`** or **`Full`** permission voluntarily submit change lists for review.
   - Direct check in - The **`Submit`** permission and change lists are not used, and users all check in changes without review.

7. Create development groups and implement your review policies by assigning appropriate permissions to control what actions users and groups can perform on particular repository documents and folders.

   In this example, we propose:
   - A Compliance Committee who can modify the reference models and shared resource files contained in the library and can submit changelists for inclusion in the glossary model.
   - A Terminology Committee who can modify the glossary model.

- For each project, lead architects are able to check in changes to the project (and submit changelists for inclusion in the glossary), and regular architects can submit changelists to the project.
- A cross-project stakeholder groups with read-only access (though we could imagine sub-folders within each project for which these users would have write or submit permission for storing their own documents).

Group	Library	Glossary Model	Project A	Project B
Administrators	Full	Full	Full	Full
Compliance Committee	Write	Submit	Read	Read
Terminology Committee	Read	Write	Read	Read
Project A Leads	Submit	Submit	Write	Read
Project A Team	Read	Read	Submit	Read
Project B Leads	Submit	Submit	Read	Write
Project B Team	Read	Read	Read	Submit
Business Stakeholders	Read	Read	Read	Read

8. Create as many users as necessary either manually (see Creating Repository Users [page 530]) or via LDAP (see Managing Externally-Authenticated Repository Users [page 516]) and assign them to appropriate groups (see Adding Users and Groups to a Group [page 532]) according to their roles and project responsibilities.

   There is no limit to the number of groups to which a user or group can be assigned, and users benefit from the cumulative total of all the privileges and permissions they receive.

3.1.1 Connecting to an LDAP Server for User Authentication

A repository administrator can delegate the authentication of repository users to an LDAP server. PowerDesigner supports authentication via Active Directory and a number of other LDAP implementations. You can optionally allow automatic creation of repository accounts when an LDAP user connects to the repository for the first time.

Context

> **i Note**
>
> LDAP integration provides only authentication. Authorization is always managed via the rights and permissions granted within the repository environment.

Procedure

1. Connect to the repository and 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).
2. Select the appropriate *Server type* from the list to set default values for the other parameters.

   For Active Directory, if your environment supports anonymous binding, you may be able to connect without further configuration. Click the *Test Connection* button and follow the instructions on the dialog. If your connection succeeds then no other options are required, but you should consider enabling the *Use Secure Socket Layer (SSL)* and *Auto-create user accounts in repository* options.
3. Modify any appropriate parameters in the *General* group box:

Parameter	Description
Server type	Specifies the type of the LDAP server and sets default values for the server. The following types are available: ○ Active Directory - if your environment supports anonymous binding, you may be able to connect without further configuration. Click the *Test Connection* button and follow the instructions on the dialog. ○ Netscape Directory Server ○ OpenLDAP ○ Oracle Directory Server ○ Other If you edit any parameters and want to revert to the defaults, click the *Default Settings* button.

Provider URL	Specifies the URL for the LDAP provider. By default, for Active Directory, the nearest LDAP server is automatically detected and used for authentication, initializing this field to:
	`LDAP://_ldap.<domain>:389`
	For other servers, this field is initialized to:
	`LDAP://ldap.<domain>:389`
	and you should replace `ldap` with the name or IP address of your LDAP server.

Use Secure Socket Layer (SSL)	Specifies to connect to the LDAP server using SSL, changing the LDAP provider port to the standard secure `636`. If you have deployed PowerDesigner Web, you must obtain and register your certificate authority root certificate in the Java installation. You must obtain and register your certificate authority root certificate in the Java installation. **i Note** PowerDesigner users may need to contact their administrator to obtain a certificate and use `\Windows\System32\certmgr.msc` to register it on their workstations. Right-click *Trusted Root Certification Authorities* and select *All Tasks* > *Import*, then follow the wizard instructions.

Parameter	Description
Default search base	Specifies the level at which the query begins its search for users in the LDAP tree. By default this is initialized to the domain components (DCs) of the LDAP server. For example: `dc=acme, dc=com` You could include the location of the User directory such as `OU=Users, dc=devpd, dc=local` . If the location of the User directory is not specified here, then you must include it in the *Authentication Search Base*.
Anonymous bind	[default] Specifies that the LDAP server supports anonymous access. If you deselect this parameter, you must specify a bind user distinguished name (DN) and password for an account that has permissions to query the LDAP server. **i Note** If the Bind user DN is in the same DN as the *Authentication search base* then you can simply enter the user id for the search. Otherwise, you must enter the full DN for that account. For example, if the *Default search base* is `ou=people,dc=Onebridge,dc=qa`, and you have a user `cn=csitest,cn=users,dc=Onebridge,dc=qa`, then the Bind DN must be `cn=csitest,cn=users,dc=Onebridge,dc=qa`.
Auto-create user accounts in repository	Specifies that any users corresponding to the LDAP authentication search filter can connect to the repository, and will have an account created for them in the repository when they do so. If you do not select this option, then an administrator must create an account for each user before they can connect.

4. Modify any appropriate parameters in the *Authentication* group box:

Parameter	Description

Search filter	Specifies the LDAP query that selects users for authentication. By default this is initialized to (for Active Directory):
	`(&(objectClass=person)(userPrincipalName={uid}))`
	and for other servers:
	`(&(objectClass=person)(cn={uid}))`
	To determine an alternative filter, you must know the properties of the users defined in the Active Directory, and which property (for example, `name` or `samAccountName`) is being used as the login name.

Parameter	Description
Search base	Specifies the location of the User list in your LDAP server. By default this is initialized to the same value as the *Default search base*. If the default search base does not include your users you must specify an appropriate search base here. Users may be in a common node such as `cn=Users` or an organization unit such as `OU=Users`. To determine the correct search base, you should use an LDAP browser to look at the full distinguished name of a user. Note that your Bind DN may be a user in a different node in the tree than general users so it is very important that you have the correct information for each.
Search scope	Specifies the scope of the authentication search. You can choose between: ○ `subtree` - [default] the search begins at the level of the Search base and also searches any subnodes. ○ `onelevel` - only the level specified in the Search base is searched
Authentication method	Specifies the method to use for authentication requests. You can choose between: ○ `simple` - [default] clear text authentication. If SSL is enabled, then the password will be encrypted. ○ `DIGEST-MD5` - hashed password authentication. If you select this option, you must specify a digest format.

5. Click the *Test Connection* button and follow the instructions on the dialog to verify your connection.
6. Click *OK* to save your changes.

> **i** Note
>
> If you have not selected the *Auto-create user accounts in repository* option, you must create repository accounts for each user that you want to be able to connect. Even if you select this option, we recommend that you create appropriate user accounts in advance in order to grant appropriate rights and permissions on your various repository folders and documents. LDAP users connecting to the repository are automatically added to the **`External users`** and **`All users`** groups, and are limited, by default, to read access on the repository.

3.1.1.1 Managing Externally-Authenticated Repository Users

You can modify the default privileges and permissions for such users by changing the privileges and permissions granted to the **`External users`** group, or provide specific privileges and permissions for individual external users by pre-creating repository user accounts for them.

Context

For example, if you want to allow any user connecting to the repository to create and edit diagrams in the **`Processes`** folder and submit them for approval, you would:

- Grant the **External users** group the **Edit on Web** privilege (see Granting Privileges to Users and Groups [page 533]).
- Grant the **External users** group **Submit** permission on the **Processes** folder (see Granting Access Permissions on Repository Items [page 534]).

In many environments, you will want to grant different privileges to different groups of users, or provide them with different permissions. For example you may want to allow users to submit changes only for processes in their particular line of business based on sub-folders beneath the **Processes** folder. In this or other more complicated cases (or if you want to restrict which users can connect to the repository and have not selected the *Auto-create user accounts in repository* option), you should create accounts for your anticipated users before inviting them to connect.

Procedure

1. Select *Repository* > *Administration* > *Users* to open the List of Users.
2. Click the *Add a Row* tool to create a new user, and then click the *Properties* tool to open its property sheet.
3. Enter the user's corporate account name in the *Login name* field, select *External (managed by LDAP)*, and click the *Check Name* button to verify the login name and auto-fill the remaining fields, which are set, with the exception of *Comment*, to read-only.
4. Click the *Groups* tab and add the user to any appropriate groups (see Adding Users and Groups to a Group [page 532]).

   All externally-authenticated users are added to:
   - **All users (PUBLIC)** groups, from which they inherit, by default, **Read** permission on all the contents of the repository.
   - **External users (EXTERNAL)** - from which they inherit, by default, the **Connect** privilege.
5. Click the *Privileges* tab and specify any appropriate privileges for the user (see Granting Privileges to Users and Groups [page 533]). By default, all users are granted the **Connect** privilege, which entitles them to connect to the repository.
6. Click the *Profiles* tab and associate the user with appropriate object permissions, UI, and general preferences profiles (see Applying Profiles to Users and Groups [page 567]).
7. Click *OK* to complete the creation of the user.

3.1.2 Connecting to an SMTP Server for Notifications

A repository administrator can automate the sending of emails for passwords, changelist submissions, and other notifications to users by specifying an SMTP server. If an SMTP server is not specified, the administrator must distribute passwords manually and notifications related to the creation of PowerDesigner Web comments and the submission and approval of change lists cannot be sent.

Procedure

1. Connect to the repository and select *Repository* > *Administration* > *SMTP Parameters* (or right-click the root node, and select *Properties* to open the repository property sheet, and then click the *SMTP* tab).
2. Enter appropriate values for each of the following settings:

Parameter	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 mails.
Use Secure Socket Layer (SSL)	Specifies to connect to the SMTP mail server through SSL.
Server requires authentication	Specifies that the SMTP server requires authentication. If you select this option, then you must specify an SMTP account and password, and can select to use Secure Password Authentication (SPA).

3. Click the *Test* button to send a test email to the sender's email address.
4. Click *OK* to save your changes.

3.1.3 Defining a Password Policy

In environments where PowerDesigner manages user passwords, the repository administrator is responsible for defining a password policy to ensure that passwords are sufficiently secure and are changed at appropriate intervals.

Context

> **i Note**
>
> The password policy governs only users who are not managed by LDAP (see Managing Externally-Authenticated Repository Users [page 516]) or single sign-on (see Enabling Single Sign-On for PowerDesigner Web [page 527]).

Procedure

1. Connect to the repository and select *Repository* > *Administration* > *Password Policy* (or right-click the root node, and select *Properties* to open the repository property sheet, and then click the *Passwords* tab).

2. Select policy settings as appropriate:

Parameter	Description
Password length	Specifies the minimum and maximum permitted length of passwords. This option cannot be disabled. The minimum length for a password is 6 characters.
Password must contain	Specifies that passwords must contain at least one of each of the character types selected.
Disallow reuse of previous <x> passwords	Prevents users from reusing the specified number of old passwords.
Enforce changing of passwords after <x> days	Requires that users change their passwords after the specified number of days.
Block inactive users after <x> days without connection	Blocks users if they try to log in after the specified number of days of inactivity.
Temporarily block users for <x> minutes after <y> failures to log in	Blocks users for the specified number of minutes if they submit an invalid combination of username and password the specified number of times.
Temporary passwords issued by an administrator are valid for <x> days	Specifies the period for which temporary passwords (which are issued when a user is created or unblocked) are valid. Users attempting to use a temporary password after this time will be blocked.

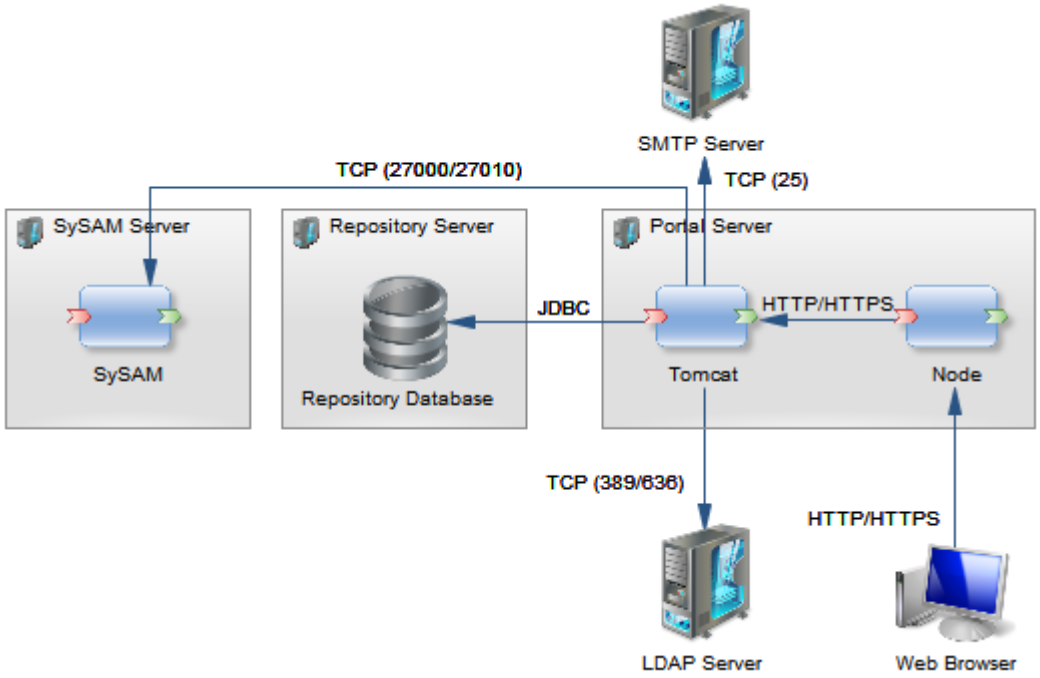
3. Click *OK* to save your changes.

   Changes made to the policy take effect immediately. If your policy becomes more restrictive, users whose passwords are no longer compliant will be instructed to change their password when next they connect.

3.1.4 Enabling Secure Connections (SSL/TLS) for PowerDesigner Web

We recommend that you encrypt the communication between the PowerDesigner Web server and client Web browsers and between its Node and Tomcat components.

Context



> **i Note**
>
> SSL/TLS support levels are determined by the application server supporting the communication with the client Web browsers and between its Node and Tomcat components. For Tomcat documentation about enabling SSL/TLS, see the SSL/TLS Configuration HOW-TO .

In this procedure, we generate or modify the following files:

- `pdca.crt/pdweb.crt` - Temporary certificates for a certification authority and the PowerDesigner Web server to allow you to test enabling secure connections. These must be replaced by legitimately signed certificates to allow your users to connect securely to PowerDesigner Web without a security warning (see step 8 [page 525]).
- `server.jks` [generated] - To contain the `tomcat.crt` certificate.
- `cacerts` [modified] - To allow Tomcat to trust Node.
- `configExt.json` [modified] - To configure secure connections for Node.
- `server.xml` [modified] - To configure secure connections for Tomcat.

> **Note**
>
> In this procedure, we give detailed instructions for producing PEM format certificates and configuring Node and Tomcat. If your environment uses DER format certificates or a different Java application server, then you should adapt the steps accordingly.

Procedure

1. Prepare your environment for this change:
   a. Download and install OpenSSL and add the `openssl` command to your path.
   b. Locate the `keytool` utility in your Java installation and add it to your path.

      If you have installed a new Tomcat 8 server with PowerDesigner Web, your `keytool` is located by default at `C:\Program Files\SAP\PowerDesigner Portal 16\sapjvm_8_jre\bin\keytool.exe`.
   c. Locate your Java installation truststore and obtain its password.

      If you installed a new Tomcat 8 server with PowerDesigner Web, your truststore is located by default at `C:\Program Files\SAP\PowerDesigner Portal 16\sapjvm_8_jre\lib\security\cacerts` with a default password of **`changeit`**.
2. Prepare a folder to contain your keys, certificates, and other files, and navigate into it.

   In this procedure, we will use the `C:/ssl` folder to contain the generated files.
   a. Copy the following code into a file and save it in your folder as `pdWebSecure.cfg`:

```
# OpenSSL configuration file for enabling secure connections for the
PowerDesigner Web server
# Define the following properties to identify the server:
# - FQDN(fully qualified server name),
# - ServerName(non-qualified name),
# - IpAddress
# - CommonName(Subject name) and other Subject properties
FQDN = <FullyQualifiedServerName>
ServerName = <ServerName>
IPAddress = <ServerIPAddress>
CommonName = <FullyQualifiedServerName>
CountryName = <TwoLetterCountryCode>
StateName = <State>
City = <City>
Organization = <Organization>
OrganizationUnit = <OrganizationUnit>
# If appropriate, modify the ALTNAMES line to add DNS aliases to the CSR.
# All DNS names must resolve to the same IP address as the FQDN.
ALTNAMES = DNS:$FQDN , DNS:$ServerName , IP:$IpAddress
# --- Do not modify below this line---
[ req ]
default_bits = 2048
default_md = sha256
prompt = no
encrypt_key = no
distinguished_name = dn
req_extensions = req_ext
[ dn ]
CN = $CommonName
C = $CountryName
ST = $StateName
```

```
L = $City
O = $Organization
OU = $OrganizationUnit
[ req_ext ]
subjectAltName = $ALTNAMES
```

b. Edit the `pdWebSecure.cfg` file to provide appropriate values for the variables defined at the beginning of the file, and then save your changes:

   - `FQDN = <FullyQualifiedServerName>` - Must contain the fully qualified name of the server.
   - `ServerName = <ServerName>` - Must contain the unqualified name of the server.
   - `IPAddress = <ServerIPAddress>` - Must contain the IP address of the server.
   - `CommonName = <FullyQualifiedServerName>` - Must contain the fully qualified name of the server.
   - `CountryName`, etc - Should contain appropriate values for the server.
   - `ALTNAMES = DNS:$FQDN , DNS:$ServerName` - If appropriate or required for your environment, add any additional DNS aliases to the end of the line.

c. Copy the following code into a file and save it in your folder as `pdWebSecure.bat`:

```
rem "-------Batch file to enable secure connections for the PowerDesigner
Web server"
rem "-------Generate a temporary PowerDesigner certification authority,
which will be replaced by your organization's real CA"
openssl genrsa -out pdca.key 2048
openssl req -new -key pdca.key -out pdca.csr -subj "/CN=pdca"
openssl x509 -req -days 3650 -in pdca.csr -signkey pdca.key -out pdca.crt
rem "-------Generate a certificate for the PDWeb server, issued by the
temporary PowerDesigner CA"
openssl genrsa -out pdweb.key 2048
openssl req -new -key pdweb.key -out pdweb.csr -config pdWebSecure.cfg -
extensions req_ext
openssl req -in pdweb.csr -text -noout
openssl x509 -req -days 3650 -in pdweb.csr -CA pdca.crt -CAkey pdca.key -
CAcreateserial -out pdweb.crt -extfile pdWebSecure.cfg -extensions req_ext
openssl x509 -in pdweb.crt -text -noout
rem "-------Generate the server.jks keystore"
openssl pkcs12 -export -in pdweb.crt -inkey pdweb.key -out
tempkeystore.p12 -name pdweb -CAfile pdca.crt -caname ca -password
pass:<KeystorePassword>
keytool -importkeystore -deststorepass <KeystorePassword> -destkeystore
server.jks -srckeystore tempkeystore.p12 -srcstoretype PKCS12 -
srcstorepass <KeystorePassword> -noprompt
keytool -list -keystore server.jks -storepass <KeystorePassword>
rem "-------Import the temporary PowerDesigner CA to the cacerts
truststore"
keytool -delete -alias ca -keystore "C:\Program Files\SAP\PowerDesigner
Portal 16\sapjvm_8_jre\lib\security\cacerts" -storepass
<TruststorePassword> -noprompt
keytool -importcert -keystore "C:\Program Files\SAP\PowerDesigner Portal
16\sapjvm_8_jre\lib\security\cacerts" -alias ca -file pdca.crt -storepass
<TruststorePassword> -noprompt
pause
```

d. Edit the `pdWebSecure.bat` to provide appropriate values for the following passwords, and then save your changes:

   - `<KeystorePassword>` - Choose a secure password for the `server.jks` keystore and enter it four times in the commands in the `Generate the server.jks keystore` section.
   - `<TruststorePassword>` - Enter the password for the `cacerts` truststore two times in the commands in the `Import the PowerDesigner CA to the cacerts truststore` section.

> **Note**
>
> If your truststore is not located at `C:\Program Files\SAP\PowerDesigner Portal 16\sapjvm_8_jre\lib\security\cacerts`, you must also provide the appropriate path in the two commands in this section.

3. Stop the `SAP PowerDesigner Nodeserver` service and then stop the `SAP PowerDesigner Portal` service.
4. Open a command prompt with administrator privileges in your `C:/ssl` folder and run the following command:

```
pdWebSecure.bat
```

The command will generate all the necessary files to enable secure connections for your PowerDesigner Web installation and import the appropriate certificates into your keystore and truststore.

> **Note**
>
> The `pdweb.crt` certificate signed by the private certification authority is sufficient for testing, but will not be trusted by your users' Web browsers, and they will receive a security warning when accessing your server. To securely identify your server, you must send your certificate request to your organization's certification authority (see step 8 [page 525]).

5. Edit `<install_dir>\nodeserver\configExt.json` to enable encryption for Node and point to the generated files:

```
{
            "node": {
              "secure": true,
              "keyFile": "C:/ssl/pdweb.key",
              "passphrase": null,
              "certFile": "C:/ssl/pdweb.crt",
              "caFile": null,
              "useSSO": false,
              "port": 3030,
              "home": "C:/Program Files/SAP/PowerDesigner Portal 16",
              "loggingLevel": null,
              "useCluster": true,
              "cpus": null,
              "maxCpus": null,
              "useCache": true
            },
            "web": {
              "applicationContext": "/powerdesigner-web",
              "resourcesPath": "/resources"
            },
            "connections": {
              "repository": {
                "secure": true,
                "caFile": "C:/ssl/pdca.crt",
                "host": "<FullyQualifiedServerName>",
                "port": 8443,
                "oDataPath": "/odata.svc",
                "oData2Path": "/odata2.svc",
                "maxTimeout": 1800000
                }
              }
            }
```

Where:

- In the **node** list:
  - **"secure": true** - Enables encryption between client Web browsers and the Node server.
  - **"keyFile": "C:/ssl/pdweb.key"** - Points to the generated `pdweb.key` file.

    > **Note**
    >
    > For this and other paths, you should use only forward slashes as in the example to ensure that the JSON is correctly interpreted.

  - **"passphrase": null** - Not required for the self-signed certificate.
  - **"certFile": "C:/ssl/pdweb.crt"** - Points to the generated `pdweb.crt` file.
  - **"useSSO": false** - Single sign-on is not supported in this configuration. To enable it, see Enabling Single Sign-On for PowerDesigner Web [page 527].
  - **"port": 3030** - Specifies the port to which your users will connect with encryption.
  - **"home": "C:/Program Files/SAP/PowerDesigner Portal 16"** - Specifies the portal install directory.
  - **"loggingLevel": null** - Specifies the verbosity of the log files. We recommend that you set the level to **null** by default. If you encounter problems, it may be helpful to change it to **"debug"**.

    > **Note**
    >
    > The most useful log files are located at `C:\Program Files\SAP\PowerDesigner Portal 16\log`.

  - Other options can keep their existing values.
- In the **connections/repository** list:
  - **"secure": true** - Enables encryption between Node and Tomcat.
  - **"caFile": "C:/ssl/pdca.crt"** - Points to the certification authority certificate.
  - **"host": "<FullyQualifiedServerName>"** - Specifies the fully qualified name of the server on which both Node and Tomcat are installed.

    > **Note**
    >
    > We recommend that you set the value to **<FullyQualifiedServerName>**. If node is unable to resolve the server name, it may be helpful to change it to **<IPAddress>**.

  - **"port": 8443** - Specifies the port that Node will use to connect to Tomcat, and must be the same as the port specified in `server.xml` (see next step).
  - Other options can keep their existing values.

6. Modify `<install_dir>\Tomcat\conf\server.xml` to enable encryption for Tomcat:
   a. Create a **<Connector** element with the following values:

```
<Connector
    port="8443"
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    clientAuth="want"
    sslProtocol="TLS"
    keystoreFile="C:/ssl/server.jks"
    keystoreType="JKS"
```

```
    keystorePass="<secure_password>"
    compression="on"
    compressionMinSize="2048"
    noCompressionUserAgents="gozilla, traviata"
    compressableMimeType="text/html,text/xml,application/xml,application/
atom+xml,application/x-font-ttf,application/javascript,application/
json,text/css"/>
```

Where:

- **port** - must match the port set in <install_dir>\nodeserver\configExt.json for connections/repository/port.
- **sslProtocol** - is set to the list of TLS versions that are accepted by the server. For example:
  - **TLS** - Accept any version of TLS.
  - **TLSv1.2** - Accept only TLS v1.2.
  - **TLSv1.1+TLSv1.2** - Accept only TLS v1.1 or TLS v1.2.
- **clientAuth** is set to **want** to allow a standard login if the certificate cannot be found.
- **keystoreFile** points to your generated server.jks keystore.
- **keystorePass** is the <secure_password> defined for your generated server.jks keystore.
- The compression options are the standard options set by the PowerDesigner Web installer to optimize performance.

b. Comment out any other connector elements to prevent any non-secure access.

7. Restart and test your environment:
   a. Start the SAP PowerDesigner Portal service and then start the SAP PowerDesigner Nodeserver service.
   b. Navigate to the login page using http://<server>:3030 to confirm that it is no longer accessible.
   c. Navigate to the login page using https://<server>:3030 to confirm that the encrypted connection is accessible with a security warning.

> **i Note**
>
> If you encounter problems, the most useful log files are located at C:\Program Files\SAP\PowerDesigner Portal 16\log. To obtain more verbose logging information, open <install_dir>\nodeserver\configExt.json, set **"loggingLevel": "debug"**, and restart the server.

8. Replace the self-signed pdweb.crt certificate with a certificate signed by your company's certification authority:
   a. Obtain the following files and save them to C:/ssl:
      - Your organization's root certification authority certificate.
      - A certificate (*.crt) and key (*.key) signed by your company's certification authority to identify the PowerDesigner Web server.

> **i Note**
>
> It is commonly necessary to edit the received certificate file so that it contains, in order, the contents of:
> - The signed certificate for your PowerDesigner Web server
> - The certificate of one or more intermediate certification authorities
> - The certificate of your organization's root certification authority

b. Edit <install_dir>\nodeserver\configExt.json to use the new signed PowerDesigner Web certificate and your organization's root certification authority certificate:

```
{
            "node": {
              "secure": true,
              "keyFile": "C:/ssl/pdweb.key",
              "passphrase": <passphrase>,
              "certFile": "C:/ssl/pdweb.crt",
              "caFile": null,
              "useSSO": false,
              "port": 3030,
              "home": "C:/Program Files/SAP/PowerDesigner Portal 16",
              "loggingLevel": null,
              "useCluster": true,
              "cpus": null,
              "maxCpus": null,
              "useCache": true
            },
            "web": {
              "applicationContext": "/powerdesigner-web",
              "resourcesPath": "/resources"
            },
            "connections": {
              "repository": {
                "secure": true,
                "caFile": "C:/ssl/<official_ca_file>.crt",
                "host": "<FullyQualifiedServerName>",
                "port": 8443,
                "oDataPath": "/odata.svc",
                "oData2Path": "/odata2.svc",
                "maxTimeout": 1800000
                }
              }
            }
```

Where:

- **"keyFile": "C:/ssl/pdweb.key"** - Points to the new signed PowerDesigner Web `.key` file.

  > **i** Note
  >
  > For this and other paths, you should use only forward slashes as in the example to ensure that the JSON is correctly interpreted.

- **"passphrase": <passphrase>** - If required in your environment.
- **"certFile": "C:/ssl/pdweb.crt"** - Points to the new signed PowerDesigner Web `.crt` file.
- **"caFile": "C:/ssl/<official_ca_file>.crt"** - Points to your organization's certification authority certificate.
- Other options can keep their existing values.

c. Run the following command to overwrite the temporary keystore file with a new one using the signed certificate and your organization's certification authority certificate:

```
openssl pkcs12 -export -in pdweb.crt -inkey pdweb.key -out
tempkeystore.p12 -name pdweb -CAfile <official_ca_file>.crt -caname ca -
password pass:<KeystorePassword>
```

Then run the following command to overwrite the `server.jks` keystore with a new one containing the signed certificate:

```
keytool -importkeystore -deststorepass <KeystorePassword> -destkeystore
server.jks -srckeystore tempkeystore.p12 -srcstoretype PKCS12 -
srcstorepass <KeystorePassword> -noprompt
```

d. Run the following command to add your organization's certification authority certificate to the jre truststore:

```
keytool -importcert -keystore "C:\Program Files\SAP\PowerDesigner Portal
16\sapjvm_8_jre\lib\security\cacerts" -alias ca -file
<official_ca_file>.crt -storepass <TruststorePassword> -noprompt
```

e. Start the `SAP PowerDesigner Portal` service and then start the `SAP PowerDesigner Nodeserver` service.

f. Navigate to the login page using `https://<server>:3030` to confirm that there is no longer a security warning.

3.1.5 Enabling Single Sign-On for PowerDesigner Web

In environments where X.509 client authentication is in place, an administrator can enable PowerDesigner Web to authenticate users by the user certificate stored on their client machine. The client browser sends a certificate issued by a trusted certificate authority to identify the user and they are logged in automatically, without the need to manually enter their credentials.

Procedure

1. If you have not already done so, enable SSL (see Enabling Secure Connections (SSL/TLS) for PowerDesigner Web [page 520]).
2. Prepare your environment for this change:
   a. Enable LDAP (see Connecting to an LDAP Server for User Authentication [page 513]).

      Enabling LDAP allows the preparation of user accounts for administrators and enhances the ability of PowerDesigner Web to obtain the real names and email addresses of your users, which may not always be available in their certificates, from the LDAP server.

   b. Create accounts for one or more External (managed by LDAP) users who will act as repository administrators (Managing Externally-Authenticated Repository Users [page 516]) and add them to the **`Administrators`** group.

   c. [optional] Customize the rights and permissions that standard users are granted by changing the default rights and permissions granted to the **`External users`** group or pre-creating user accounts for other individuals.
3. Generate a private certification authority to control the communication between the node frontend and Tomcat backend of the server.
   a. Copy the following code into a file and save it in your folder as `pdWebSSO.bat`:

```
rem "-------Batch file to enable SSO for the PowerDesigner Web server"
```

```
rem "-------Generate private CA to manage communications between node and
Tomcat"
openssl genrsa -out privateca.key 2048
openssl req -new -key privateca.key -out privateca.csr -subj "/CN=ssoca"
openssl x509 -req -days 3650 -in privateca.csr -signkey privateca.key -out
privateca.crt
rem "-------Generate frontend certificate for node.js client issued by
internal CA ..."
openssl genrsa -out frontend.key 1024
openssl req -new -key frontend.key -out frontend.csr  -subj "/CN=frontend"
openssl req -in frontend.csr -text -noout
openssl x509 -req -days 3650 -in frontend.csr -CA privateca.crt -CAkey
privateca.key -CAcreateserial -out frontend.crt
openssl x509 -in frontend.crt -text -noout
rem "-------Generate a trust keystore for tomcat server ... "
keytool -genkey -alias tempalias -keyalg RSA -keysize 2048 -dname
"CN=temp" -keystore truststore.jks -storepass <TruststorePassword> -
keypass <TruststorePassword> -noprompt
keytool -delete -alias tempalias -keystore truststore.jks -storepass
<TruststorePassword> -noprompt
rem "-------Import internal ca to tomcat trust keystore in order to trust
node server as its one and only client"
keytool -importcert -keystore truststore.jks -alias privateca -file
privateca.crt -storepass <TruststorePassword> -noprompt
keytool -list -keystore truststore.jks -storepass <TruststorePassword>
pause
```

b. Edit the `pdWebSSO.bat` to specify a secure password, `<TruststorePassword>`, for the `truststore.jks` keystore, enter the password in the five places indicated, and then save your changes:

c. Stop the `SAP PowerDesigner Nodeserver` service and then stop the `SAP PowerDesigner Portal` service.

d. Open a command prompt with administrator privileges in your `C:/ssl` folder and run the following command:

```
pdWebSSO.bat
```

The command will generate all the necessary files to enable SSO for your PowerDesigner Web installation.

4. Edit `<install_dir>\nodeserver\configExt.json` to enable SSO :

```
{
            "node": {
              "secure": true,
              "keyFile": "C:/ssl/pdweb.key",
              "passphrase": <PassPhrase>,
              "certFile": "C:/ssl/pdweb.crt",
              "caFile": <SSO_CA_Certificate>,
              "useSSO": true,
              "port": 3030,
              "home": "C:/Program Files/SAP/PowerDesigner Portal 16",
              "loggingLevel": null,
              "useCluster": true,
              "cpus": null,
              "maxCpus": null,
              "useCache": true
            },
            "web": {
              "applicationContext": "/powerdesigner-web",
              "resourcesPath": "/resources"
            },
            "connections": {
              "repository": {
```

```
                "secure": true,
                "keyFile": "C:/ssl5/frontend.key",
                "certFile": "C:/ssl5/frontend.crt",
                "caFile": "<Global Root Certificate>",
                "host": "<Fully_Qualified_Server_Name>",
                "port": 8443,
                "oDataPath": "/odata.svc",
                "oData2Path": "/odata2.svc",
                "maxTimeout": 1800000
                }
            }
        }
```

Where:

- **"caFile": <SSO_CA_Certificate>** - Points to your organization's SSO certification authority certificate.
- **"useSSO": true** - Enables SSO.
- **"keyFile": "C:/ssl5/frontend.key"** - Points to the new key for the node client generated by the private certification authority.
- **"certFile": "C:/ssl5/frontend.crt"** - Points to the new certificate for the node client generated by the private certification authority.
- Other options can keep their existing values.

5. Modify `<install_dir>\Tomcat\conf\server.xml` to enable Tomcat to use the new truststore by adding the following three lines to your secure **<Connector** element:

   - **truststoreFile** points to your generated `truststore.jks` truststore.
   - **truststoreType** is set to `JKS`.
   - **truststorePass** specifies your `<TruststorePassword>`.

   Your connector should now be as follows

```
<Connector
    port="8443"
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    clientAuth="want"
    sslProtocol="TLS"
    keystoreFile="C:/ssl/server.jks"
    keystoreType="JKS"
    keystorePass="<secure_password>"
    truststoreFile="C:/ssl/truststore.jks"
    truststoreType="JKS"
    truststorePass="<TruststorePassword>"
    compression="on"
    compressionMinSize="2048"
    noCompressionUserAgents="gozilla, traviata"
    compressableMimeType="text/html,text/xml,application/xml,application/atom
+xml,application/x-font-ttf,application/javascript,application/json,text/
css"/>
```

6. Restart and test your environment:
   a. Start the `SAP PowerDesigner Portal` service and then start the `SAP PowerDesigner Nodeserver` service.
   b. Navigate to the login using the `https://` prefix. You should bypass the login screen and go straight to the homepage.

3.1.6 Creating Repository Users

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

Context

The following standard users are automatically created in the PowerDesigner repository:

- **`ADMIN`** - Has all available privileges and implicit **`Full`** permission on all repository folders.
- **`_ADMIN`** - Is created as a temporary emergency user for connecting to a repository when no administrator is able to access it (see Obtaining Emergency Access to the Repository [page 539]).

> **i Note**
>
> This procedure is for creating users authenticated by the repository. PowerDesigner also supports delegating authentication of users to an LDAP server (see Connecting to an LDAP Server for User Authentication [page 513]) and authentication by X509 certificate (see Enabling Single Sign-On for PowerDesigner Web [page 527]).

Procedure

1. Select *Repository* > *Administration* > *Users* to open the List of Users.
2. Click the *Add a Row* tool and then click the *Properties* tool to open the new user's property sheet.
3. Enter the following properties as appropriate:

Property	Description
Login Name	Specifies the account name used to connect to the repository. **i Note** Account names and passwords managed by PowerDesigner are case-sensitive and must only contain standard ASCII characters.
External (managed by LDAP)	By default, users must enter a password managed by PowerDesigner to connect to the repository. To allow them to enter their standard corporate password, select *External (managed by LDAP)* (see Managing Externally-Authenticated Repository Users [page 516]).
Full Name	Specifies the real name of the user.
Email	Specifies the email address of the user. If you have specified an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) this address will be used to send the password.
Comment	Specifies any additional information about the user.

Property	Description
Status	Specifies the status of the user, which can be: ○ Active - the user can access the repository. ○ Inactive - the user is no longer active in the repository (see Deactivating Users [page 537]) and may not connect. The user remains in the List of Users and can be reactivated at any time. ○ Blocked - the user has violated one of the rules of the password policy (see Defining a Password Policy [page 518]), and cannot access the repository until an administrator unblocks her account (see Unblocking Blocked Users [page 536]).

4. If you have not enabled delivery of passwords by email (see Connecting to an SMTP Server for Notifications [page 517]), click the *Reset Password* button and then select the *Show* checkbox to display the password and note it for transmission to the user.
5. Click the *Groups* tab and add the user to any appropriate groups (see Adding Users and Groups to a Group [page 532]).

   All users are added to the **`Public`** group, from which, by default, they inherit the **`Read`** permission on all the contents of the repository (see Granting Access Permissions on Repository Items [page 534]).
6. Click the *Privileges* tab and specify any appropriate rights for the user (see Granting Privileges to Users and Groups [page 533]). By default, all users are granted the **`Connect`** right, which entitles them to connect to the repository.
7. Click the *Profiles* tab and associate the user with appropriate object permissions, UI, and general preferences profiles (see Applying Profiles to Users and Groups [page 567]).
8. Click *OK* to complete the creation of the user. If an SMTP server is configured, the password will be sent to the user at the recorded email address.

   Users must log in with their temporary password before the delay specified in the password policy (see Defining a Password Policy [page 518]). When they first log in they will be required to change the temporary password.

3.1.7 Creating Repository Groups

The repository administrator is responsible for creating groups of users in the repository. Users are added to groups in order to simplify the granting of privileges and permissions. You can create hierarchies of groups. For example, you could insert the **`Designers`**, **`Quality Assurance`**, and **`Documentation`** groups into the **`R&D`** group, to which you assign permissions to documents that all these groups must use.

Context

The following standard groups are automatically created in the PowerDesigner repository:

- Administrators, [ADMN] - Has, by default, all available privileges and implicit **`Full`** permission on all repository folders.
- All users [PUBLIC] - Has, by default, **`Read`** permission on the repository root. All users belong to this group and can thus, by default, check out any document.

- External users [EXTERNAL] - Has, by default, no privileges or permissions. Users authenticated via LDAP (see Connecting to an LDAP Server for User Authentication [page 513]) or single-sign on (see Enabling Single Sign-On for PowerDesigner Web [page 527]) are automatically added to this group when they connect for the first time.

Procedure

1. Select *Repository* > *Administration* > *Groups* to open the List of Groups.
2. Click the *Add a Row* tool and then click the *Properties* tool to open the new group's property sheet.
3. Click the *Members* tab and add any appropriate users and groups to the group (see Adding Users and Groups to a Group [page 532]).
4. Click the *Parent Groups* tab and add the group to any appropriate groups (see Adding Users and Groups to a Group [page 532]).
5. Click the *Rights* tab and specify any appropriate rights for the group (see Granting Privileges to Users and Groups [page 533]). By default, groups do not have any rights.
6. Click the *Profiles* tab and associate the group with an appropriate object permissions, UI, and general preferences profile (see Applying Profiles to Users and Groups [page 567]).
7. Click *OK* to complete the creation of the group.

3.1.7.1 Adding Users and Groups to a Group

You can add users and groups as members of a group from either the user or the group property sheet.

Context

There are three ways to add members to a group:

- To add a user to a group from the user's property sheet, select the *Groups* tab, which lists the groups to which the user belongs. Click the *Add Groups* tool to open a list of groups, select one or more, and then click *OK* to add the user to them.

  > **i** Note
  >
  > Click the *Show All Parent Groups* tool to show all the parent groups of the groups of which the user is a member, and from which she may inherit additional rights, permissions, and profiles.

- To add a group to a parent group from the child group's property sheet, select the *Parent Groups* tab, which lists the groups to which the group belongs. Click the *Add Groups* tool to open a list of groups, select one or more, and then click *OK* to add the group to them.
- To add a user or group to a group from the parent group's property sheet, select the *Members* tab, which lists the users and groups which are members of the group. Click the *Add* tool to open a dialog with lists of users and groups on sub-tabs, select one or more, and then click *OK* to add them to the group.

> **Note**
>
> Click the *Show All Children Members* tool to show all the groups and users that belong to the members of the group, and who will inherit rights, permissions, and profiles assigned to the group.

3.1.7.2 Deleting a Group

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

Procedure

1. Select *Repository* > *Administration* > *Groups* to open the List of Groups.
2. Select a group in the list and click the *Delete* tool. The group is removed from the list, and:
   - Any document permissions granted to the group are revoked.
   - Any members of the group lose whatever privileges or permissions they had inherited from the group.
   - The group is removed from membership of any groups to which it belonged.
3. Click *OK* to close the List of Groups.

3.1.8 Granting Privileges to Users and Groups

A new user has only the **`Connect`** privilege assigned by default and belongs only to the **`PUBLIC`** group. The repository administrator can grant additional privileges to the user either directly or by adding her to other groups.

Context

> **Note**
>
> The privileges a user or group receive when they become members of a group (see Adding Users and Groups to a Group [page 532]) are cumulative. For example, a user with the **`Manage Branches`** privilege will not lose this privilege when he becomes a member of a group that has only the **`Freeze Versions`** and **`Lock Versions`** privileges.

User privileges are associated with document permissions (see Granting Access Permissions on Repository Items [page 534]) to define the actions a user can effectively perform on a document.

> **i** Note
>
> Some user privileges are automatically implied by permissions:
>
> - A user with the **Write** permission on a repository document automatically has the **Lock Versions** and **Freeze Versions** privileges for it.
> - A user with the **Full** permission on a document can unlock or unfreeze a version he did not lock or freeze.

Procedure

1. Select *Repository* > *Administration* > *Users* (or *Repository* > *Administration* > *Groups*), select the appropriate user or group in the list, click the *Properties* tool, and click the *Rights* tab.
2. Select the check boxes corresponding to the privileges you want to assign:
   - **Connect** - Connect to the repository.
   - **Edit on Web** - Create and edit diagrams in PowerDesigner Web.
   - **Edit Extensions on Web** - Create and edit custom properties in PowerDesigner Web.
   - **Freeze Versions** - (see Freezing and Unfreezing Document Versions [page 275]).
   - **Lock Documents** - (see Locking and Unlocking Documents [page 276]).
   - **Manage Branches** - (see Branching Version Trees [page 281]).
   - **Manage Configurations** - (see Grouping Document Versions in a Configuration [page 278]).
   - **Manage All Documents** - Perform any action on any document version. Implicitly includes **Full** permission on all repository documents.
   - **Manage Users & Permissions** - Create, modify, and delete repository users and groups, grant them privileges, and add them to groups. Users with this privilege can list all repository documents and set permissions on them without needing explicit **Full** permission.
   - **Manage Repository** - Create, upgrade, and delete the repository database.
3. Click *OK* to save your changes and close the user property sheet.

3.1.9 Granting Access Permissions on Repository Items

The repository administrator or a user with **Full** permission on a document or folder can grant permissions on it. Permissions can be granted on the repository root, folders, models, model packages, and external application files, but not on individual model diagrams or objects.

Context

A user wanting to check documents out of and into the repository must have at least the following permissions:

- Check out - **Read** permission. When you create a user, she is inserted into the **Public** group, which by default is granted **Read** permission on the repository root.

- First check in (to create a document in the repository) or creating a folder - **Write** permission on the target location (the repository root or a folder) or **Submit** to propose the addition.
  The user who creates a repository item has, by default, **Full** permission over it, though this permission can be reduced by an administrator. Other permissions are cascaded down from the parent folder or root, so that if another user has **Write** permission on the parent folder, then he will have **Write** permission on the new item.
- Subsequent check ins (to update an existing document) - **Read** permission on the target location and **Write** permission on the document (including, for PowerDesigner models, all impacted packages).

> **i** Note
>
> Object permissions should be viewed in conjunction with the privileges granted to users or groups (see Granting Privileges to Users and Groups [page 533]).

Procedure

1. Right-click the item in the Repository Browser, and select *Permissions* to open the item's property sheet at the *Permissions* tab.

   > **i** Note
   >
   > A PowerDesigner project (see Projects and Frameworks [page 46]) acts as both a repository document and a folder. Use the project permissions to control working with the project file itself (project diagram and any framework or dependency matrices) and the folder permissions to grant access to the models contained within the project, as you would for a repository folder.

2. Click the *Add* button to open a list of available users and groups, select one or more, and click *OK* to add them to the list.
3. For each user or group, select the permission you want to grant in the *Granted Permission* column:
   - **List** - View the document or folder in the repository browser and in search results. Without this permission, the folder or document is hidden from the user.
   - **Read** - Also check documents out from the repository and compare documents.
   - **Submit** - Also check the document into a changelist for review by a user with **Write** permission.
   - **Write** - Also check in (with or without a changelist), freeze, and lock document versions.
   - **Full** - Also move and delete documents, manage permissions granted to users and groups, and remove locks on documents.

     > **i** Note
     >
     > Administrators, who have implicit **Full** permission on all repository objects, will only receive models for review if they have been granted explicit **Write** permission on them.

4. [optional] Click the *Copy Permissions to All Children* tool to propagate changes to the item's children.

   When you create a folder or check in a model or project, the permissions defined on its parent folder are propagated to it. However, subsequent changes made to the permissions for the parent are not applied to its children unless you click this tool. For example, if you grant **Write** permission on the **Major Project** folder, to the **Development Team 2** group, then they will not automatically be granted **Write** access on its existing contents.

5. [optional] Click the *Show All Authorized Users* tool to display all the users and groups who have a permission on the item, including those inheriting permissions from groups.

   Consult the read-only *Effective Permission* column to see the highest level of permission that each user or group has either directly or via a group. A user who has no directly granted permission on an item may inherit a **`Write`** permission by virtue of membership in a group, and will appear in the list when the *Show All Authorized Users* tool is pressed.

   > **i** Note
   >
   > If you delete a group from the list while the *Show All Authorized Users* tool is pressed, it and its members remain visible in the list with their granted and effective permissions reduced to **`<none>`**, until you click *Apply* or *OK*.

6. Click *OK* to save your changes.

3.1.10 Unblocking Blocked Users

The repository administrator or a user with the **`Manage Users & Permissions`** right can unblock users blocked for password policy violations.

Procedure

1. Select *Repository* > *Administration* > *Users* to open the List of Users, select the user in the list and click the *Properties* tool to open his property sheet.
2. Click the *Reset Password* button. If an SMTP server is configured, the password will be sent to the user at the recorded email address. Otherwise, select the *Show* checkbox to display the password and note it for transmission to the user.

   Users must log in with their temporary password before the delay specified in the password policy (see Defining a Password Policy [page 518]). When they first log in they will be required to change the temporary password.

3.1.11 Deactivating Users

The repository administrator or a user with the **Manage Users & Permissions** privilege can deactivate users. Inactive users cannot connect to the repository and the information held about them is deleted except for the log of their checkins and other repository actions.

Context

> ⚠ Caution
>
> A user cannot deactivate himself, even if he has the **Manage Users & Permissions** privilege.

Procedure

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

3.1.12 Monitoring Repository Activities

Users with the appropriate rights can review lists of repository activities.

- *List of Activities* - [requires the **Manage All Documents** right] View events concerning repository documents. Press Ctrl + Alt + V or select *Repository* > *Activities*:
  - **Check In** - User checks in a new document or a new version of a document. This event is also triggered each time a user saves a draft diagram on the Web.
  - **Change List Submission** - User submits a change list for review (see Change Lists [page 259]).
  - **Change List Approval** / **Rejection** / **Return for Revision** - A reviewer approves, rejects or returns a change list to the user for revision.
  - **Freeze** / **Unfreeze** - A document version is frozen or unfrozen (see Freezing and Unfreezing Document Versions [page 275]).
  - **Lock** / **Unlock** - A document is locked for editing or the lock is released manually or by a check in (see Locking and Unlocking Documents [page 276]).
  - **Document Move** - User moves a document from one repository folder to another (see Moving Documents in the Repository [page 258]).
  - **Version Deletion** / **Document Deletion** - User deletes a document version or a document (see Deleting Documents, Document Versions, and Folders [page 295]). If a version of the document being deleted is included in a configuration (see Grouping Document Versions in a Configuration [page 278]), then though it will no longer be available through the repository Browser, the version will not be deleted from the configuration, and the **Document Removal** action will be shown.

- **`Branch Creation`** - User creates a branch (see Branching Version Trees [page 281]).
- **`Integration`** - User integrates a document from one branch to another (see Integrating Changes from One Branch to Another [page 284]).
- **`Repository Upgrade`** - Administrator upgrades the repository to a new software version (see *Installation Guide* > *Installing the Repository* > *Upgrading the Repository*).
- **`Automatic Fix`** - An existing document is updated to reflect changes required by a new software version.

> **i Note**
>
> Click the *Customize Columns and Filter* tool to customize the list. Select an item in the list and click the *Properties* tool to open the property sheet of the affected object. You can use the *Versions* tab of the property sheet to check out or lock the document version.

- *Security Log* - [requires the **`Manage Users & Permissions`** right] View events concerning user accounts. Press `Ctrl` + `Alt` + `B` or select *Repository* > *Administration* > *User Activities*:
  - **`User Created`** - User's account is created, either by an administrator or automatically (if the *Auto-create user accounts in repository* option or single sign-on is enabled, see Connecting to an LDAP Server for User Authentication [page 513] and Enabling Single Sign-On for PowerDesigner Web [page 527]).
  - **`Login`** - User enters correct user name and password combination.
  - **`Failed Login`** - User enters incorrect user name and password combination.
  - **`Logout`** - User logs out.
  - **`Password Changed`** - User changes password.
  - **`Password Reset`** - User's password is changed by an administrator.
  - **`User Blocked for Failed Logins`** - User is blocked for having entered too many incorrect user name and password combinations (see Defining a Password Policy [page 518]).
  - **`User Blocked for Inactivity`** - User is blocked for having tried to login after too long a period of inactivity (see Defining a Password Policy [page 518]).
  - **`User Unblocked`** - User is unblocked by an administrator (see Unblocking Blocked Users [page 536]).
  - **`User Deactivated`** - User is deactivated by an administrator (see Deactivating Users [page 537]).
  - **`User Activated`** - User is reactivated by an administrator.
  - **`User Accessed`** - User's account information is accessed by an administrator.
  - **`User Updated`** - User's account information is updated by an administrator.
  - **`User Deleted`** - User's account is deleted by an administrator

> **i Note**
>
> Click the *Customize Columns and Filter* tool to customize the list.

3.1.13 Querying the Repository Using SQL

Administrators can run basic SQL SELECT queries against the repository through the Execute Query window. For more complex queries, you should use your DBMS query editor.

Context

> **i Note**
>
> You cannot execute SQL queries when connecting to the repository via the repository proxy.

Procedure

1. Connect to the repository and then select *Repository* > *Administration* > *Execute SQL* to open the Execute Query window.
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.

3.1.14 Obtaining Emergency Access to the Repository

In the event that no administrator is able to log in to a running repository, you can create an emergency administrator account to regain access.

Procedure

1. If the repository proxy is not already in use in your environment, install it on the repository server or a nearby machine for which you are an administrator (see *Installation Guide* > *Installing the Repository Proxy*).
2. Launch the Repository Proxy Configuration utility (`pdproxyconf16.exe`).

   > **i Note**
   >
   > On a Windows 7 machine, you must open the utility by right-clicking its icon or menu item and selecting *Run as administrator*.
3. If an instance is not already available for the repository in question, create one, specifying the appropriate data source, user name, password, and port to connect to the repository database.

4. Select the appropriate instance and click the *Create Emergency User* button to open the Emergency Password dialog, and then select *Show password* to see an automatically generated password, for use with the special `_ADMIN` account.
5. Open PowerDesigner and select *Repository* > *Connect*, select the appropriate repository, enter `_ADMIN` as the login name and the generated password in the Password field and click *OK*.

   > **Note**
   >
   > If you do not access the repository within 15 minutes, you must repeat the procedure to generate a new temporary password.

6. Reset the administrator's password or create a new administrator account as appropriate (see Creating Repository Users [page 530]).
7. Select *Repository* > *Disconnect* to disconnect the emergency user from the repository. The emergency user is deleted after disconnection.

3.2 Deploying an Enterprise Glossary and Library

An administrator can deploy a glossary and a library of enterprise reference models, resource files, and other documents, which will be pushed to and automatically updated for all users. By default, only administrators can edit the glossary or add models to or edit the library. You can provide access to other users using the standard repository permissions.

3.2.1 Deploying an Enterprise Glossary

The glossary model (GLM) helps you capture and organize the terminology to be used for naming your model objects. An administrator deploys the glossary, and users enable it in their models to provide autocompletion for object names and model checks to ensure compliance. The glossary appears in the Browser *Glossary* tab and updates are pushed to users each time they connect to the repository, or on demand. Only one glossary is permitted per repository.

For information about using the glossary, see The Glossary [page 38].

3.2.1.1 Creating Terms, Synonyms, and Related Terms

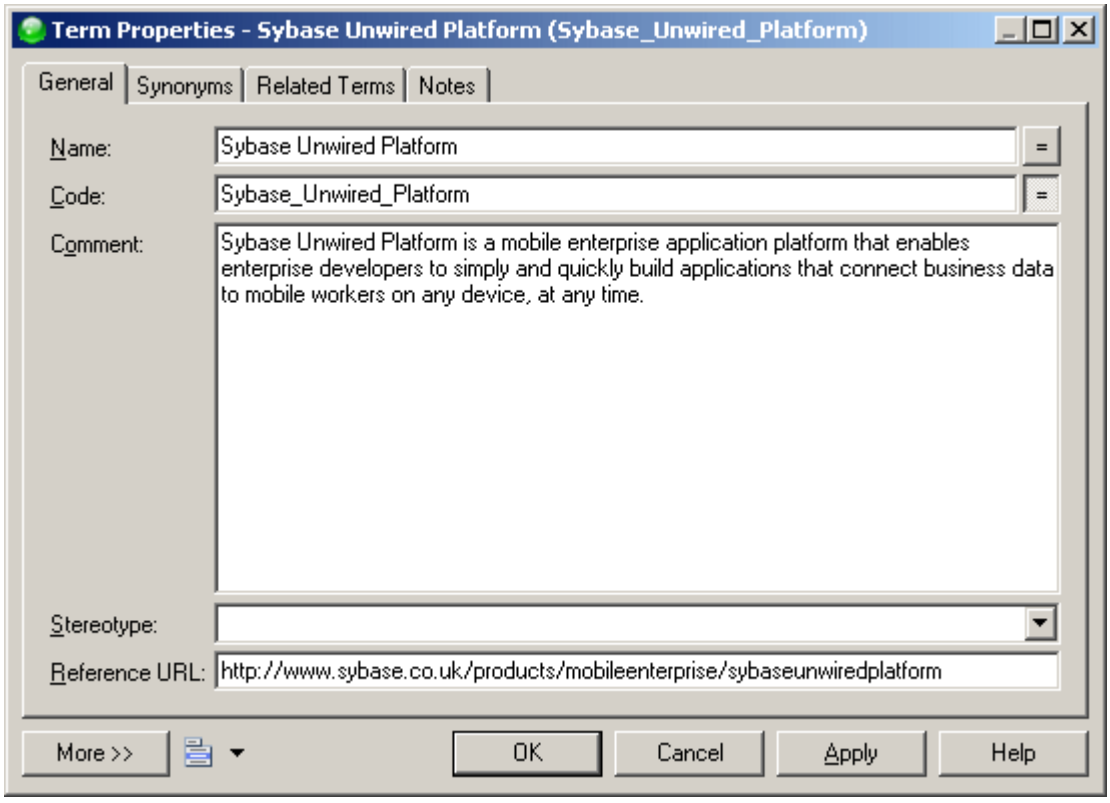
An administrator can define new glossary terms in the *Glossary* tab of the Browser.

Context

> **i** Note
>
> You can quickly populate the glossary by performing a mass import of terms from a model (see Importing Terms from a Model [page 543]) or a CSV or Excel file (Importing Terms from an Excel or CSV File [page 544]), or from the results of the `List of Missing Terms` glossary compliance model check (see Checking Glossary Compliance [page 154]). You can also add terms in a glossary terms list (see Glossary Terms Lists [page 546]).

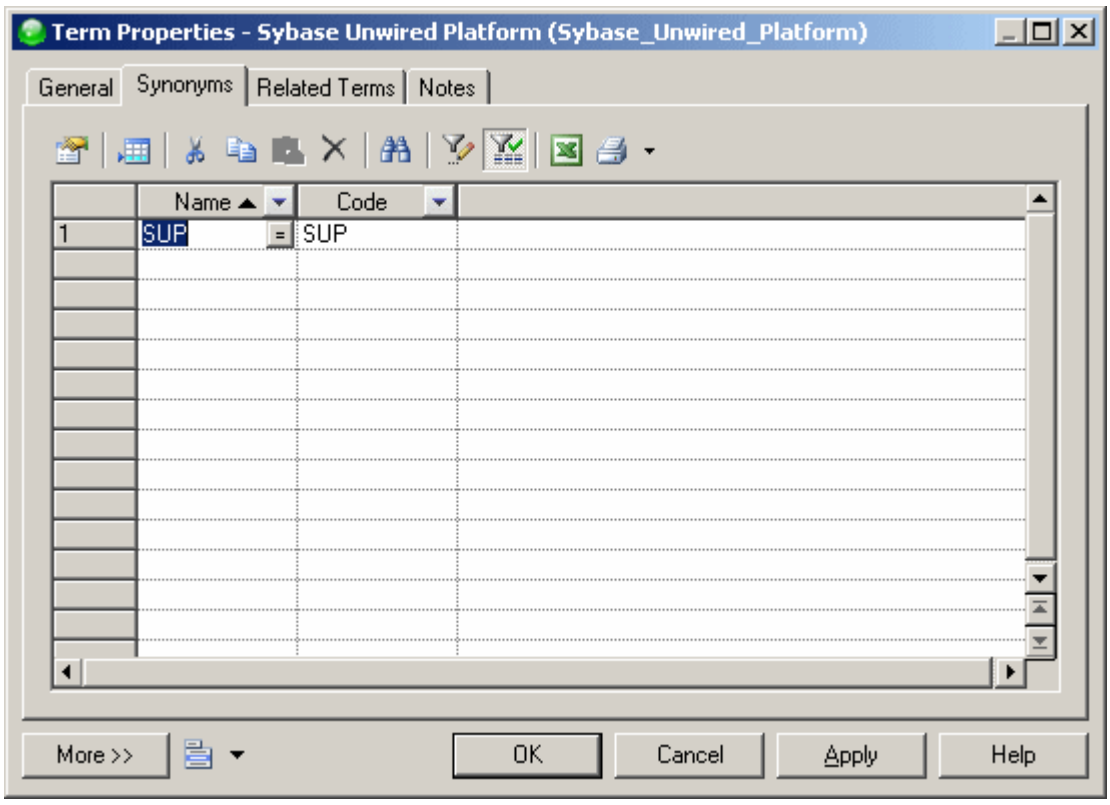
Procedure

1. Select the *Glossary* tab of the Browser, right-click the *Glossary* node, and select *New* > *Term*.
2. Enter the *Name* and *Code* of the term, which should be approved for use in your organization's models.
3. [optional] Enter a description of the term in the *Comment* field, and a *Reference URL* leading to a site giving further information about the term.



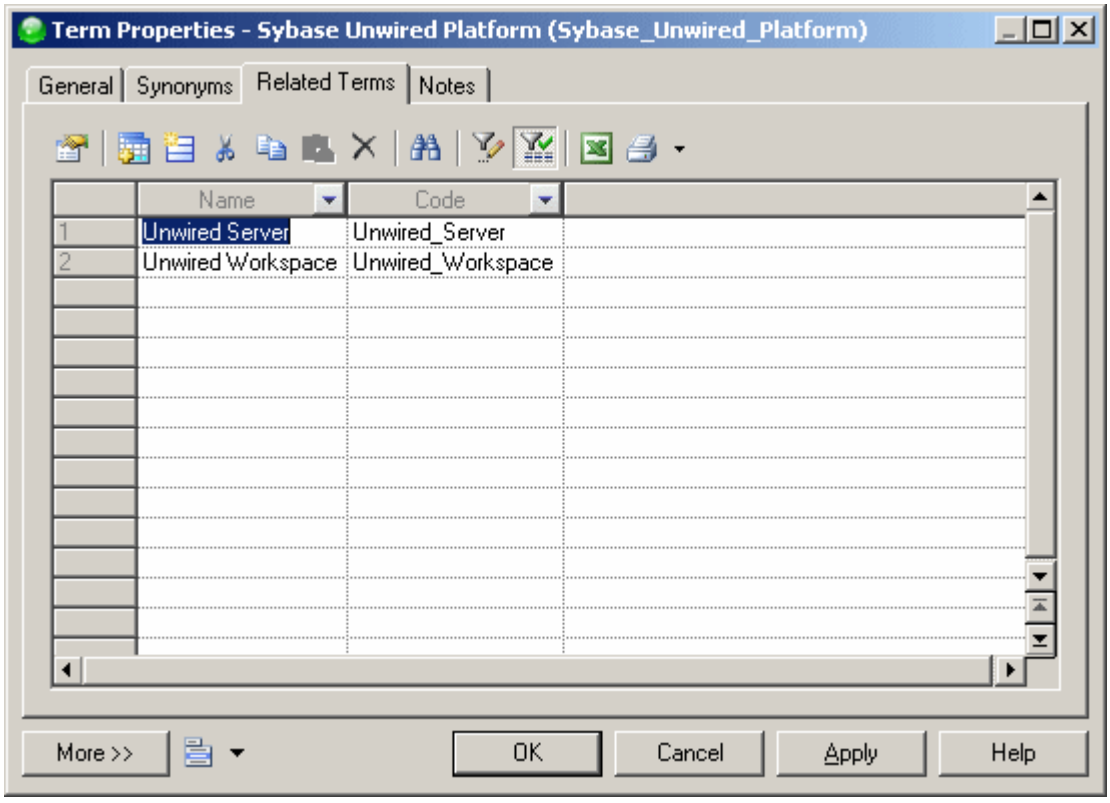
4. [optional] Click the *Synonyms* tab and enter one or more synonyms for the term.

> **i Note**
>
> Synonyms are words that have a similar meaning to the term, but are not approved for use in object names. For example, **Customer** may be specified as a synonym for the approved term **Client**. When naming an object, if a user begins to enter a synonym, the approved term is suggested in the autocompletion list. Modelers can use the `<Object>` *name contains synonyms of glossary terms* model check to find and replace synonyms in object names (see Checking Glossary Compliance [page 154]).



5. [optional] Click the *Related Terms* tab and use the *Add Objects* and *Create an Object* tools to associate other terms with the term.

> **i Note**
>
> Related terms are other glossary terms that are in some way associated with the present term. For example, **Bank** may be entered as a related term for the term **Current Account**.



6. Click *OK* to save your changes and add the term to the glossary.

3.2.1.1.1 Importing Terms from a Model

You can import the names of objects from any kind of PowerDesigner model as terms to quickly populate your glossary.

Context

Procedure

1. Right-click the *Glossary* node, and select *Import from Model* to open the Import Object Names as Glossary Terms Wizard.
2. Select a model open in the workspace from which to import object names, and then click *Next*.
3. Select the kind of object whose names you want to import as glossary terms and, optionally, a property of the object to specify the category of the imported terms, and then click *Next*.

4. Select the objects whose names you want to import as glossary terms, and then click *Finish*.

   The selected objects' names are imported as glossary terms. To import other objects' names as glossary terms, restart the wizard.

   > **i** Note
   >
   > Though a repository can only contain one glossary, in multi-repository environments or in other situations where you want to import terms from another glossary file, right-click the *Glossary* node, and select *Import from Another Glossary Model*. Navigate to the glossary file and click *Open* to open the Merge Models dialog (see Merging Models [page 369]), which will allow you to review and modify the proposed changes before clicking *OK* to complete the import.

3.2.1.1.2 Importing Terms from an Excel or CSV File

You can import terms from an Excel or CSV file to quickly populate your glossary.

Context

To simplify the import of terms from an Excel file, we recommend that you create a worksheet called **`Term`**, with the following format:

Name	Code	Comment	Reference URL	Synonyms	Related Terms	Category
Client	CLIENT	Specifies...	www.dictionary.com	Customer, User	Account, Invoice	Sales
Supplier	SUPPLIER	Specifies...	www.dictionary.com	Provider, Seller	Account, Invoice	Purchasing
etc	etc	etc	etc	etc	etc	etc

Using these exact column headings allows the Excel import to automatically recognize them as the appropriate properties. Only the *Name* column is required. You can enter multiple values in a column by separating them with commas.

Procedure

1. Right-click the *Glossary* node, and select *Import from Excel File* to open the Excel Import wizard.
2. Click the *Options* button to open the Import Options dialog, select *Auto-map columns to properties*, and click *OK* to return to the wizard.
3. Click the *Select File* button, navigate to and select the file you want to import, and click *OK* to return to the wizard.

4. Click *Next* to begin the wizard. If you have named your columns as suggested above, the wizard recognizes and imports the terms and all their properties without further input from you.

> **i** Note
>
> For detailed information about the PowerDesigner Excel Import Wizard, see Importing Objects from Excel Files [page 95].

3.2.1.2 Organizing Terms in Categories

You can organize terms in the glossary by placing them in categories. You can create categories under the root or under other categories, and drag and drop terms and categories to move them between categories.

To create a category, right-click the glossary root or a category and select *New* > *Category*, or use the tools on the *Categories* tab of a category property sheet.

To add terms to a category, drag and drop them onto the category node in the Browser or use the *Add* and *Create* tools on the *Terms* tab of the category property sheet.

> **⚠** Caution
>
> If you delete a category, you also delete any categories and terms that it contains.

3.2.1.3 Adding Other Views and Objects to, and Extending the Glossary

Although the Glossary's primary purpose is to manage enterprise terminology, you can also add business rules and create reports on glossary objects. You can create lists and dependency matrices to display the glossary terms and their relationships with other objects. You can also extend the glossary using all the standard extension mechanisms including adding new properties to glossary terms and creating new types of objects.

These features are available from the glossary contextual menu. Right-click the glossary and select *New* > :

- *Glossary Terms List* - to create a terms list in the canvas (see Glossary Terms Lists [page 546]).
- *Dependency Matrix* - to create a dependency matrix (see Dependency Matrices [page 199]).
- *Business Rule* - to create a business rule (see Business Rules [page 137]).
- *Report* - to create a report on the glossary (see Reports [page 295]).
- *List Report* - to create a list report on terms or other glossary objects (see Creating a List Report [page 303]).
- *Extension* - to add an extension file (see Extension Files [page 411]) to the glossary to allow you to extend glossary objects or define new extended objects. You can also add extended attributes to a glossary term, synonym, or related term directly from its property sheet using the standard mechanism (see Adding New Properties to an Object [page 168])

3.2.1.4 Glossary Terms Lists

You can create glossary terms lists to help users browse the glossary and to simplify glossary editing. You can create terms directly in a list and define custom filters for each list. To create a list, right-click the Glossary node and select *New* > *Glossary Terms List*.

The following tools are available on the glossary terms list toolbar:

Tools	Description
	Properties [Alt+Enter] - Opens the property sheet of the selected term.
	Add a Row [Ctrl+N] - Create a new term.
	Show Titles and Texts and *Show Titles Only* - Shows the titles and descriptions or only the titles of all terms in the view.
	Show Current Title and Text / *Show Current Title Only* - Toggle between showing the title and description or only the title of the selected term.
	Cut (Ctrl + X), *Copy* (Ctrl + C), and *Paste* (Ctrl + V) - Perform the standard clipboard actions.
	Delete [Ctrl+D] - Deletes the selected term.
	Customize Columns and Filter - Opens a dialog to change the columns displayed in the list or define a filter (see Customizing Object List Columns and Filtering Lists [page 123]).
	Enable/Disable Filter - Toggles the filter specified in the Customize Columns and Filter dialog.
	Find (Ctrl + F) - Opens a dialog to search for text.
	Format (Ctrl + M) - Opens a dialog to specify fonts and styles.
	Bold, *Italic*, and *Underline* - Format the selected text.
	Align Left, *Align Center*, and *Align Right* - Align the selected text.
	Bullets - Indent and bullet the selected text.
	New Glossary Terms List - Creates a new list.
	Open Glossary Terms List - Opens a dialog to select another terms list to view.
	Export to Excel - Saves the list to a *.xls, *.xlsx, or *.csv format (specify the format in the *Save as type* field in the Save as dialog).

> **Note**
>
> Right-click the glossary node in the Browser and select *Model Options* to set default formats for the term names and descriptions in your glossary terms lists (see Glossary Model Options [page 548]).

3.2.1.5 Checking the Glossary

To perform a model check on the glossary, right-click the glossary root and select *Check Model*. PowerDesigner provides default model checks to verify the validity of terms, synonyms, and categories.

For detailed information about working with model checks, see Checking Models [page 148]. By default, these checks are provided for glossary objects:

Check	Description and Correction
Name and code uniqueness	Term, synonym, and category names and codes must be unique in the glossary. • Manual correction: Modify the duplicate name/code. • Automatic correction: Appends a number to the duplicate name/code.
Missing term definition	Terms should be defined. • Manual correction: Enter a definition in the Description field. • Automatic correction: None.

3.2.1.6 Deploying the Glossary

Once you have created your glossary and added your terms, check it into the repository to automatically deploy it to all users connecting to the repository.

Procedure

1. Right-click the Glossary root, and select *Check In* to open the Check In Document dialog (see Checking Documents into the Repository [page 247]).
2. Enter an appropriate comment and click *OK* to complete the check in.

   The glossary is now deployed to the repository and will be pushed to all users the next time they connect.

   > **Note**
   >
   > You can continue to make modifications to your local glossary model, and each time that you check it in, your changes are pushed to users the next time they connect to the repository.

3.2.1.7 Glossary Model Properties

The model property sheet displays information about the glossary model.

To open the glossary property sheet, double-click its entry in the Browser *Glossary* tab. The *General* tab contains the following properties:

Property	Description
Name / Code / Comment	Identify the model. The name should clearly convey the model's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the model. By default the code is auto-generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Filename	Specifies the location of the glossary model file.
Conjunctions	Specifies a comma-separated list of small words that are excluded from glossary compliance checks. For example: `and, of, to, the`.
Name / Code separators	Specify lists of characters that indicate the end of a word in the name and code fields to provide bounding of words for the compliance checking and auto-completion. By default, the characters `.` (space and dot) are set as separators for the name field and `_` (underscore) for the code field.
Author	Specifies the author of the model. If you enter nothing, the *Author* field in diagram title boxes displays the user name from the model property sheet *Version Info* tab. If you enter a space, the *Author* field displays nothing.
Version / Repository	Specify the version of the model. You can enter any value in the first field, while the second field displays the repository version.

3.2.1.8 Glossary Model Options

You can set glossary model options by right-clicking the glossary node and selecting *Model Options*.

You can set the following options on the *Model Settings* page:

Option	Description
Name/Code case sensitive	Specifies that the names and codes for all objects are case sensitive, allowing you to have two objects with identical names or codes but different cases in the same model. If you change case sensitivity during the design process, we recommend that you check your model to verify that your model does not contain any duplicate objects.

Option	Description
External Shortcut Properties	Specifies the properties that are stored for external shortcuts to objects in other models for display in property sheets and on symbols. By default, *All* properties appear, but you can select to display only *Name/Code* to reduce the size of your model. **i Note** This option only controls properties of external shortcuts to models of the same type (PDM to PDM, EAM to EAM, etc). External shortcuts to objects in other types of model can show only the basic shortcut properties.

Click the *Terms Fonts* page to control the fonts and formatting used for the various types of text in your glossary terms lists (see Glossary Terms Lists [page 546]).

3.2.2 Deploying an Enterprise Library

The library is a special folder at the root of the repository, which can be used to push models, resource files, model and report templates and external documents to all users. An administrator controls access to the library using standard permissions. The library folder appears at the top of the Browser *Local* tab and updates are pushed to users each time they connect to the repository. Library models are for reference or to provide objects for reuse while resource files are made available for selection wherever appropriate in the PowerDesigner interface.

For information about using the library, see The Library [page 42]

3.2.2.1 Sharing Models and External Documents via the Library

An administrator or other user with **`Write`** permission on the **`Library`** folder or its subfolders can check models, model templates, resource files, and other files into the library. Users with **`Submit`** permission can propose changes to the library. Once one or more documents are added to the library, users connecting to the repository are invited to check them out into the library folder in their local workspace.

Context

i Note

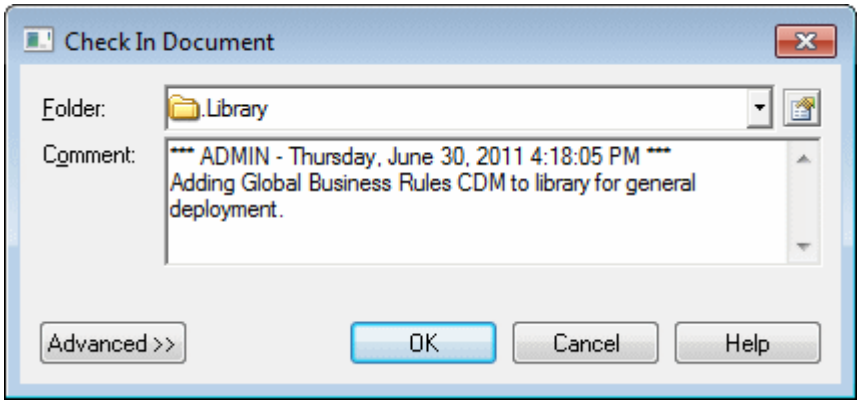
For information about managing permissions, see Granting Access Permissions on Repository Items [page 534].

Procedure

1. Select a model or other type of file (such as a Word document or PDF) that you want to deploy to all users, and connect to the repository.

   > **i** Note
   >
   > For information about checking resource files and model templates into the library, see Sharing Resource Files via the Library [page 551] and Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552].

2. Right-click the document in the *Local* tab of the Browser and select *Check In*.
3. Select the **`Library`** folder (or a subfolder beneath it) as the destination for your check in and enter an appropriate comment:



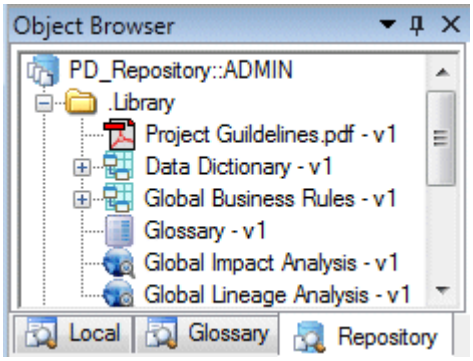
4. Select whether to check into a change list (see Change Lists [page 259]). You can choose from:
   - **`<None>`** - [if you have **`Write`** permission on the target folder] Allows you to check the documents in directly without review.
   - **`<New change list>`** - To create a new change list to contain the checked in documents.
   - **`Change list_<xxxx>`** - To add the documents to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list.

     > **i** Note
     >
     > If you are checking in a new version of a document that is already contained in a draft change list, then this field is read-only.

5. Enter a comment to explain the purpose of your check in, select any other appropriate parameters (see Check-In Parameters [page 252]), and click *OK* to check in the document.

   If you check into a change list, you can add other documents to it as necessary through other check ins before submitting it (see Creating, Building, and Submitting a Change List [page 260]). Once the change list has been approved by the reviewers, your changes are integrated into the official repository versions of the documents and made available to other users. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) you will receive an email at the moment of submission of your change list and when it is approved, rejected, or returned to you for further revision.

The document is checked into the library in the repository:



The next time a user connects to the repository (or selects *Repository* > *Synchronize Library*, they will be invited to check the model out into the library in their local workspace (see Connecting to a Repository [page 19]).

> **i Note**
>
> Branching (see Branching Version Trees [page 281]) is not supported for the library.

3.2.2.2 Sharing Resource Files via the Library

Users with **`Submit`** permission or higher can check DBMS definition files, extension files and other PowerDesigner resource files into the library. Resource files are pushed to repository users in the same way as other library documents, and appear automatically in the new model dialog, resource file lists, and anywhere in the interface where a resource file can be selected.

Procedure

1. Expand the `Library/Resources` folder in the *Repository* browser tab to view the available sub-folders:
   - Extensions
   - Model Languages - for DBMS and process, object, and XML language definition files
   - Report Languages

> **i Note**
>
> PowerDesigner recursively scans all folders in the library to populate resource file lists with shared resources, but we recommend that you group your files inside the `Resources` subfolder. All resource files, including those without specific sub-folders are supported, and you can create additional sub-folders (and sub-sub-folders) as appropriate. For information about templates and conversion tables, which must be saved to specific sub-folders, see Sharing Model, Report, and Project Templates and Conversion Tables via the Library [page 552]).

2. Select *Tools* > *Resources* > `<type>` to open the list of resource files containing the file that you want to share (see PowerDesigner Resource Files [page 408]).
3. Select the resource file to check in from the list and click the *Check In* tool.
4. Select the folder inside the library into which you want to check in the file.

   > **i Note**
   >
   > If you check the resource into a folder outside the library and it is referenced by any models, PowerDesigner will try to resolve the reference when these models are checked out, but the resource will not be automatically synchronized in users' library folders and so will not be available for selection in lists of resource files.

5. Select whether to check into a change list (see Change Lists [page 259]). You can choose from:
   - **`<None>`** - [if you have **`Write`** permission on the target folder] Allows you to check the documents in directly without review.
   - **`<New change list>`** - To create a new change list to contain the checked in documents.
   - **`Change list_<xxxx>`** - To add the documents to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list.

     > **i Note**
     >
     > If you are checking in a new version of a document that is already contained in a draft change list, then this field is read-only.

6. Enter a comment to explain the purpose of your check in, select any other appropriate parameters (see Check-In Parameters [page 252]), and click *OK* to check in the document.

   If you check into a change list, you can add other documents to it as necessary through other check ins before submitting it (see Creating, Building, and Submitting a Change List [page 260]). Once the change list has been approved by the reviewers, your changes are integrated into the official repository versions of the documents and made available to other users. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) you will receive an email at the moment of submission of your change list and when it is approved, rejected, or returned to you for further revision.

   The next time users connect to the repository (or select *Repository* > *Synchronize Library*, they will be invited to check the resource out into their local library, and it will immediately be available for use wherever a resource file can be selected.

   > **i Note**
   >
   > The Resources folder and its sub-folders are only displayed inside the library folder in the *Local* browser tab if you have at least **`Submit`** permission on them.

3.2.2.3 Sharing Model, Report, and Project Templates and Conversion Tables via the Library

Users with **`Submit`** permission or higher can check model, report, and project templates and conversion tables into special sub-folders in the library. These files are pushed to repository users in the same way as other

library documents, and appear automatically for selection anywhere in the interface where a template or conversion table can be selected.

Procedure

1. Expand the `Library/Resources` folder in the *Repository* browser tab to display the following sub-folders:
   - Model Templates (see Model Templates [page 14])
   - Report Templates (see Report Templates [page 357])
   - Project Templates (see Project Templates [page 81])
   - Conversion Tables (see Conversion Tables [page 161])
2. Right-click the appropriate folder and select *Add*. Navigate to the file you want to check into the library and click *Open*.

   > **Note**
   >
   > You must check these files into the correct folder or they will not be available for selection in PowerDesigner dialog boxes.

3. Select whether to check into a change list (see Change Lists [page 259]). You can choose from:
   - **<None>** - [if you have **Write** permission on the target folder] Allows you to check the documents in directly without review.
   - **<New change list>** - To create a new change list to contain the checked in documents.
   - **Change list_<xxxx>** - To add the documents to an existing draft change list. Click the *Properties* tool to the right of this field to access the property sheet of the change list.

     > **Note**
     >
     > If you are checking in a new version of a document that is already contained in a draft change list, then this field is read-only.

4. Enter a comment to explain the purpose of your check in, select any other appropriate parameters (see Check-In Parameters [page 252]), and click *OK* to check in the document.

   If you check into a change list, you can add other documents to it as necessary through other check ins before submitting it (see Creating, Building, and Submitting a Change List [page 260]). Once the change list has been approved by the reviewers, your changes are integrated into the official repository versions of the documents and made available to other users. If an administrator has connected the repository to an SMTP server (see Connecting to an SMTP Server for Notifications [page 517]) you will receive an email at the moment of submission of your change list and when it is approved, rejected, or returned to you for further revision.

   The next time users connect to the repository (or select *Repository* > *Synchronize Library*, they will be invited to check the file out into their local library, and it will immediately be available for use wherever a template or conversion table can be selected.

   > **Note**
   >
   > The Resources folder and its sub-folders are only displayed inside the library folder in the *Local* browser tab if you have at least **Submit** permission on them.

3.2.3 Controlling Library and Glossary Write Permissions

Since library models and other documents are intended to be a reference for users, many administrators will want to closely control who can edit them. You control access to the Library folder with standard repository permissions. By default, all users can see the Library folder and have updates pushed to them as necessary, but only administrators have the right to modify the glossary and other library models.

Context

The following are common uses of permissions on the Library folder and individual models within it:

- To allow a restricted group of users to modify library models - Create a **`Compliance Committee`** or **`Library Administrators`** group with **`Write`** permission on the Library folder. Sometimes a separate **`Terminology Committee`** or **`Glossary Administrators`** group is created and **`Write`** permission on the glossary model is restricted to this subset of users.
- To allow other users to propose changes to library models via change lists - Grant them **`Submit`** permission on the Library folder. Changes proposed are only applied to the library models if they are approved by your administrators (see Change Lists [page 259]).
- To hide library models while they are being developed - Remove the **`All Users`** group from the permissions list of individual models until they are ready for deployment.

For detailed information about setting repository permissions, see Granting Access Permissions on Repository Items [page 534].

3.3 Customizing the PowerDesigner Interface

You can customize the New Model dialog to guide model creation through templates and use profiles to hide types of models, objects, and object properties, and to customize menus, commands, and preferences.

3.3.1 Customizing the New Model Dialog

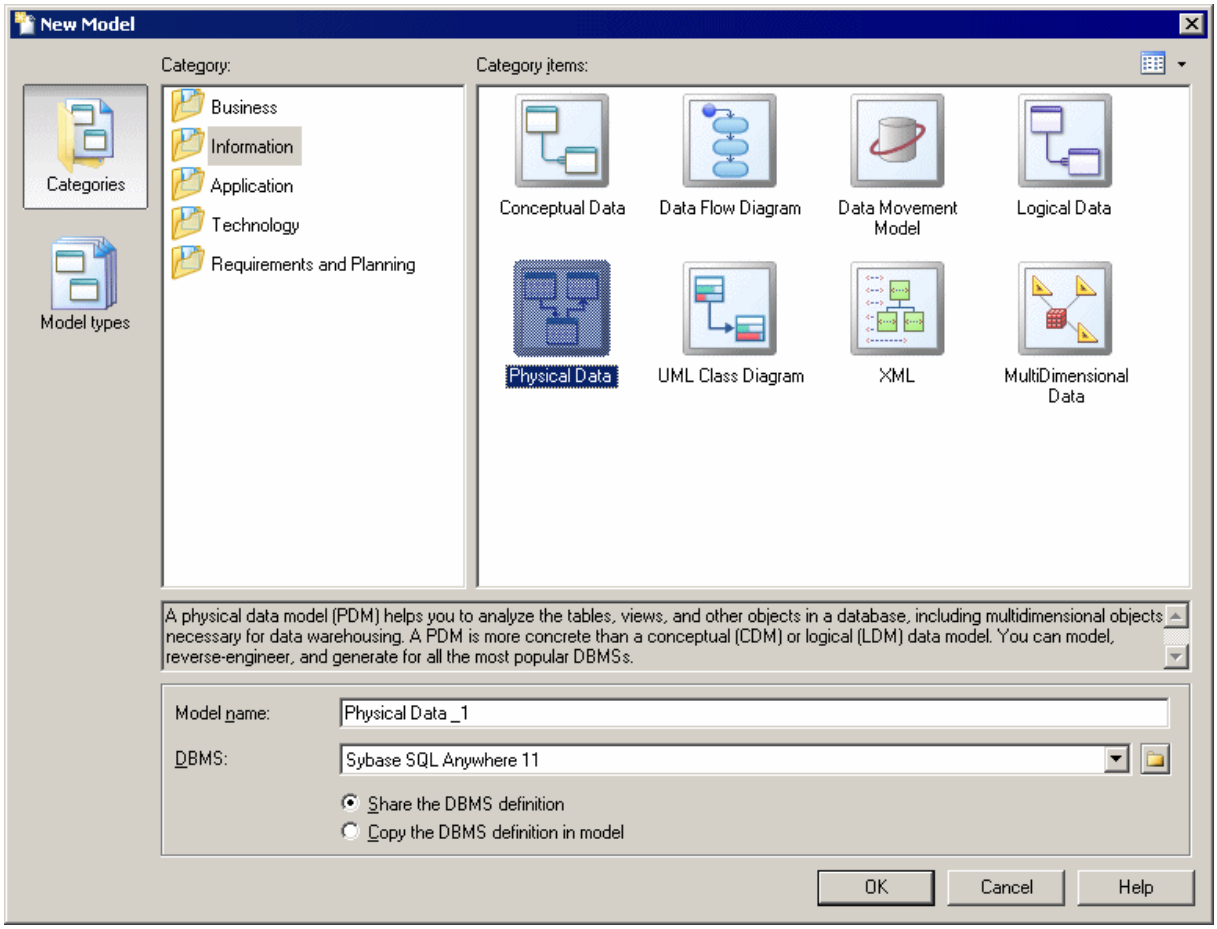
A model category set is a PowerDesigner resource file that contains one or more categories, which in turn contain templates to guide users through model creation in the New Model dialog. Using categories and templates for model creation simplifies the model creation process by preselecting appropriate targets, extensions, and diagrams, and by hiding irrelevant choices from users.

To view the list of model category sets, select *Tools* > *Resources* > *Model Category Sets*. For information about the tools available in resource file lists, see PowerDesigner Resource Files [page 408].

You can deploy model category sets to users (see Sharing Resource Files via the Library [page 551]) to ensure that everyone on your team is creating models using the same centralized targets and extensions checked out automatically from the repository. You can create different category sets for different users so that, for

example, data architects will have different choices from business analysts when they open the New Model dialog.

In the following example, the Information category is selected and provides eight templates to choose from:



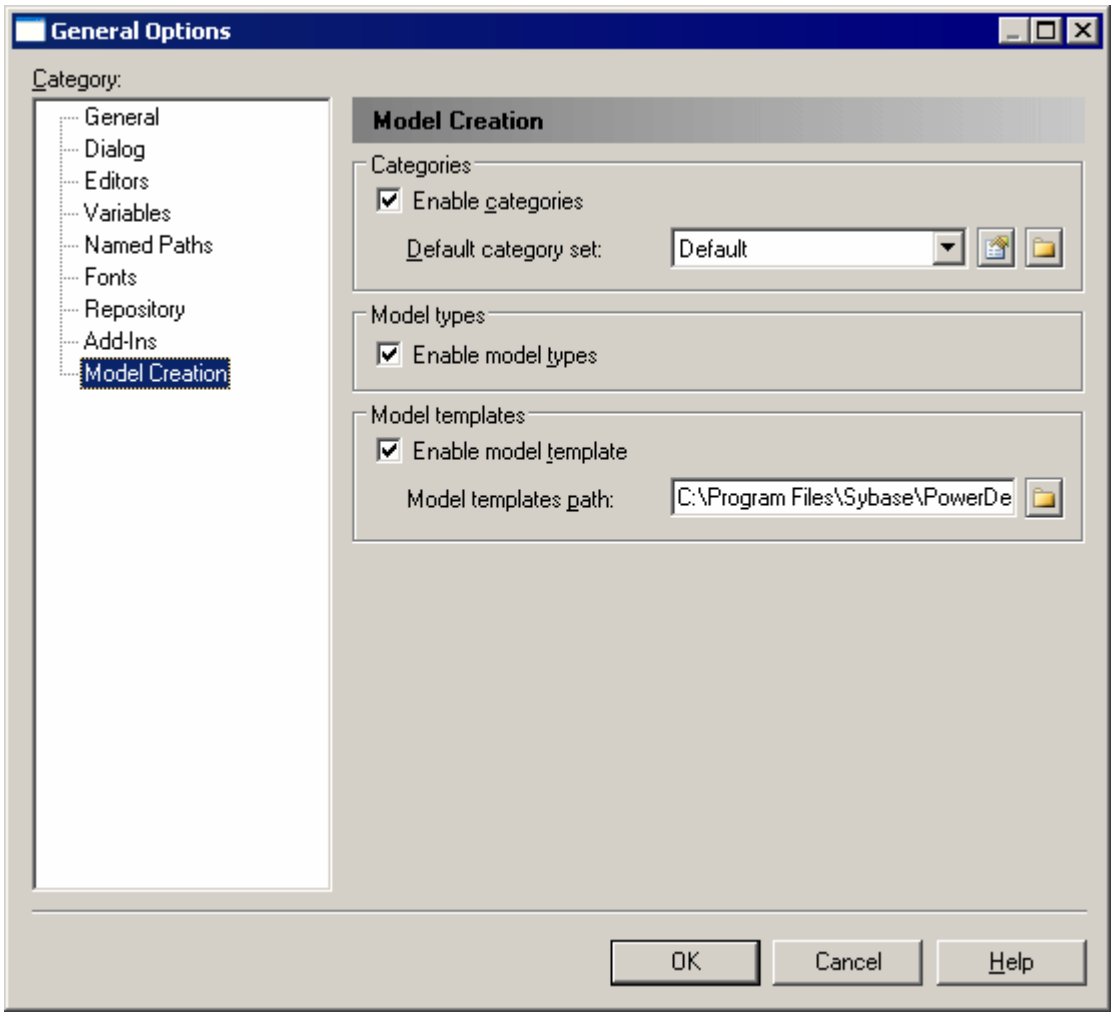
> **i Note**
>
> You can also hide model types from certain types of users with object permission profiles (see Controlling the Availability of Models, Objects, and Properties [page 562]).

3.3.1.1 Selecting a Model Category Set to Display in the New Model Dialog

The model category set you select in the Model Creation general options will be displayed in the New Model window.

Procedure

1. Select *Tools* > *General Options*, and click the *Model Creation* category in the left hand pane.



2. Verify that the *Enable categories* check box is selected, and select the category set you want to display from the Default category set list.
3. [optional] Click the *Path* tool to the right of this field to browse to another folder containing model category sets or the *Properties* tool to open the currently-selected set in the Resource Editor.
4. Click *OK* to close the General Options dialog.

The selected category set and the templates it contains will be displayed in the New Model dialog next time you open it.

3.3.1.2 Creating a Model Category Set

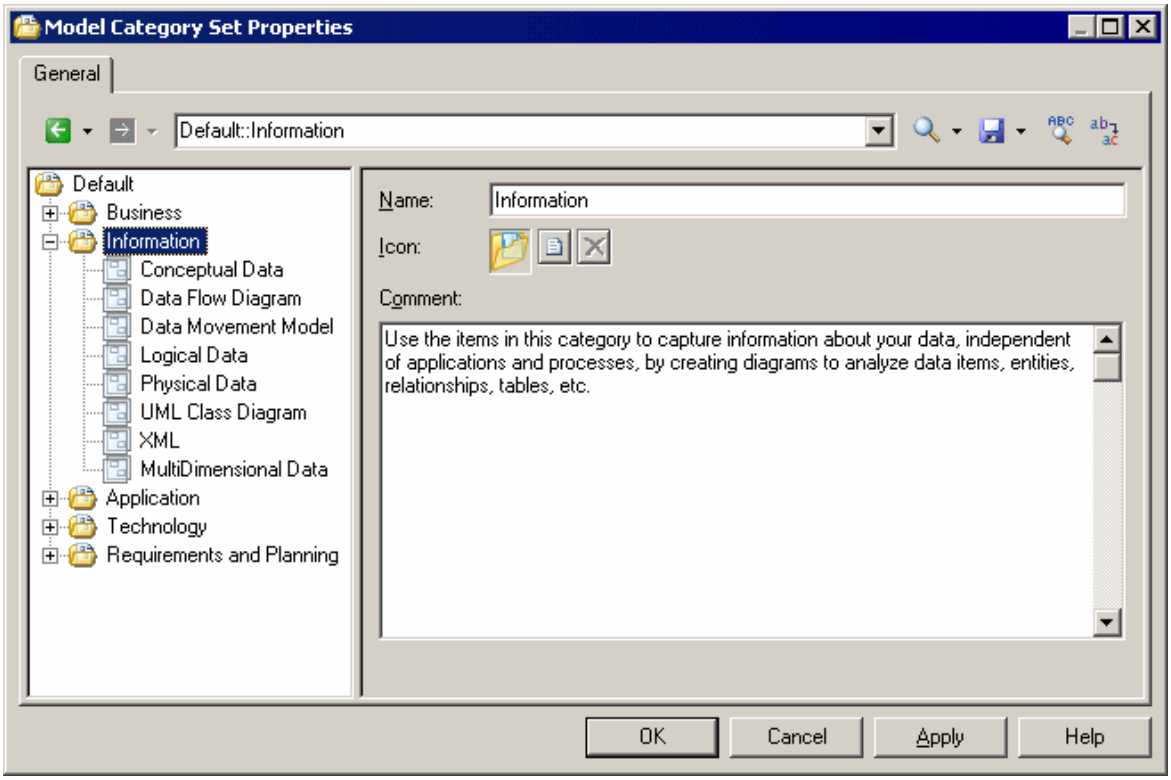
You can create as many different model category sets as you need, creating different sets for different types of users, for example.

Procedure

1. Select *Tools* > *Resources* > *Model Category Sets* to open the List of Model Category Sets.
2. Click the *New* tool to open the New Model Category Set dialog.
3. Enter a name for your model category set, and select a source to copy it from. You can choose between:
   - <Default template> – creates an empty set that you complete by adding categories and templates.
   - Another model category set – makes a copy of an existing set that you can edit as necessary.
4. Specify a name and a location to create the set and click *Save*.

   The model category set is created and opened for review in the Resource Editor.
5. For each category that you need, right-click the root node, and select *New*. In the right-hand pane, enter a name for the category and, optionally, click the *Browse for icon* tool to change the default icon.

   > **i** Note
   >
   > You can reorder the categories if necessary by drag and drop.
6. When you have created your categories, you add templates to them by right-clicking the category and selecting *New* and then one of the following:
   - Model - creates a simple model template that can specify a default target, extensions, and an empty first diagram (see Adding a Model Template to a Model Category Set [page 558]).
   - Model from Template - which points to an actual model file that, in addition to specifying a target and extensions, can contain multiple diagrams, display preferences, general and model options, etc (see Adding a Model from Template to a Model Category Set [page 559])
7. When you have finished creating categories and templates, click OK to close the resource editor and click the Save tool in the List of Model Category Sets to save your edits.



You can test your category set by selecting it in the General Options dialog (see Selecting a Model Category Set to Display in the New Model Dialog [page 556]) and selecting *File* > *New Model*.

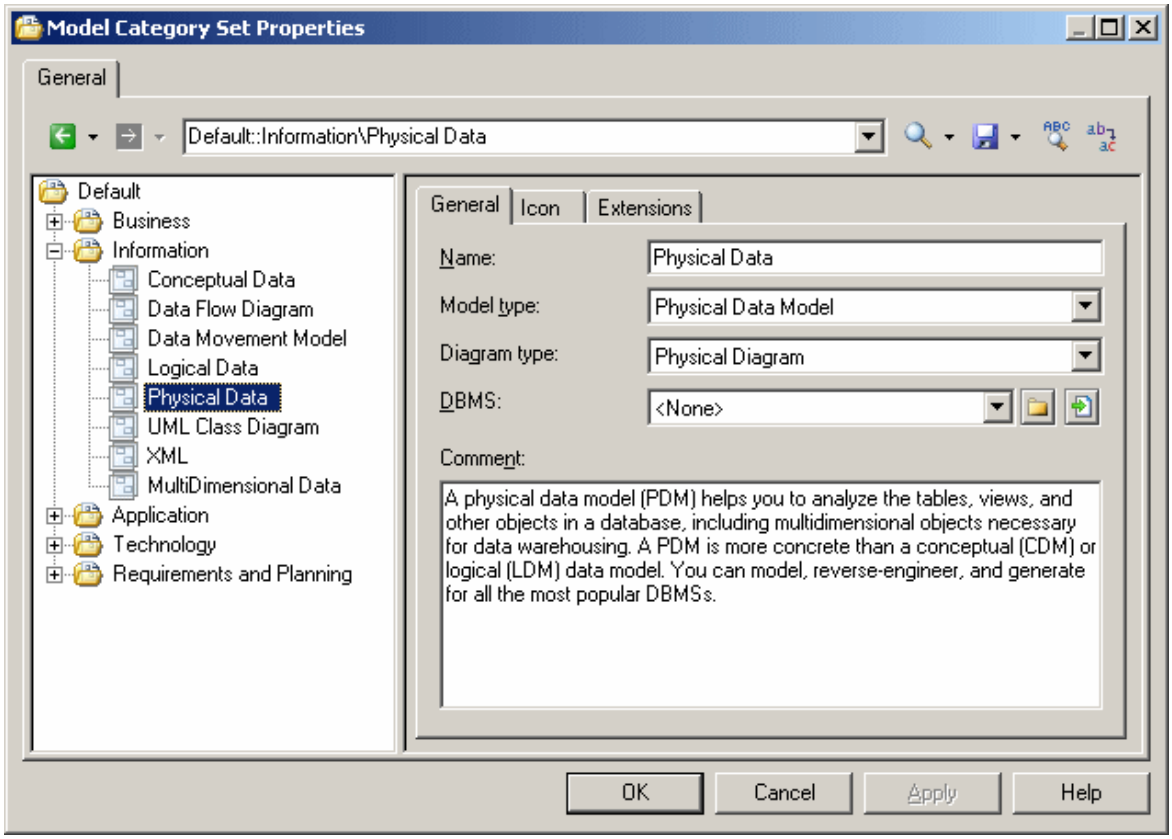
3.3.1.2.1 Adding a Model Template to a Model Category Set

Model templates are simple templates that can specify a default target, extensions, and an empty first diagram.

Procedure

1. Right-click the category to which you want to add the template, and select *New* > *Model*.
2. On the *General* tab, enter a template name and select a model and diagram type.
3. [optional, for model types with targets] Specify a target (DBMS, object or process language, etc). If you select None, then the user can select any available target language in the New Model dialog.
4. [optional] On the *Icon* tab, click the *Browse for icon* tool to change the default template icon.
5. [optional] On the *Extensions* tab, click the *Select Extensions* tool to attach one or more extensions to the model (see Attaching Extensions At Model Creation Time [page 13]).
6. [optional] Select the *Allow additional extensions at model creation* check box if you want to allow users to attach additional extensions to those you have specified. The *Extensions* button in the New Model dialog is not available if you do not select this check box.

In the following example, the Physical Data template is specified to create a Physical Data Model with a physical diagram, and since no DBMS is specified, the user will be able to choose one at model creation time:



3.3.1.2.2 Adding a Model from Template to a Model Category Set

Models from templates are templates that point to existing model files that, in addition to specifying a target and extensions, can contain multiple diagrams, display preferences, general and model options, etc.

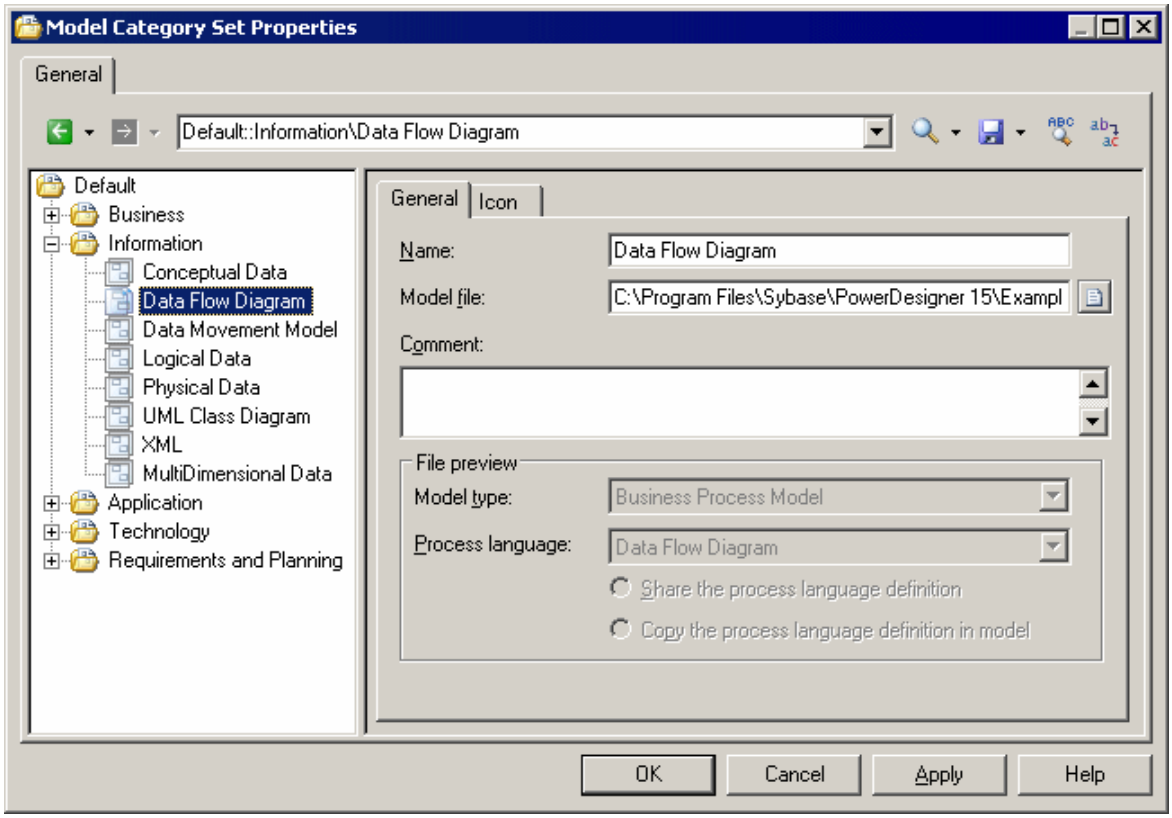
Context

In order to add a model from template to a category, you must have a model file that contains all the appropriate parameters ready to point to.

Procedure

1. Right-click the category to which you want to add the template, and select *New* > *Model from Template*.
2. On the *General* tab, enter a template name and click the *Select File* tool to the right of the Model file field to browse to and select the model file to use as the template.
3. [optional] On the Icon tab, click the *Browse for icon* tool to change the default template icon.

   In the following example, the Data Flow Diagram template is specified to create a Business Process Model from an existing dfd.bpm model, and will inherit its pre-configured options, preferences, extensions, and objects:



3.3.1.3 Pushing Model Category Sets to Users

You can deploy your model category sets to the library to make them available to other users.

Procedure

1. Select *Tools* > *Resources* > *Model Category Sets* to open the List of Model Category Sets.
2. Select the category set you want to push to other users and click the *Check In* tool.
3. Select the folder inside the library into which you want to check in the category set, and enter an appropriate comment.

   > **i** Note
   >
   > If you check the category set into a folder outside the library, it will not be available for selection in the General Options dialog. For more information about the library, see Deploying an Enterprise Library [page 549].

4. Click *OK* to check in the category set.

   The next time users connect to the repository (or select *Repository* > *Synchronize Library*, they will be invited to check the category set out into their local library, and it will immediately be available for selection when they click *Tools* > *General Options*, and click the *Model Creation* category in the left pane.

3.3.2 Using Profiles to Control the PowerDesigner Interface

An administrator can create profiles in the repository to customize and simplify the PowerDesigner interface for users by hiding types of models, objects, and object properties, by reconfiguring menus, toolbars, and toolboxes, and by setting appropriate defaults for options and preferences. You can define multiple profiles to customize the interface in different ways for different kinds of users and apply them as appropriate to individual user accounts or to repository groups.

Three types of profiles are available to administrators to customize the PowerDesigner interface for users:

- Object permissions profiles - to hide types of models or make them read-only, and to hide model objects and object properties. Use these profiles to simplify the modeling environment by presenting users with only the concepts that they need:
  - Hiding a module - prevents the creation or opening of models of this type.
  - Hiding an object - removes it from the Toolbox, Browser, *Model* menu, *New* contextual menu, and property sheets.
  - Hiding an attribute or collection - removes it from object symbols and property sheets.
- UI preferences profiles - to customize the commands available in menus, toolbars, and toolboxes. Use these profiles to reorganize and simplify the PowerDesigner interface for users. They provide the same kind of customization as users can perform using the *Tools* > *Customize Menus and Tools* command (see Customizing Menus, Toolbars, and Toolboxes [page 405]).

- General preferences profiles - to set default display preferences and model and general options. Use these profiles to guide users toward making appropriate option and preference choices. They provide the same kind of customization as users can perform using user profiles (see User Profiles [page 381]).

3.3.2.1 Controlling the Availability of Models, Objects, and Properties

An administrator can define object permissions profiles to hide models, objects, and properties, or make models available in read-only form for users or group of users. By default, all types of models, objects, and properties are available and you hide them (or make them read-only) by adding entries to an object permissions profile.

Procedure

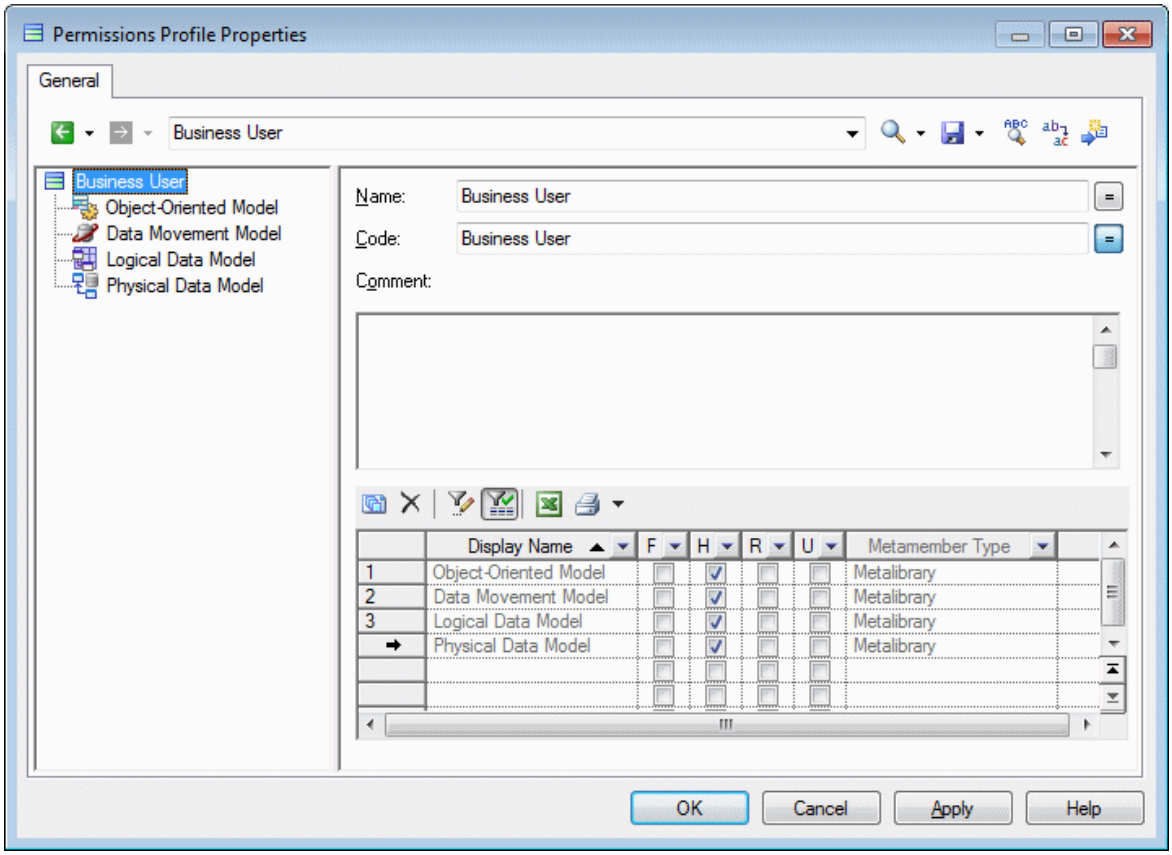
1. Connect to the repository and then select *Repository* > *Administration* > *Object Permission Profiles*.
2. Click the *Create an Object* tool, enter a name for the profile, select a template to start from, and click *OK* to create and open it in the Resource Editor.
3. [optional] Click the *Import Items* tool to import an existing profile, review the items to be imported in the Merge dialog, and then click *OK* to complete the import.

   For detailed information about working with this dialog, see Merging Models [page 369]. You can repeat this step as many times as necessary to import additional profiles.

   > **i Note**
   >
   > PowerDesigner provides a number of example profiles, which you can combine as necessary and edit to support your particular workflow (see Example Permission Profiles [page 565]).

4. To modify the status of model types (modules) for users associated with this profile, right-click the root element and select *Add Modules to Customize* (or click the equivalent tool in the toolbar), select the modules and click *OK* to add them to the profile. Then select the appropriate status:
   - *Undefined* - [default] The module's availability is not specified in the profile. This status is useful when the profile will be combined with others, and these other profiles will determine the availability of the module. The module can be customized by hiding objects, object attributes, and collections.
   - *Enable* - The module is available to users associated with the profile, and can be customized by hiding objects, attributes, and collections.
   - *Disable (Hide)* - The module and all models of this type are hidden to users associated with the profile. No further customization is possible.
   - *Read-Only* - The module is available to users associated with the profile, but they may not edit models of this type. No further customization is possible.

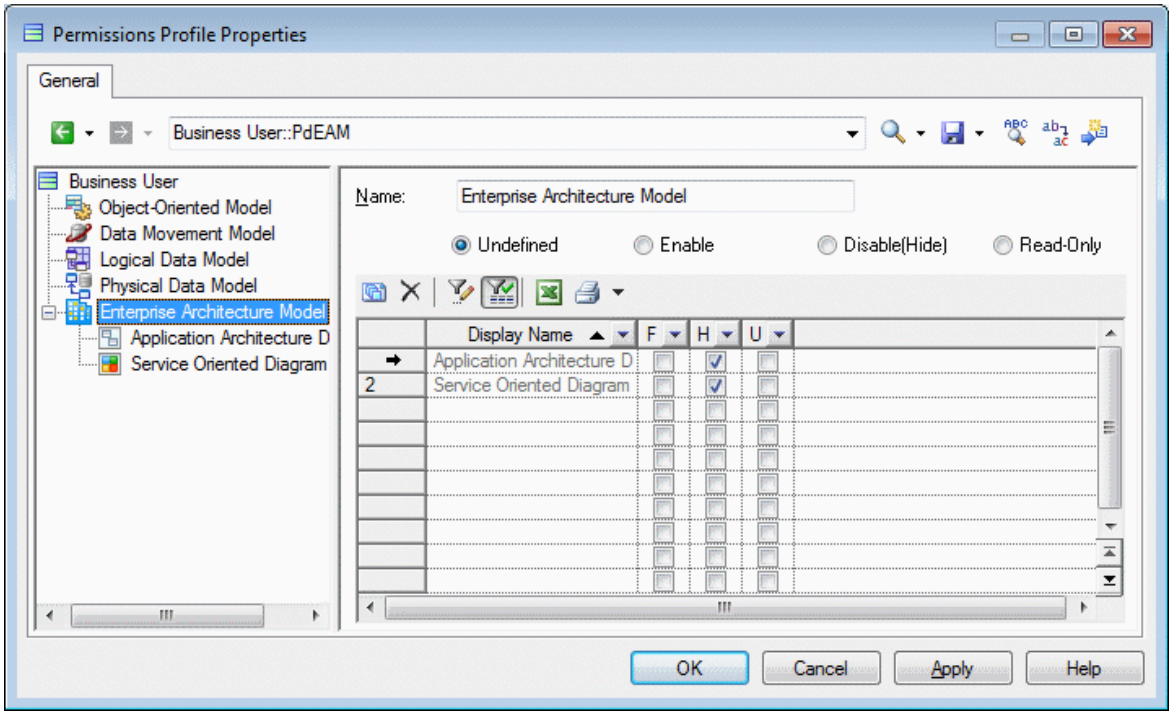
   In this example, the OOM, DMM, LDM, and PDM are added to the profile to hide them from business users:



5. To modify the status of object types (metaclasses) for users associated with this profile, right-click the parent module, select *Add Metaclasses to Customize* (or click the equivalent tool in the toolbar), select the metaclasses and click *OK* to add them to the profile. Then select the appropriate status:
   - *Undefined* - [default] The metaclass's availability is not specified in the profile. This status is useful when the profile will be combined with others, and these other profiles will determine the availability of the metaclass. The metaclass can be customized by hiding attributes and collections.
   - *Enable* - The metaclass is available to users associated with the profile, and can be customized by hiding attributes and collections.
   - *Disable (Hide)* - The metaclass and all objects of this type are hidden to users associated with the profile. No further customization is possible.

> **i Note**
>
> To add a metaclass defined in a DBMS, language definition, or extension file to a permission profile, enter it by hand in the format **`<metaclass>::<stereotype>`**. For example, to hide SAP IQ events, enter **`ExtendedObject::Event`**, and select the checkbox in the *H* column.

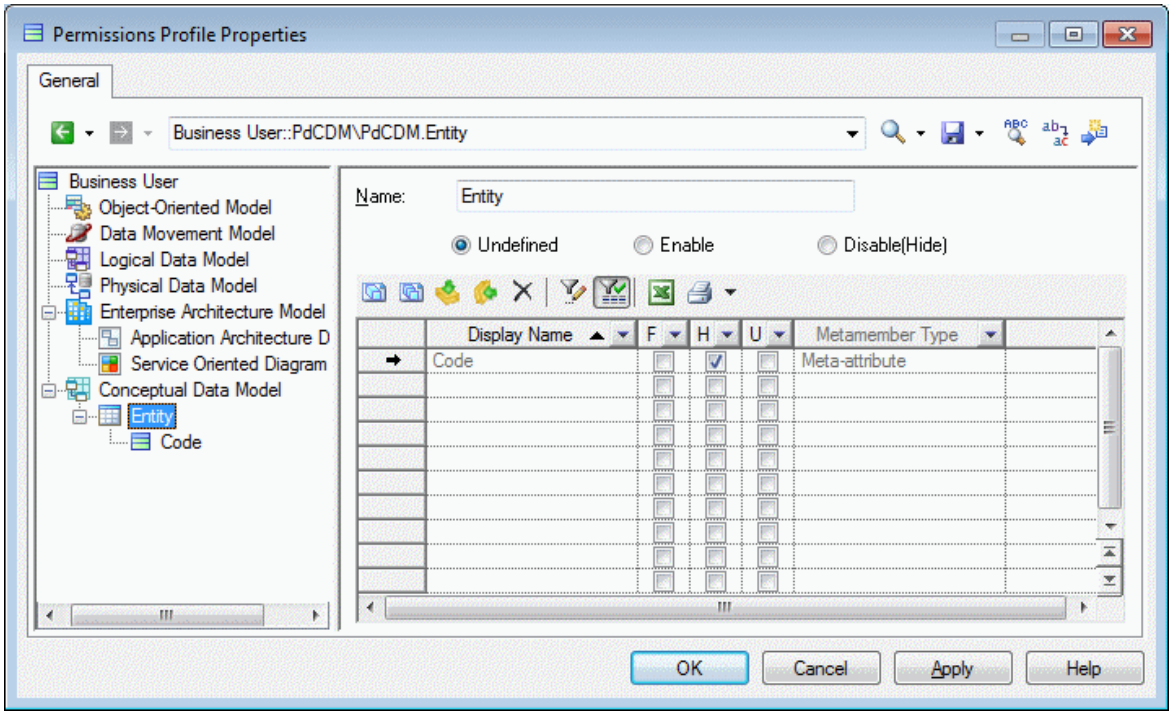
In this example, the Application Architecture Diagram and Service Oriented Diagram metaclasses are added to the profile to hide them from business users:



6. To modify the status of object properties (attributes and collections) for users associated with this profile, right-click the parent metaclass and select *Add Meta-attributes to Customize* or *Add Metacollections to Customize* (or click the equivalent tool in the toolbar), select the attributes or collections and click *OK* to add them to the profile. Then select the appropriate status:
   - *Undefined* - The attribute or collection's availability is not specified in the profile.
   - *Enabled* - The attribute or collection is available to users associated with the profile.
   - *Disabled (Hidden)* [default] - The attribute or collection is hidden to users associated with the profile.

> **i Note**
>
> To add an extended attribute defined in a DBMS, language definition, or extension file to a permission profile, enter it by hand. For example, to hide the **DBSpace** attribute for SAP IQ tables, enter **DBSpace**, and select the checkbox in the *H* column.

In this example, the entity Code attribute is added to the profile to hide it from business users:



> **i Note**
>
> You can propagate customizations of properties to other metaclasses, including to metaclasses in other types of models. Select a metaclass with customized properties and click the *Apply to All Metaclasses in This Module* or *Apply to All Metaclasses in All Modules* tool or right-click a meta-attribute or metacollection and select:
>
> - *Apply to All Metaclasses in This Module* - to propagate its customization to all other metaclasses that include this property in the present module. PowerDesigner will add the minimum number of metaclasses necessary to apply the customization by exploiting inheritance in the metamodel. This often involves adding the Named Object and a handful of other metaclasses.
> - *Apply to All Metaclasses in All Modules* - to propagate its customization to all other metaclasses that include this property in all the modules that are included in the profile. PowerDesigner will not add additional modules to the profile, so if you want to customize the property everywhere it appears, you must first add all relevant modules.

7. Click *OK* to save the profile in the repository.

3.3.2.1.1 Example Permission Profiles

PowerDesigner provides a set of permission profiles that you can merge into your own profiles, combining, editing, or adding to them as appropriate for your team's working practices.

There are four kinds of example profiles available in the `<install_dir>/Resource Files/Profiles` directory:

- Hide Objects - Remove one or more types of objects from the Toolbox, Browser, *Model* menu, *New* contextual menu, and property sheets:

    - Hide Objects - Business Rules
    - Hide Objects - EAM Application Objects
    - Hide Objects - EAM Business Objects
    - Hide Objects - EAM Technology Objects
    - Hide Objects - Traceability Links
- Hide Properties - remove attributes and collections from object symbols and property sheets:
    - Hide Properties - Codes for Non-Data Objects
    - Hide Properties - Keywords
    - Hide Properties - Mappings
    - Hide Properties - Related Diagrams
    - Hide Properties - Stereotype
    - Hide Properties - Technical Properties
- Lock - Prevent the user from changing special model properties:
    - Lock - Extensions (removes extension files from the Browser, *Model* menu, and *New* contextual menu to prevent users changing extensions)
    - Lock - Model Options (removes the *Model Options* command from the *Tools* menu to prevent users changing model options)
- Role - Control the availability of modules for various types of users:
    - Role - Developer (hides the BPM and EAM, and makes the CDM, LDM, and RQM read-only)
    - Role - Enterprise Architect (hides or makes read-only all models except the EAM)
    - Role - Information Architect (hides the RQM, and makes the BPM, EAM, and OOM read-only)
    - Role - Process Architect (hides or makes read-only all models except the BPM)

3.3.2.2 Controlling Defaults for Menus, Toolbars, and Toolboxes

An administrator can define UI preferences profiles to control the default configuration of menus, toolbars, and toolboxes for users or group of users.

Procedure

1. Connect to the repository and then select *Repository* > *Administration* > *UI Preference Profiles*.
2. Click the *Create an Object* tool, enter a name for the profile, select a template to start from, and click *OK* to create and open it in the Customize Menus and Tools dialog.
3. Select and modify the menus, toolbars, and toolboxes for the appropriate views using the various panels in the dialog (see Customizing Menus, Toolbars, and Toolboxes [page 405]).
4. Click *OK* to save the profile in the repository.

3.3.2.3 Controlling Defaults for Options and Preferences

An administrator can define general preferences profiles to control the default values for display preferences, model options, general options, and other preferences for users or group of users.

Procedure

1. Connect to the repository and then select *Repository* > *Administration* > *General Preference Profiles*.

   These profiles are the same as those that individual users can use to customize their options and preferences (see User Profiles [page 381]).
2. Click the *Add a Row* tool, enter a name, and click the *Properties* tool, to create the profile and open it in the resource editor.
3. Click the *Create an Object* tool, enter a name for the profile, and select a source from which to copy from. You can choose between:
   - Another user profile – Makes a copy of the existing profile.
   - **<Registry>** - Copies all your PowerDesigner defaults from your Windows registry.
   - **<Model>** - Opens the Model for User Profile dialog, which allows you to specify preferences to copy from an open model (see Copying Preferences from a Model [page 385]).
4. Click *OK* to open the profile in the Resource Editor.
5. [optional] Click the *Import Items* tool to import an existing profile, review the items to be imported in the Merge dialog (see Merging Models [page 369]), and then click *OK* to complete the import.

   You can repeat this step as many times as necessary to import additional profiles.
6. Edit the profile as necessary (see Reviewing and Editing User Profiles in the Resource Editor [page 386]).
7. Click *OK* to save the profile in the repository.

3.3.2.4 Applying Profiles to Users and Groups

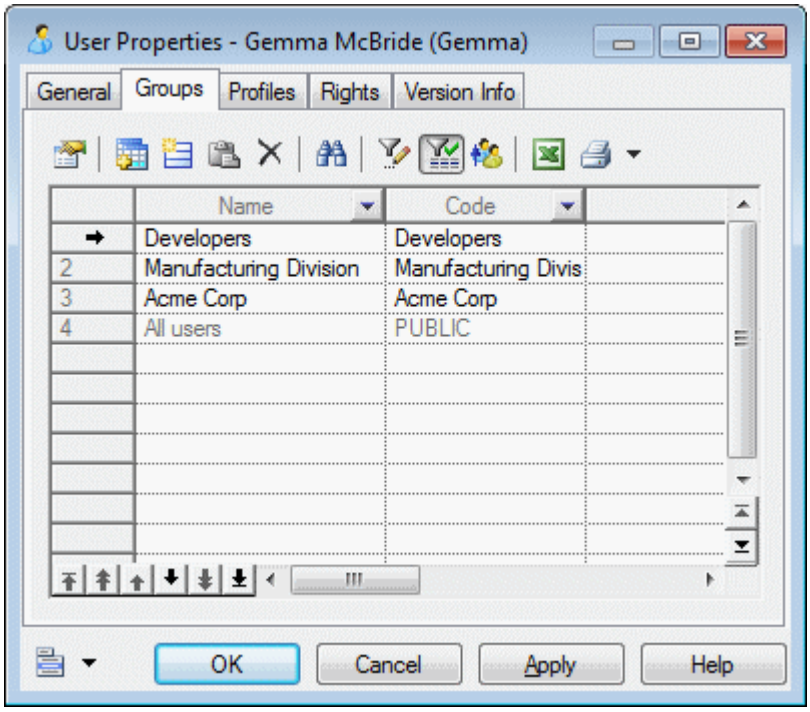
An administrator applies object permissions profiles, UI preferences profiles, and general preferences profiles to a user or group via their repository property sheet *Profiles* tab.

Context

> **i Note**
>
> Though you can apply only one of each type of profile to each user or group, you can apply multiple profiles to a user or group by inserting them into multiple groups or by using a hierarchy of groups (see Adding Users and Groups to a Group [page 532]).

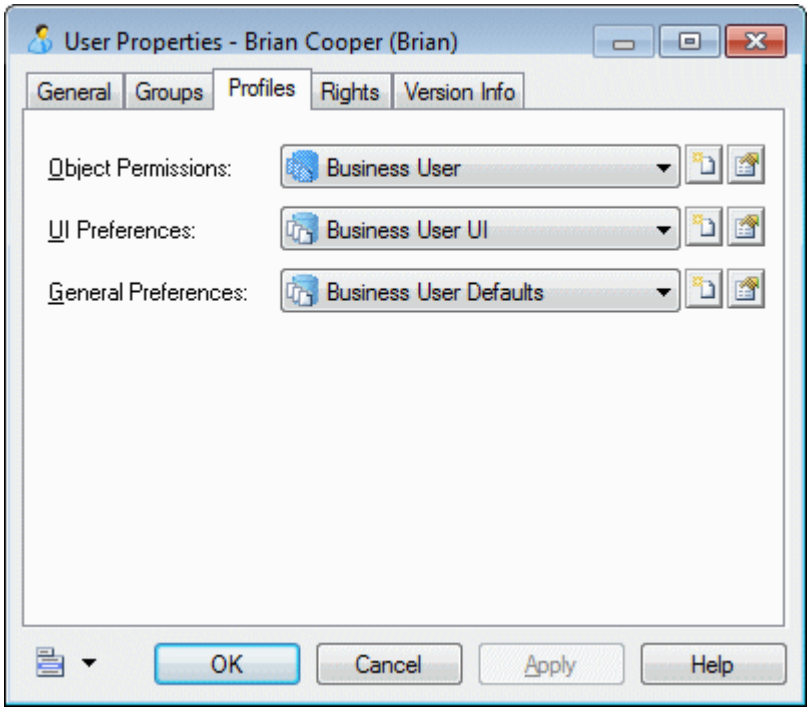
Where a user or group is associated with multiple profiles through membership in multiple groups, the following rules apply:

- Object Permissions Profiles - The most permissive superset of permissions is applied and the order of group membership is unimportant. For example, if the user is associated with one profile which hides the EAM module and another which enables it, then it will be visible for him.
- UI and General Preference Profiles - Preferences are applied starting from the bottom of the list on the *Groups* tab and finishing with any profile applied directly to the user, with the last applied preferences being used in the case of conflicts. In the following example, any UI or preference profile applied to the All users group will be applied, followed by those for the Acme Corp, Manufacturing Division, and Developers groups, and finally any profile applied to Gemma herself:



Procedure

1. Create appropriate object permission, UI preference, and general preference profiles in the repository.
2. Select *Repository* > *Administration* > *Users* (or *Repository* > *Administration* > *Groups*) and select the user or group to whom you want to apply the profile.
3. Click the *Properties* tool to open the user's property sheet, and then click the *Profiles* tab.
4. Select the appropriate profile in one or more of the *Object Permissions*, *UI Preferences*, and *General Preferences* fields, and click *OK*.



When the user next connects to the repository, the selected profiles will be pushed to their installation of PowerDesigner.

> **i** Note
>
> Permissions profiles are enforced (the user cannot access types of models or objects that are hidden from her), and the defaults specified in the UI and general preference profiles are re-applied each time the user connects to the repository (if the *Apply profiles* option is selected in the user's repository general options (see Repository Options [page 396]).

3.4 PowerDesigner Security Guide

This Security Guide provides an overview of the security concepts used and recommended in administering a PowerDesigner environment. It is aimed at technology consultants, security consultants, and system administrators.

As it becomes more and more important for organizations to protect their critical data from unauthorized access and to ensure compliance with a growing number of rules and regulations, the demands on security are also on the rise. As a repository for your enterprise metadata, PowerDesigner can contain sensitive information about your organization and its systems. It is therefore essential that you integrate PowerDesigner into your infrastructures securely and that you protect your data in PowerDesigner.

This guide provides an overview of the security-relevant information that applies to PowerDesigner:

- Auditing and Logging [page 570]
- Data Protection and Privacy [page 570]

- Identity and Access Managment [page 571]
- Network and Communication Security [page 573]
- Cookies [page 575]
- Secure Delivery, Configuration, and Change Management [page 575]
- Operational Security [page 576]

3.4.1 Auditing and Logging

Auditing provides you with visibility on who did what in the PowerDesigner repository (or tried to do what) and when.

Auditing allows you to monitor and record selected actions performed in the PowerDesigner repository. Although auditing does not directly increase your system's security, if wisely designed, it can help you achieve greater security in the following ways:

- Uncover security holes if too many privileges were granted to some user
- Show attempts to breach security
- Protect the system owner against accusations of security violations and data misuse
- Allow the system owner to meet security standards

The following forms of auditing are available:

- View lists of events concerning repository documents and user accounts (see Monitoring Repository Activities [page 537]).
- View the repository proxy log file (see *Installation Guide* > *Installing the Repository Proxy*).
- If you are experiencing problems with memory usage or repository database performance, contact customer support to obtain access to further debugging features.

3.4.2 Data Protection and Privacy

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy acts, it is necessary to consider compliance with industry-specific legislation in different countries.

> **i** Note
>
> In the majority of cases, compliance with data privacy laws is not a product feature. SAP software supports data privacy by providing security features and specific functions relevant to data protection, such as functions for the deletion of personal data. SAP does not provide legal advice in any form. The definitions and other terms used in this guide are not taken from any given legal source.

3.4.3 Identity and Access Managment

Every user who wants to work with the repository must have a repository user with the necessary privileges and permissions. After a successful logon, the user's authorization to perform the requested operations on the requested objects is verified.

Repository privileges give users access to general repository features, while permissions give them access to particular locations in the repository. The following privileges and permissions are available:

Privileges (Entire Repository)	Permissions (Per Folder or Item)
• **`Connect`** - Connect to the repository. • **`Edit on Web`** - Create and edit diagrams in PowerDesigner Web. • **`Edit Extensions on Web`** - Create and edit custom properties in PowerDesigner Web. • **`Freeze Versions`** - (see Freezing and Unfreezing Document Versions [page 275]). • **`Lock Documents`** - (see Locking and Unlocking Documents [page 276]). • **`Manage Branches`** - (see Branching Version Trees [page 281]). • **`Manage Configurations`** - (see Grouping Document Versions in a Configuration [page 278]). • **`Manage All Documents`** - Perform any action on any document version. Implicitly includes **`Full`** permission on all repository documents. • **`Manage Users & Permissions`** - Create, modify, and delete repository users and groups, grant them privileges, and add them to groups. Users with this privilege can list all repository documents and set permissions on them without needing explicit **`Full`** permission. • **`Manage Repository`** - Create, upgrade, and delete the repository database.	• **`List`** - View the document or folder in the repository browser and in search results. Without this permission, the folder or document is hidden from the user. • **`Read`** - Also check documents out from the repository and compare documents. • **`Submit`** - Also check the document into a changelist for review by a user with **`Write`** permission. • **`Write`** - Also check in (with or without a changelist), freeze, and lock document versions. • **`Full`** - Also move and delete documents, manage permissions granted to users and groups, and remove locks on documents. **i Note** Administrators, who have implicit **`Full`** permission on all repository objects, will only receive models for review if they have been granted explicit **`Write`** permission on them.

Privileges and permissions can be granted to repository users either directly, or indirectly through groups.

User Administration Tools

In order to manage repository users and groups, you must have the **`Manage Users`** privilege. In order to grant permission on a repository folder or document, you must have **`Full`** permission on the folder or document.

See Controlling Repository Access [page 510].

Standard Users and Groups

The following standard users are automatically created in the PowerDesigner repository:

- **ADMIN** - Has all available privileges and implicit **Full** permission on all repository folders.
- **_ADMIN** - Is created as a temporary emergency user for connecting to a repository when no administrator is able to access it (see Obtaining Emergency Access to the Repository [page 539]).

The following standard groups are automatically created in the PowerDesigner repository:

- Administrators, [ADMN] - Has, by default, all available privileges and implicit **Full** permission on all repository folders.
- All users [PUBLIC] - Has, by default, **Read** permission on the repository root. All users belong to this group and can thus, by default, check out any document.
- External users [EXTERNAL] - Has, by default, no privileges or permissions. Users authenticated via LDAP (see Connecting to an LDAP Server for User Authentication [page 513]) or single-sign on (see Enabling Single Sign-On for PowerDesigner Web [page 527]) are automatically added to this group when they connect for the first time.

Password Policy

Passwords for the user name/password authentication of repository users are subject to a password policy. You can change the default password policy in line with your organization's security requirements. You cannot deactivate the password policy.

See Defining a Password Policy [page 518].

LDAP

PowerDesigner supports delegating user authentication to an LDAP server.

See Connecting to an LDAP Server for User Authentication [page 513].

Single Sign-On

In environments where X.509 client authentication is in place, PowerDesigner Web supports authentication of users by the user certificate stored on their client machine.

See Enabling Single Sign-On for PowerDesigner Web [page 527].

Controlling Write Access through Change Lists

PowerDesigner supports the enforcement of reviewing proposed changes to repository documents through the use of the `Submit` permission, which requires users to pass through a review process before publication.

See Change Lists [page 259].

Secure Locations and Trusted Files

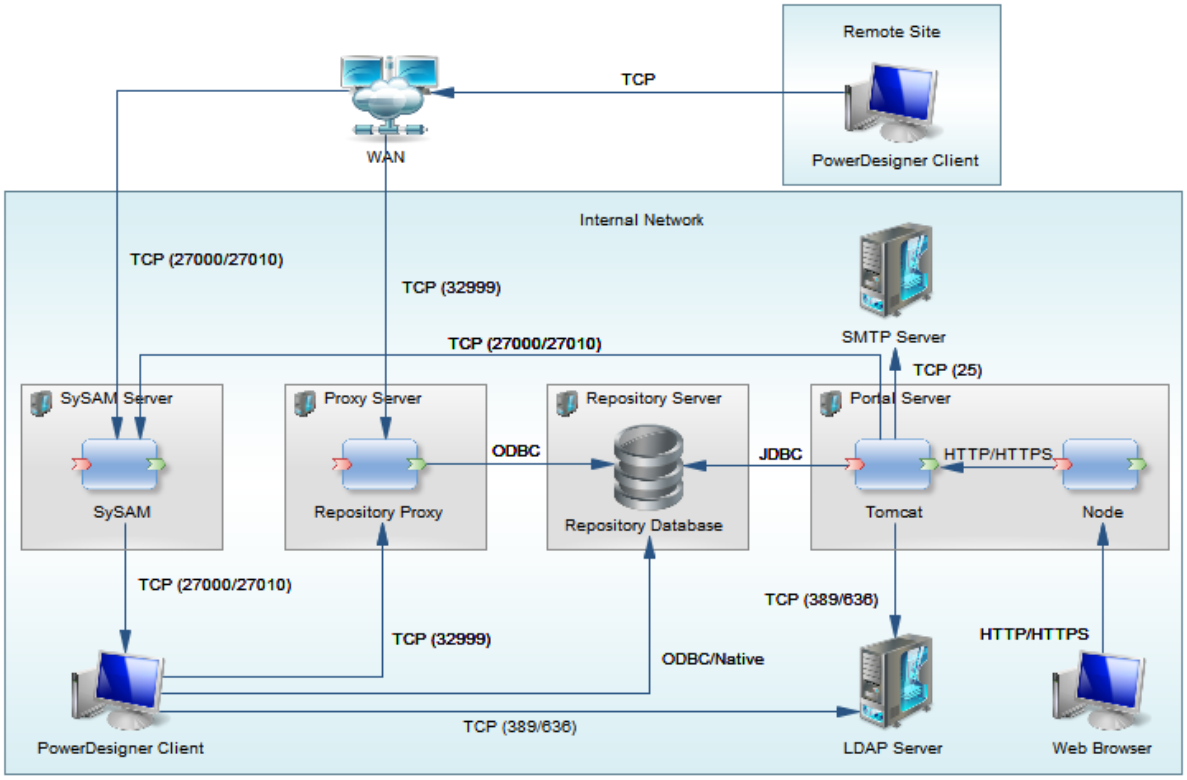
To reduce the risk posed by malicious scripts, you can instruct PowerDesigner to notify you when an untrusted script embedded in or called from a model or resource file is run, to prompt you for permission to run an untrusted script, or to block such scripts from running altogether.

See Security Options [page 398].

3.4.4 Network and Communication Security

The components of a PowerDesigner landscape communicate through different network communication channels. It is recommended security practice to have a well-defined network topology to control and limit network access to the PowerDesigner repository to only those communication channels needed for your scenario, and to apply appropriate additional security measures, such as encryption, where necessary.

This can be achieved through different means, such as separate network zones and network firewalls, and through the configuration options. The exact setup depends on your environment, your implementation scenario, and your security requirements and policies.



The network connections shown in the diagram can be encrypted as follows:

- PowerDesigner Client to:
  - Repository - encryptable through options in the appropriate connection profile.
  - Proxy - encrypted by default.
  - SySAM - not encryptable.
  - LDAP Server - encryptable by enabling SSL (see Connecting to an LDAP Server for User Authentication [page 513]).
- Proxy Server to Repository - encryptable through options in the appropriate connection profile.
- Web Browser to Node - encryptable by enabling SSL (see Enabling Secure Connections (SSL/TLS) for PowerDesigner Web [page 520]).
- Node to Tomcat - encryptable by enabling SSL (see Enabling Secure Connections (SSL/TLS) for PowerDesigner Web [page 520]).
- Tomcat to:
  - Repository - encryptable through options in the appropriate connection profile.
  - LDAP Server - encryptable by enabling SSL (see Connecting to an LDAP Server for User Authentication [page 513]).
  - SMTP Server - can be encrypted by enabling SSL (see Connecting to an SMTP Server for Notifications [page 517]).
  - SySAM - not encryptable.

Several mechanisms can be used to protect security-relevant data stored in and read from the PowerDesigner repository. PowerDesigner supports encryption of both passwords and data for direct and proxy connections

to a repository and for connections to other databases using most of the supported DBMSs via one or both of ODBC and JDBC connections.

The following table provides information and examples for encrypting passwords and data with each type of connection:

Type	Encryption
ODBC	Password and data encryption can generally be enabled in the ODBC Configuration dialog. For example, to encrypt the password for SAP SQL Anywhere, select the *Encrypt password* option on the *Login* tab, and to encrypt data, select an encryption method on the *Network* tab.
JDBC	It is not possible to control password and data encryption through the PowerDesigner connection profile interface, but you may be able to enable encryption directly in your database client.

For more information about using connection profiles, see Connecting to a Database [page 22]. For detailed information about encryption and other parameters for your database, see your DBMS documentation.

PowerDesigner Web supports reverse engineering from and generating to SAP Solution Manager v7.2 servers via HTTP connections. We recommend that you encrypt your connections, if possible, by selecting the *Use https connection* option.

3.4.5 Cookies

Session cookies are required for each PowerDesigner Web client session and are deleted when the session is closed. Additional persistent cookies are used to store the most recent choices for content language and user interface language.

3.4.6 Secure Delivery, Configuration, and Change Management

PowerDesigner and PowerDesigner Web require third party components and will install them if they are not present in your environment.

PowerDesigner Dependencies

PowerDesigner depends on the following frameworks and libraries:

- Microsoft .NET Framework 4.0.

  > **i** Note
  >
  > After installation is complete, we recommend that you check for and install the latest security updates.

- `o2003pia.msi`, `o2007pia.msi`, `o2010pia.msi` - Microsoft Office Primary Interop Assemblies for MS Office 2003, 2007, and 2010 to support PowerDesigner MS Office plugins.
- `msxml6-KB2758696-fra-x86.exe` or `msxml6-KB2758696-fra-amd64.exe` - to support the PowerDesigner XSM.
- `NCo3016_Net40_x86.msi` or `NCo3016_Net40_x64.msi` - SAP DotNetConnector3
- `vjredist.exe` - Microsoft Visual J# 2.0 redistribuable to support the PowerDesigner Auto-Layout feature.
- `vcredist_x86.exe` or `vcredist_x64.exe` - Microsoft Visual C++ 2008 and 2010 Runtime Libraries
- `vstor_redist.exe` - Visual Studio Tools for Office

PowerDesigner Web Dependencies

PowerDesigner Web depends on the following components:

- A recent Node.js 6.x Active LTS version.
- Java 8 (The installer will install the SAP JVM 8.x and a Tomcat 8 server.)

> **Note**
>
> After installation is complete, we recommend that you check for and install the latest security updates.

3.4.7 Operational Security

It is possible to securely scale your PowerDesigner Web installation through the use of load balancers such as NGINX or HAProxy. Refer to your load balancer documentation for security recommendations

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

www.sap.com/contactsap

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

THE BEST RUN SAP