SYBASE®

An **SAP®** Company

Core Features Guide

PowerDesigner® 16.1

Windows

Contents

Contents

CHAPTER 14: Generating Models and Model Objects 433

CHAPTER 15: Shortcuts and Replicas 451

PART I

The PowerDesigner Interface

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

CHAPTER 1   **Getting Started with PowerDesigner**

PowerDesigner® is a graphical enterprise modeling solution supporting standard methodologies and notations and providing automated code reverse engineering and generation through customizable templates. It provides a scalable enterprise repository solution with strong security and versioning capabilities to aid multi-user development, powerful reporting capabilities, and is highly extensible.
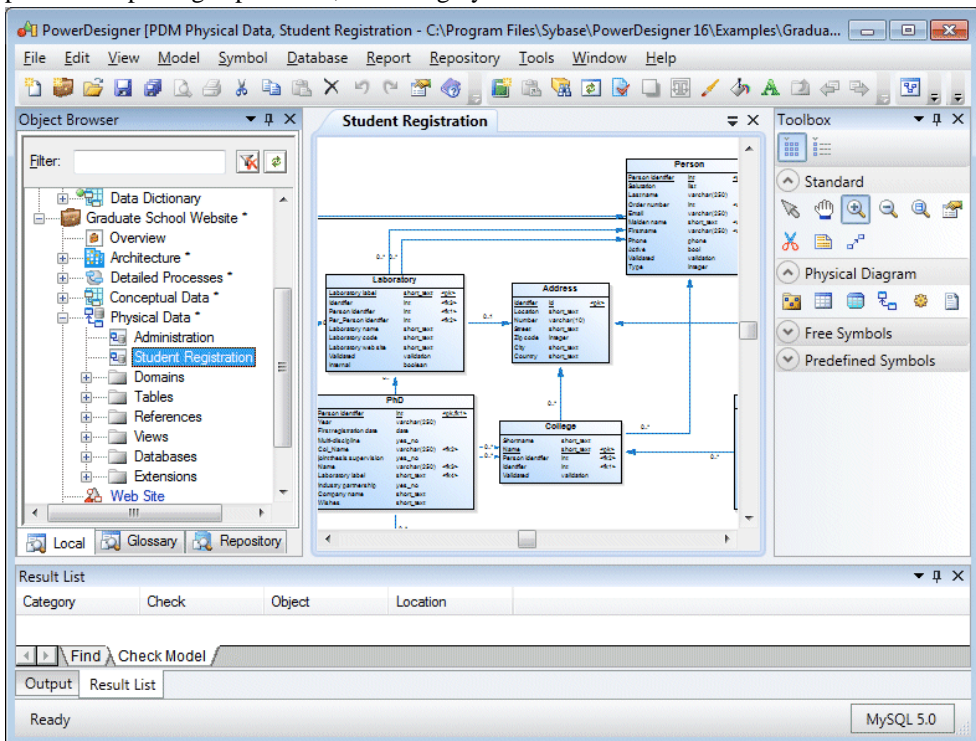
The default PowerDesigner window displays the following views:

- The *Browser* - displays your models and the objects belonging to them (see *Chapter 2, The Browser* on page 39). The Browser also allows you to view a PowerDesigner repository (see *Chapter 8, The Repository* on page 291) and may display an enterprise library of shared models (see *The Library* on page 43) and an enterprise glossary (see *The Glossary* on page 40).
- The *Canvas* - is a tabbed view that displays your open diagrams (see *Diagrams* on page 175) and report outlines (see *Chapter 6, Reports* on page 225).

- 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* on page 95).
- 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* on page 409.

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* on page 97).
- *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* on page 115).

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* on page 6).
  - **Create Project** - Opens the New Project dialog to let you create a project (see *Creating a Project* on page 54).
  - **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 Sybase® website.
        - • **PowerDesigner Newsgroup** - Opens the PowerDesigner newsgroup page on the Sybase website.
    - • **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.

Creating a Model

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

Note: A project can provide a convenient environment for working with multiple interconnected models and other files (see *Chapter 3, Projects and Frameworks* on page 53).

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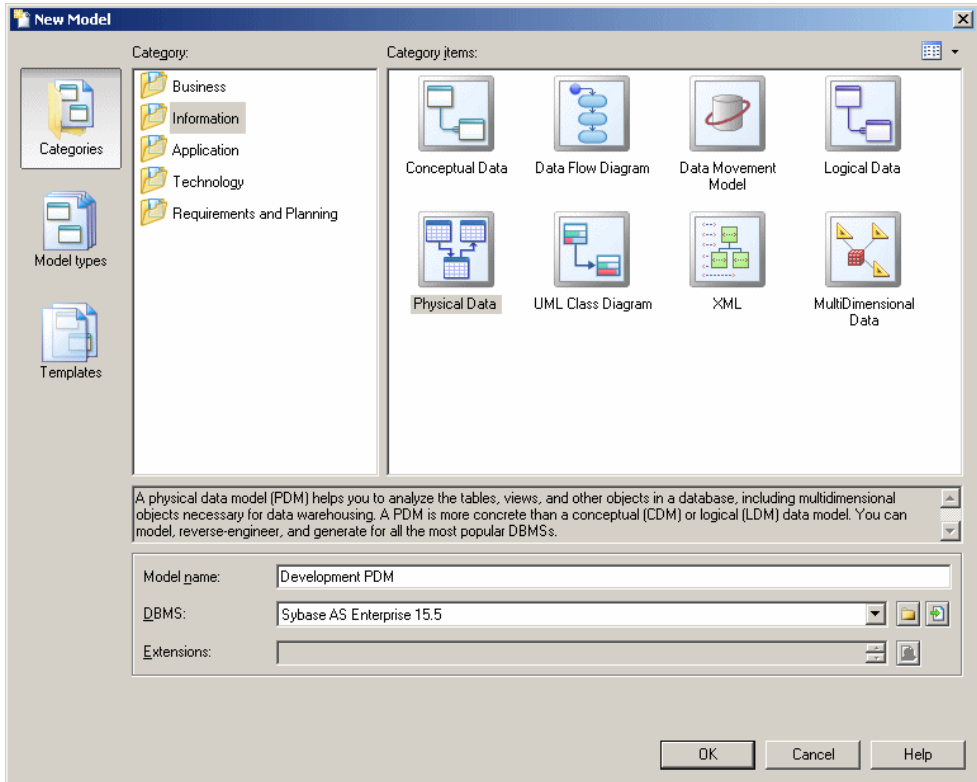
1. Select **File > New Model** to open the New Model dialog.
2. Click a button, and then select a category or model type (as appropriate) in the left-hand pane.

   The following table lists the PowerDesigner model types:

Icon	Model definition	File ext	Back-up ext
	Requirements Model. 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.	.rqm	.rqb
	Enterprise Architecture Model. An *enterprise architecture model (EAM)* helps you analyze and document your organization and its business functions, along with the applications and systems that support them and the physical architecture on which they are implemented.	.eam	.eab

Icon	Model definition	File ext	Back-up ext
	Business Process Model. 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). You can use BPEL, BPMN, and many other process languages.	.bpm	.bpb
	Conceptual Data Model. 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.	.cdm	.cdb
	Logical Data Model. 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).	.ldm	.ldb
	Physical Data Model. 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.	.pdm	.pdb
	Data Movement Model. 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.	.dmm	.dmb
	Object Oriented Model. 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.	.oom	.oob
	XML Model. 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.	.xsm	.xsb
	Free Model. A *free model (FEM)* provides a context-free environment for modeling any kind of objects or systems. It is generally associated with a set of extensions, which allow you to define your own concepts and graphical symbols.	.fem	.feb

Icon	Model definition	File ext	Back-up ext
🔹	Multi-Model 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.	.mmr	.bmr

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 .

Note: Sample models are available in the Example Directory.

Note: For information about creating new models in the PowerDesigner Eclipse and Visual Studio plugins, see *Creating a Model in Eclipse* on page 358 and *Creating a Model in Visual Studio* on page 370.

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.

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.

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

PowerDesigner Models and Diagrams

PowerDesigner provides a wide range of models and diagrams for all your modeling needs.

- 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.
  - A *requirements document view* displays a list of written requirements in a hierarchic grid.
  - A *traceability matrix view* displays the links between requirements and objects from other types of models, external files or other requirements.
  - A *user allocation matrix view* displays the links between requirements and the users and groups who will fulfill them.
- An *enterprise architecture model (EAM)* helps you analyze and document your organization and its business functions, along with the applications and systems that support them and the physical architecture on which they are implemented.
  - A *process map* provides a graphical view of your business architecture, and helps you identify your business functions and high-level processes, independent of the people and business units who fulfill them.
  - An *organization chart* provides a graphical view of your organization as a tree structure, and helps you analyze and display the relationships between organization units (divisions, groups, teams, etc), individuals, and roles.
  - A *business communication diagram* provides a graphical view of your organization, and helps you analyze, the relationships, flows, and other connections between business functions, organization units, roles, and sites.
  - A *city planning diagram* provides a graphical view of the big picture of your enterprise architecture, using the metaphor of planning the infrastructure of a city to represent the organization of systems, applications, etc into architectural areas.
  - A *service-oriented diagram* provides a graphical view of your business and application services and the relationships between them, and helps you associate applications and other application layer objects with business services and processes to assist with SOA design.

- An *application architecture diagram* provides a high-level graphical view of the application architecture, and helps you identify applications, sub-applications, components, databases, services, etc, and their interactions.
  - A *technology infrastructure diagram* provides a high-level graphical view of the physical architecture required to support the application architecture.
- 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). You can use BPEL, BPMN, and many other process languages.
  - A *business process diagram* (or process flow diagram) provides a graphical view of the control flow (the sequence of execution) or data flow (the exchange of data) between processes at any level in your system.
  - A *process hierarchy diagram* (or functional decomposition diagram) provides a graphical view of the functions of a system and helps you decompose them into a tree of sub-processes.
  - A *process service diagram* provides a graphical view of the services, operations, and interfaces available in your system.
- 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.
  - A *conceptual data diagram* provides a graphical view of the conceptual structure of an information system, and helps you identify the principal entities to be represented, their attributes, and the relationships between them.
- 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).
  - A *logical data diagram* provides a graphical view of the structure of an information system, and helps you analyze the structure of your data system through entities and relationships, in which primary identifiers migrate along one-to-many relationships to become foreign identifiers, and many-to-many relationships can be replaced by intermediate entities.
- 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.
  - A *physical data diagram* provides a graphical view of your database structure, and helps you analyze its tables (including their columns, indexes, and triggers), views, and procedures, and the references between them.
  - A *multidimensional data diagram* provides a graphical view of your datamart or data warehouse database, and helps you identify its facts, cubes and dimensions.

- 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.
  - A *data movement diagram* provides a high-level graphical view of the movement of your information, including data sources, replications, and ETL operations.
  - A *data transformation diagram* provides a graphical view of the inputs, outputs, and steps involved in a data transformation task.
  - A *transformation control flow diagram* provides a graphical view of the order in which a series of data transformation tasks is linked together in a control flow.
- 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.
  - A *use case diagram* is a UML diagram that provides a graphical view of the requirements of your system, and helps you identify how users interact with it.
  - A *class diagram* is a UML diagram that provides a graphical view of the classes, interfaces, and packages that compose a system, and the relationships between them.
  - An *object diagram* is a UML diagram that provides a graphical view of the structure of a system through concrete instances of classes (objects), associations (instance links), and dependencies.
  - A *composite structure diagram* is a UML diagram that provides a graphical view of the classes, interfaces, and packages that compose a system, including the ports and parts that describe their internal structures.
  - A *package diagram* is a UML diagram that provides a high-level graphical view of the organization of your application, and helps you identify generalization and dependency links between the packages.
  - A *sequence diagram* is a UML diagram that provides a graphical view of the chronology of the exchange of messages between objects and actors for a use case, the execution of an operation, or an interaction between classes, with an emphasis on their chronology.
  - A *communication diagram* is a UML diagram that provides a graphical view of the interactions between objects for a use case scenario, the execution of an operation, or an interaction between classes, with an emphasis on the system structure.
  - An *interaction diagram* is a UML diagram that provides a high-level graphical view of the control flow of your system as it is decomposed into sequence and other interaction diagrams.
  - An *activity diagram* is a UML diagram that provides a graphical view of a system behavior, and helps you functionally decompose it in order to analyze how it will be implemented.
  - A *statechart diagram* is a UML diagram that provides a graphical view of a State Machine, the public behavior of a classifier (component or class), in the form of the

changes over time of the state of the classifier and of the events that permit the transition from one state to another.

- A *component diagram* is a UML diagram that provides a graphical view of the dependencies and generalizations among software components, including source code components, binary code components, and executable components.

- A *deployment diagram* is a UML diagram that provides a graphical view of the physical configuration of run-time elements of your system.

- 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.

  - An *XML diagram* provides a graphical view of the elements that comprise an XML schema definition in a tree format.

- A *free model (FEM)* provides a context-free environment for modeling any kind of objects or systems. It is generally associated with a set of extensions, which allow you to define your own concepts and graphical symbols.

  - A *free diagram* provides a context-free graphical environment for modeling any kind of objects or systems.

- 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.

  - A *multimodel report (MMR)* is a PowerDesigner report that can document any number of models. 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.

Model Templates

A model template is a shell model that contains a set of model options, display preferences, extensions, and/or objects, which you can reuse in multiple models of the same type. For example, a BPM model template can be used to create other BPMs, but not an OOM or PDM. Model templates are stored in the model templates directory, which is, by default, the PowerDesigner installation directory.

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 (select **File > New Model**, or click the New Model tool, and then select the Templates button.

Note: The New Model dialog is highly customizable and its display is controlled by the Model Creation general options. 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. For information about these options, see *Customizing the New Model Dialog* on page 406.

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.

For information about using the New Model dialog, see *Creating a Model* on page 6.

Accessing the Repository

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

In the following diagram, the workstations connect to the Repository database in various ways:
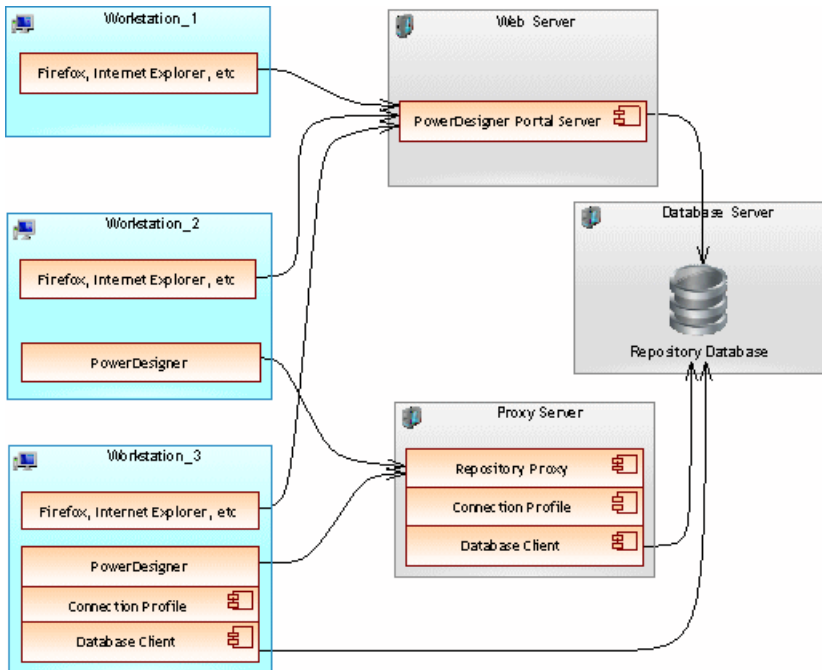
These connection types are further explained in the following sections:

• Web connection [if the PowerDesigner Portal is installed] – see *Chapter 9, The PowerDesigner Portal* on page 331

- Direct connection – see *Setting up PowerDesigner for Direct Access* on page 15
- Proxy connection [if the repository proxy is installed] - see *Setting up PowerDesigner for Access Via the Proxy* on page 17

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*.

Setting up PowerDesigner for Direct Access

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

If your administrator has installed the repository proxy, see *Setting up PowerDesigner for Access Via the Proxy* on page 17.

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

1. Select **Repository > Repository Definitions** to open the List of Repository Definitions.

2. Click the **Add a Row** tool and type a name in the **Repository Name** column to identify the definition for future use.

3. Type your user name in the **Repository User** column. If you are an administrator and are creating the first repository definition that will be used to initialize the repository (see *Chapter 18, Controlling Repository Access* on page 521), you must use the ADMIN repository user name.

4. Press F4 or click the Ellipsis button in the **Data Source Name** column to open the Select a Data Source dialog:
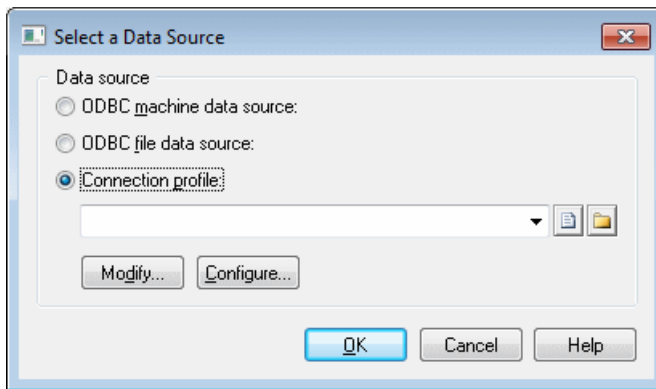
5. Select one of the following options:

   - ODBC machine data source

- ODBC file data source
- Connection profile (for native or 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* on page 23.

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:
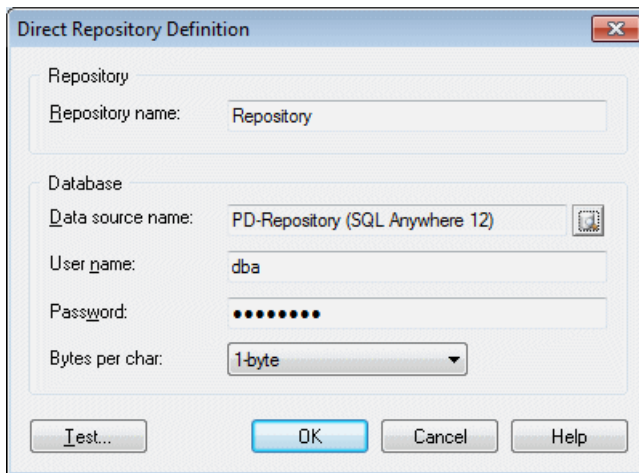
This window has two group boxes:
- **Repository** – specifies the name of the repository definition.
- **Database** – specifies the name of the data source that hosts the repository database, and the user name and password that you will use to connect to it. The user name and password are optional, but 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 - [default] For SBCS (Single-Byte Character Set)
  - 2-byte - For DBCS (Double-Byte Character Set)
  - 3-byte - For Unicode or MBCS (Multi-Byte Character Set)

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

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

Setting up PowerDesigner for Access Via the Proxy

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

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

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:
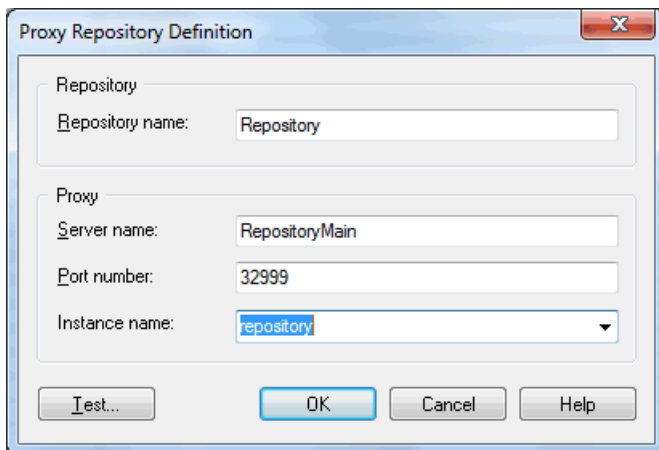
This window has two group boxes:
- **Repository** – specifies the name of the repository definition.
- **Proxy** – specifies the name of the server where the repository proxy is installed, the port number, and the name of the repository instance that you will connect to.

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.

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.

For information about creating a repository definition, see *Accessing the Repository* on page 14.

1. Select **Repository > Connect** (or double-click the root in the Browser **Repository** tab) to open the Connect dialog.
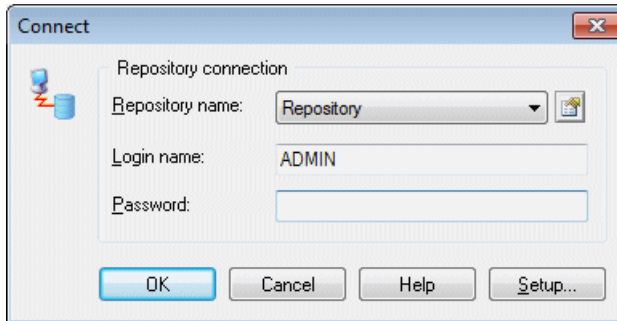
2. Select a repository to connect to and enter your login name and password. To review the repository definition, click the **Properties** tool.

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* on page 43) are not up to date, the Library Synchronization dialog will open to invite you to update them:
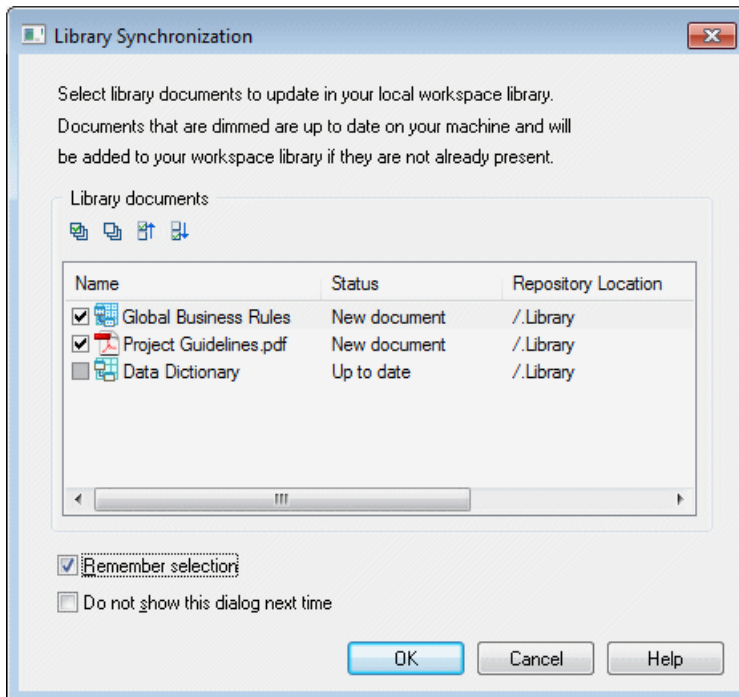
4. Select documents to update in your local library folder. Select **Remember selection** to retain the same selection next time you connect to the repository and select **Do not show this dialog next time** to have the library silently update from now on.

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.

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

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.

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 General Options* on page 407.

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.

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* on page 526).

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* on page 533).

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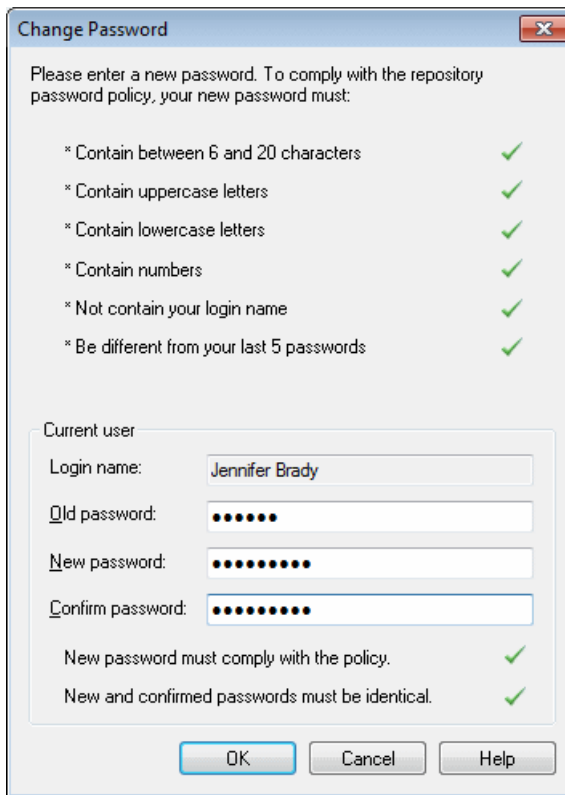
When your new password conforms to all the policy rules and the values entered in the
New password and **Confirm password** fields are identical, the **OK** button becomes
available.

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

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 Modeling*) and data movement model (see
Data Movement Modeling) 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
Chapter 8, The Repository on page 291) requires a database to store your models and other
design documents.

PowerDesigner supports various forms of connection, and your choice will depend on the interface that you have already installed:

You have...	Configure a connection of type...
ODBC driver	ODBC machine or file data source (see *Configuring ODBC Machine and File Data Sources* on page 21).
DBMS client	Native connection profile (see *Configuring Connection Profiles* on page 23).
JDBC driver	JDBC connection profile (see *Configuring Connection Profiles* on page 23).

Configuring ODBC Machine and File Data Sources

There are three types of Open Database Connectivity (ODBC) data sources:

Data source	Description
Machine	Data source is created on the client machine, and is available to the user currently logged onto the system. Machine data sources are stored in the part of the registry containing settings for the current user.
System	Data source is created on the client machine, and is available to all users regardless of whether a user is logged onto the system or not. System data sources are stored in the part of the registry containing settings for the current machine.
File	Data source is stored as a file. A file data source has the extension .dsn. It can be used by different users if it is placed in the default location for file data sources. File data sources are usually managed by database administrators.

You define data sources using the Windows ODBC Administrator, which you can access directly from PowerDesigner:

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.
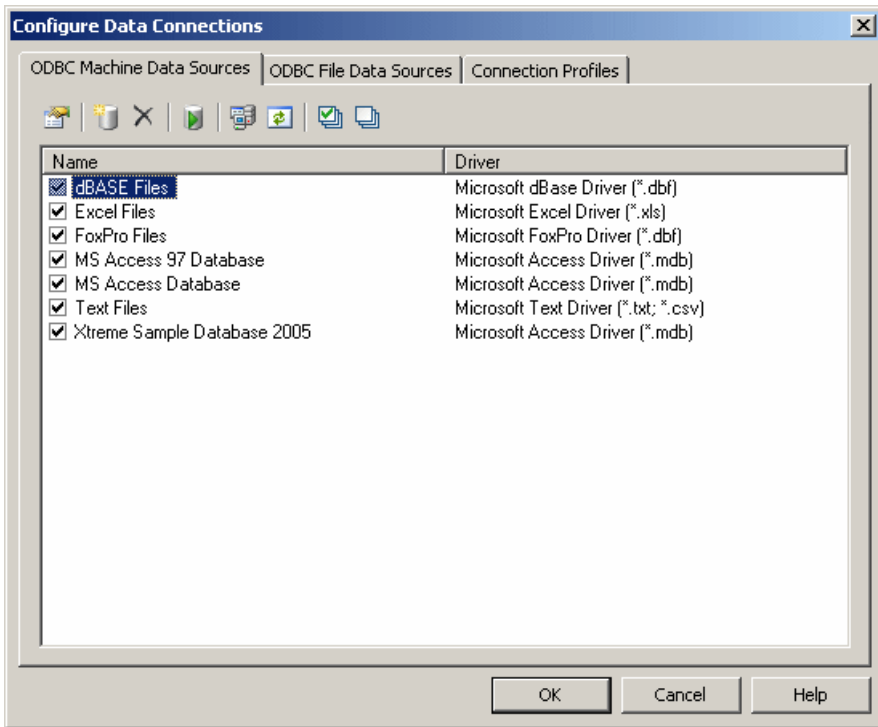
2. Click the Add Data Source tool (see *Data connection tools* on page 28) to open the Create New Data Source window, and select a File, User, or Machine data source as appropriate and click Next.

3. Select the appropriate driver for your DBMS and click Next and then Finish to access a driver-specific configuration dialog:
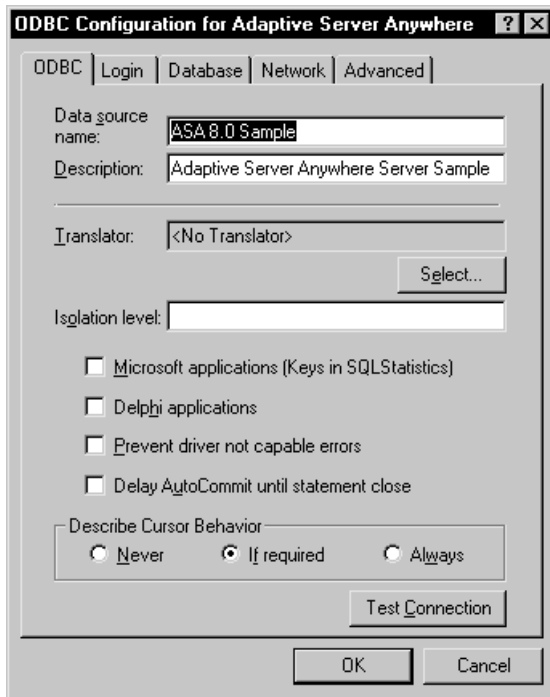
4. Enter the appropriate parameters and then click OK to return to the Configure Data Connections window.

5. Click OK to return to your model.

Configuring Connection Profiles

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

1. Select **Database > Configure Connections** to open the Configure Data Connections Window, and click the Connection Profiles tab:
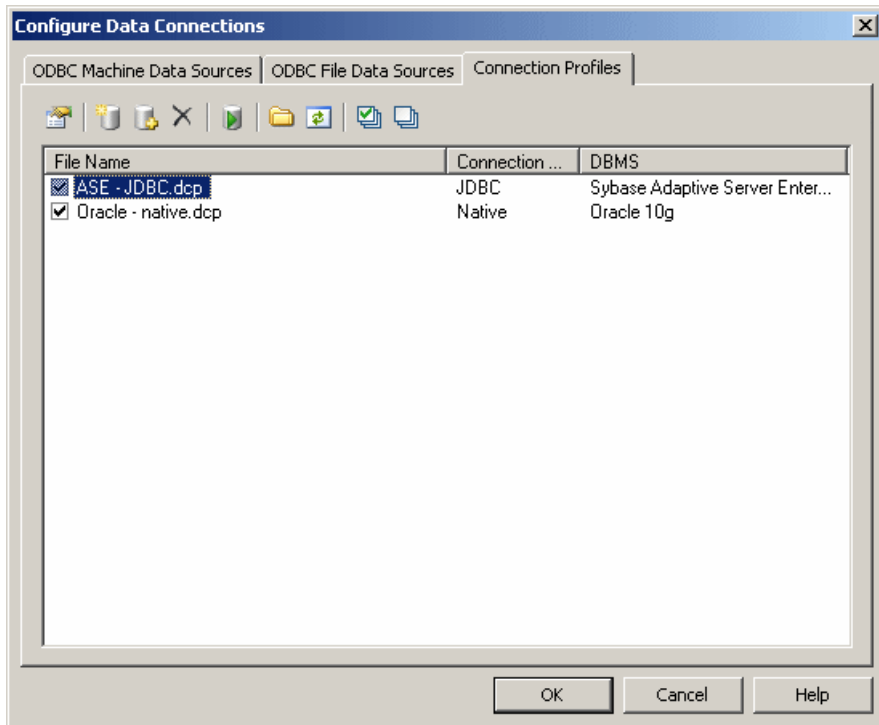
2. Click the Add Data Source tool (see *Data connection tools* on page 28) to open a Connection Profile Definition window:

Connection Profile Definition   ✕

Connection profile name:	Oracle - native.dcp
Directory:	C:\Program Files\Sybase\PowerDesigner 15\Connection Profiles\ 📁
Description:	
Connection type:	Native ▼
DBMS type:	Oracle 11g ▼
Service:	Delphes
User name:	mledier
Password:	••••••••
Connection parameters:	TrimSpaces=0 ...

Test Connection...   OK   Cancel   Help
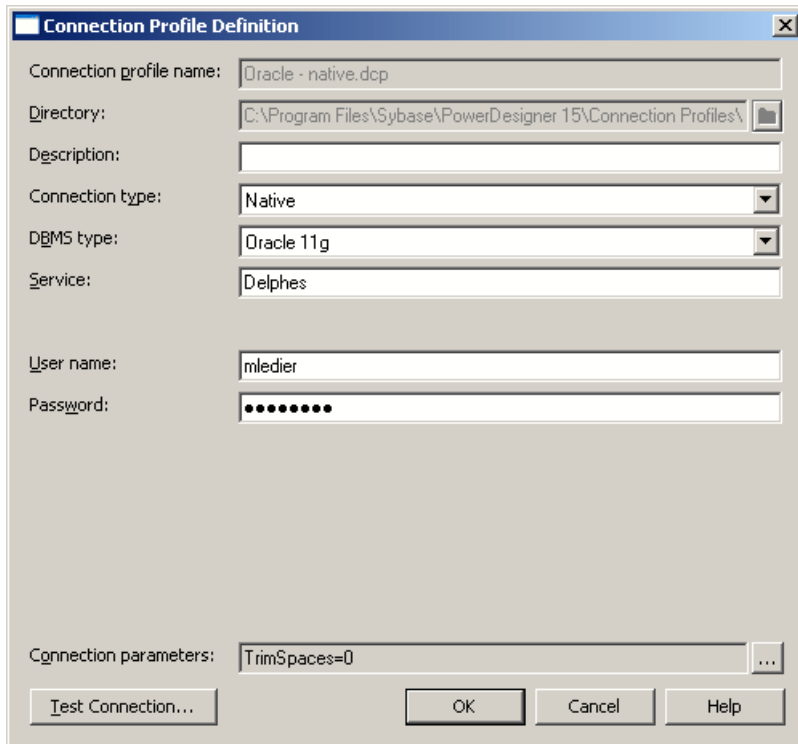
3. Enter the properties (see *Connection profile properties* on page 26) appropriate to your method of connection.

4. Click the Test Connection button to open the Test Connection window, and click OK to test your connection.

Test Connection   ✕

Data source:	Oracle - native.dcp
User name:	mledier
Password:	•••••••

OK   Cancel
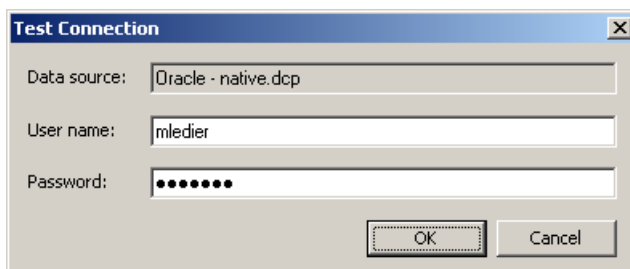
5. Click OK to close the profile definition, and return to the Configure Data Connections window

6. Click Ok to return to your model

Connection Profile Properties

The Connection Profile Definition window lets you specify all the necessary properties for connecting to your database.

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. You can choose between: • Native • JDBC The choice of connection type will affect the remaining fields to be completed.
DBMS type	Specifies the DBMS to which the connection profile will connect. The list includes only those DBMSs supported for the specified connection type.
Server name	[Native only] Specifies the name of the database server to connect to.
Database name	[Native only] Specifies the name of the database to connect to.
Service	[Oracle Native only] Specifies the service name of the database to connect to.
User name	Specifies the username to use when connecting.
Password	Specifies the password to use when connecting.
JDBC driver class	[JDBC only] Specifies the driver class to use for the connection.
JDBC connection URL	[JDBC only] Specifies the connection URL to use for the connection.
JDBC driver jar files	[JDBC only] Specifies the driver jar file to use for the connection.
Connection parameters	Specifies advanced connection parameters. Click the parenthesis tool to the right of this field to access the Connection Parameters window.

Connection Parameters Window

You access this window by clicking the parenthesis tool to the right of the Connection parameters field in the Connection Profile Definition window.

1. Click the Add a Row tool to create a new parameter.
2. Enter a name and value pair in the two columns.

3. Click OK to return to the Connection Profile Definition window. All the parameters are listed in the read-only Connection parameters field.

Native Connection Profile Example

In the example below, I have installed the Oracle DBMS client, and so create a Native connection to the "Delphes" service.

JDBC Connection Profile Example

In the example below, I have installed the Sybase ASE JDBC driver. I specify the appropriate driver class in the jar file, and assemble the connection URL to create a JDBC connection to my database, "Green", on my server "Rainbow".

Connection Profile Definition	
Connection profile name:	ASE - JDBC.dcp
Directory:	C:\Program Files\Sybase\PowerDesigner 12\Connection Profiles\
Description:	
Connection type:	JDBC
DBMS type:	Sybase Adaptive Server Enterprise
Server name:	
Database name:	
User name:	graham
Password:	•••••••
JDBC driver class:	com.sybase.jdbc3.jdbc.SybDriver
JDBC connection URL:	jdbc:sybase:Tds:Rainbow:5000/Green
JDBC driver jar files:	C:\Program Files\Sybase\PowerDesigner 12\jconn3.jar
Connection parameters:	Driver='com.sybase.jdbc3.jdbc.SybDriver',URL='jdbc:sybase:Tds:
Test Connection...	OK Cancel Help

Data Connection Tools

The Configure Data Connections window provides tools to manipulate your connections to data sources.

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.

Tool	Description
▶	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.

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.

1. Select **Database > Connect** to open the Connect to a Data Source window:
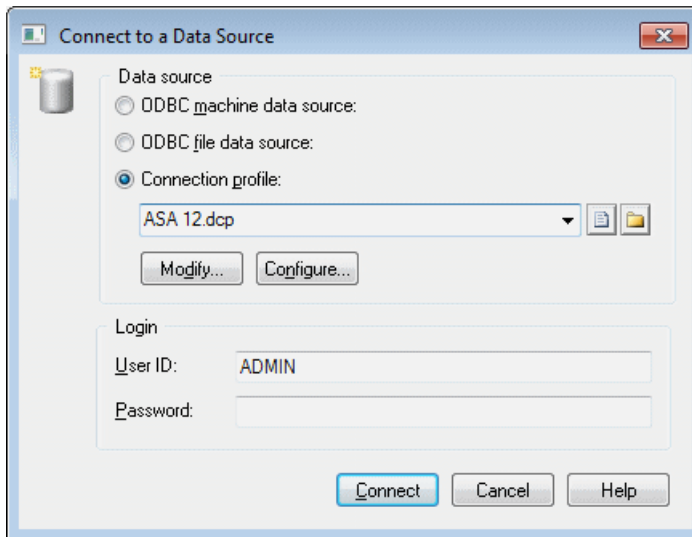
2. Select one of the following radio buttons, depending on your chosen method for connecting to your data source:

   • ODBC machine data source (see *Configuring ODBC Machine and File Data Sources* on page 21)

- ODBC file data source (see *Configuring ODBC Machine and File Data Sources* on page 21) - use the tool to the right of the data source field to browse to a new file
- Connection profile (see *Configuring Connection Profiles* on page 23) - 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.

Note: You stay connected until you disconnect or terminate the shell session.

Running SQL Queries Against Your Database

You can use the Execute SQL Query dialog to query your database.

In addition to the standard tools (see *Text Editor Tools* on page 210), 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**)

License Management

You can evaluate PowerDesigner by obtaining a fully featured trial version, which expires 15 days after installation. If you need more than 15 days to evaluate PowerDesigner, you can obtain an extension from Sybase, which will be provided as a standalone local license granted for a limited period of time. To continue to use PowerDesigner after the trial period, you must purchase a regular license from Sybase or a reseller.

PowerDesigner is licensed as a package of modules, or types of model. Various packages are available, each regrouping some or all of the modules. Your permission to use these modules is controlled by the Sybase licensing system, SySam (see *Installation Guide > Planning Your PowerDesigner® Installation* or visit *http://www.sybase.com/sysam*).

The following types of license are available:

- *Standalone seat (local)* – installed on a specific machine and cannot be used on other machines. PowerDesigner reviews the local license file and authorizes only the use of modules for which a valid license is present. This type of license is convenient for smaller

teams as it does not require a license server, but does not allow you to centralize license management.

- *Standalone seat (served)* – assigned to a specific machine but served from a license server. You must be connected to the license server to obtain the license when you start PowerDesigner, and connect to the server at least once every 30 days in order to retain it. This type of license is convenient for larger teams, when you want the security of centralized license activation and management.

Note: A 15 day grace period is granted for users with standalone licenses when the license server cannot be contacted or the license file cannot be read. There is no grace period for floating licenses.

- *Floating license (served)* – licenses can be requested from the license server by any machine, and are returned after use, or after 3 hours of inactivity. This type of license is convenient for teams of all sizes where usage patterns allow the sharing of licenses.

  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 (see *Obtaining a Mobile License* on page 33) for up to 30 days.

License Management Wizard

The License Management Wizard provides the same licensing options as the PowerDesigner setup program. Note that you do not need to be administrator on your machine to access the License Management Wizard.

You may need to access the 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 standalone local license) from Sybase.
- *Obtaining a trial of a new package* – You must rerun setup to install the additional modules before launching the wizard.
- *Moving from a standalone local to a standalone served license or from a standalone to a floating license* – You must be able to connect to the license server to make this change.

Note: The PowerDesigner plugin for Sybase Workspace is licensed with the other components of Workspace, and the following procedure is not available for Workspace users. For more information about licensing in Workspace, see the Sysam documentation provided with Workspace.

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

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.
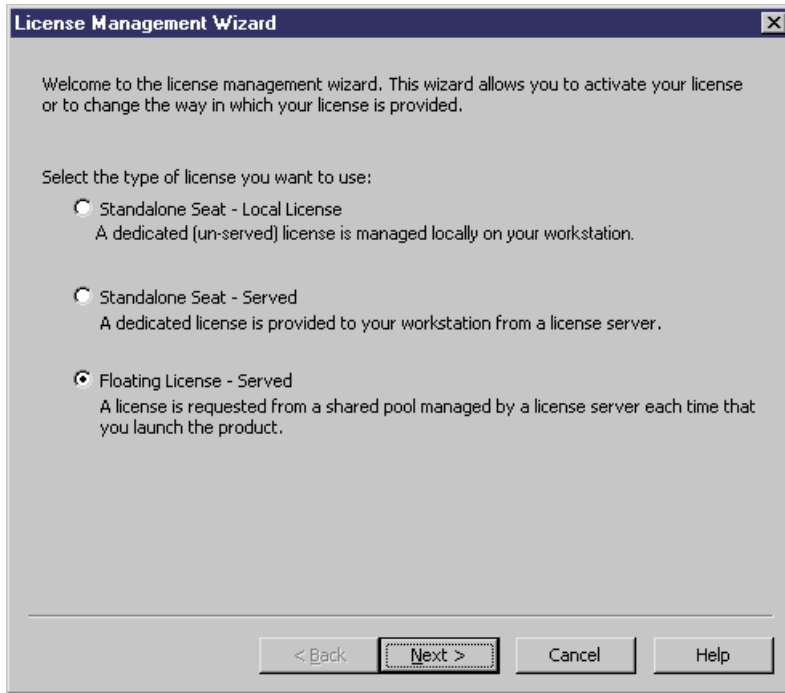
2. Select a type of license (see *License Management* on page 30) and then click **Next**:
   - [local licenses] You must provide a valid license key. Follow the instructions to load the key and then click **Finish** to validate your license and exit the wizard.
   - [if you select to use a served license] Specify the name of the license server (and if your administrator has specified one, a port number), and click **Next**

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 a standalone local license, the license file information is added to the license folder as a new license file.
   - If you have selected a served option (standalone or floating) a new license file is created to store the server name and port number.

- If you have selected a different package, the new models of the package will be available the next time you start PowerDesigner.

Obtaining a Mobile License

When you are using a floating 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.

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**.

Troubleshooting Licensing Problems

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

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 *License Management Wizard* on page 31). Re-load your local license key or select the appropriate served license type.

PowerDesigner Documentation

Full documentation is provided in the form of context-sensitive help (available by pressing **F1**) and as PDFs.

The following manuals are provided with your PowerDesigner installation:

- *New Features Summary*
- *Installation Guide*
- *Core Features Guide*
- *Requirements Modeling*
- *Enterprise Architecture Modeling*
- *Business Process Modeling*
- *Data Modeling*
- *Data Movement Modeling*
- *Object-Oriented Modeling*
- *XML Modeling*
- *Customizing and Extending PowerDesigner*

This documentation is available:

- as online help by pressing **F1**
- as PDFs by selecting **Help > Online Documentation**
- for download from *http://sybooks.sybase.com*

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).

Modeling Roles

PowerDesigner provides a unique set of enterprise modeling tools that bring together the standard techniques and notations of business process modeling, data modeling and UML application modeling with other powerful features to assist you in analyzing, designing, building, and maintaining your applications, using software engineering best practices.

The PowerDesigner enterprise modeling solution 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. 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.

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, including business analysts, analysts and designers, database administrators, developers, and testers, each of whom will use a different combination of PowerDesigner components:

- *Business Analysts* – define the architecture of the organization, the business requirements and high-level business flows.

  They may use an *Enterprise Architecture Model (EAM)* for providing a big picture of the organization, for 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.

  They may use a *Requirements Model (RQM)* for defining business requirements to be refined into technical requirements by Analysts and Designers.

  They may also use a *Business Process Model (BPM)* to define the high-level business process flows that describe existing and new systems, and to simulate business processes to reduce time and resource and increase revenue.

- *Data Analysts and Designers* - will map technical requirements to business requirements. Going deeper into the analysis, you can define Use Cases (using an OOM, see below), and map them to requirements. You can write functional specifications and define more precisely the nature and details of each process, the application and its data structure. You will use a BPM, and additionally a *Conceptual Data Model (CDM)*, which permits real normalized data structures with many-to-many and super/sub-type relationships, and provides a clear view of business data across all systems, making system information accessible to business users, system architects, and business analysts.

- *Database Administrators* use the well-defined data structure to optimize, denormalize, and create the database. You will use a *Physical Data Model (PDM)*, which is a representation of a real database and associated objects 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.

  A PowerDesigner PDM can be used to generate all of the database code for any of the 50 supported RDBMSs. The PDM can be created by reverse engineering from a script or from a live server through a standard ODBC connection. By maintaining a PDM and a CDM, you can ensure that your final implementation exactly matches your system requirements, and that your analysis and design efforts are reflected exactly in your actual systems.

  You may also use a *Logical Data Model (LDM)*, which can act as a bridge between a CDM and a PDM. More technically precise than a CDM, an LDM allows you to resolve many-to-many and super/sub-type relationships, de-normalize your data structures, and define indexes, without specifying a particular RDBMS.

  If you are responsible for database replication, you will also use a *Data Movement Model (DMM)*, which provides a global representation of the replication of information from a source database to one or several remote databases.

- *Developers* will write technical specifications in an RQM, and will model the application, defining object structures and behaviors, and Object/Relational mappings.

  You will use an *Object-Oriented Model (OOM)*, which uses standard UML diagrams and notation to represent your objects and their interactions. It can be reverse-engineered from,

and used to generate code for Java, .NET and many other languages. Close integration with your BPM, CDM, and PDM can greatly simplify the maintenance and development of your system.

You may also use an *XML Model (XSM)* to graphically model the complex structure of an XML file. Its diagram and tree views give you a global and schematic view of all the document elements, and this type of model can be used to generate DTDs, and XSDs directly from a PDM or OOM.

- *Team Leaders* will have an interest in all the models, and will want to ensure that all the requirements, design objects and documents are linked together via traceability links to allow for impact analysis and change management.

  You will establish a *PowerDesigner Enterprise Repository* as a central point of storage. The repository supports metadata sharing, versioning, impact analysis, and reporting for models and other system documents, has a robust security model, and supports true enterprise scalability from a single repository instance. You may also deploy a library containing a glossary and other enterprise models and documents to ensure that all the team are using the same terminology and concepts.

  You can ensure that up-to-date and accurate documentation is produced and widely available. The full-featured *Report Editor* allows you to automate the production of detailed reports (in RTF and HTML formats) on any or all of the components of your system for sharing design information within the project team and across the whole company.

  A *Free Model (FEM)* can be used to create diagrams to explain the architecture of your system and applications, the use-case scenarios of applications, flowcharts, and other graphics.

- *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.

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 Modeling* 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 Modeling* 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 Modeling* 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 Modeling* 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 Modeling* guide where you will learn how to:
  - Define databases, replication processes, event scripts, and other replication objects
  - Reverse engineer from and generate to the 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 Modeling* 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 Modeling* 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

CHAPTER 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.

You can create objects from the Browser (see *Creating Objects from the Browser* on page 96) and drag, drop, and copy objects in the Browser, or between the Browser and the Canvas (see *Moving, Copying, and Deleting Objects* on page 130).

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**.

A typical hierarchy of objects in the Browser may include:

*   *Workspace* - The root of every Browser tree, which contains and organizes all your models and supporting documents (see *Workspaces* on page 46).
*   *Library* - A folder containing enterprise models that are pushed automatically to all users when they connect to the repository (see *The Library* on page 43)
*   *Branches* - Allow concurrent development of multiple versions of documents (see *Branching Version Trees* on page 325).
*   *Projects* – Act as containers for models and other documents, allowing you to save them as a single unit in the repository (see *Chapter 3, Projects and Frameworks* on page 53).
*   *Folders* - Organize models and other files (see *Folders* on page 46).
*   *Models* - Are the basic design unit in PowerDesigner and contain one or more diagrams and any number of model objects (see *Models* on page 47).
*   *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* on page 48).
*   *Diagrams* - Graphical views of model objects (see *Diagrams* on page 175).
*   *Model objects* – Include all items belonging to a model (see *Chapter 4, Objects* on page 95).
*   *Reports* – Are generated to document your models (see *Chapter 6, Reports* on page 225).

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:
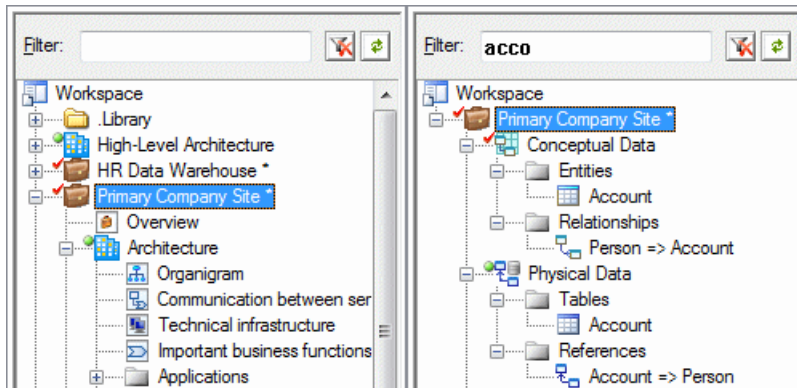
The Glossary

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* on page 560), dependency matrices (see *Dependency Matrices* on page 186), and reports (see *Chapter 6, Reports* on page 225) - to present glossary terms in various ways.
*   Business Rules - see *Business Rules* on page 142.

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:
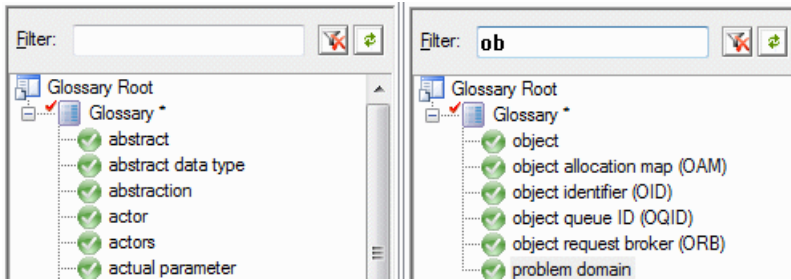
Note: You can only edit the glossary if you have Write permission on the glossary model (see *Deploying an Enterprise Glossary* on page 555).

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.

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.

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.
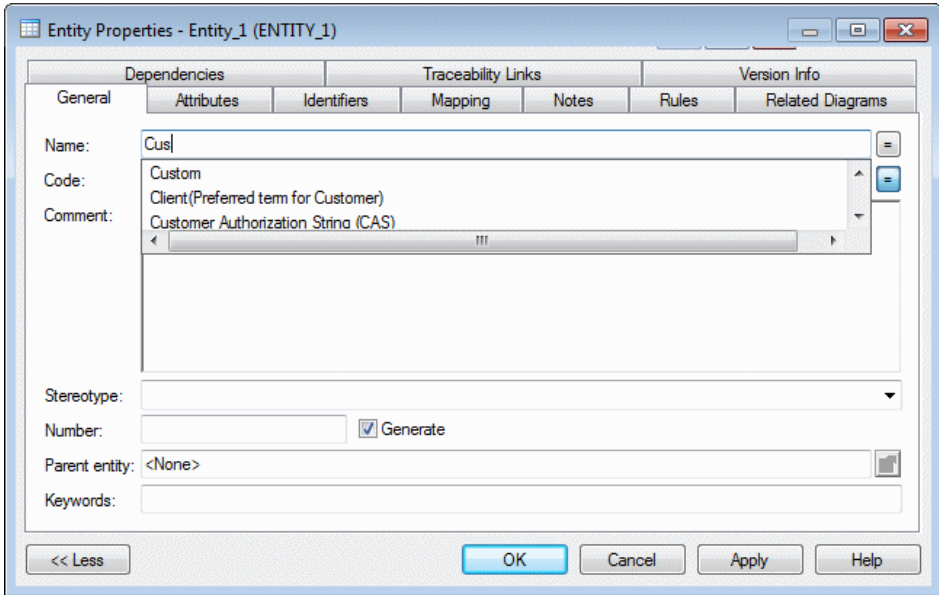
If you click the object's **Dependencies** tab, the link to the chosen term is listed there:
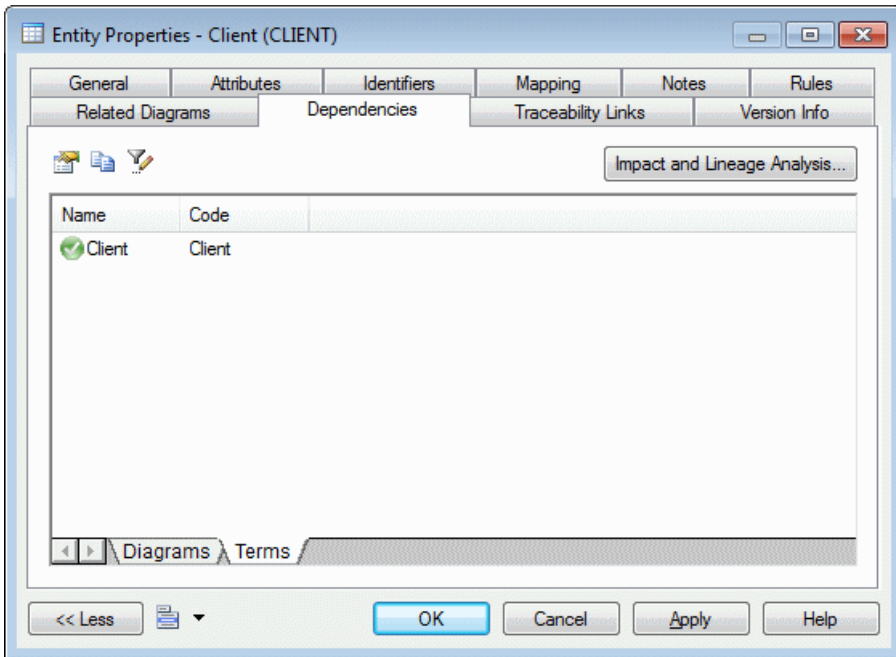
3. [optional] Perform a model check to see which objects have names containing terms that are not included in the glossary (see *Checking Glossary Compliance* on page 153).

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* on page 210.
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.

Note: Only an administrator or other user with Write permission on the glossary can edit terms (see *Creating Terms, Synonyms, and Related Terms* on page 555).

The Library

In environments with a repository, administrators can deploy an enterprise library containing reference models, resource files, and other documents to share generic modeling objects between projects. 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:
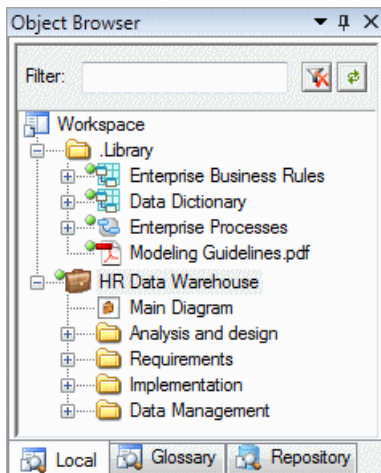
You can rename your local library folder, move it in the Browser tree or detach it from your workspace. To ensure that you have the very latest version of library documents, select **Repository > Synchronize Library**.

The library documents and resource files are automatically updated whenever you connect to the repository (see *Connecting to a Repository* on page 17) and you can control which ones will be pushed to your workspace.

Resource files, such as DBMS or language definition files and extensions, do not appear in the Browser, but appear automatically in the new model dialog, resource file lists, and anywhere in the interface where a resource file can be selected.

Note: By default, library documents are checked out to a subfolder inside `My Documents/ PowerDesigner Libraries`, controlled by the `_SHARED` named path (see *Defining Named Paths* on page 400). 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* on page 563).

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.

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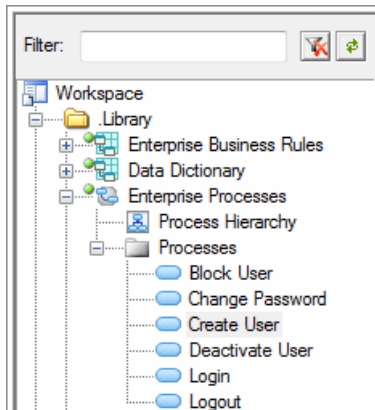
2. Right-click on the object in the library model and drag and drop it into your model diagram where you want to reuse it (or onto a model or package in the Browser).

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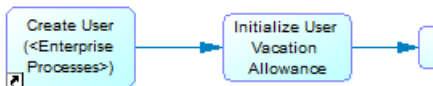
If the library object subsequently changes, any updates will cascade automatically to your model.

Note: For detailed information about working with shortcuts and replicas, see *Chapter 15, Shortcuts and Replicas* on page 451.

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* on page 310.

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.

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.

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**.

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.

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* on page 6). For a list of available types of models, see *PowerDesigner Models and Diagrams* on page 10. 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.

Note: By default, when you open a workspace, its models are closed and bear a red dot in their bottom-left corners. 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.

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.

Packages

A package is a piece of a model. When working with large models, you can split them into smaller subdivisions in order to avoid manipulating the entire model at once. Packages can be used to organize your model into different tasks and subject areas, and to assign parts of it to different development teams.

You can create as many packages as you need in a model. The name of each package must be unique in the model. You can decompose a package into other packages, and there is no limitation to the decomposition depth.

Packages can contain the same kinds of items as models:

- Model objects
- Other packages
- Diagrams, in order to have different views of the contents of the package. Each package has a default diagram

In the example below, a package called "Lending Records" (with a diagram of the same name) has been added to the model "Library OOM":
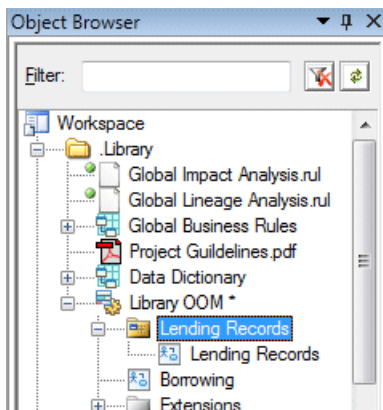
Note that you cannot save a package individually. When you save the model you also save all the packages it contains.

Creating a Package

You can create package from the Toolbox, Browser, or **Model** menu.

- Use the package tool in the diagram Toolbox
- Select **Model > Packages** to access the List of Packages, and click the Add a Row tool
- Right-click the model or package in the Browser and select **New > Package**

When you create a package in models with multiple kinds of diagrams, you may be required to specify the type of diagram to create in the package.

For general information about creating objects, see *Chapter 4, Objects* on page 95.

Package Properties

To view or edit a package'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.

Note: The same properties (except where noted) are also available for project folders.

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.
Default diagram	[package only] Specifies the diagram to display by default when you open the package.
Use parent namespace	[package only] Option that defines the package as being the area in which the name of an object must be unique in order to be used.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

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.

- For the CDM, PDM, DMM, and FEM - the entire model is, by default, a single namespace, and all the packages use their parent namespace. PowerDesigner applies checks on uniqueness at the model level.
- For the OOM, RQM and BPM – each package is, by default, a separate namespace. PowerDesigner applies checks on uniqueness at the package level.

Depending on the type of model where you create a new package, the Use Parent Namespace check box is selected in the package property sheet.

Note: The XML model does not support packages. For more information on namespaces in the XSM, see *XML Modeling*.

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:
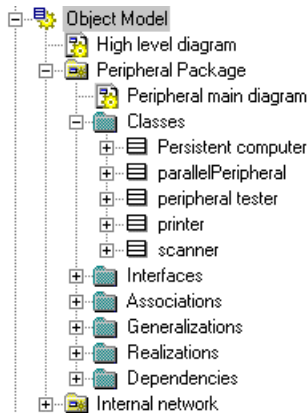
If you were to select the Use Parent Namespace check box in the property sheet of Peripheral Package, the internal names of objects in this package would no longer be prefixed by the name of the package. The parent of the folder, in this case the model itself, becomes the namespace, and you could not create a Printer class in the Internal Network package, because PowerDesigner would verify its uniqueness in the entire model.

1. Open the property sheet of the package.
2. Select or clear the Use Parent Namespace check box.
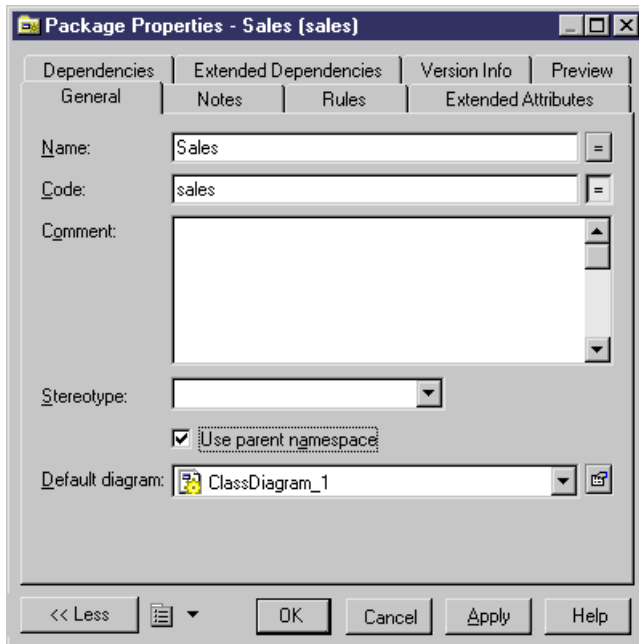
3. Click OK to return to the model diagram.

CHAPTER 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* on page 55), which show the connections between models and other documents:
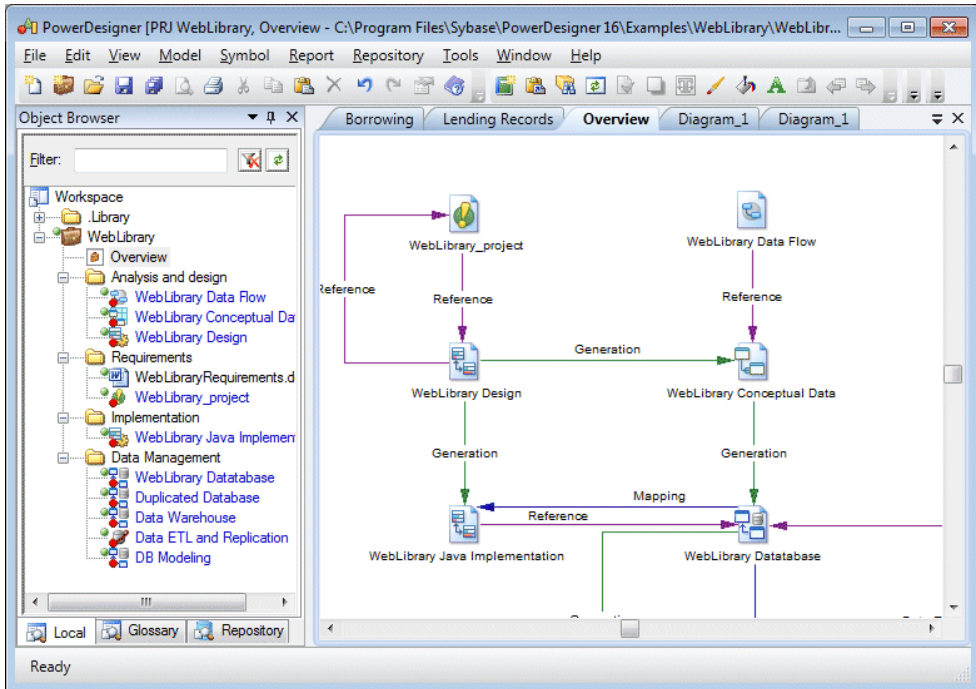
Your project can also contain one or more framework diagrams and/or a framework matrix, which direct how your project must be modeled and list the documents that are needed (see *Completing Framework Diagrams and Matrices* on page 62).

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* on page 84).

Note: Projects and framework diagrams and matrices are only available with Enterprise versions of PowerDesigner.

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.

1. Select **File > New Project** to open the New Project dialog box.
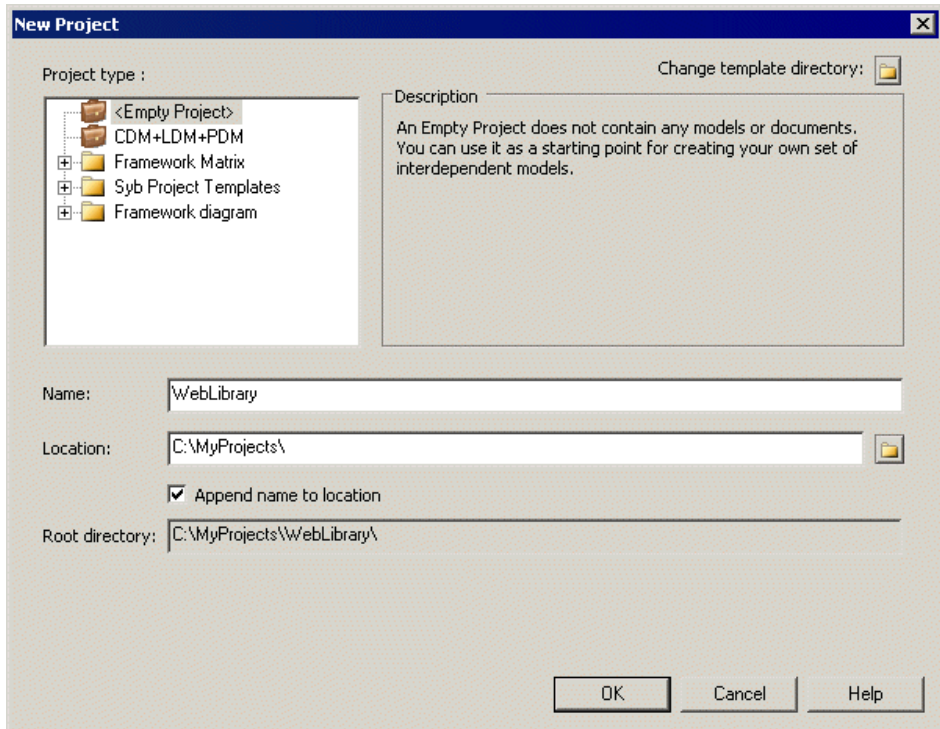
2. Select a type of project in the tree. You can choose to create:

- An empty project.
- A project based on a template (see *Project Templates* on page 84). 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 *Building a Project* on page 55 and *Completing Framework Diagrams and Matrices* on page 62).

Building a Project

You build your project by creating (or adding existing) models and files. 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.

In order to profit from the convenience of the project as a container, you should create (or place) all the associated models and files inside the project directory. However, you can also link to files outside the project directory. Such files are listed under the project node in the Browser, but display small shortcut icons on their symbols to indicate that they are located outside the project folder.

In the following example, Object Model is inside the project and Physical Model is outside the project:
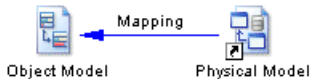
You can, at any time, right-click a document in the Browser or its symbol in the diagram, and select Move to Project Directory to move it inside the project.

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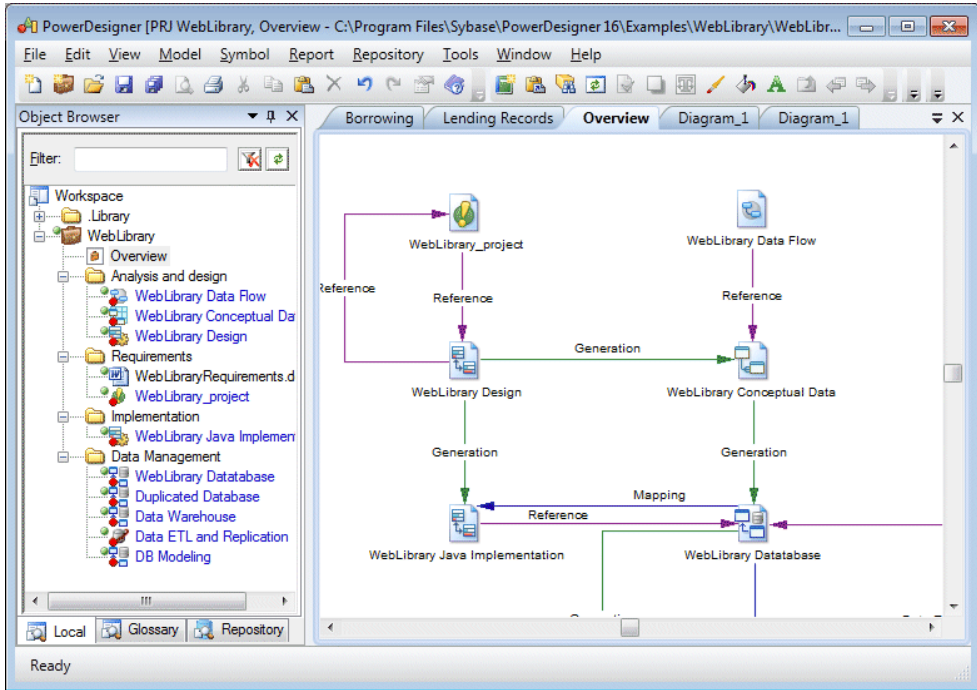
The following tools are available in the Toolbox of a project diagram:

Tool	Description
	New Model – Creates a PowerDesigner model (PDM, OOM, LDM etc.). For more information, see *Creating Models in a Project* on page 56.
	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.). For more information, see *Adding existing model and file documents to a project* on page 57.
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* on page 59).

Creating Models in a Project

You can create new models in your project using the **New Model** tool or from the project contextual menu.

• Click the **New Model** tool in the Toolbox, click in the diagram to open the New Model dialog box, enter a model name and any appropriate information, and then click **OK**.

- Right-click a project or a project folder, select **New > Model** to open the New Model dialog box, enter a model name and any appropriate information, and then click **OK**.

Adding Existing Model and File Documents to a Project

You can add existing model or file documents to your project using the Toolbox, Browser, or drag and drop.

- Click the **Add Project Document** tool in the Toolbox, click in the diagram to open a standard Open dialog box, browse to and select one or more documents in your file system, and then click **Open**.
- Drag and drop one or more documents from:
  - The file system to the Browser or to the project diagram. If a folder has sub-folders, the hierarchy is preserved when it is added to the project. However, since projects are not synchronized with the file system, if you subsequently add files to these folders outside PowerDesigner, they will not automatically be displayed in the Browser.
  - A project diagram to another project diagram.
  - The Browser to the project diagram.
- Right-click a project or a project folder in the Browser, and select **Add** to open a standard Open dialog box, browse to and select one or more documents, and then click **Open**.

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.

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.

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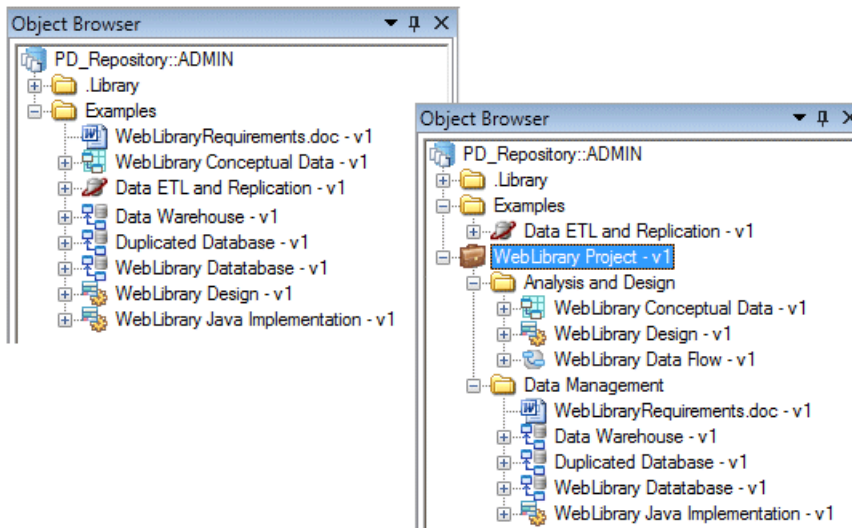
Opening Project Documents

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.

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.

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* on page 446).
- Mapping – displays the mapping links between models in the Mapping Editor (see *The Mapping Editor* on page 476).
- 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:
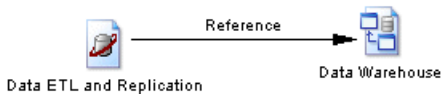
Rebuilding Dependency Links

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.

Rebuilding dependency links can be useful to update:

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

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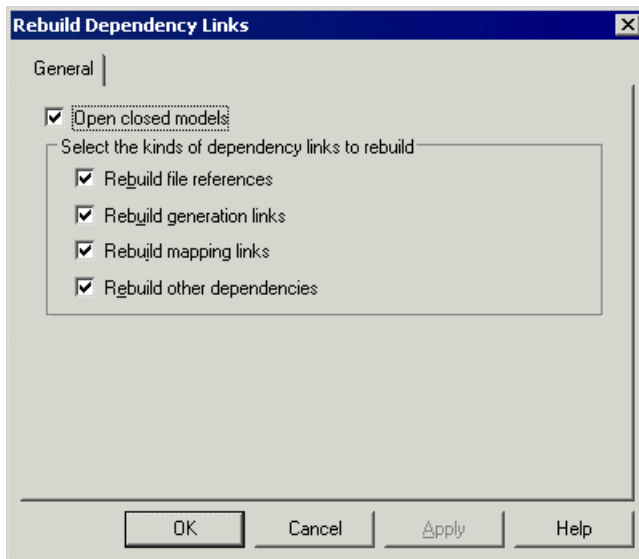
2. [optional] Clear the Open closed models check box to rebuild only those links in a limited number of models.
3. Select the check boxes that correspond to the dependency links you want to rebuild.
4. Click OK to close the dialog box and return to the diagram. Any missing links are updated in the diagram.

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).

Note: For information about changing the format of symbols, see *Format Display Preferences* on page 217. Note that only framework architects (see *Designing Framework Diagrams and Matrices* on page 69), are allowed to modify the framework diagram and matrix format display preferences (see *Symbol Format Properties* on page 200).
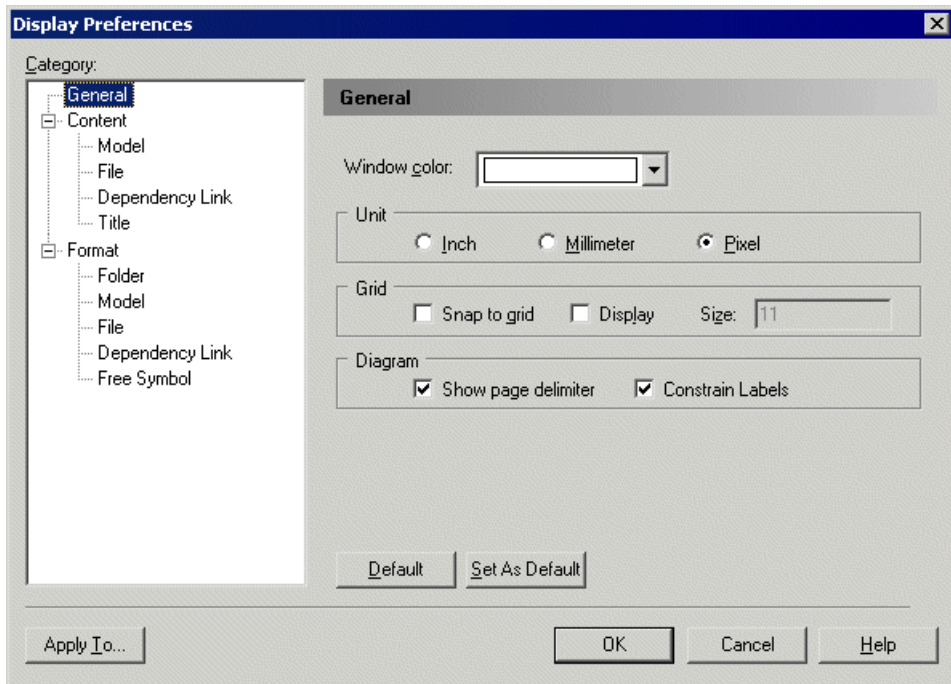
To set display preferences for a particular object, select it under the Content category. The following table lists the display preferences available. The objects available to be customized in the Display Preferences window depend upon the current diagram type.

Preference	Description
Code	Displays the code of the object.
Comment	Displays the comment of the object.

Preference	Description
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.

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**.

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

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.

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 unique-ness	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

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.
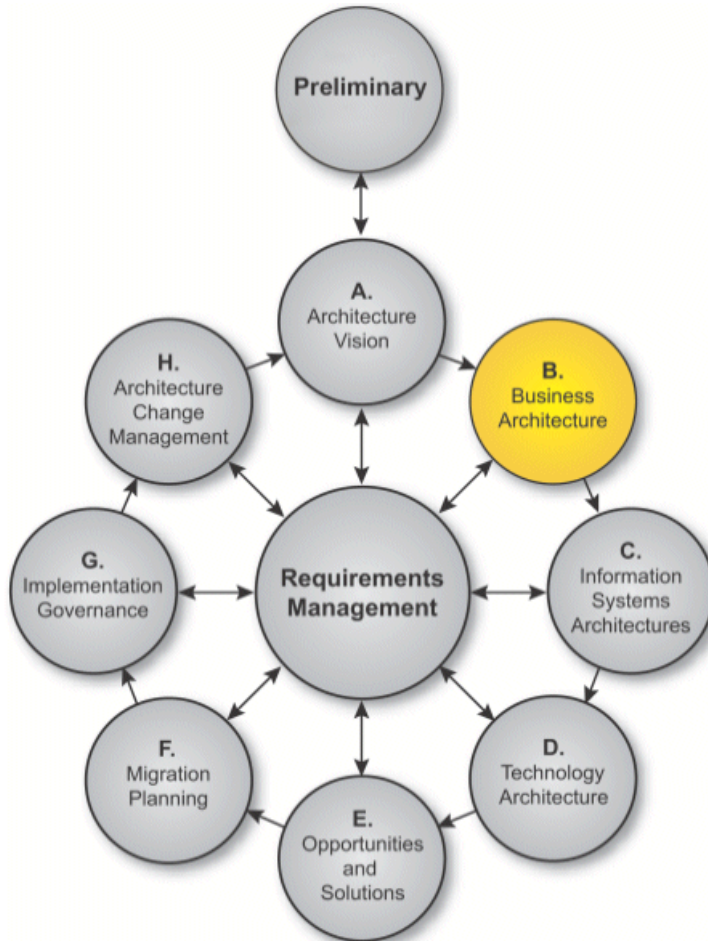
- Framework matrix - lets you model a framework in a matrix, in which columns usually represent various aspects of the enterprise that can be described or modeled, and the rows represent various viewpoints from which the aspects can be described. Each cell formed by the intersection of a column and a 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* on page 69). 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* on page 186).
- • 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).

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* on page 68).
- • Right-click a node or cell – to opens contextual menu (see *Working with the Framework Node and Cell Contextual Menu* on page 69).
- • Double-click a framework matrix element [framework architects only] – to open the property sheet of the element (see *Framework Matrix Properties* on page 89).

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.

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.
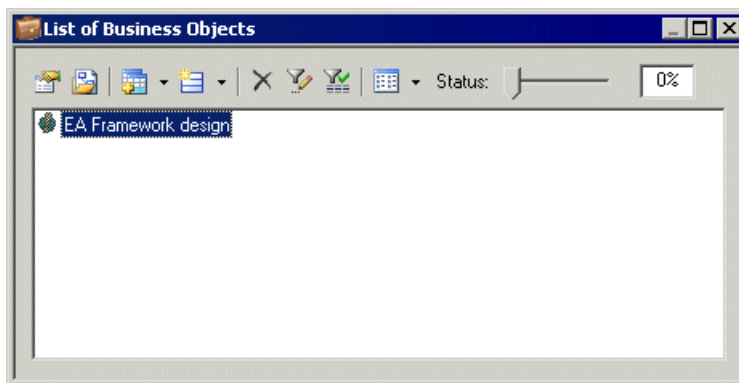
3. Click one of the following tools:

- • **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 a Model* on page 433).
- 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* on page 186).
- 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.

> **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.

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.

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* on page 72.

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.

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* on page 74.

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.

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.
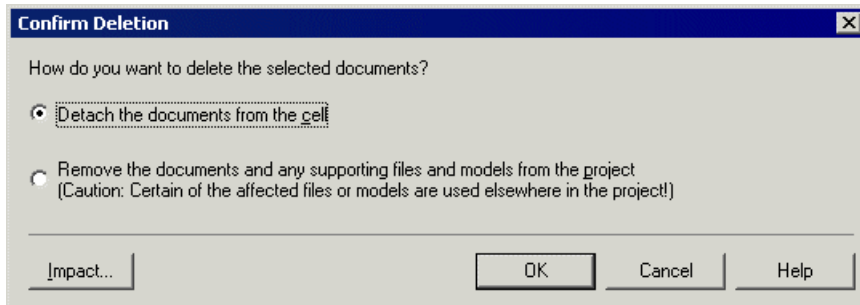
2. Choose one of the following delete options:

   - 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.

Working with Floating Lists

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

Note: Floating lists are not available when you select the Documents display preference (see *Project Display Preferences* on page 60).

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.
	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 a Framework Node and Cell* on page 67.
	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* on page 120).
	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.

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* on page 84):

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.

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.
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.

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.

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* on page 70).

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* on page 71).

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 (see *Project and Framework Templates* on page 84).

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.

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* on page 69).

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.

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**.

   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* on page 89).

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 Area**s 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* on page 71).

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* on page 84).

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.

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* on page 90).

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* on page 92).

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* on page 72).

   • **Diagram** – a PowerDesigner model diagram (see *Example: Specifying a Diagram Action* on page 73).

   • **List** – a list of PowerDesigner model objects (see *Example: Specifying a List Action* on page 74).

   • **Model Generation** –a PowerDesigner model generated from another model (see *Example: Specifying a Model Generation Action* on page 76).

   • **Object Generation** –a PowerDesigner model populated with one type of objects generated from another model (see *Example: Specifying an Object Generation Action* on page 77).

   • **File** – an external file (see *Example: Specifying a File Action* on page 78).

- **Matrix** – a PowerDesigner dependency matrix showing links between model objects (see *Example: Specifying a Matrix Action* on page 80).
- **Script** - a VB script that can define any scriptable action (see *Example: Specifying a Script Action* on page 81).

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* on page 65). Note that only fully-defined actions are available for selection.
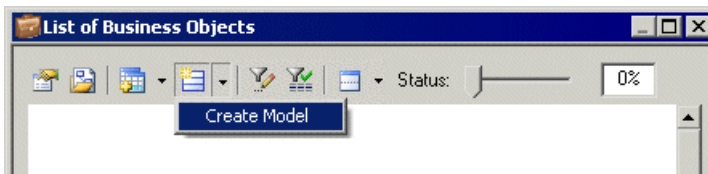
Example: Specifying a Model Action

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.

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.

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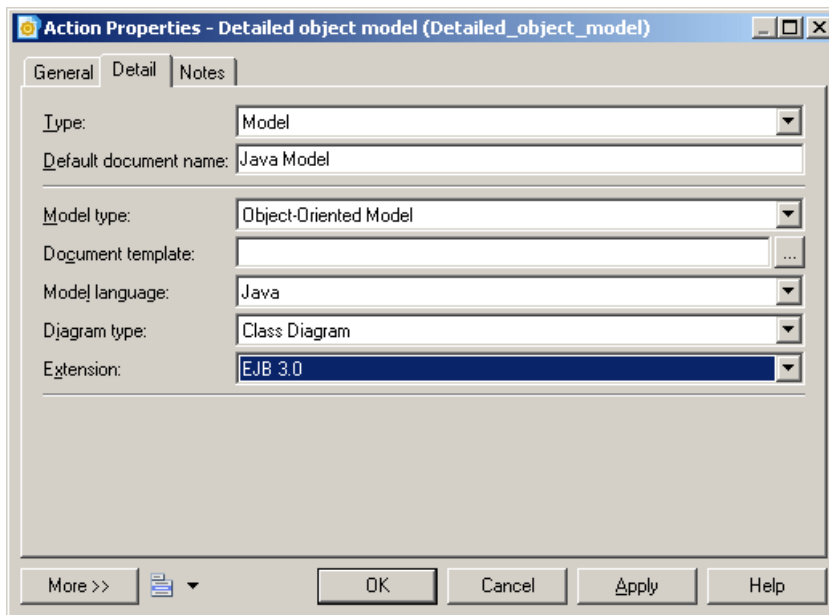
6. Click OK to close the action and node property sheets and return to the framework diagram.

The Model action is now available for use by framework diagram users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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* on page 13.
Diagram type	*Business Process Diagram*
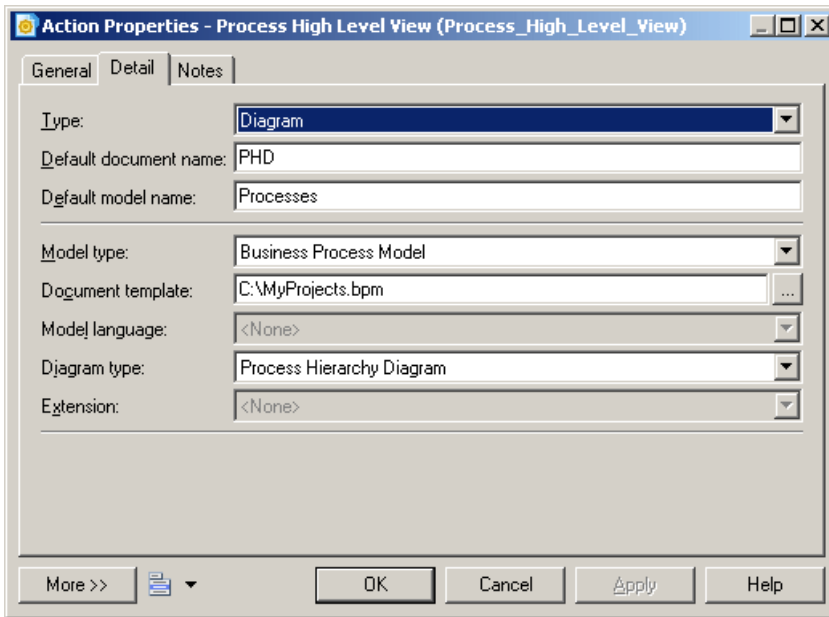
6. Click **OK** to close the action and cell property sheets and return to the framework matrix.

The Diagram action is now available for use by framework matrix users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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>*
List object type	*PDM::Cluster* (click the **Select Object** tool, navigate to Oracle Version 11g, and select the Cluster node)
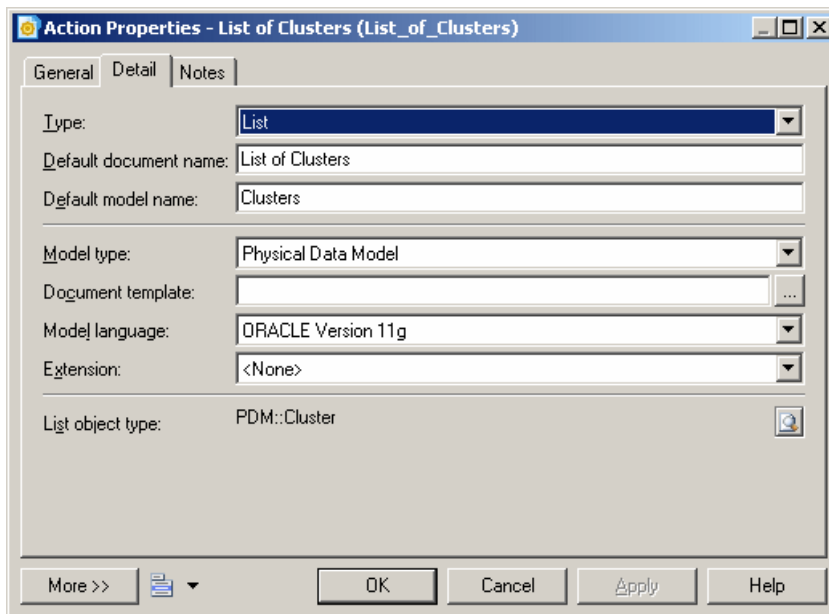
6. Click **OK** to close the action and node property sheets and return to the framework diagram.

The List action is now available for use by framework diagram users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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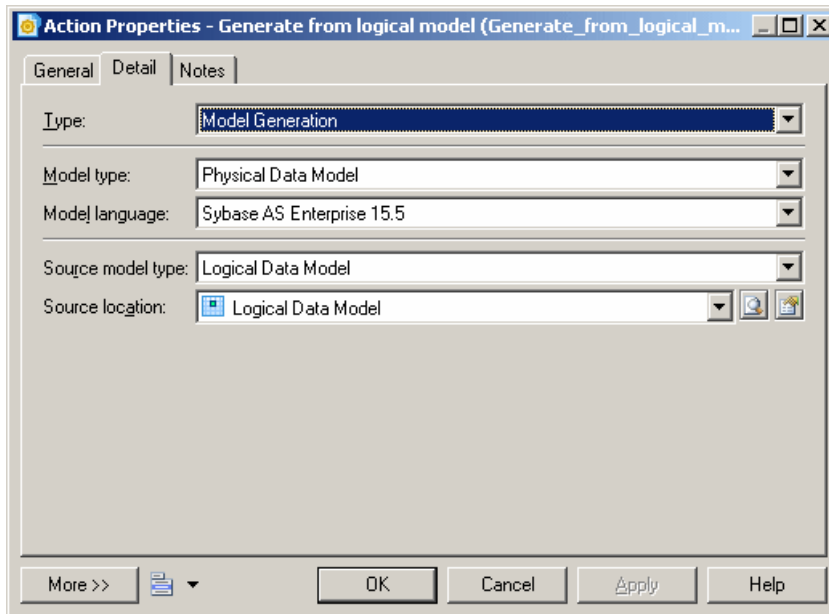
6. Click **OK** to close the action and cell property sheets and return to the framework matrix.

The Model Generation action is now available for use by framework matrix users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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*

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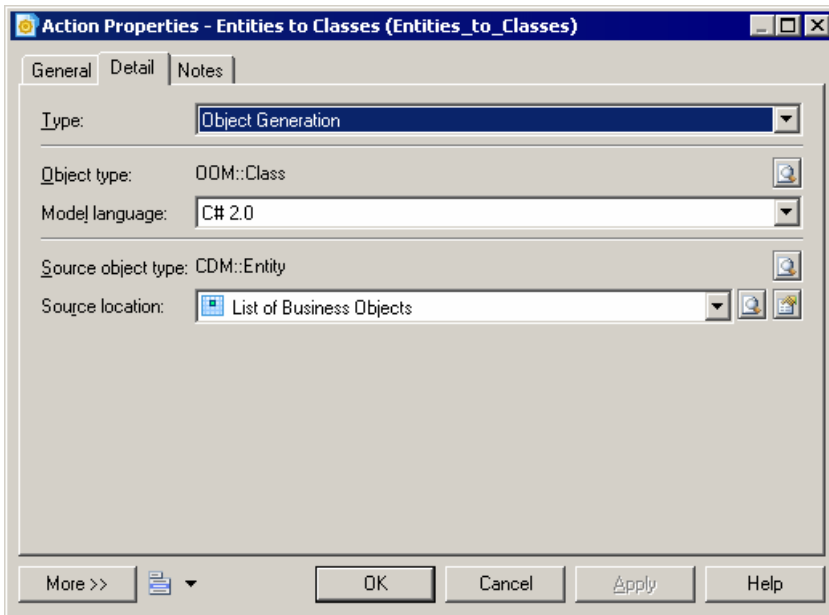
6. Click **OK** to close the action and cell property sheets and return to the framework matrix.

The Object Generation action is now available for use by framework matrix users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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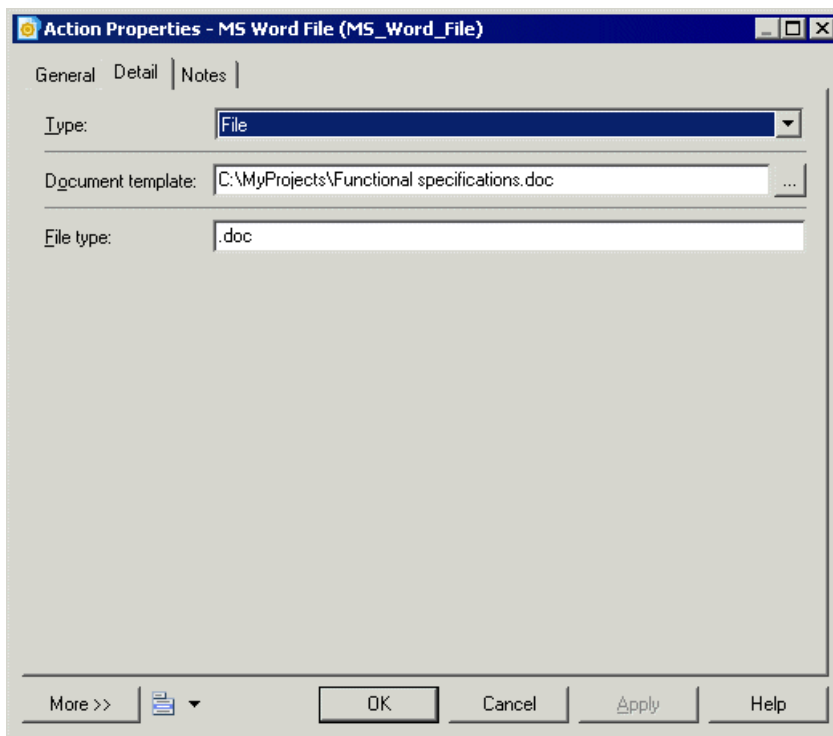
6. Click **OK** to close the action and node property sheets and return to the framework diagram.

   The File action is now available for use by framework diagram users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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.

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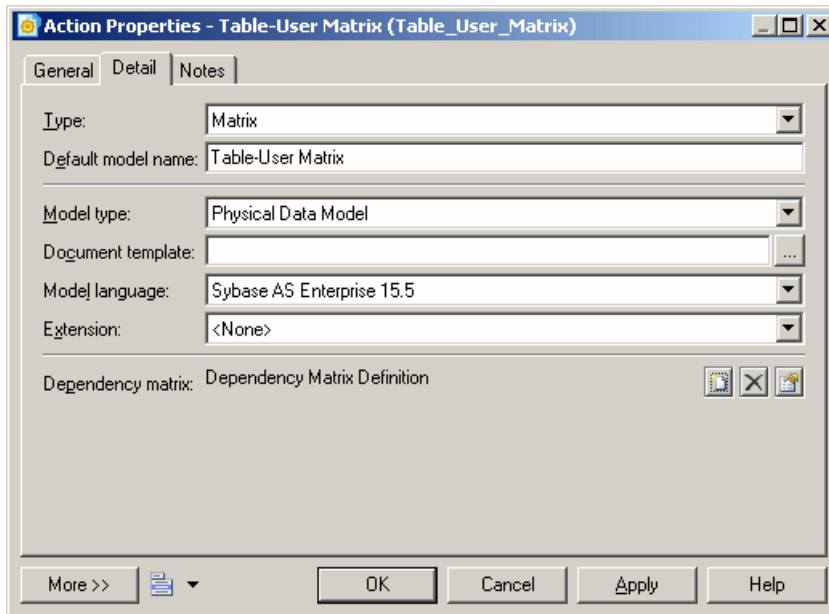
7. Click **OK** to close the action and cell property sheets and return to the framework matrix.

The Matrix action is now available for use by framework matrix users (see *Attaching a Document to a Framework Node and Cell* on page 65).

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.

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
```

>>

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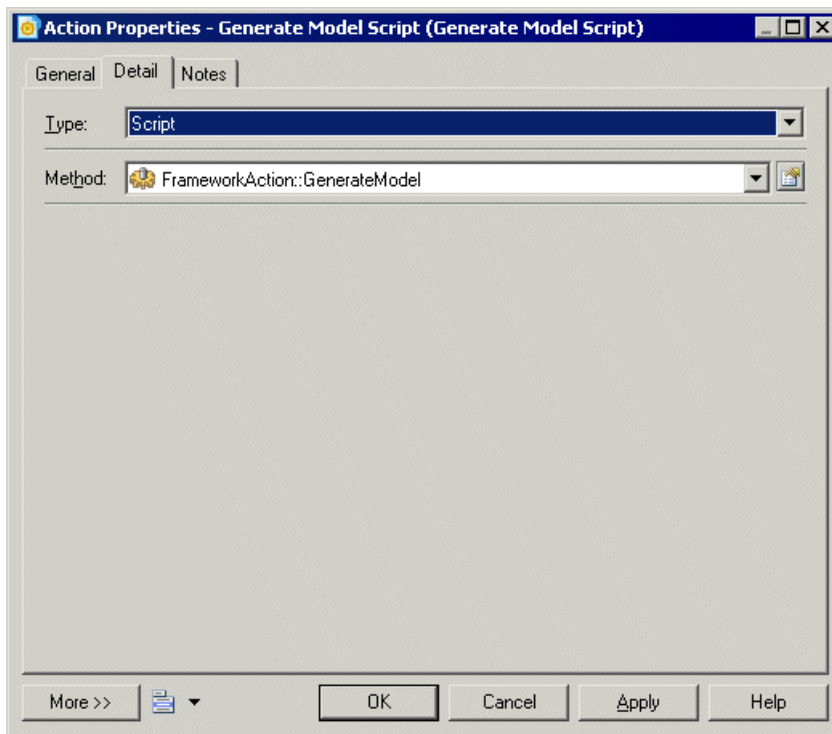
5. Click **OK** to close the action and node property sheets and return to the framework diagram.

The Script action is now available for use by framework diagram users (see *Attaching a Document to a Framework Node and Cell* on page 65).

Styling Framework Nodes and Cells

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

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* on page 200).
Change Image	Opens the Select Image dialog box to let you insert an image in the node or cell (see *Browsing for Images* on page 205).
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* on page 69.

Project and Framework Templates

A project template can provide predefined content, rules, and formatting for your project, and can also include a framework matrix or framework diagram.

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

1. Create a project (see *Creating a Project* on page 54).
2. [optional] Add any appropriate models or files (see *Building a Project* on page 55).
3. [optional] Create one or more framework diagrams (see *Creating a Framework Diagram* on page 69).
4. [optional] Create a framework matrix (see *Creating a Framework Matrix* on page 70).

5. [optional] Specify any appropriate display preferences (see *Project Display Preferences* on page 60).

6. Select **File > Save As Template**

7. [optional] Specify a template directory in which to save the template. By default, the project template directory is selected. You can click the **Change Template Directory** tool to select another location.

8. Select a template category from the list or enter a template category name that will be available for when you create other projects. You can use the \ character to create sub-directories, or right-click the project in the Browser, and select **Save As Template** to open the Template Wizard.
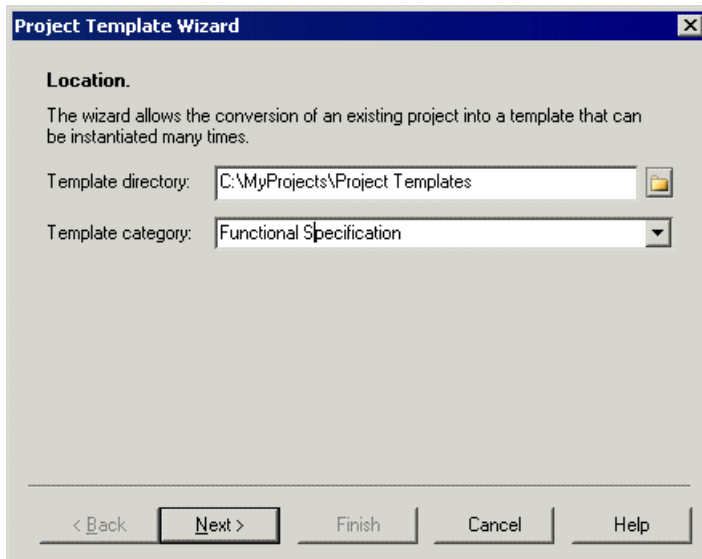
9. Click **Next**, and enter a description for the new template that will be displayed in the New Project Wizard when you create projects with the new template. You can modify a template description by right-clicking a project template in the Browser, and selecting **Edit Template**.

Project Template Wizard ☒

Template Description.

Enter a description for the template to display in the New Project Wizard.

This template is a guide for writing specifications for high-level features.

< <u>B</u>ack   <u>N</u>ext >   Finish   Cancel   Help
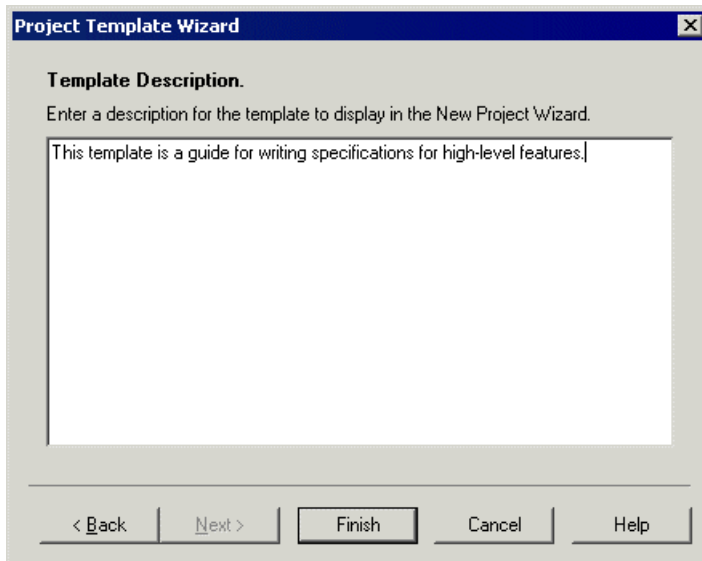
10. Click **Finish** to close the wizard and create the template.

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

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.

11. [optional] Select **New > Project** to open the New Project dialog box. The template you have created is available for selection in the Project type tree.
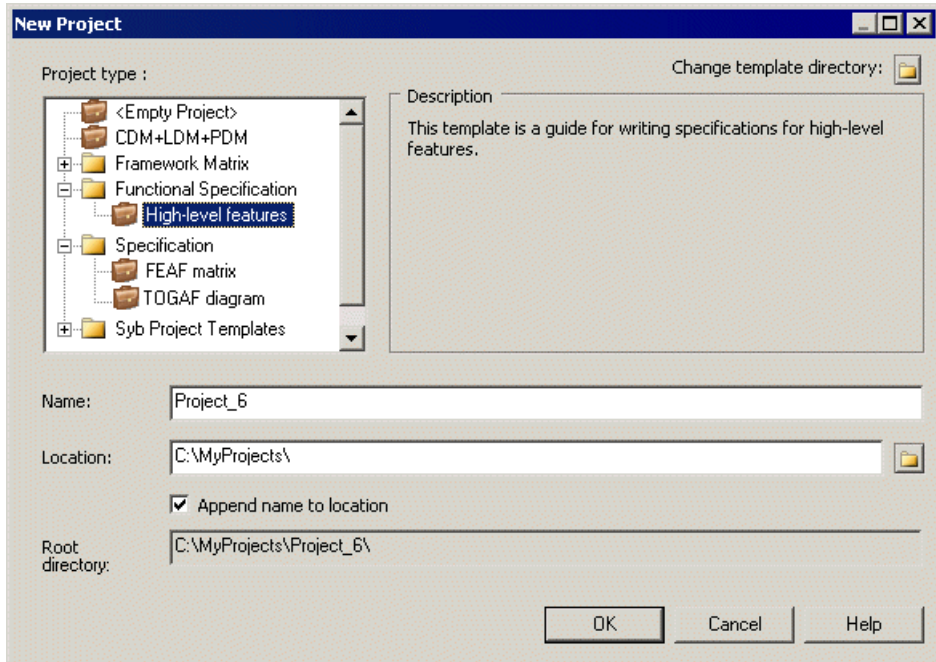
Project and Framework Properties

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

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.

Property	Description
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.

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.

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.

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 Matrices and Diagrams* on page 69.

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* on page 90.
- Columns tab – displays a sortable list of columns, in which you can create, edit or delete columns. See *Framework column and row properties* on page 90.
- 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* on page 90.

- 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* on page 90.

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 Matrices and Diagrams* on page 69).

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.

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* on page 69.

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 de-couple 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 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* on page 92).

- 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 and Cell* on page 65).

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.

Note: For information about enabling the Framework Design Mode, see *Designing Framework Diagrams and Matrices* on page 69.

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 de-couple 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* on page 71).

Property	Description
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.
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.

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 Matrices and Diagrams* on page 69.

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.

CHAPTER 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* on page 175) supports certain specific types of objects, and each object created in a diagram is represented by a symbol (see *Symbols* on page 190). 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* on page 137).

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.

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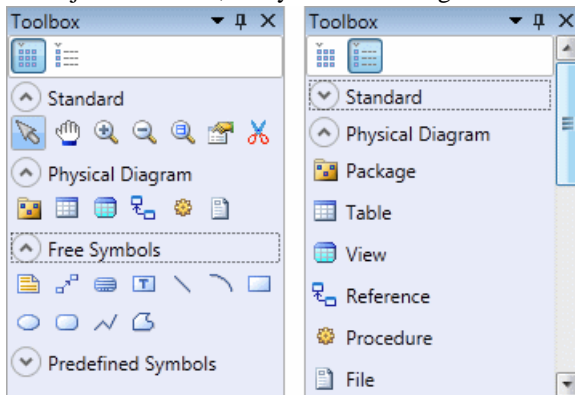
To release the tool, simply select another tool or right-click in free space. To revert to a previously selected tool, hold down **Ctrl** and perform a double right-click.

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 – allows you to select, move and resize the entire diagram. Double-click to display the entire diagram, centered.
	Zoom in
	Zoom out
	Open the diagram of a composite object or packages
	Delete a symbol and, optionally, its associated object.
	Open the property sheet of an object
	Create a package

- *Type* **Diagram** - containing tools specific to the current diagram. For details, see the appropriate modeling guide diagram chapter.
- **Free Symbols** - containing tools for drawing free shapes and links to decorate your diagrams (see *Decorative Symbols* on page 209).
- **Predefined Symbols** - containing tools for drawing standard flowchart shapes.

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* on page 395).

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.

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* on page 97).
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.

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.

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**.

For detailed information about object lists, see *Object Lists* on page 115.

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* on page 107) and must be unique within a namespace (see *Object Namespaces* on page 115). If the enterprise glossary is enabled in the model, then autocompletion and glossary compliance checking is available (see *The Glossary* on page 40).

- **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 by unselecting the equal sign button to the right of this field. You control this synchronization of codes with names with the **Name to Code mirroring** general option (see *Dialog Box General Options* on page 397) and can modify the transformations performed on codes with naming conventions.
- **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* on page 139), and in impact and lineage analysis (see *Chapter 17, Impact and Lineage Analysis* on page 499).

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.

- **Notes** – lists additional information about the object. See *Notes Tab* on page 101.
- **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* on page 142.
- **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* on page 106
- **Dependencies** - lists all the objects that depend on the object. See *Dependencies Tab* on page 106
- **Traceability Links** - lists all the objects on which the object depends. See *Creating Traceability Links* on page 429
- **Requirements** – lists the project requirements that the object is intended to satisfy. By default, this tab does not appear. See *Requirements Tab* on page 105

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 *Chapter 19, Customizing the PowerDesigner Interface* on page 539).

You can control the form of the symbols, the background color, and the text format with display preferences (see *Display Preferences* on page 211). 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.

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 called "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, found object is displayed centered and selected.
Find in Browser	Finds the object in the Browser and highlights it.
Impact Analysis	Opens the Impact Analysis dialog box.
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. For more information, see *Adding a New Attribute to an Object* on page 155.
New List of Associated Objects	Opens the New List dialog, which lets you add new lists of associated objects to your object property sheets. For more information, see *Adding a New List to an Object* on page 158.
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. For more information, see *Managing Extensions* on page 159.
Find in Metamodel Objects Help	Opens the MetaModel Objects Help for the metaclass on which the object is based.

Command	Description
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 throughout 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 Notes, 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, all the tabs except Implementation are checked and are displayed:
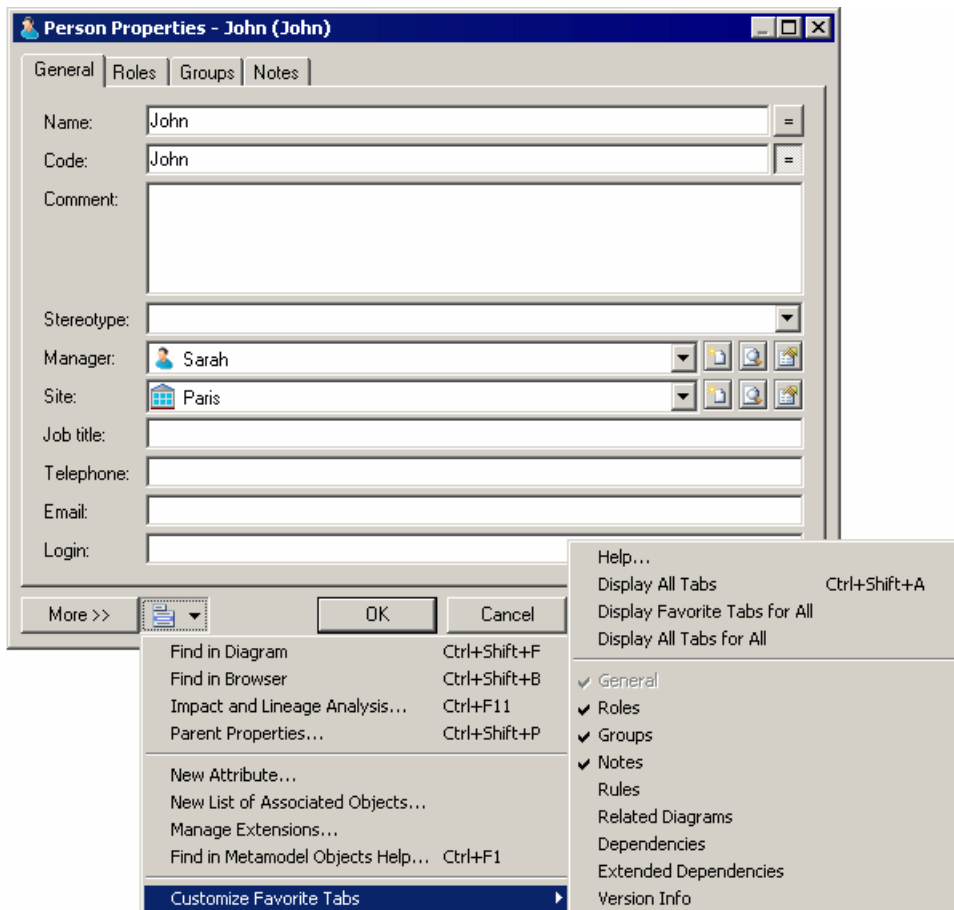
Any changes in the display of tabs are immediately applied to the current property sheet but not to other property sheets that are currently open. The changes become the default setting for any property sheet of the same type.

Note: You can choose to display tabs on one or several rows in property sheets by selecting **Tools > General Options > Dialog**, and selecting the Tabs on one row option or the Tabs on one several rows option in the Property sheets groupbox.

Notes Tab

The Notes tab in an object property sheet contains the sub-tabs Descriptions and Annotations, intended to hold additional information about the object.

- **Descriptions** - in general, includes important information that does not fit into the General tab. 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 in 2002.

- **Annotations** - contains notes regarding 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 are editable directly in the tab with the internal PowerDesigner RTF editor. For information about the tools available, see *Text Editor Tools* on page 210.

You can insert the content of an existing text or RTF file in the RTF editor to use it as a standard for your descriptions or annotations. This can be very helpful to standardize objects notes as you can have a description or annotation RTF file for each object type, and open it when needed.

1. Open the object's property sheet, click the **Notes** tab, and select either the **Description** or **Annotation** sub-tab.

2. Click the **Editor Menu** tool and select **Insert**, browse to the file you want to insert and click **Open**.

   The content of the file is displayed in the field.

Note: To see which objects have notes, open the appropriate object list and review the [**N**]otes checkbox. For information about customizing the display of a list, see *Customizing Object List Columns and Filtering Lists* on page 120.

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.
🖨	**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.

Tools	Description
	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 > Extending Generation and Creating Separate Generation Targets*.
	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.

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 "("
^	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"

Character	Description
?	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.

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.

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

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.

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.

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

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.

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.

Dependencies can be of two types:

- Internal - links to other objects within the model. These dependencies are saved in the model and appear in the **Dependencies** tab of an object property sheet
- External - links to objects in other models such as shortcuts and generation links (see *Creating Links Between Models* on page 428). The display of external dependencies is dependent on the the other models being open in your workspace unless all the linked models are checked into the repository and you have checked your model out with the you check your model out with the **Check out dependencies** option (see *Checkout Parameters* on page 305).
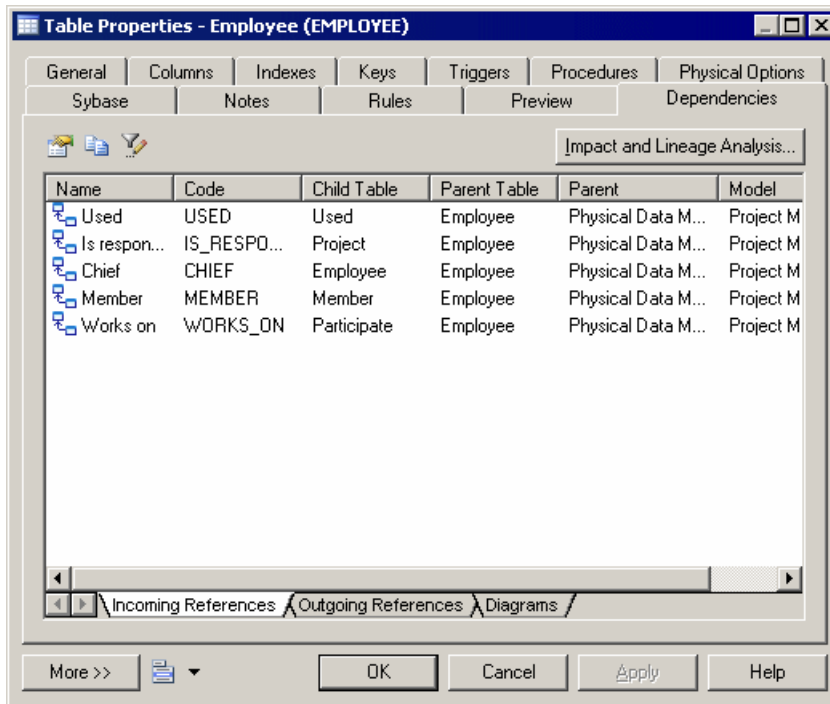
Naming Conventions

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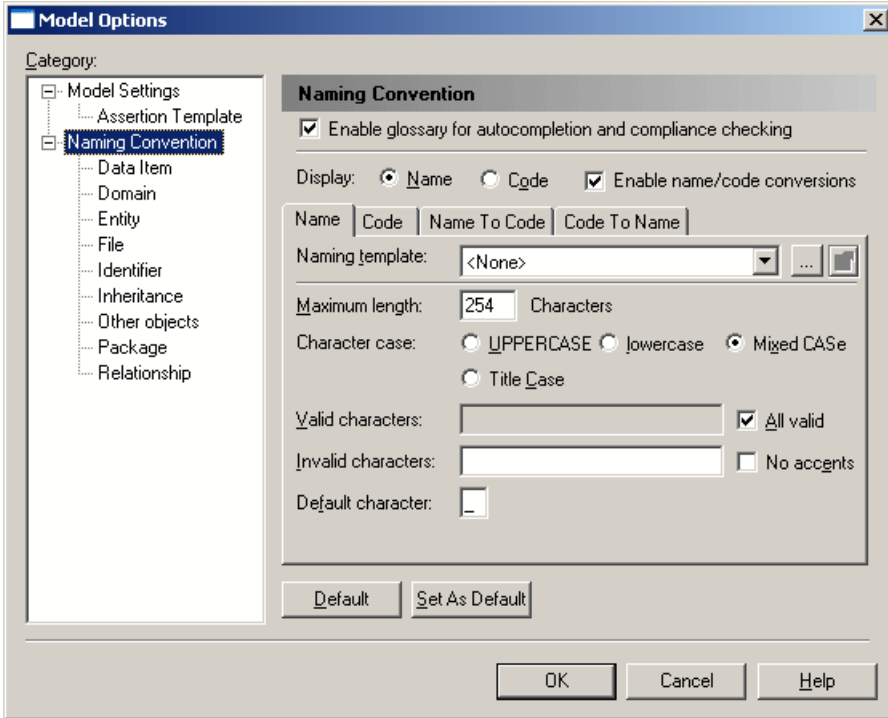
To set naming conventions to control the names and codes of all objects, select **Tools > Model Options** and click the **Naming Convention** node. To set naming conventions for a particular kind of object, select its node under the **Naming Convention** node.

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

Option	Description
Use glossary for autcompletion 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* on page 40), 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* on page 153).
	For information about administering the glossary, see *Deploying an Enterprise Glossary* on page 555.
Display	Specifies whether object names or codes appear in the Browser and on diagram symbols. You can override this setting for diagram symbols by setting the appropriate display preferences (see *Display Preferences* on page 211).

Option	Description
Enable name/ code conversions	By default, when you enter an object name, the object code is autogenerated by applying the naming conventions specified on the **Code** subtab. Select this option to additionally apply the conversion scripts and conversion tables specified on the **Name to Code** tab (see *Name and Code Conversion Scripts* on page 110).
	You can decouple an object's code from its name by clicking to release the = button to the right of the **Code** field in the object property sheet and entering your own code (which will still be subject to the naming conventions). You can autogenerate a name from a code by clicking the = button to the right of the **Name** field.

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* on page 110). 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)
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.

Option	Description
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.

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.

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* on page 107), 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.

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.

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.
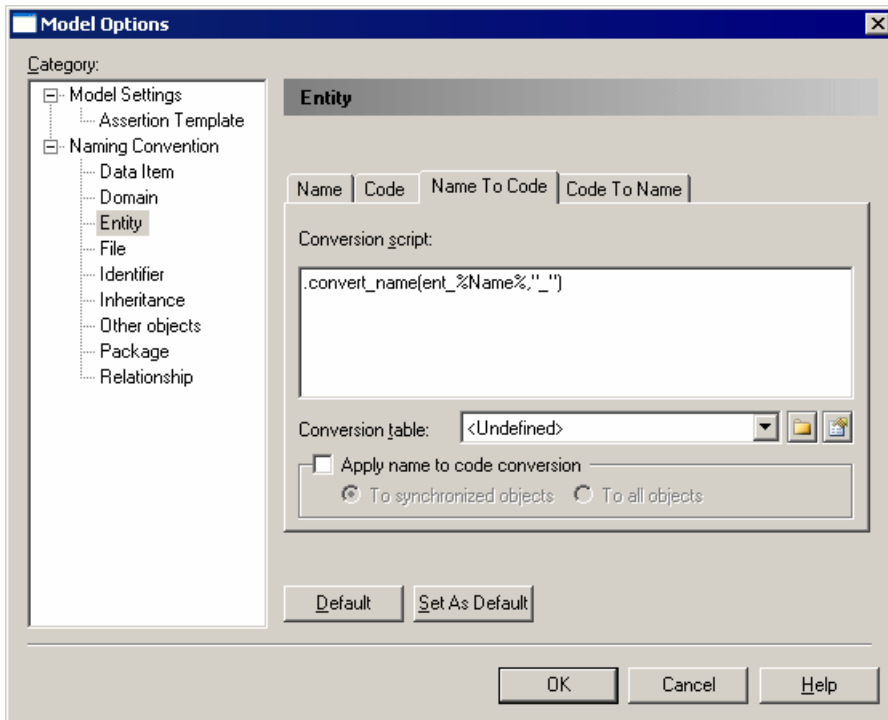
The following options are available on each of these tabs:

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 *Creating a Conversion Table* on page 113). 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* on page 43) 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.

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 Window Detail Tab* on page 435).

Creating a Conversion Table

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 could be converted to cust. To review the example conversion table provided with PowerDesigner navigate to and open **install_dir**/Resource Files/Conversion Tables/stdnames.csv.

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.

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:
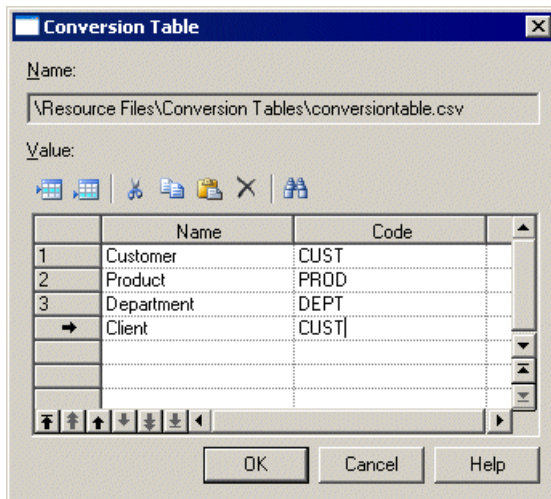
   You can create as many conversion Name-Code pairs as required. The values in these columns are not case-sensitive.

   Note: It may be that several different names are converted 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.

.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"[,"separator_pattern"],case])
```

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

```
.convert_code (expression[,"separator"[,"separator_pattern"]])
```

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 **pattern-separator** is found in the code. For example, "_" (underscore). Type: Text
pattern-separator	[optional] Declaration of the different separators likely to exist in a code, 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

In the following example, the .convert_name macro will convert a class name Corporate Customer into clsCorporateCust if the stdnames conversion table is selected:

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

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
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.

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:
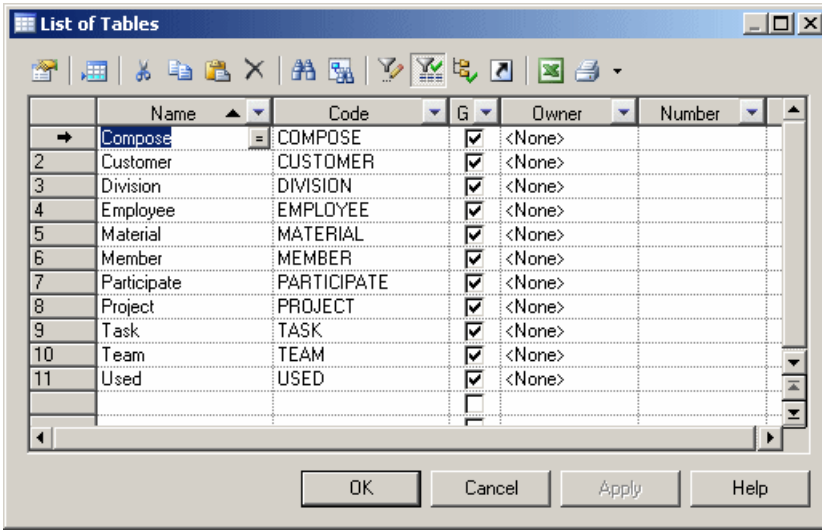
The properties of the listed objects are organized in columns. You can order the list by a particular columns values by clicking on its column header. To filter the values in a column, click the down-arrow on the right of the column header and enter your filter (see *Filtering a List using In-Column Filters* on page 118).

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* on page 120).

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.

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 General Options* on page 397). 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* on page 97).
	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.

Tool	Description
	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* on page 126).
	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* on page 120).
	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.
	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* on page 453) 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* on page 178).
	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:
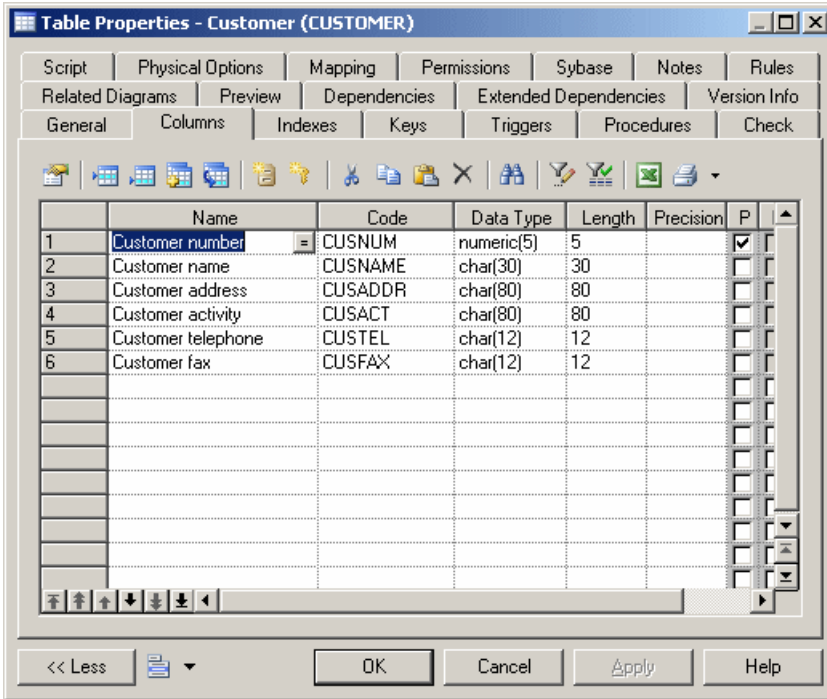
These lists provide the same features as the lists of objects available from the **Model** menu. Depending on the properties of the listed objects, the list may have a natural order that you can control. For example, a list of columns has such an order, and you can move objects in the list by selecting them and then clicking on one of the arrows at the bottom left corner of the list.

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.

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.

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*.
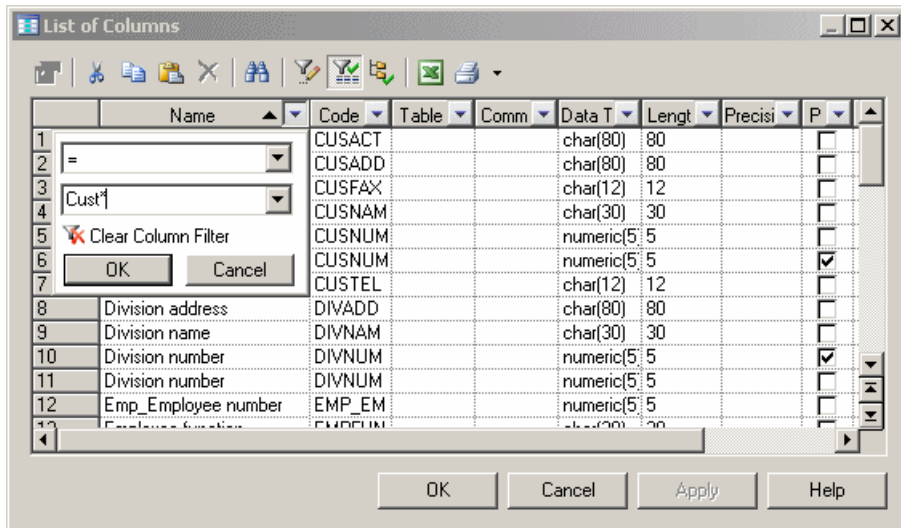
For an explanation of the available operators and the syntax for the expressions, see *Defining a Filter Expression* on page 124.

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:
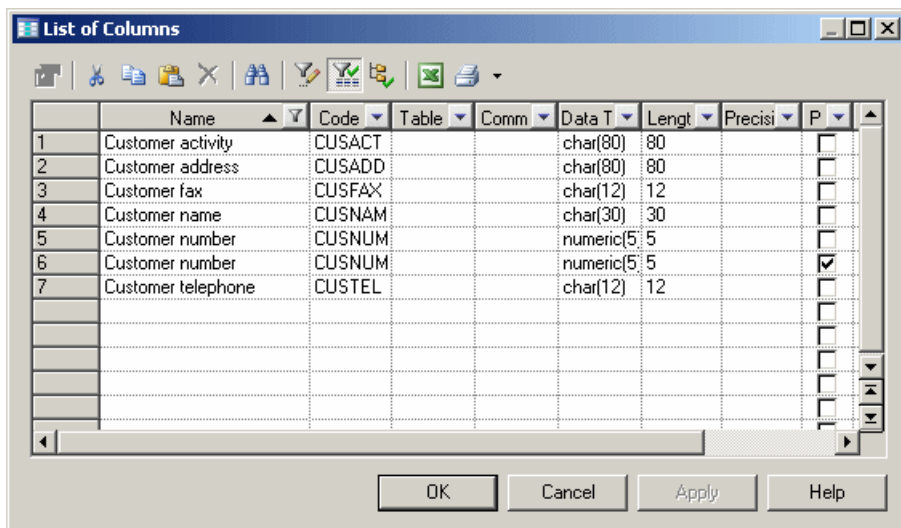
3. [optional] Further restrict the rows displayed by defining a filter on any number of other columns. In this example, both the Name=Cust* filter and a Length<80 filter are applied:
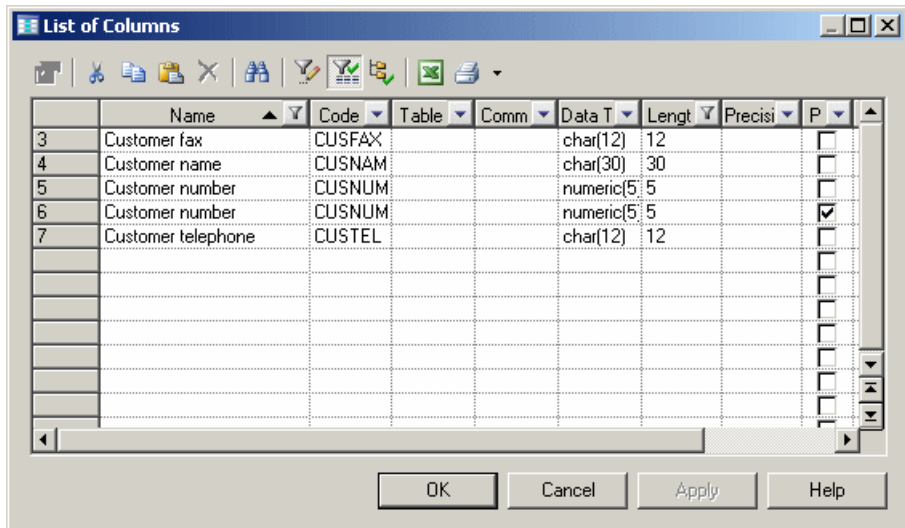
Note: The filter criteria are stored in the Customize Columns and Filter dialog and remain applied until you remove them. To remove all filters, click to release the **Enable/Disable Filter**. To remove a filter from a single column, click the filter button in the column header, and then click the **Clear Column Filter** button.

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.

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.
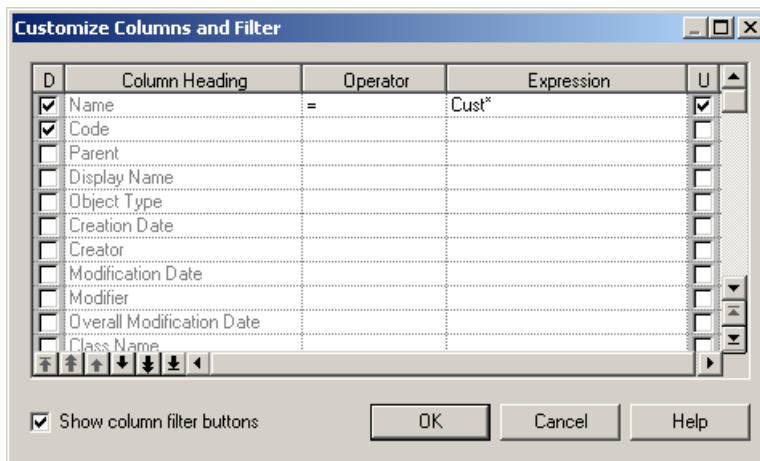
The [**U**]sed checkbox is automatically selected when you enter a filter expression. To disable the expression but keep it in memory, clear the checkbox.

- Enable or disable in-column filtering directly in the list (see *Filtering a List using In-Column Filters* on page 118) 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.

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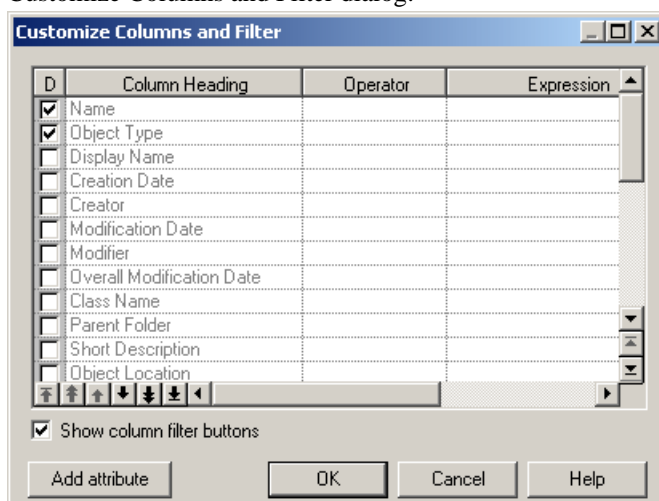
Note: The **Add Attribute** button is only available when the list can contain multiple types of objects.

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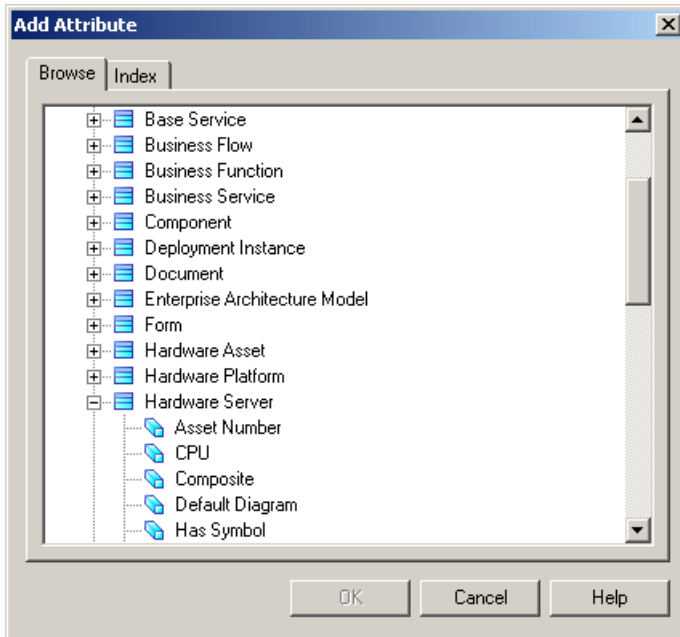
When you select an attribute and click **OK**, it is added to the list in the Customize Columns and Filter dialog and all instances of the attribute (if it is available for multiple objects) are removed from the Add Attribute dialog. You can add as many attributes as necessary and apply filters on them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

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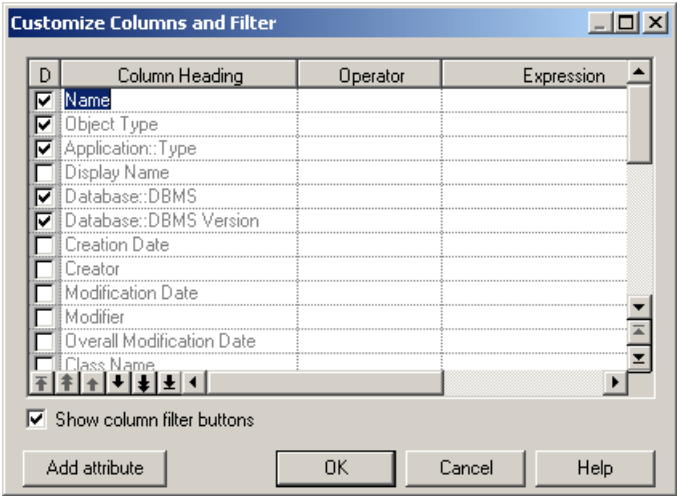
When I return to the list tab, my new attributes are displayed as columns with values where appropriate. Both my applications and databases have **Type** properties, but only my databases display values for the **DBMS** and **DBMS Version** columns:
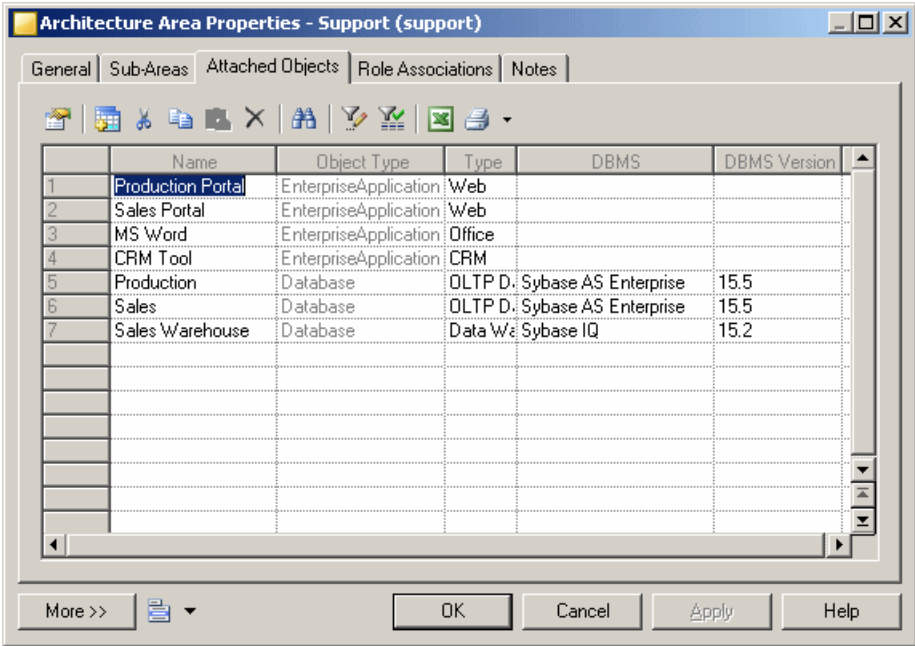
Defining a Filter Expression

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	O,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?

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.

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.

Tool	Description
⬆	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

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.
📇	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 |

Tool	Description
🔗	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* on page 120), 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.

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 Modeling > Working with 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.

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:
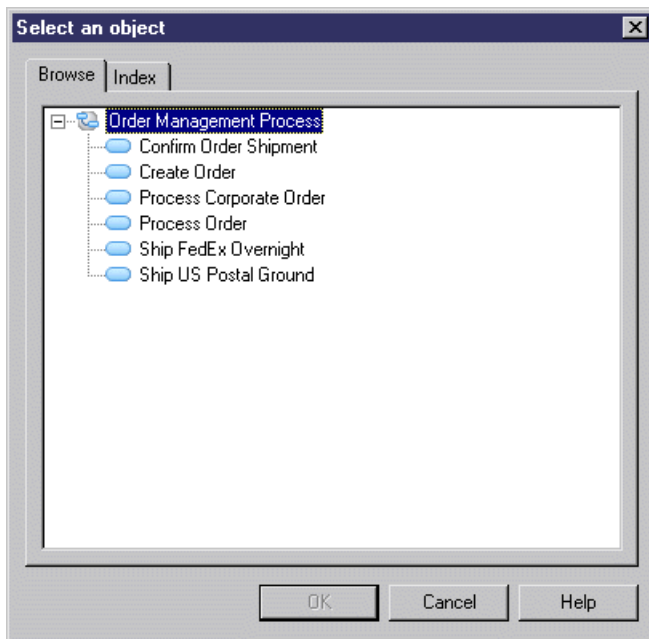
The **Index** tab lets you search for an object by entering all or part of its name (or code depending on the **Display Name/Code** model option). Objects are sorted alphabetically in the list and dynamically update as you type. Select an object and click the **Properties** button to view its property sheet:
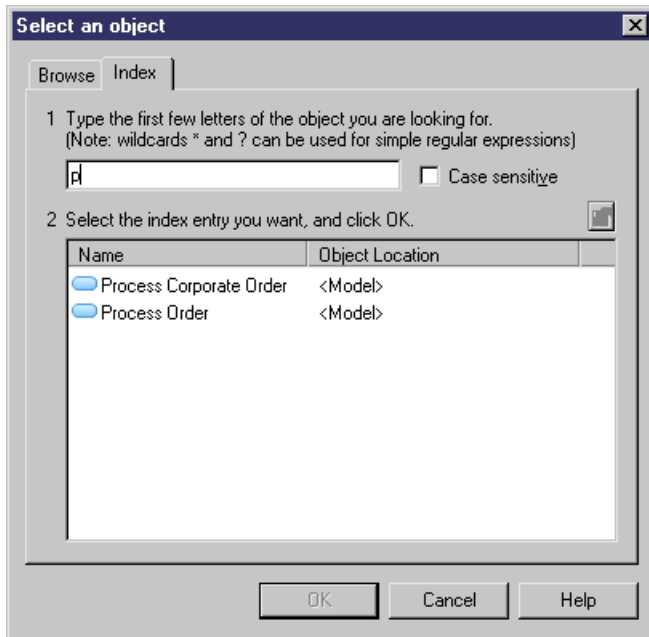
You can select an object on either tab and then click the other tab to retain the choice and view it in that context. Double-click an object, or select it and click **OK**, to commit the selection and close the selection dialog.

Moving, Copying, and Deleting Objects

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

Dragging and Dropping Objects

You can drag and drop objects to copy, move, create a shortcut or a replica in the PowerDesigner modeling environment.

You can drag and drop objects from the Browser, the diagram or the Result list to the Browser or diagram window but not to the Result list.

By default you move an object (from one package or model to another) in the Browser by drag and drop.

You can change this behavior temporarily by pressing one or more of the following keys:

Drag and drop with...	Result
[no key}	Move (in the Browser) or paste as shortcut (between diagrams)
Shift	Move
Ctrl	Copy
Shift+**Ctrl**	Shortcut creation
Shift+**Alt**	Replication creation

Note: If you select an object in the Browser or diagram and then right-click and drag it, when you release the right mouse button, a contextual menu opens listing all the available drop actions.

To modify the default Browser drag and drop behavior, select **Tools > General Options** and select the appropriate radio button:
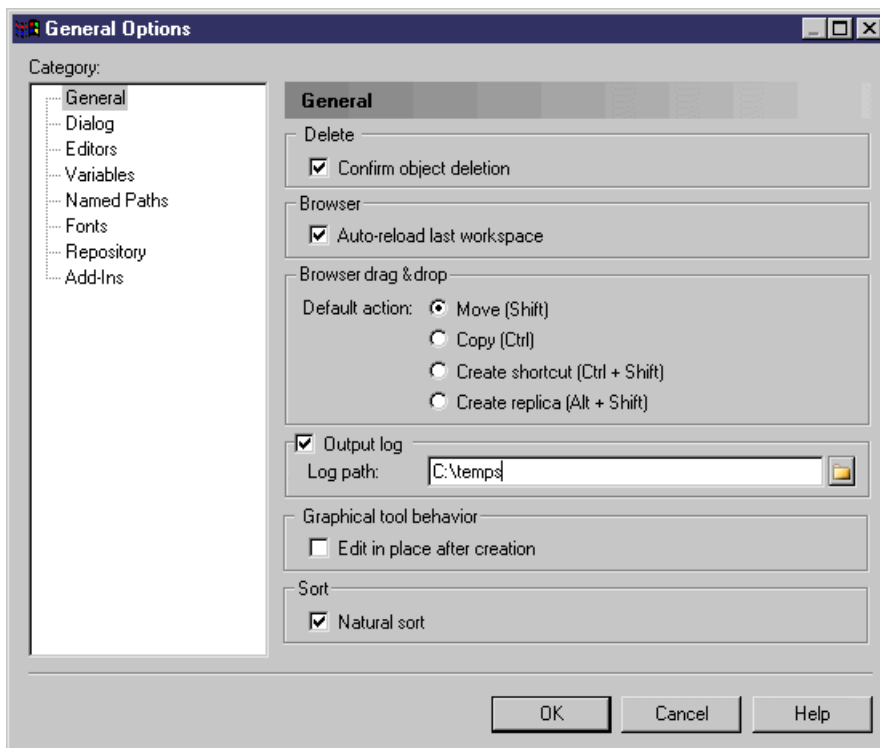
Copying and Pasting Objects

You can copy objects from diagrams, the Browser, object lists, and the Results List and paste them into diagrams, the Browser, object lists, and external applications.

When you copy an object, you copy not only its properties but also the properties of its related objects to the clipboard. For example, if you copy a CDM entity, you also copy the attributes and 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.

Copying Objects

To copy an object, select it in a diagram, the Browser, an object list, or the Results List, and then do one of the following:

* Select **Edit > Copy** from the PowerDesigner menu bar.
* Press **Ctrl+C**.
* Right-click and select **Edit > Copy**.
* Open a list of objects, select one or several lines in the list and Press **Ctrl+C**.

Pasting Objects

Note: By default, when you drag and drop objects, you move them, but if you hold down the **Ctrl** key during a drag and drop, you can create a copy. For more information, see *Dragging and dropping objects* on page 130.

To paste an object in PowerDesigner, click in a diagram, the Browser, or an object list, and then do one of the following:

* Select **Edit > Paste** from the PowerDesigner menu bar.
* Press **Ctrl+V**.
* Right-click and select **Edit > Paste**.
* Open a list of objects, select a line in the list and press **Ctrl+V**.

If you paste the selected object into an external application, the following will occur:

Copied item	Paste result
Diagram symbol	Image of the symbol (MS Word, PaintBrush)
List item from an object list	List in CSV format (MS Word, Excel)
Item from the Check Result list	List in CSV format (Excel)
Item from the Find Result list	List in CSV format (Excel)

Pasting Objects as Shortcuts

When you copy an object you can paste it as a shortcut that references the original object (see *Chapter 15, Shortcuts and Replicas* on page 451).

Note: By default, when you drag and drop objects, you move them, but if you hold down the **Ctrl+Shift** keys during a drag and drop, you can create a shortcut. For more information, see *Dragging and dropping objects* on page 130.

To paste an object as a shortcut in PowerDesigner, click in a diagram, the Browser, or an object list, and then do one of the following:

* Select **Edit > Paste As Shortcut** from the PowerDesigner menu bar.
* Right-click the Browser target or the diagram window and select **Edit > Paste As Shortcut**.

When pasting as a shortcut into a diagram, the following rules are used. If the object copied is in:

* The same model or package and the same diagram - a new graphical synonym of the object is created.
* The same model or package but a different diagram - a new shortcut symbol or graphical synonym is created.
* A different model or package:
  * If the shortcut does not exist, a shortcut is created (with a symbol)
  * If the shortcut already exists without a symbol, a symbol is created (if you are pasting into a diagram)
  * If a shortcut and a symbol already exist, a graphical synonym is created (if you are pasting into a diagram)

Managing Paste Conflicts

When you paste an object in PowerDesigner, checks are applied in order to verify that no conflict occurs between the identifying properties of the objects. The identifying criteria depends on the type of object, for some objects it is just the code, for others, the name and the code.

For more information about the identifying properties of an object, see *Object namespaces* on page 115.

When a paste conflict occurs, PowerDesigner automatically renames the name and/or the code of the object in the following way:

Source object	First renaming	Second renaming
Name	Name2	Name3
CODE	CODE2	CODE3

A message is displayed in the Output pane to warn you that the object was renamed.

When a paste conflict occurs on a CDM entity, the entity is renamed according to the data item options set in the model.

Data item options	Result of copying an entity
Unique Code Allow Reuse	New entity with new name and code New identifier with new name and code Reuses other attributes
Unique Code only	New entity with new name and code New identifier with new name and code New attributes with new names and codes
Allow Reuse only	New entity with new name and code New identifier with same name and code Reuses other attributes
None	New entity with new name and code New identifier with same name and code New attributes with same names and codes

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:
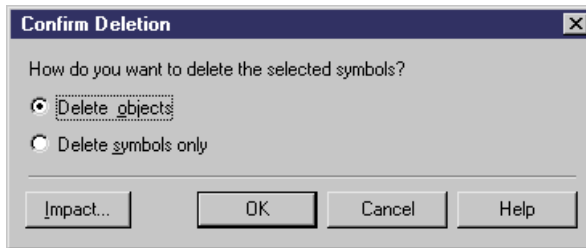
You can review the impact that deleting the object would have on other objects in your environment by clicking the **Impact** button on the confirmation dialogs (see *Chapter 17, Impact and Lineage Analysis* on page 499). You can undo a deletion by clicking the **Undo** tool.

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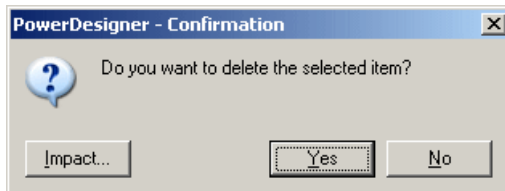
Click **OK** to delete the object, any sub-objects it contains, and any associated diagram symbols.

• 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.

Note: You can suppress the display of the Confirmation dialogs by deselecting the **Confirm Object Deletion** general option (see *General Options* on page 395).

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 Modeling*.

Moving Objects from Package to Package

You can move an object from package to package using drag and drop feature.

Moving objects is different from cutting and pasting items, since you do not duplicate objects.

For entities containing data items, the following situations can occur when you move the entity:

Data items	Namespace	Move result
Only used by selected entity	Move within the same namespace	The data items are moved with the entity
Reused among different entities	Move within the same namespace	Shortcuts of data items are created for reused data items
Used only by one entity or reused among different entities	Change namespace	Data items are copied in the other namespace

For more information about shortcut and copy rules, see *Creating Shortcuts* on page 453.

Moving objects is restricted as follows:

- Models must be compatible (same resource file)
- Global objects (business rule, domain, storage etc.) cannot be moved into a sub-package

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.

Shortcuts creation rules in PowerDesigner also apply to moving objects between packages.

For more information about shortcuts creation, see *Creating Shortcuts* on page 453.

To move an object with a non-oriented link:

Source package	Moving result
	In the source package, a shortcut of the moved entity is created: In the target package, the moved entity is displayed:
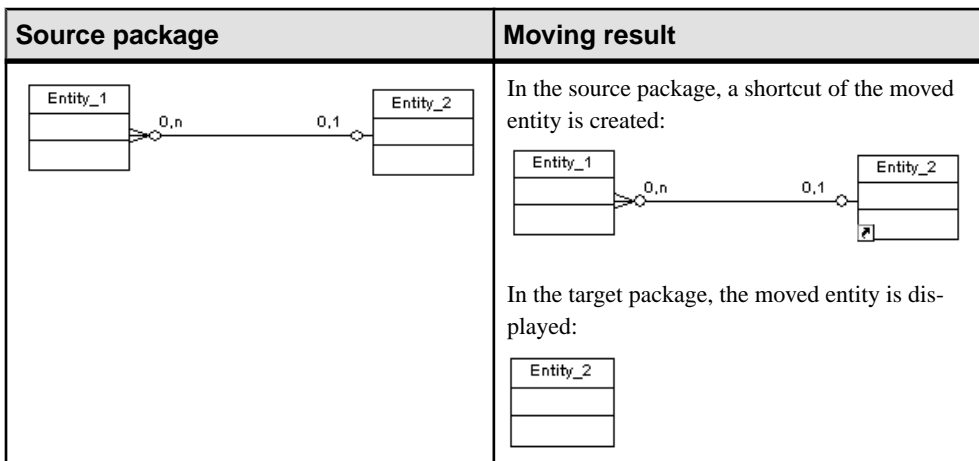
To move an object with an oriented link:

Source package	Moving result
	Example 1: Move TABLE_2 In the source package, a shortcut of the parent table (TABLE_2) is created: In the target package, the parent table (TABLE_2) is displayed: Example 2: Move TABLE_1 In the source package, only the parent table (TABLE_2) is displayed: In the target package, the child table (TABLE_1) is displayed with the reference link and a shortcut is created for the parent table (TABLE_2):

1. Select an object.
2. Press **Shift** while dragging the object to the target package.

   The object is moved to the new destination and a shortcut is created either in the source package or in the destination package depending on the link type.

Creating Graphical Synonyms for Object Symbols

A graphical synonym is an additional symbol for an object. Sometimes, creating multiple symbols for an object in a diagram can improve readability by reducing the length or

complexity or links. 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. You can even create graphical synonyms of graphical synonyms and of object shortcuts.

You can create graphical synonyms of graphical synonyms and of object shortcuts. Each graphical synonym displays the name of the object followed by a colon and the number of the synonym. In the following example, `Employee : 1` and `Employee : 2` both represent the `Employee` table:

Employee : 1	
id	<Undefined>
first name	<Undefined>
family name	<Undefined>

Employee : 2	
id	<Undefined>
first name	<Undefined>
family name	<Undefined>

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.

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.

Graphical Synonym Limitations
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

Note that 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:
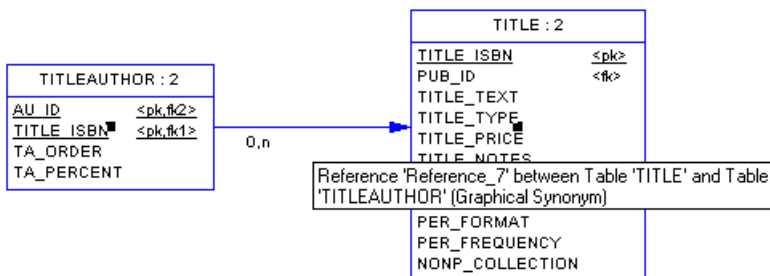
Finding Objects

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

Note: For information about searching in the repository and PowerDesigner Portal, see *Finding Objects in the Repository* on page 314 and *PowerDesigner Portal Search* on page 336.
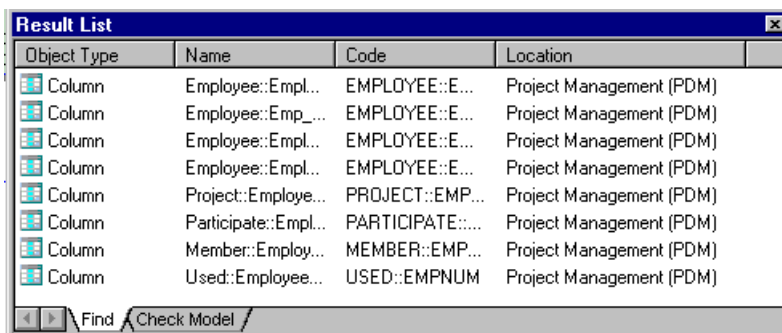
1. Select **Edit > Find Objects** (or press **Ctrl+F**) to open the Find Objects dialog.
2. Specify the appropriate constraints on these tabs:
   * **Name and Location** - to search on the name and location of PowerDesigner objects.
   * **User and Date** - to search on the user and date of creation/modification 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* on page 140.
3. Click **Find Now**.

   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* on page 97).
* **Impact and Lineage Analysis** - to analyze the object (see *Chapter 17, Impact and Lineage Analysis* on page 499).

- **Find in Browser** - to highlight the object in the Browser (see *Chapter 2, The Browser* on page 39).
- **Find in Diagram** - to open the diagram (see *Diagrams* on page 175) 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 (see *Copying and Pasting Objects* on page 132). Select a destination in the Browser or a diagram and select **Edit > Paste** or **Edit > Paste as Shortcut**.

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* on page 124.
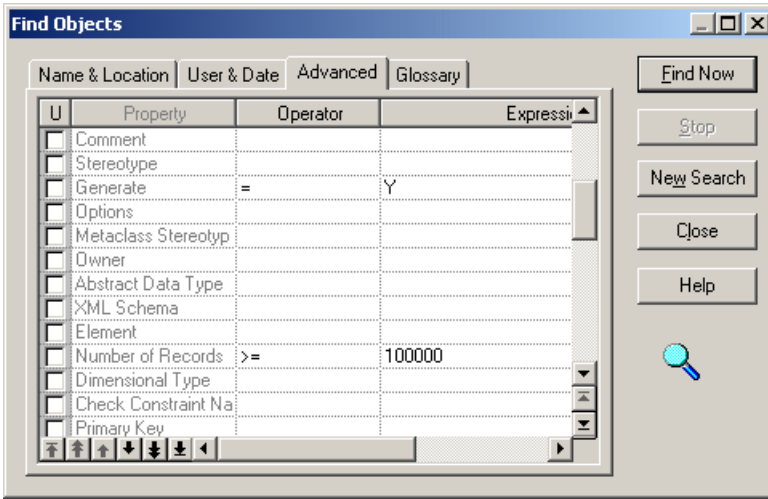
Note: If you do not select an object type on the **Name & Location** tab, then you can only search on the name and code properties on this tab.

Glossary Tab

The **Glossary** tab allows you to specify additional search constraints in relation to glossary objects (see *The Glossary* on page 40) with which objects may be associated:

Parameter	Description
Search by	Specifies that the objects searched for must be associated with a glossary object whose name contains the specified string.
Search Range	Specifies the glossary objects to search on. You can select one or more of **Keywords**, **Terms** [default], **Related Terms**, and **Synonyms**.
Case sensitive	Specifies that the search string must match the case of the glossary object.

Business Rules

A business rule is a rule that your business follows. It is a written statement specifying what an information system must do or how it must be structured. It could be a government-imposed law, a customer requirement, or an internal guideline.

Business rules often start as simple observations, for example "customers call toll-free numbers to place orders." During the design process they develop into more detailed expressions, for example what information a customer supplies when placing an order or how much a customer can spend based on a credit limit.

You can attach business rules to your model objects to guide and document the creation of your model. For example, the rule "an employee belongs to only one division" can help you graphically build the link between an employee and a division.

Business rules complement model graphics with information that is not easily represented graphically. For example, some rules specify physical concerns in the form of formulas and validation rules. These technical expressions do not have a graphical representation.

In the case of the PDM and OOM, you can generate business validation rules attached to domains as check parameters.

Before you create business rules, formulate your rules by asking yourself the following questions:

- What business problems do I want to address?
- Are there any procedures that my system must respect?
- Do any specifications dictate the scope of my project?
- Do any constraints limit my options?
- How can each of these procedures, specifications, and constraints be described?
- How can each of these descriptions be classified? Possible classifications are definitions, facts, formulas, requirements or validation rules

Creating a Business Rule

You can create a business rule from the **Model** menu, from the Browser of from a 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.
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 Tab - 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 Tab - This tab is only available for business rules with a type of OCL Constraint. The Object Constraint Language is the UML expression language. Enter your OCL expression in the text field.

Applying a Business Rule to a Model Object

You can apply business rules that you have created to your model objects.

1. Open the property sheet of an object, and then click the Rules tab:
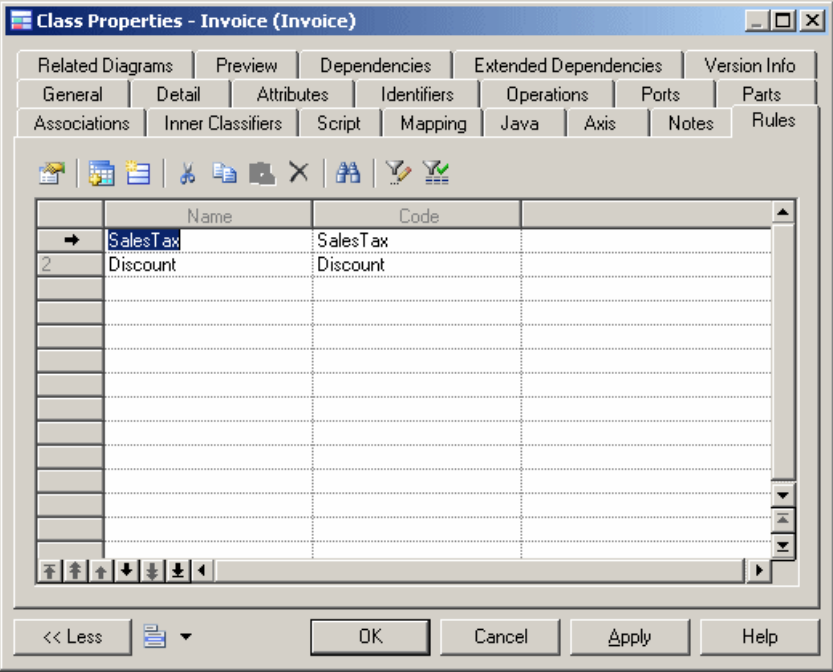
2. Click the Add Rules tool to open a selection window listing all the business rules available in the model:
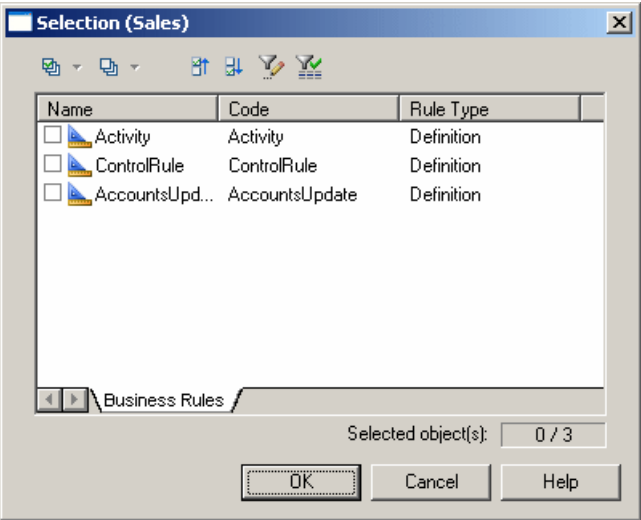
3. Select the business rules you want to add to the object, and then click OK to return to the object's property sheet.

4. Click OK to close the object property sheet and return to the model.

> **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.

File Objects

A file object is a representation in a PowerDesigner model of an external application file (for example, a Java file, script SQL, or MS Word file). The file itself can be kept external to the model or embedded within and saved with it.

For example you could:

- Attach a file object to any PowerDesigner object to enrich its description
- Attach a generated OOM class to a target Java file

You open the file in its associated editor by double-clicking its diagram symbol or browser entry. To change the associated editor, right-click the file object and select **Open With > Choose Program** (see *Specifying Text Editors* on page 399).

Creating a File Object

You can create a file object from the Toolbox, Browser, or **Model** menu.

- 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**.
- Drag a file from Windows Explorer and drop it in the diagram or Browser.

File Object Properties

To view or edit a file 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.
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
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	[External and URL types only] Specifies the path or URL to the file.
Extension	Specifies the extension of the file object, which is used to associate it with an editor. 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 key-words, separate them with commas.

Attaching a File to a Model Object

You can attach file objects to other model objects through traceability links.

• Right-click an object symbol in the diagram and select **Edit > File > Add File**.
• Use the **Link/Traceability Link** tool in the toolbox to draw a link from the object symbol to the file object symbol.
• Select the **Traceability Links** tab of the object property sheet, click the **Add Objects** tool, select the file, and click **OK** (see *Creating Traceability Links* on page 429).

The connection between the object and the file object takes the form of a traceability link, and is visible:

• In the form of a link in the diagram.
• On the **Dependencies** tab of the file object property sheet and on the **Traceability Links** tab of the dependent object property sheet.
• Under the **File** command in the contextual menu of the dependent object symbol.

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.

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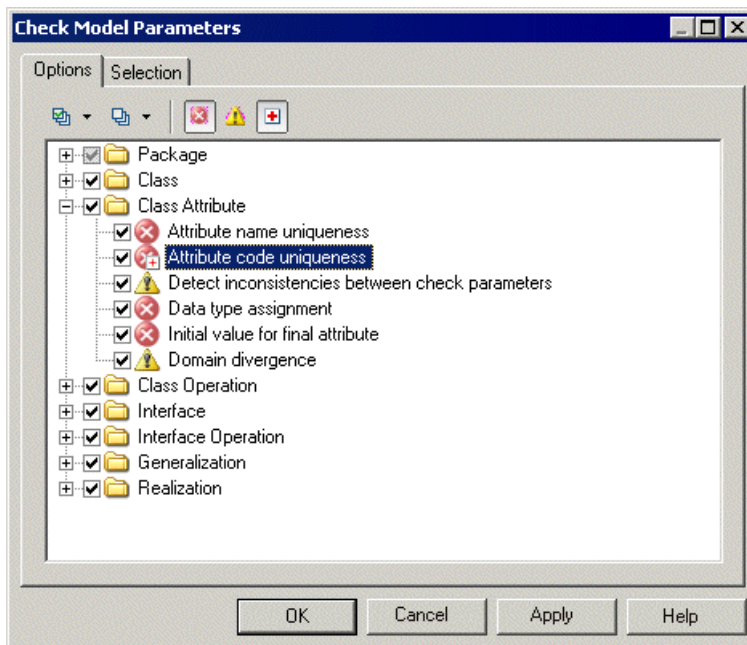
2. [optional] Select or deselect types of objects to check, and expand object nodes to enable, disable, vary the severity of, and enable or disable automatic correction of individual checks with the following tools:

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.

Tool	Description
	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:
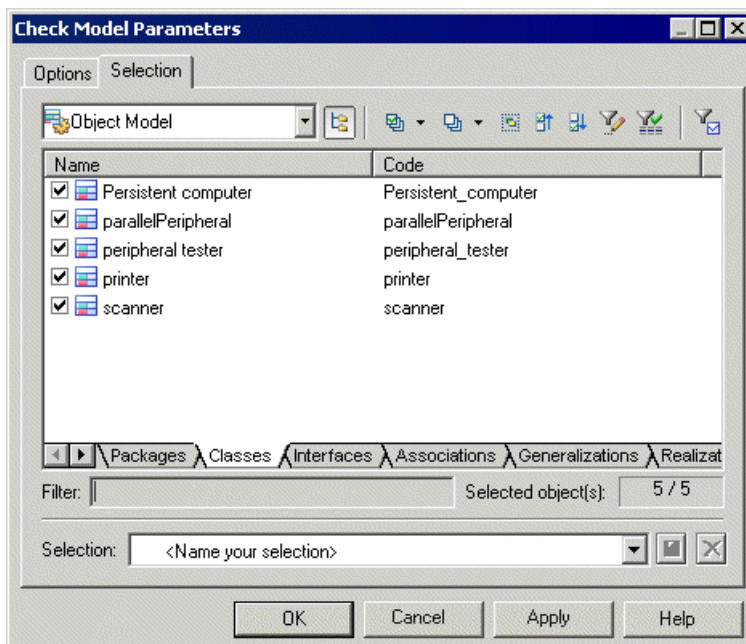
Note: If you had previously selected symbols in your diagram, you can select them for checking with the **Use Graphical Selection** tool. For detailed information about these tools, see *Adding an Item from a Selection List* on page 126.

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* on page 150.

Result List			
Category	Check	Object	Location
⚠ Class	Operation implementation	Operation 'Perip...	<Model>::Periph...
Interface Ope...	Return type assignment	Operation 'Perip...	<Model>::Periph...
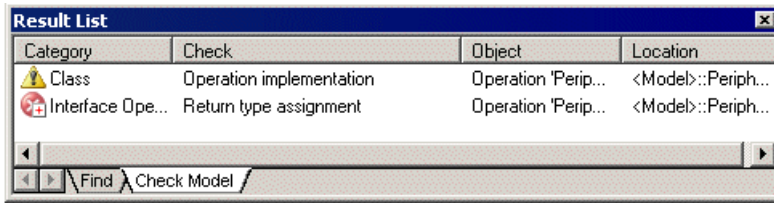
Find ⟩ Check Model

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.

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.

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

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.

Check	Description and Correction
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.

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.
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

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

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 autcompletion and compliance checking** for your naming conventions (see *Naming Conventions* on page 107):

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.

The Free Model (FEM)

A *free model (FEM)* provides a context-free environment for modeling any kind of objects or systems. It is generally associated with a set of extensions, which allow you to define your own concepts and graphical symbols.
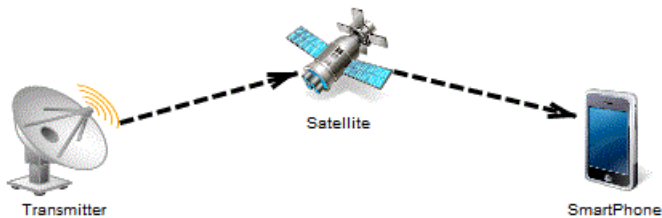
To create a new, empty FEM - Select **File > New Model**, and choose **Free Model**.
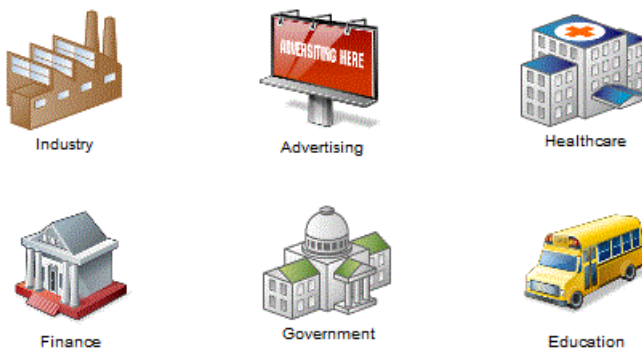
For information about extending objects from their property sheets, see *Extending Objects* on page 155. For detailed information about all the extension mechanisms available in PowerDesigner, see *Customizing and Extending PowerDesigner > Extension Files*.

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.

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*.

Adding New Properties to an Object

To quickly add new properties to an object, use the dialogs available from the property sheet menu. Properties added to an object are available for all other objects of that type.

Adding a New Attribute to an Object

If the standard PowerDesigner attributes are insufficient for your modeling requirements, you can add new ones using the New Attribute dialog available from the property sheet menu.

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:

New Attribute

Name: Login

Comment:

Data type: (Text)

Default value:

List of values: Complete ☐

Text Format: Text

☐ Open Resource Editor on Finish   Create Another...

< Back   Next >   Finish   Cancel   Help
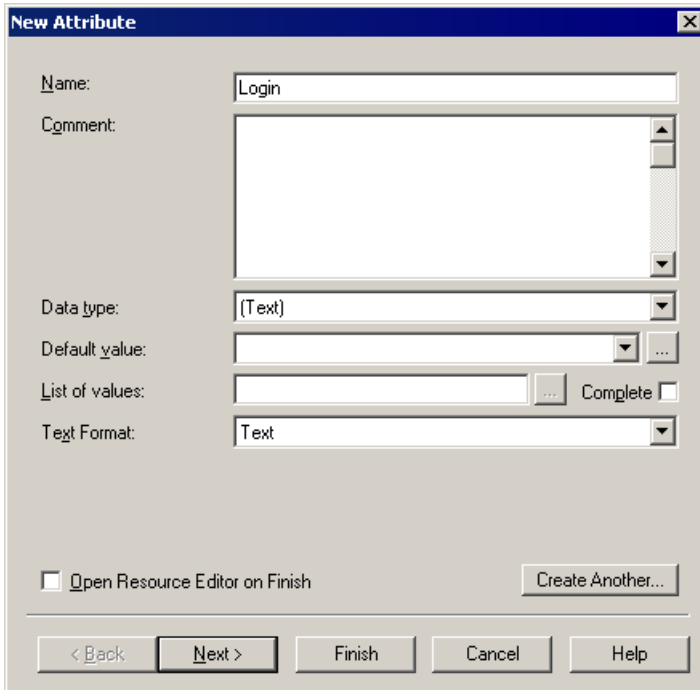
2. Enter a **Name** and **Datatype** for the new attribute and complete any other appropriate fields.

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.
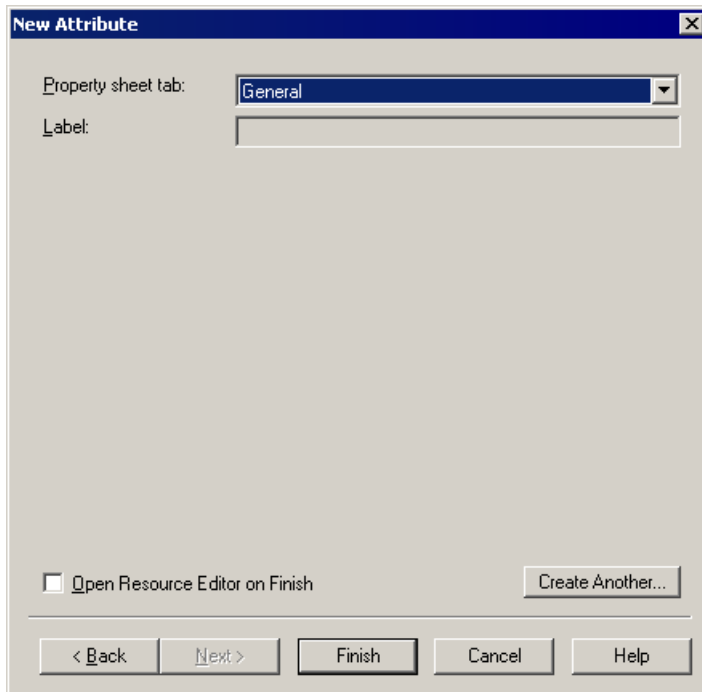
By default, the attribute is inserted on the tab from which you launched the New Attribute dialog, if the tab is editable.

4. [optional] If you want to view the new attribute in the PowerDesigner Resource Editor, select the **Open Resource Editor on Finish** checkbox.

All extensions are stored in PowerDesigner resource files. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files*.

5. Click **Finish** to close the dialog and create the new attribute.

The attribute appears on the specified property sheet tab:
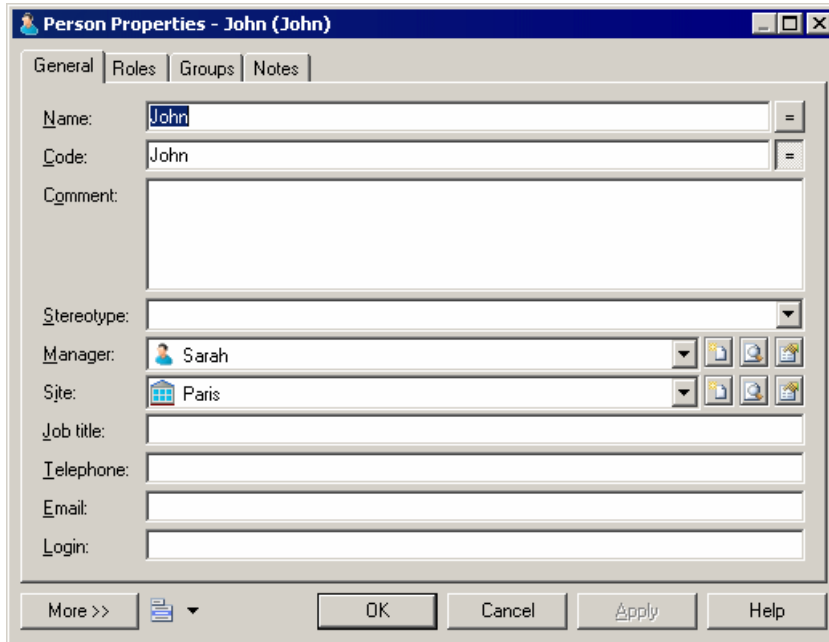
Adding a New List to an Object

Many PowerDesigner objects are associated with multiple instances of another type of object. For example, a table can have multiple columns, a class can have many operations, and a person can be in many groups. You can model new kinds of associations by adding new lists to your object property sheets.

1. Open the property sheet of the object you want to extend, click the menu button, and select **New List of Associated Objects**.
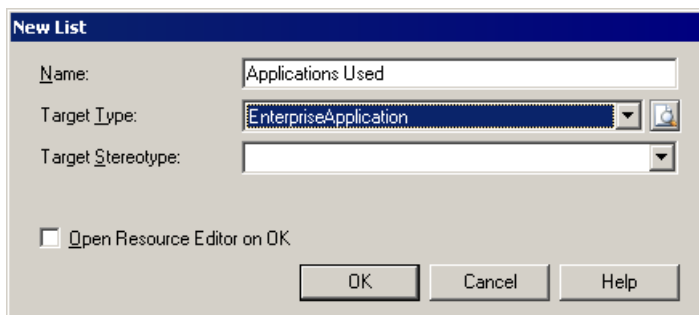
2. Enter a **Name** for your list. This name is used as the name of the property sheet tab on which the list appears.

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, but the association should make sense for your modeling needs.

4. [optional] To limit the objects of the selected type that your object can be associated with, specify a stereotype. If you specify a stereotype here, you will only be able to associate your object with other objects bearing this stereotype.

5. [optional] If you want to view the new collection in the PowerDesigner Resource Editor, select the **Open Resource Editor on OK** checkbox.

   All extensions are stored in PowerDesigner resource files. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension 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:
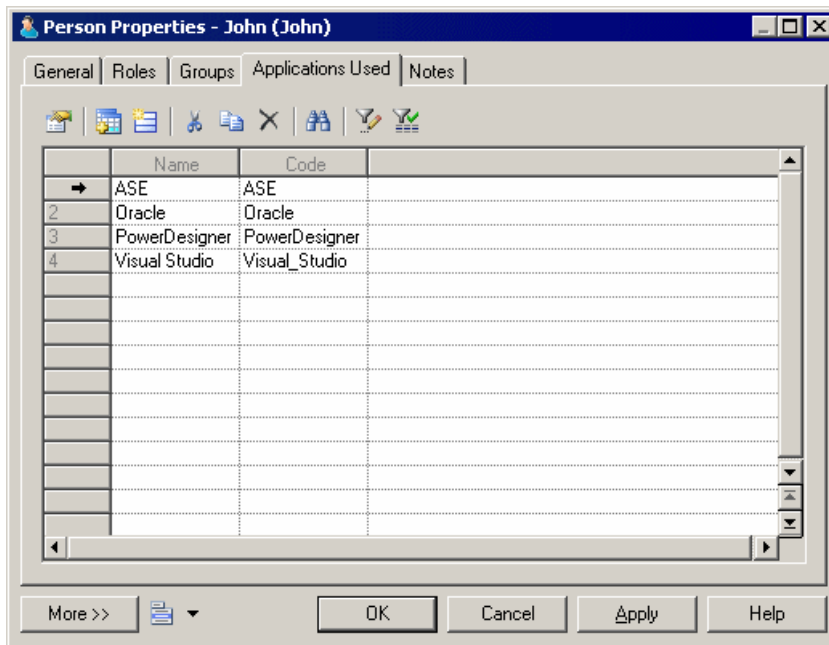
Managing Extensions
To review your object extensions in the Resource Editor, select **Manage Object Extensions** from the property sheet menu.

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.

---

1. Open the property sheet of the object, click the menu button, and select **Manage Object Extensions**.
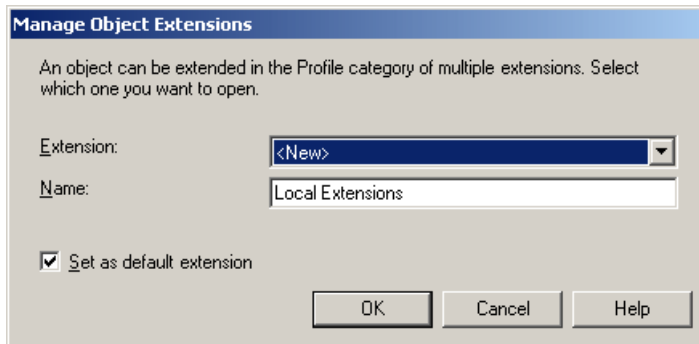
2. Specify the **Extension** that you want to open, from the list of files attached to the model.

   An object can be extended in the Profile category of multiple extensions. You can select an existing file or create a new one.

3. [optional] Select the checkbox to set the chosen extension as the default.

4. Click OK to open the chosen extension in the Resource Editor.

   For detailed information about editing files in the Resource Editor, see .

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 *Using the Free Model* on page 154), 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.
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.

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)*.

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 > Templates and 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.

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:
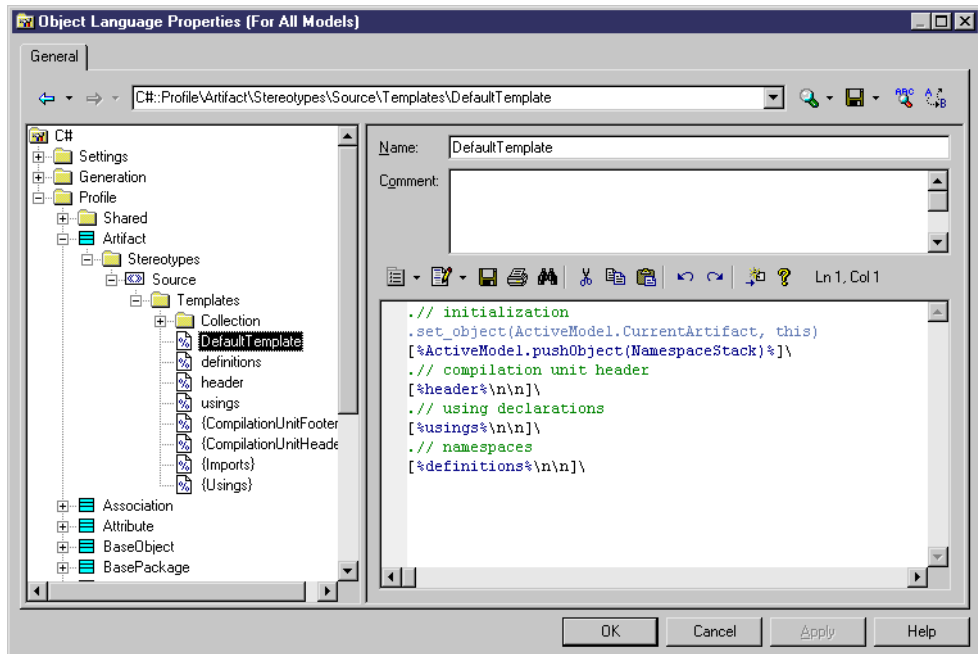
Example

You define the following artifact criterion in an extension:
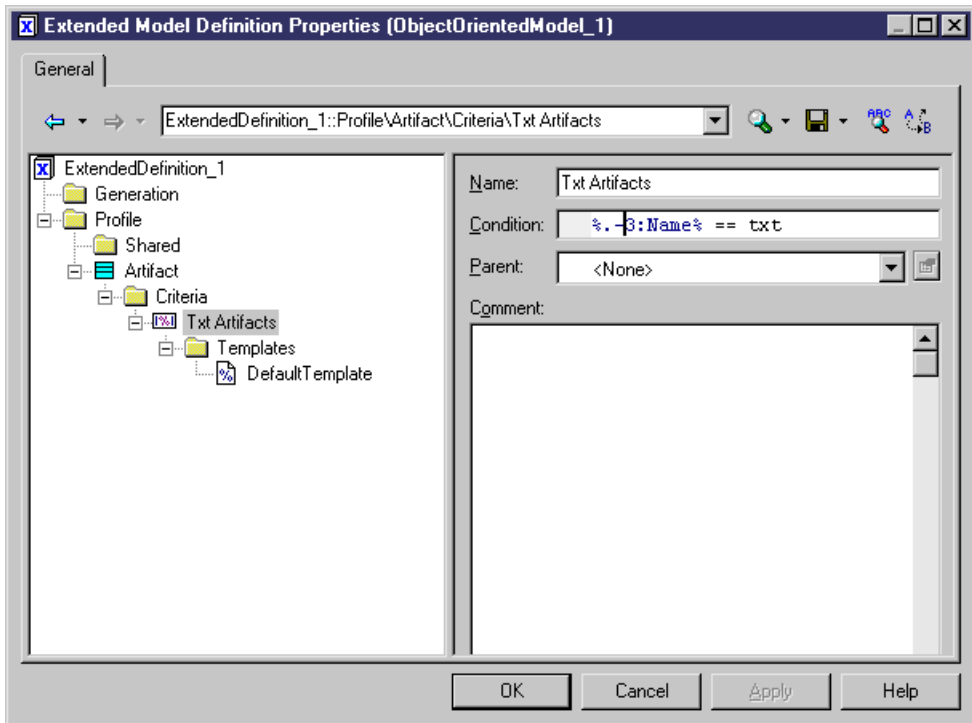
This criterion verifies that the last 4 characters of the artifact name are .TXT. You also need to create the default template in order to define the content of the generated file. In this example, DefaultTemplate is defined as follows:

```
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:
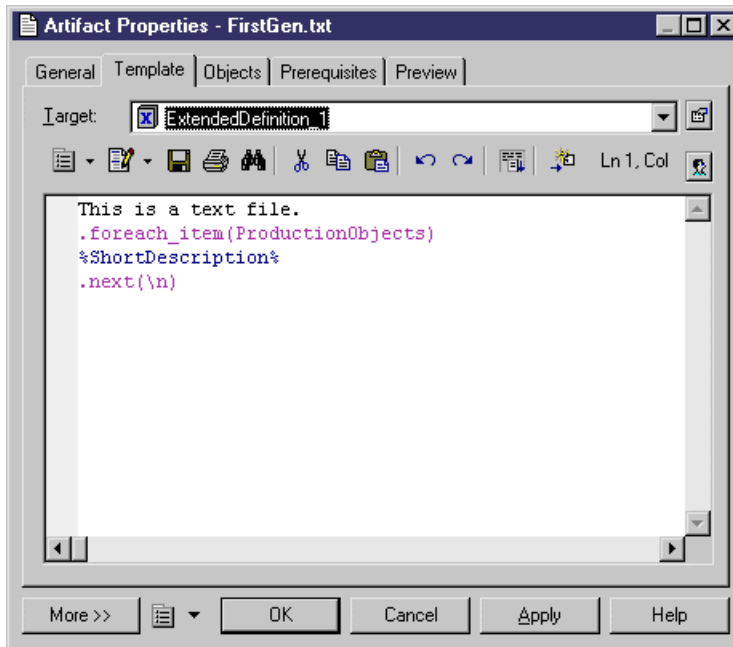
You can now define instances of objects you want to include in the file generated from the current artifact:
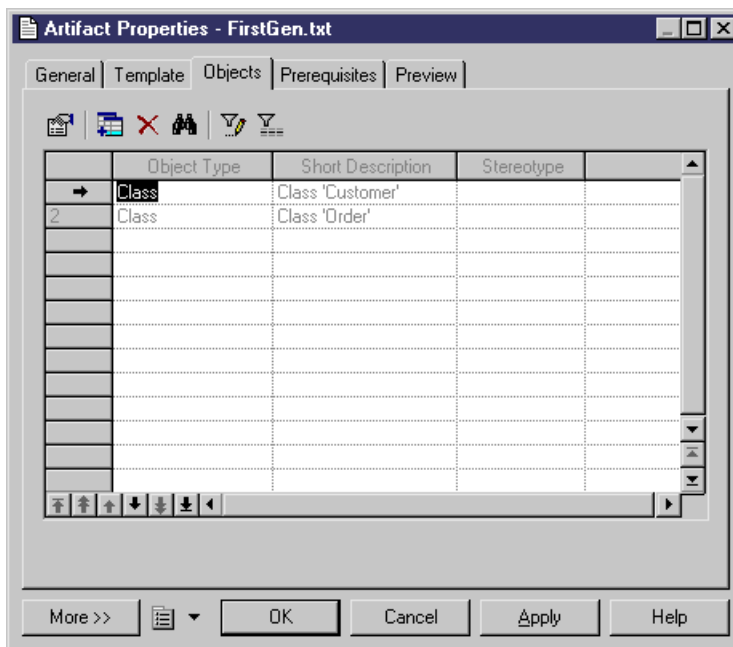
Defining an Artifact

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.

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.

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.
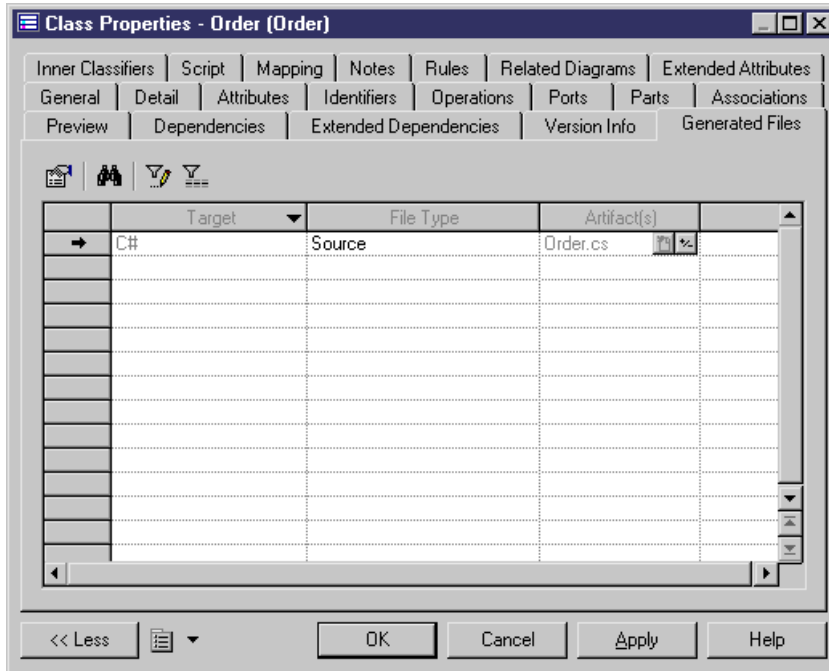
4. Click Apply and click the Properties tool to define artifact properties.

5. Click OK in both property sheets.

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.
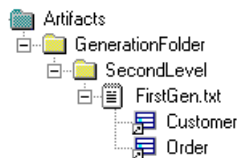
Generated Files Tab

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.
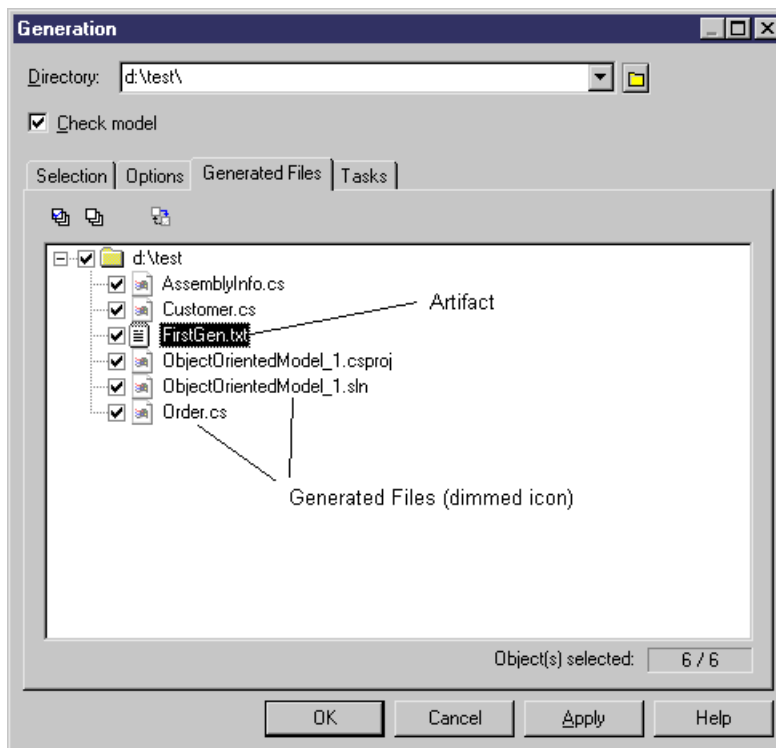
You can modify generation options from this dialog box, and you can also check artifact generation completeness: if an exclamation mark is displayed on the artifact icon it means that one or several prerequisite artifacts are missing.

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.
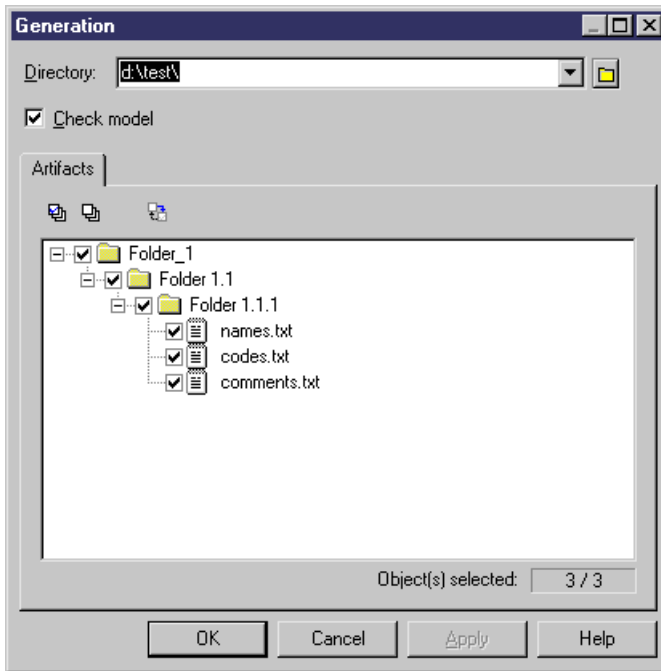
You can enforce dependencies in the Generation dialog box to make sure all prerequisites are selected for generation.

Importing Objects from Excel Files

The Excel Import extension allows you to launch a wizard that guides you through mapping tables of objects to be modeled from Excel files (v2003 and higher) to PowerDesigner objects and properties, and to import the contents of the files 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.

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 separate tables of objects containing 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* on page 171. An example Excel file for importing into a PDM is provided at `install_dir\Examples\excel_import.xls`.

You can create a new model by importing objects from an Excel file or import your objects into an existing model.

1. Open the Excel Import Wizard:

   • To a 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* on page 6), 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* on page 418), then right-click the model in the Browser, and select **Import Excel File**.

Excel Import Wizard

Welcome to the Excel Import Wizard. This wizard guides you through importing tables of objects defined in an Excel spreadsheet. To get started, select an Excel file to import.

`C:\temp\tables.xlsx`

Options...

Next>   Cancel   Help

2. Click the **Select File** tool, browse to and select the Excel file to import, and click **Open** to return to the wizard.
3. [optional] Click the **Options** button to open the Import Options dialog (see *Excel Import Options* on page 173), specify any appropriate options and then click **Close** to return to the wizard.
4. Click **Next** to go to the Import Table page:
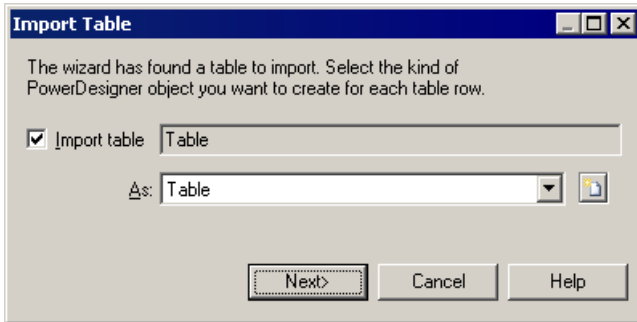
On this page, you must either:

• 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.

5. Click **Next** to go to the Import Column page:
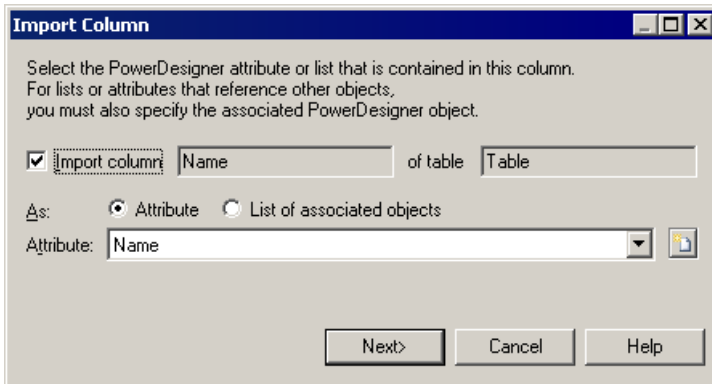
On this page, you must either:

• 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.

6. 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 sheets and all the 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 it is complete, a dialog will appear showing how many objects have been created. Click **OK** to return to your model.

In addition to the objects that you have 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.

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.

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:

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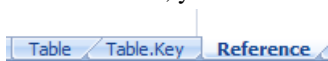
For sub-objects (objects linked to their parent by composition), you must prefix the name of the object with that of its parent object, (eg `Table.Key`) or it will not be recognized.

- **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

*   **List items in a collection in a single cell** – You can quickly list items in a collection by entering them in a single Excel cell separated by commas:

?r	Columns
	ID,Name,Password,Email
	ID,CustomerID,Date
	OrderID,ProductID,Qty

Table / Table.Key / Reference

You can change the separator by specifying another symbol in the **List value separator** import option.

*   **Represent relationships of composition using a `Parent` column** – To specify relationships of type composition (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. 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.

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

Architecure Area / Business Fur

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 tables and columns that share names with objects or properties in the PowerDesigner metamodel without requesting confirmation. If your Excel sheets and table column headings conform to the PowerDesigner metamodel, this option can save considerable time by only showing you (and requesting your input for) items that the wizard is unable to match itself.
	For information about helping the wizard match your data to the PowerDesigner metamodel, see *Preparing Your Excel File for Import* on page 171).
	Default: Unselected
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 automatically create objects referenced by the imported objects where these are not otherwise defined in your Excel file. For example, If you have a worksheet listing PDM tables, which contains a column called `Owner`, PowerDesigner will create user objects for each of the names appearing in this column unless you supply a list of users on another worksheet. Similarly, if you do supply such a list of users but not all the names in the `Owner` column are included, user objects will be created for each of the missing names.
	Default: Selected
Object identifier	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. The Qualified name separator is used to specify the location or parent of an object. For example, to specify that architecture area `Area3` is inside `Area2`, which is in turn inside `Area1`, you would enter `Area1.Area2` in the `Parent` column for `Area3`.
	Default: Dot

Option	Description
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'	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'.

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 `PD_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.

CHAPTER 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.

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.

Note: In requirements models (see *Requirements Modeling*), there are no diagrams, but rather views.

Creating a Diagram

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 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 (see also *Creating a Diagram from a Selection* on page 176).
* 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.

Creating a Diagram from a Selection

If you have selected two or more symbols in a diagram, you can create a new diagram from your selection by right-clicking in it and choosing **Create Diagram from Selection**.

Note: Since PowerDesigner objects can appear in multiple diagrams, the selected objects will now be present in both the original and the new diagram.

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
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

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**.

*   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.**
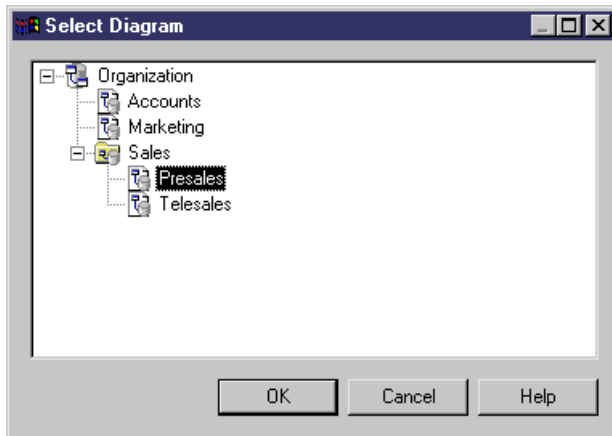
- To access diagrams beneath package or other composite object symbols:
  - Select the **Open Diagram** tool from the Toolbox and click the symbol.
  - Press **Ctrl** and double-click the symbol.
  - Right-click the symbol and select **Diagram > Open Diagram**.

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**.

Tool	Description
[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**.

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.

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.

Specifying Diagrams as Related Diagrams

Related Diagrams are diagrams that provide additional information about an object. 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, by opening the property sheet of the object and clicking the **Related Diagrams** tab.

The following tools are available to attach new related diagrams or work with existing ones:

Tool	Description
	Properties - Opens the property sheet of the selected diagram.
	Open Diagram - Opens the selected diagram.
	Add Objects - Opens an object selection dialog to select diagrams and add them to the list (see *Adding an Item from a Selection List* on page 126).
	Create an Object - Creates a new diagram, which will be added 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.

Note: You can navigate to a related diagram from an object symbol by right-clicking the symbol and selecting **Related Diagram >** *Diagram Name*. To create a new diagram and associate it to the object as a related diagram, right-click the symbol and select **Related Diagram >** *New*.

You can only attach related diagrams to objects that have a **Related Diagrams** tab in their property sheet.

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.

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**.

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.

Note: To print detailed information about model objects, create a model report (see *Chapter 6, Reports* on page 225).

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.
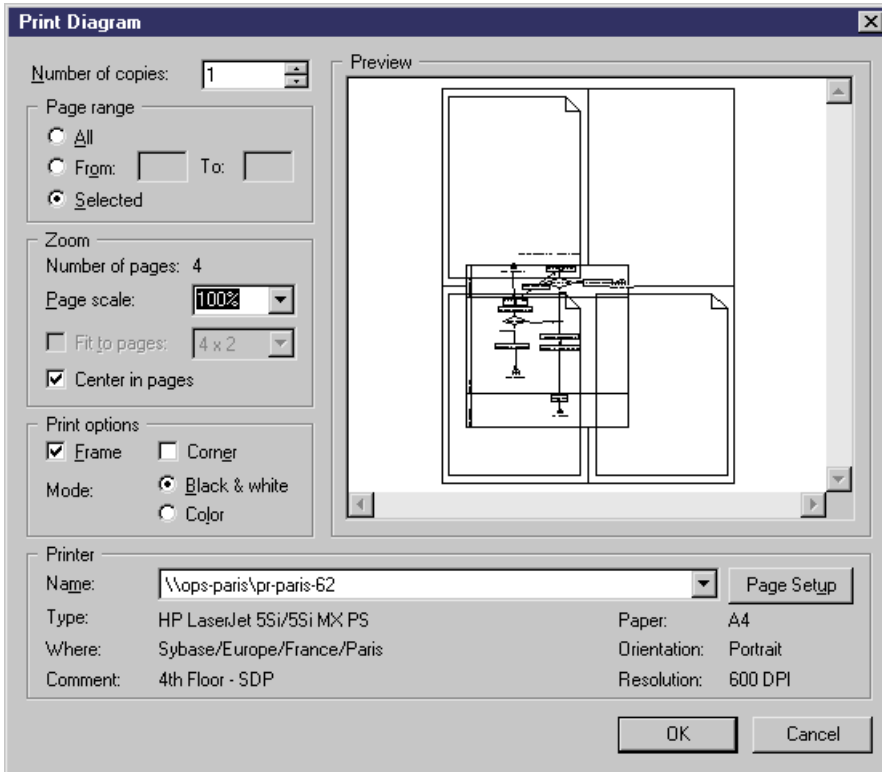
3. [optional] Specify the pages to print in the Page range groupbox or by clicking in the Preview pane (see *Print Diagram Options* on page 180). Only pages with an overlaid page frame will be printed.

4. [optional] Specify a page scale or set of pages to fit to (see *Print Diagram Options* on page 180). 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* on page 181).

6. Click **OK** to start printing.

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.

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.
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

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.

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 *Chapter 15, Shortcuts and Replicas* on page 451).

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.
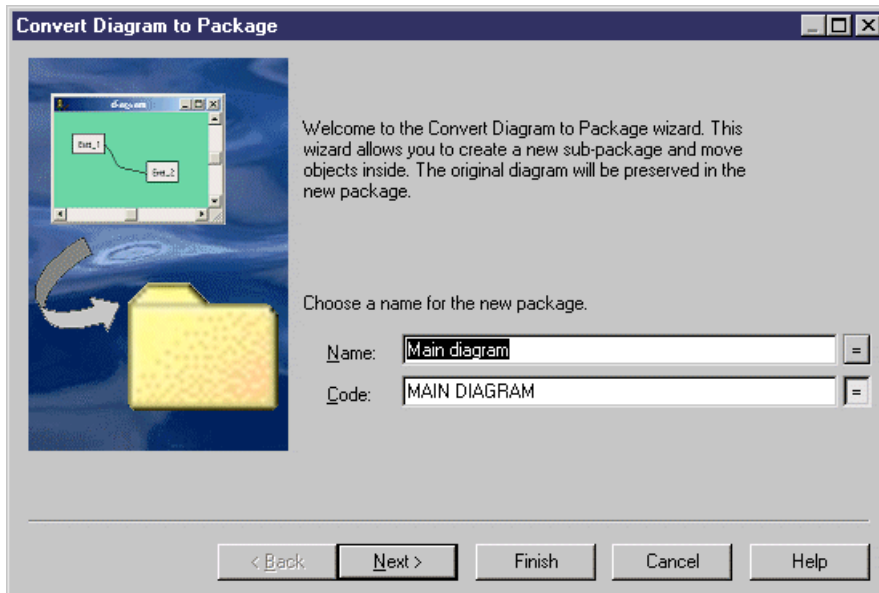
2. Click **Next** to open the Selecting Objects to Move page, which lists all the objects in the diagram available to move to the new package. Objects are organized by object type, with a sub-tab for each object type. By default, all the objects are selected.
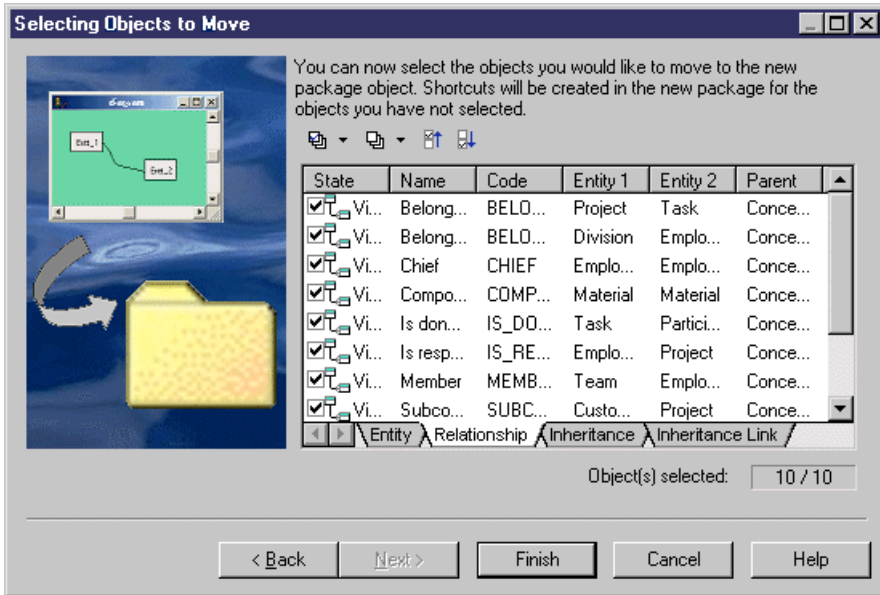
3. [optional] Deselect any objects you do not want to move to the new package. Objects deselected here will remain in the original package and be represented in the new package via shortcuts.

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.

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.

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 *Chapter 15, Shortcuts and Replicas* on page 451).

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.

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.
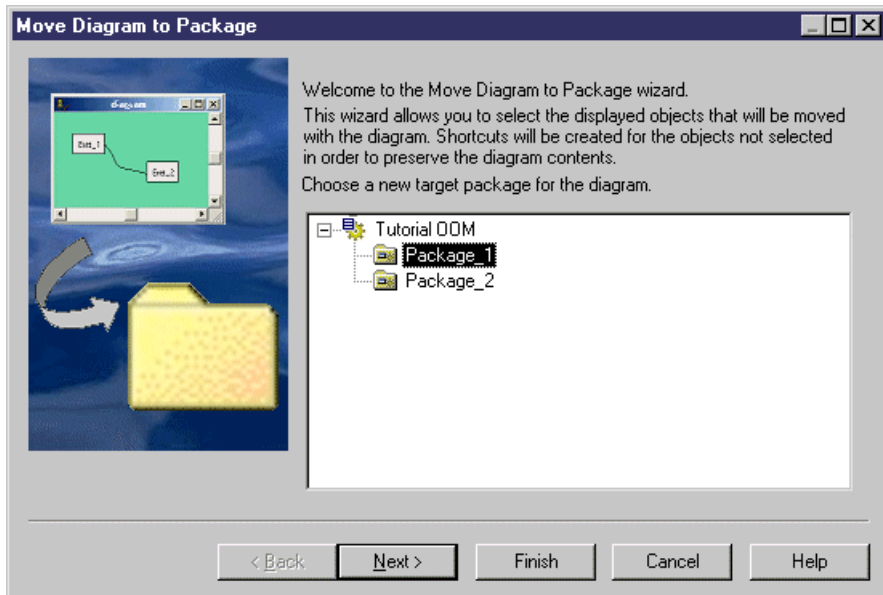
2. Select the target package to which you want to move the current diagram and click **Next** to open the Selecting Objects to Move page, which lists all the objects in the diagram available to move to the new package. Objects are organized by object type, with a sub-tab for each object type. By default, all the objects are selected.
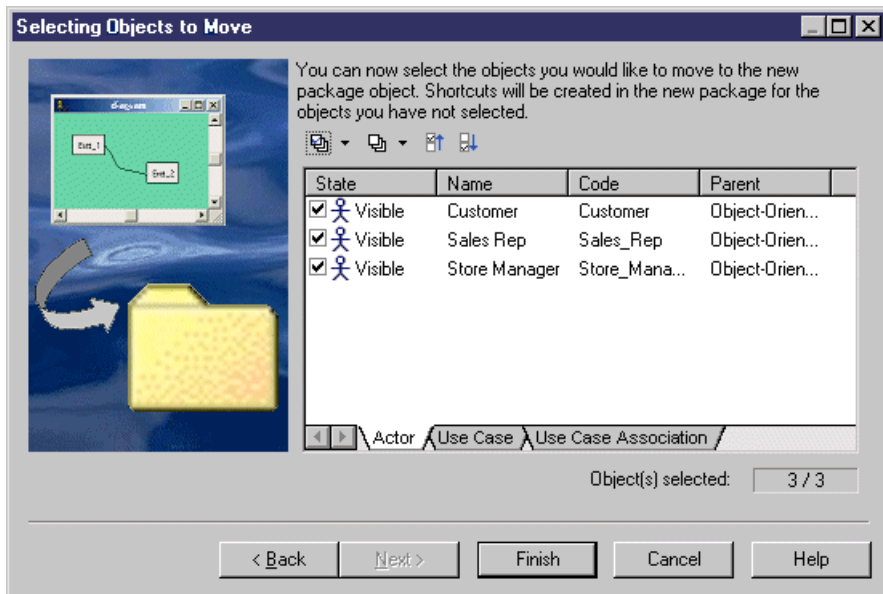
3. [optional] Deselect any objects you do not want to move to the new package. Objects deselected here will remain in the original package and be represented in the new package via shortcuts.

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.

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 namespace	The data items are moved with the entity.
Reused among different entities	Move within the same namespace	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.

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.

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:

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:
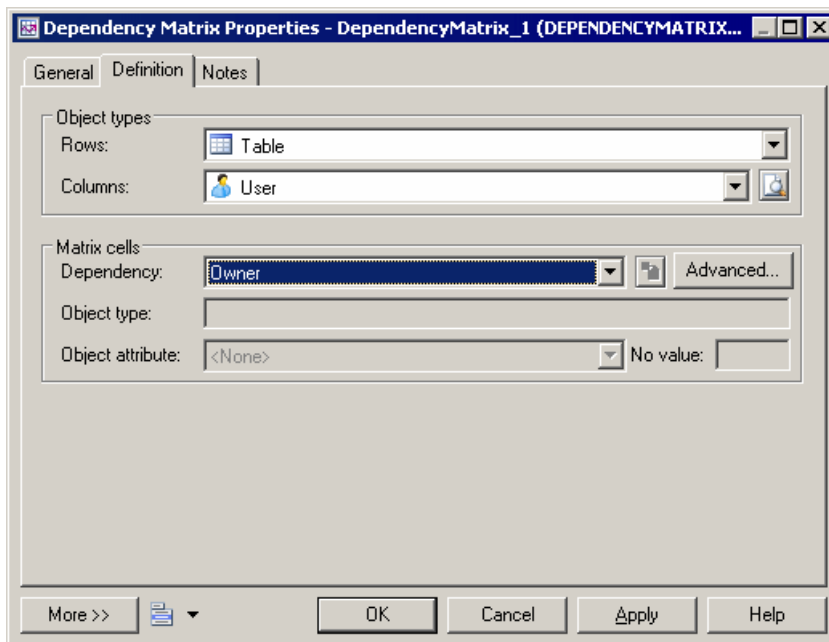
   Only direct dependencies are available from the list. To specify a more complex dependency, click the **Advanced** button to open the Dependency Path Definition dialog (see *Specifying Advanced Dependencies* on page 188).

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.

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.*

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 functions and roles, by passing from the business function through the processes it contains, to the role linked to it by a role association:
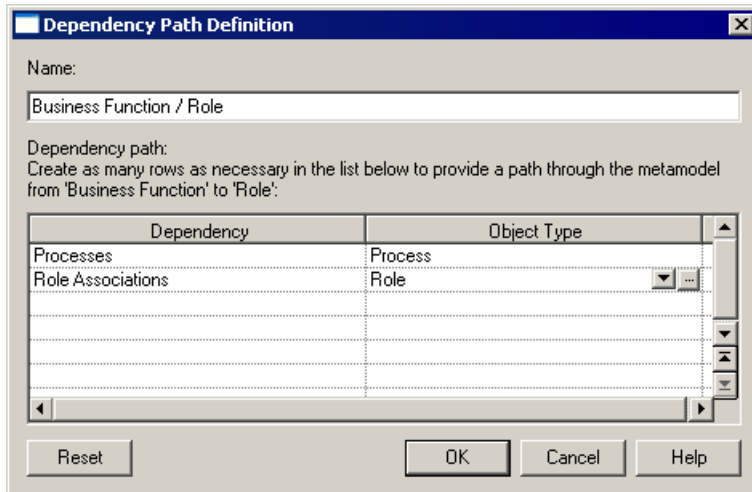
Working with Dependency Matrices

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: Register Order	Direct Sales: Send to Manufacturing	Direct Sales: Validate Order	Indirect Sales	Indirect Sales: Contact Partner	Indirect Sales: Organize Event	Manufacturing	Manufacturing: Manufacture	Manufacturing: Material Control	Manufacturing: Quality Assurance	Manufacturing: 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* on page 126).
	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.

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.

Symbols

PowerDesigner provides multiple tools and techniques for manipulating object symbols in your model diagrams.

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:
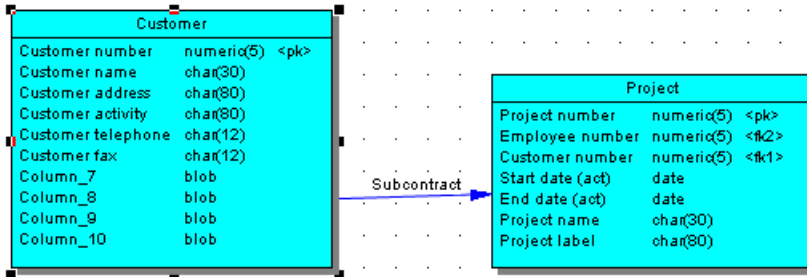
Note: To resize all the symbols in a diagram at once, click the **Grabber** tool and then drag one of the handles.

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:
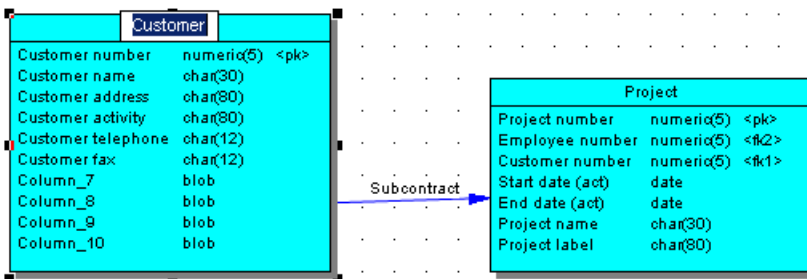
When an object property is selected for editing, you can navigate through its editable properties and on into its sub-object properties by pressing the Tab or Shift+Tab keys.

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:
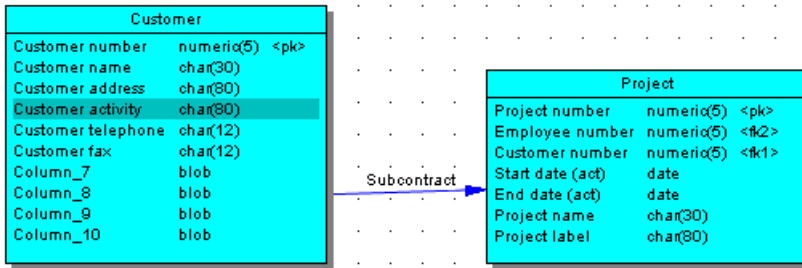
You can navigate up and down in the list of sub-objects by pressing the Up Arrow or Down Arrow keys. When a sub-object is selected, clicking on one of its properties lets you edit its value. Now the `Customer activity` column name is selected for editing:
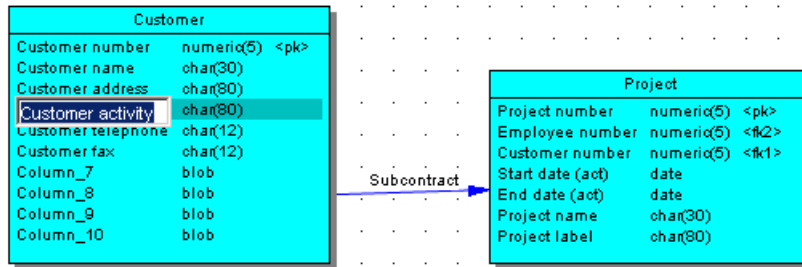
To create a new sub-object at the end of the list, press Ctrl+N. To insert a new sub-object above the selected sub-object, press Ctrl+I. To delete the selected sub-object, press Del. Right-clicking a sub-object opens a contextual menu offering useful commands.

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.

Manipulating Link Symbols

You can add and remove corners to and from link symbols, and change the objects that they connect.

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:
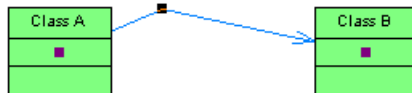
Note: To move an extremity of a link symbol from one object to another, click to select the link, and then drag the end handle to the new object.

Arranging Symbols Using the Symbol Menu

The Symbol menu allows you to do many different things with your diagram symbols.

Menu Item	Function
Format [Ctrl+T]	Opens the Symbol Format window, which allows you to control many aspects of the appearances of the selected symbols. For detailed information, see *Symbol Format Properties* on page 200.
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.

Menu Item	Function
Shadow [Ctrl +W]	Applies the standard shadow effect to the selected symbols. See also *Symbol Format Properties* on page 200.
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 the Format Display Preferences) to the selected objects.
Fit to Page	Opens the Fit to Page dialog, which lists the number of pages currently used and the display scale, and allows you to specify the number of pages to use and to center the symbols on the pages.
Auto-Layout	Automatically rearranges the symbols in the diagram See *Auto-layout* on page 195.
Align	Each of the submenu options aligns the selected symbols in a different manner. For a list of options, see *Aligning selected symbols* on page 197.
Disposition	Automatically arranges all (or selected) symbols in the diagram. There are various forms of disposition: • Horizontal [Ctrl+H] – straightens selected link objects and makes them horizontal where possible. • Vertical [Ctrl+L] - straightens selected link objects and makes them vertical where possible. • Flip Horizontal – reverses the horizontal disposition of the selected symbol • Flip Vertical - reverses the vertical disposition of the selected symbol • Arrange Symbols – distributes 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 – returns text objects associated with the selected link symbols to its default position.

Menu Item	Function
Order	Promotes or demotes the selected symbols in terms of layers within the diagram. This can be useful when you have overlapping symbols and want to have one appear above the other. The following options are available: • Bring to Front • Send to Back • Bring Forward • Send Backward By default, when you insert a free symbol (for example, a note) on a design object symbol (for example, a table), the free symbol is always inserted at the back, as free symbols are usually used as backgrounds. Priority is given to the front-most symbols. When symbols overlap, it may not be possible to select the symbol in the background, even if its handles are visible.
Group Symbols	Groups selected symbols, allowing them to be selected, moved and resized as a single block.
Ungroup Symbols	Separates selected objects that were previously grouped together.
Hide Symbols	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 Symbols	Opens the Show Symbols dialog box, in which you can select or deselect all the symbols in the diagram to show or hide them.
Protect Symbols	Protects the selected symbols, making them impossible to select and edit.
Unprotect Symbols	Unprotects the selected symbols, making them available to select and edit.

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.

Note: Auto-layout is not available for the OOM sequence diagram or any diagram containing swimlanes.

1. Select **Symbol > Auto-Layout** to open the Auto-Layout window.
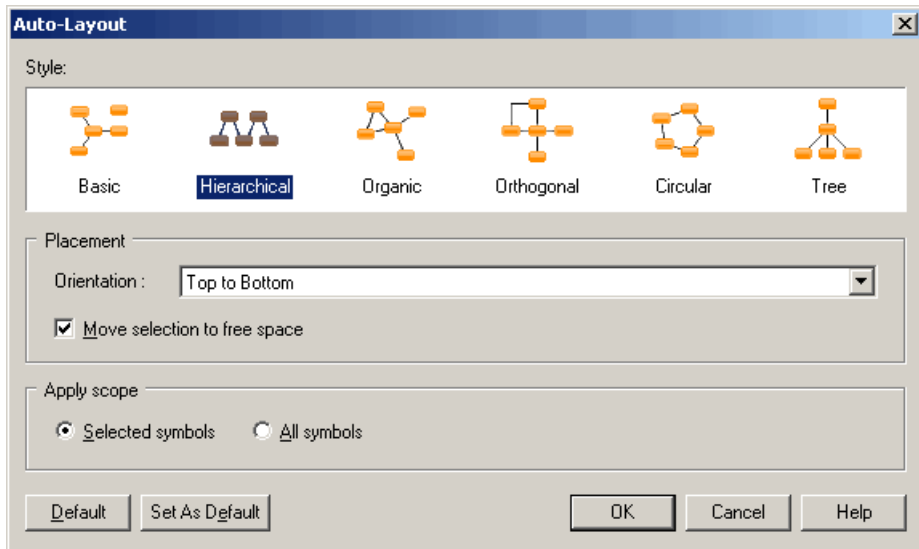
2. Only the styles relevant for the current diagram can be selected. Choose one of the available styles:

    • 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* on page 137) 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.

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.

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)
	Evenly space vertically - Assigns equal space between three or more symbols on a vertical axis

Showing and Hiding Symbols

For PowerDesigner objects that have symbols, the symbol is automatically displayed in the diagram when you create the object. Hiding some symbols can improve the readability of your diagram. When you hide a symbol you do not delete it or the object.

To open the Show Symbols dialog, select **Symbol > Show Symbols** or right-click the diagram background and select **Diagram > Show Symbols**:
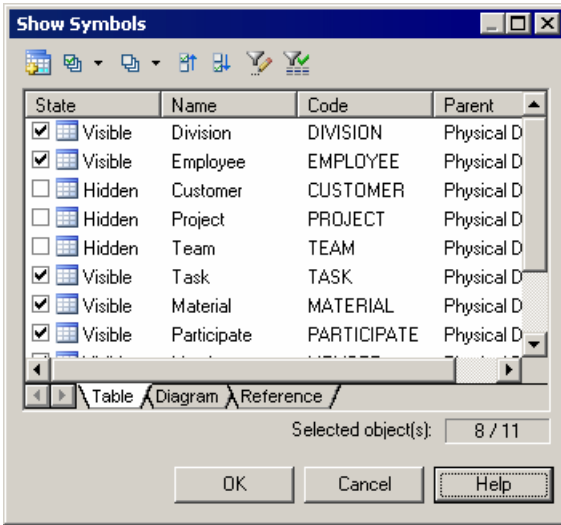
Object symbols are organized by type in sub-tabs. If shortcuts, traceability links and/or free symbols are present in the package, they appear on their own sub-tabs. You show and hide symbols by selecting or deselecting them.

The following tools are available for selecting symbols:

Tool	Description
	Allows you to select objects from other packages to display them in the current diagram.
	Selects all objects in the current tab. You can select all objects on all tabs by pressing the **Ctrl** key as you click or by clicking the arrow and selecting All Pages.
	Deselects all objects in the current tab. You can deselect all objects on all tabs by pressing the **Ctrl** key as you click or by clicking the arrow and selecting All Pages.
	Moves all selected object to the top of the list
	Moves all selected object to the bottom of the list

Note: To hide one or more symbols directly from the diagram, select them and then choose **Hide Symbols** from the **Symbol** or contextual menu. To show a symbol without using the dialog, drag the object from the Browser or another diagram into the diagram. If you drag the symbol from another package or model, a shortcut to the object will be created in the destination model.

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.

Understanding Automatic Link Completion
The following rules apply to the display of link objects in the diagram window.

General Rules

Object	Check box	Result in the diagram...
Link	Selected	Objects at both ends are automatically displayed
Any	Deselected	Link objects attached to the hidden object are automatically hidden
Non link	Selected	Link objects attached to the displayed non link object are automatically displayed if their other ending object is already displayed

CDM Special Behavior

Object	Check box	Result in the diagram...
Association	Selected	Entities and association links attached to the displayed association are automatically displayed
Association link	Selected	Entities and associations attached to the displayed association link are automatically displayed
Inheritance	Selected	Entities (parent and children) and inheritance links attached to the displayed inheritance are automatically displayed
Inheritance link	Selected	Entities (parent and children) and inheritance attached to the displayed inheritance link are automatically displayed

Object	Check box	Result in the diagram...
Parent entity	Deselected	Inheritances for which the hidden entity is the parent are automatically hidden

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* on page 211).
- 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* on page 204.

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* on page 214). 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.

Property	Description
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* on page 205). Selecting this option prevents the symbol from growing to accommodate 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.
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.

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.

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.

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* on page 205): • 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.
Display name	Specifies where the symbol name should appear (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* on page 216).

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.

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* on page 216). 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.

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.

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* on page 218).

For detailed information about extending objects and creating custom symbols for them, see *Customizing and Extending PowerDesigner > Extension Files*.

Finding a Symbol in the Diagram from an Object List

You can locate any object in a diagram from an object list using the PowerDesigner Find feature.

1. **Model >** *Object type* .
2. Select an object in the list.

An arrow is displayed at the beginning of the line.

3. Click the Find Symbol in Diagram tool.

The symbol is selected and centered in the diagram window. To see the symbol, you have to move the list dialog box.

When an object has several symbols within a model, a symbol selection dialog box is displayed that lets you select an object symbol among all instances of the object within the model diagrams. When you click OK the symbol is selected and centered in the current diagram window. To see the symbol, you have to move the list dialog box.

Note: You can also find the symbol of an object in a diagram by right-clicking the object in the Browser and select Find in Diagram (or, for an RQM, Find in Document View) from the contextual menu. The symbol of the object is centered and selected in the diagram window.

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.

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.

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.

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.

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* on page 200).
- From the Symbol Format dialog **Fill** tab, click the **Modify** button in the **Fill Effects** group box (see *Symbol Format Properties* on page 200).
- Select **Edit > Import Image** (see *Importing and Exporting Model Graphics* on page 223).
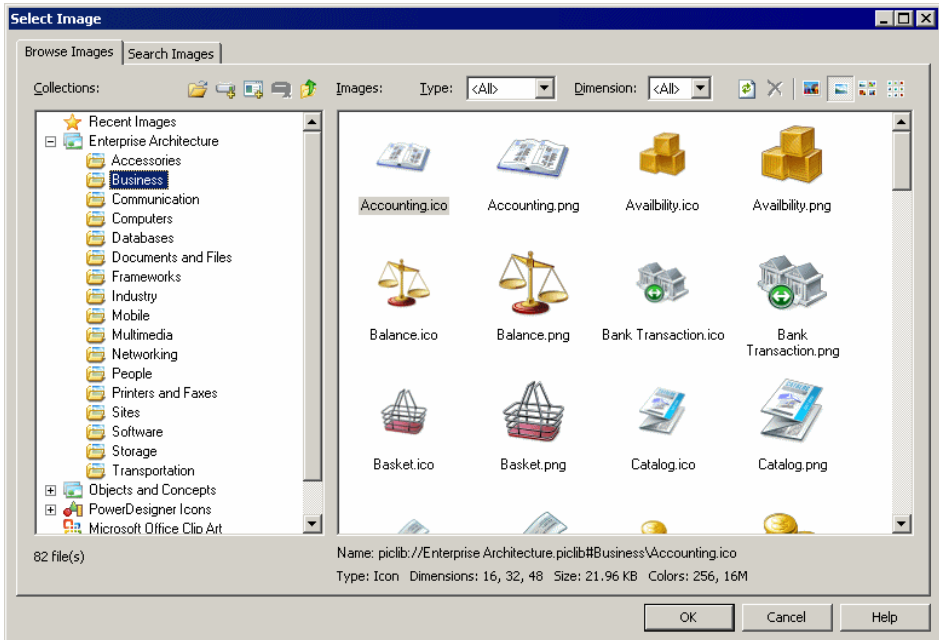
2. Click a collection in the **Collections** pane to display its available images in the **Images** pane.

   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 sub-directories 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.

Tool	Description
	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.

Searching for Images

You can search for images among your collections on the **Search Images** tab of the Select Image dialog.
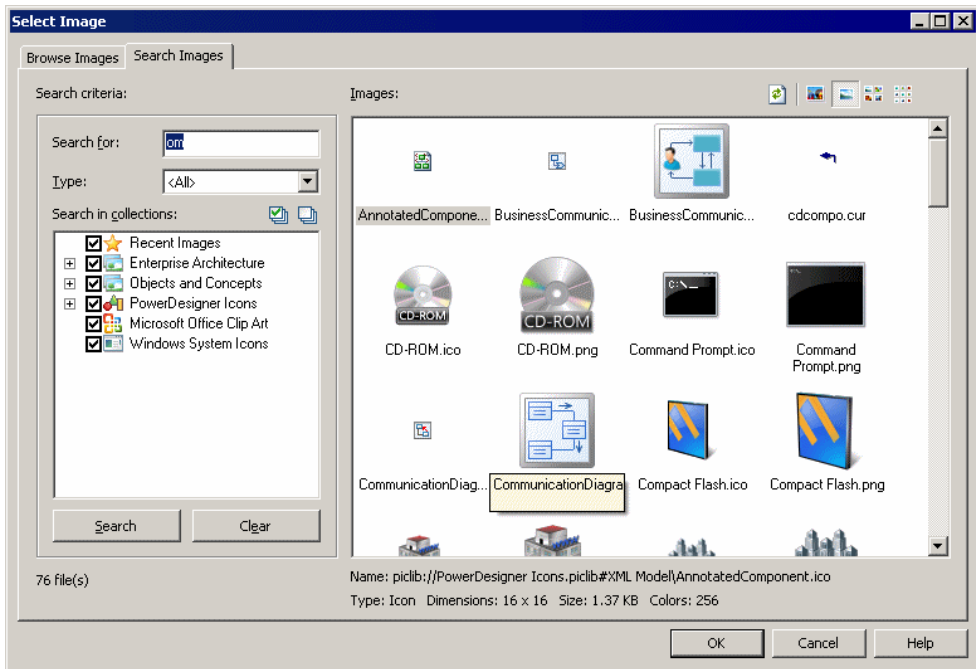
1. Specify the appropriate search criteria in the **Search Criteria** pane. You can specify:

   - **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.

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 *Creating Traceability Links* on page 429)
	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.

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* on page 210).

To change the format of the symbol, right-click it and select **Format** (see *Symbol Format Properties* on page 200).

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.

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.

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.

Tools	Description
	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.

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

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.

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* on page 214).

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**:
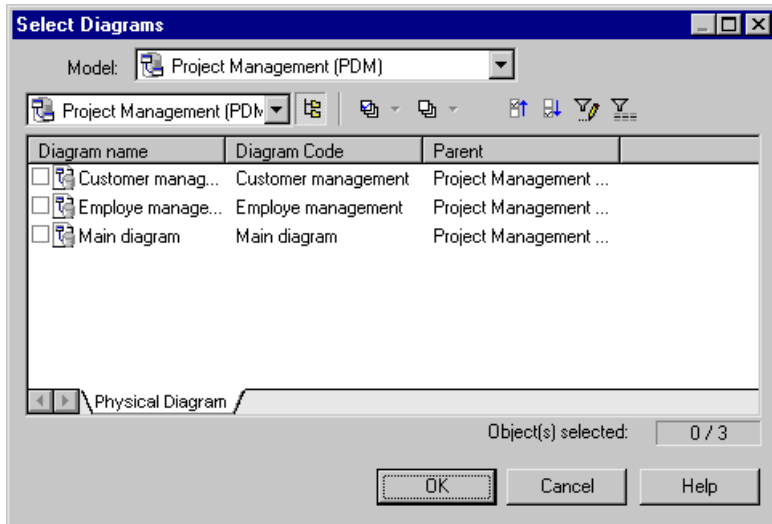
You will be asked whether you want to apply your changes to all the symbols in the selected diagrams. If you click:

*   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:

Changing Formats   ☒

Do you want to apply the format changes to:

⦿ All symbols

○ Selected symbols

○ New symbols

[OK   ]
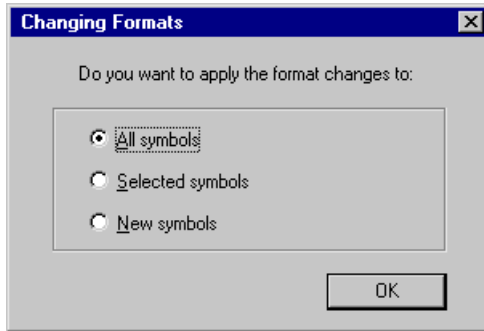
The choices in this dialog apply only to the present diagram and will not override your choices made in the Select Diagrams dialog.

6. Click **OK** to apply your changes

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*.
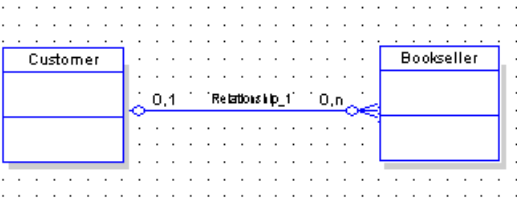
General Settings Display Preferences

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
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.
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:

Property	Description
Show page de-limiter	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.

Preference	Description
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* on page 453).

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* on page 217).

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.
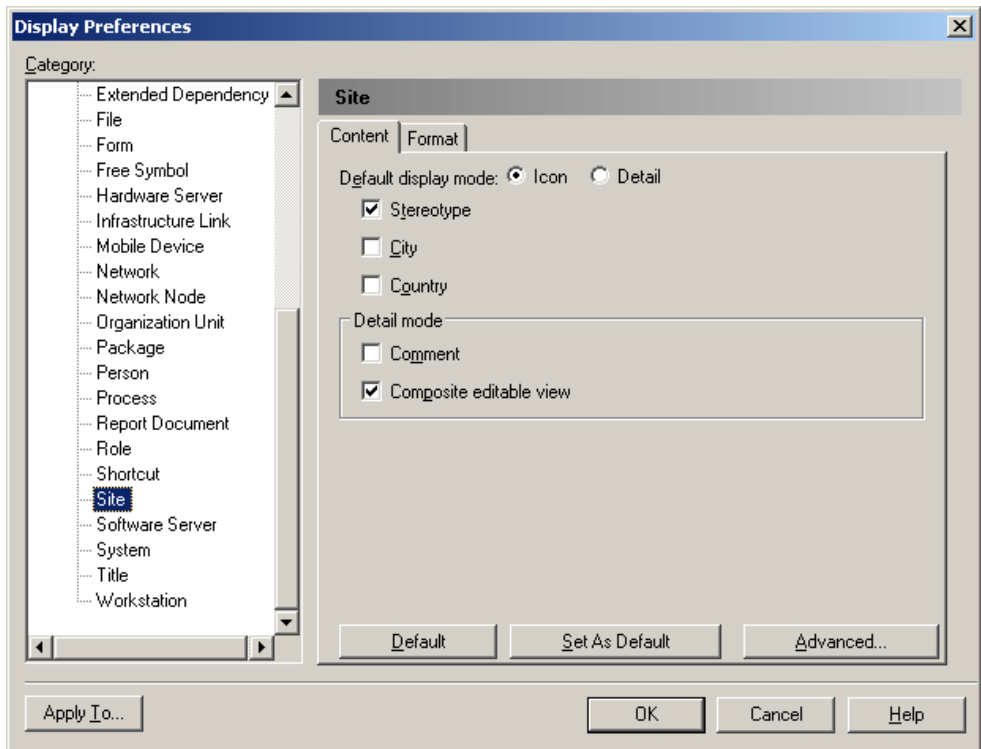
Choose the information that you want to display by selecting the appropriate checkboxes. You can customize the properties that are available to be displayed by clicking the **Advanced** button (see *Customizing Content Display Preferences* on page 218).

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. **Note:** Multi-line fields and collections (such as table columns or class operations) cannot be displayed in icon mode or on link symbols
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

Format Display Preferences

The **Format** tab allows you to control the look of object symbols.

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* on page 200).

Note: The Preview box displays the format of the new objects you are going to create not the format of existing objets. In the case of existing objects, only the attributes modified in the display preferences are modified in the diagram

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.
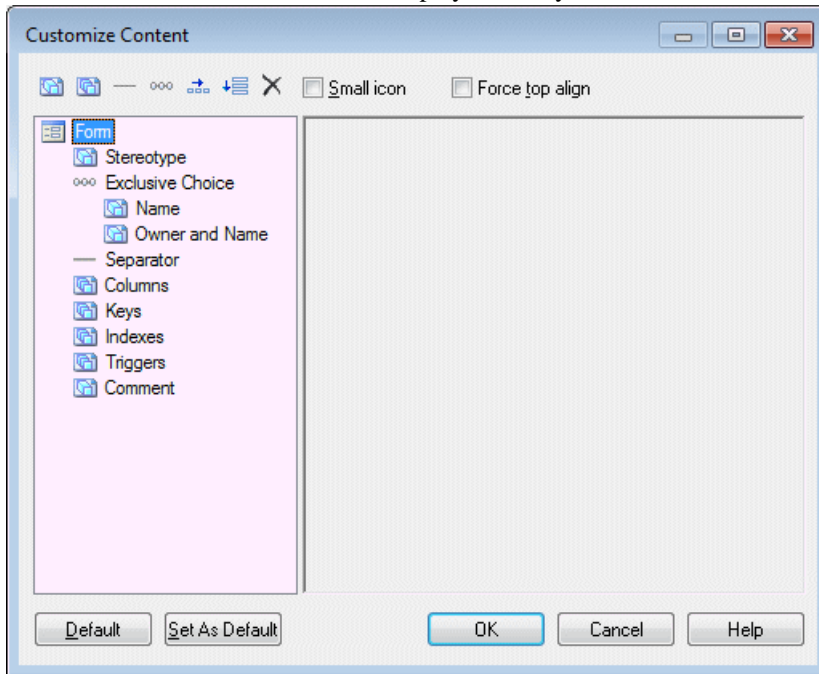
The list in the left hand pane shows the properties that are currently available to display on the symbol in the Display Preferences dialog. In the case of link symbols the list contains three sublists, Source, Center, and Destination, which allow you to specify different content to display on each of 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* on page 220).
	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* on page 221).
	Add Separator Line – inserts a separator line after the item selected in the list. Separators added to horizontal layouts display as vertical lines.

Tool	Description
⠿	**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* on page 222).
(icon)	**Add Horizontal Layout** - inserts a horizontal layout. To arrange attributes to display side by side, drag them onto the horizontal layout in the list.
(icon)	**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.

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**.
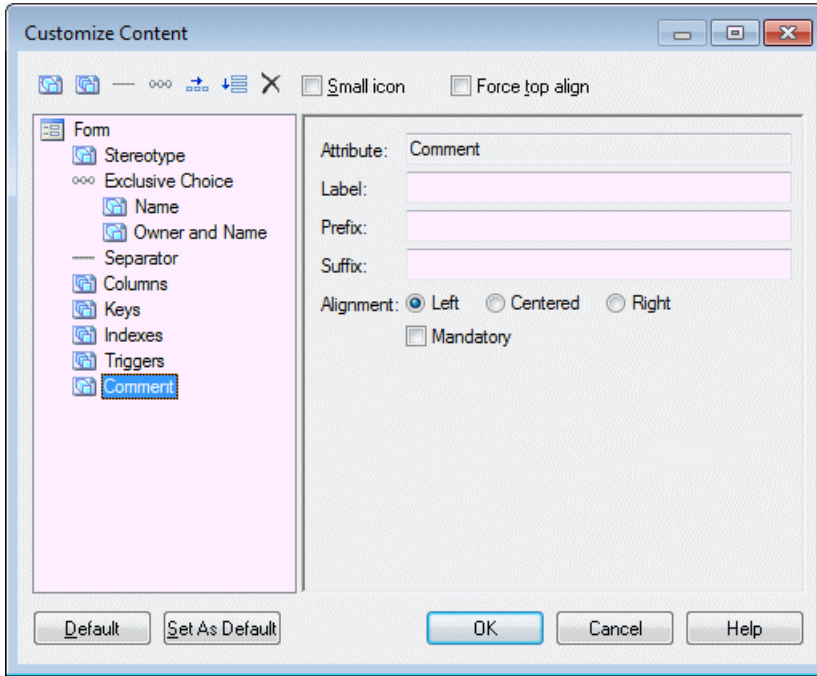
Click on an attribute in the list to set the following options:

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.

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**.
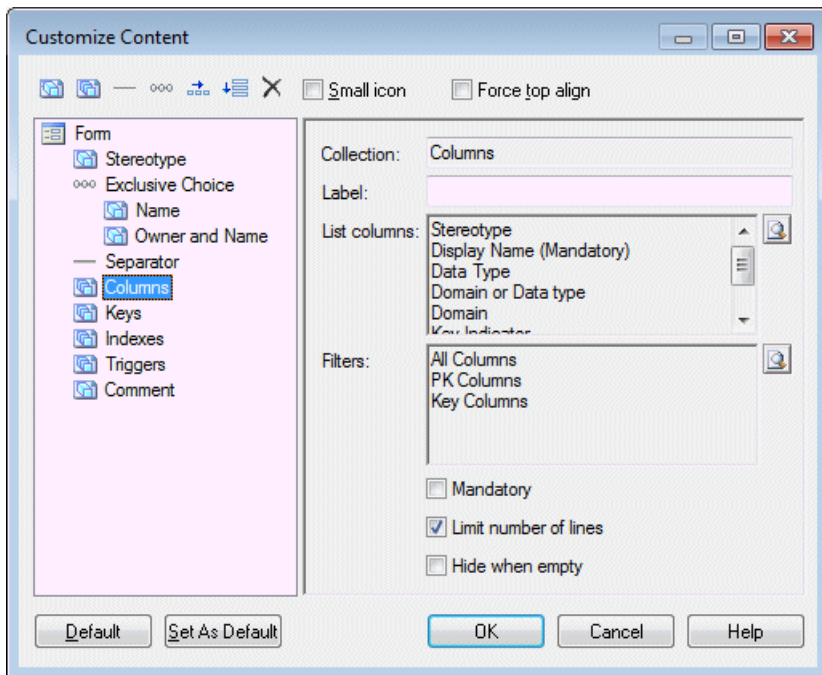
Click on a collection in the list to set the following options:

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.
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.

Option	Description
Hide when empty	Specifies that the symbol compartment containing the collection is hidden when there is nothing to display.

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.
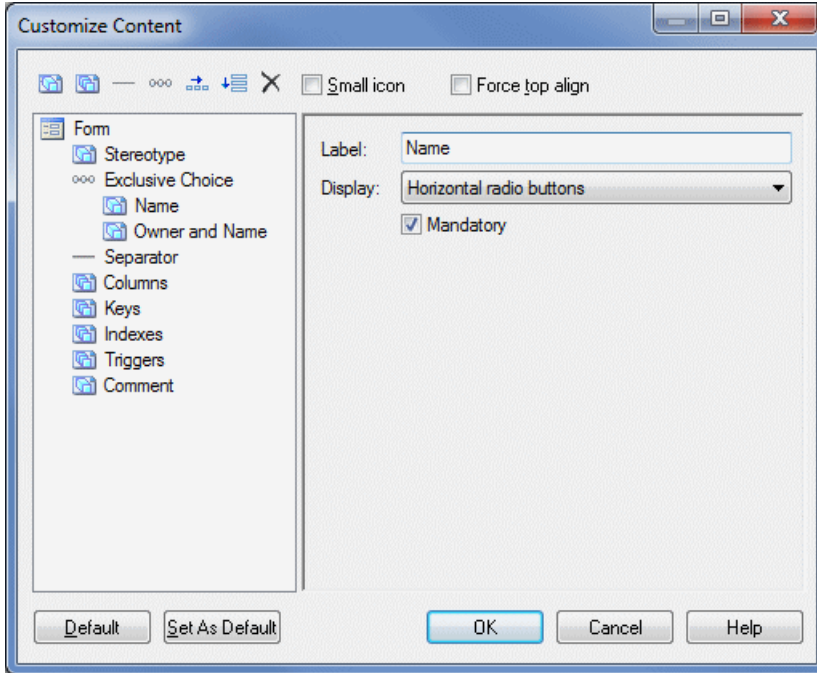
Click on an exclusive choice in the list to set the following options:

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.

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* on page 205), 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

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.

CHAPTER 6   **Reports**

Reports allow you to publish information about your model, and can be used to provide documentation for your system.

You can create the following types of report:

- A *model report* – documents the contents of a model, listing all or a selection of its objects, and showing how they are associated with one another. Listed in the browser within the Reports folder beneath its parent model, and saved with the model. See *Creating a Model Report* on page 225.
- A *list report* – documents a single object type, and can help you to see, for example, the type of each table in a physical data model (PDM). Listed in the browser within the List Reports folder beneath its parent model, and saved with the model. See *Creating a List Report* on page 236.
- A *multi-model report* – documents one or more models, and can help you to see, for example, to which table in a physical data model (PDM) an entity in a conceptual data model (CDM) corresponds. Listed in the browser as a top-level object, and saved as a *.mmr* file. See *Creating a Multi-Model Report* on page 243.

Creating a Model Report

There are two steps to creating a PowerDesigner model report:

- *Define the contents and format of the report* - PowerDesigner provides a variety of ways to define model reports:
    - *Standard Report Templates* – to generate a model report directly, without any configuration. See *Creating a Report with a Report Template* on page 226.
    - The *Report Wizard* – to easily control the generation of a model report with minimal intervention. See *Creating a report with the Report Wizard* on page 226.
    - The *Report Editor* – to have full control over the content and format of your reports. See *Creating a report with the Report Editor* on page 234.
- *Generate the report* – PowerDesigner supports generating reports in HTML or RTF, or will print them directly. See *Generating a model report* on page 235

You can combine these methods as necessary. For example, you could create an initial report with a standard template, refine your selection of objects with the wizard, and then fine-tune formatting in the editor.

Creating a Report with a Report Template

The easiest way to create a model report without any configuration is to use one of the standard model report templates. These allow you to generate a report directly in HTML or RTF format, without specifying objects or formats.

The following standard templates are available:

- *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.

You can also create your own report templates (see *Report Templates* on page 272).

Report templates are available in the Generate Report window. To generate a report using a template, see *Generating a model report* on page 235.

Creating a Report with the Report Wizard

The Report Wizard is an easy way to select which objects will appear in your report, and to control their formatting.

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 exit the wizard and create a report based on your selections so far.

The Report wizard can be launched from your model, to create a new report, or from the Report Editor window of an existing report.

1. Select **Report > Report Wizard** (or select **Report > Reports**, and then click the Report Wizard tool) to launch the wizard:
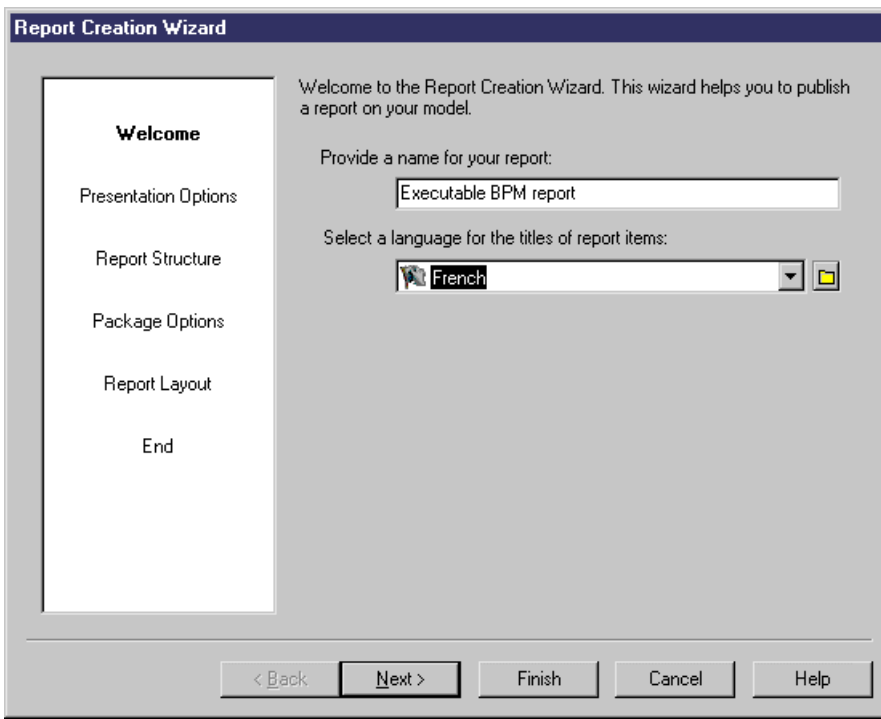
2. The Welcome page allows you to specify a name for the report and the language in which you want its titles to appear. Note that this page is not displayed if you launch the wizard from the Report Editor.

When you are satisfied, click Next:
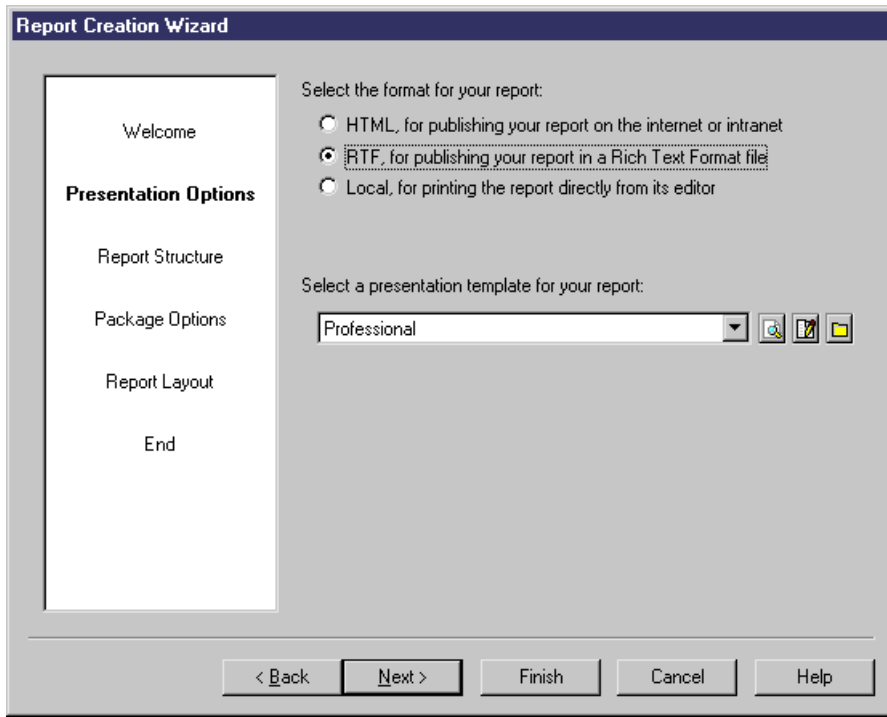
3. The Presentation Options page allows you to specify a format for your report. You can choose between:

- *HTML, for publishing your report on the internet or intranet* – you can additionally select, and preview, a presentation template using the tools to the right of the template field.
- *RTF, for publishing your report in a Rich Text Format file* – you can additionally select, preview, and edit, a presentation template using the tools to the right of the template field.
- *Local, for printing the report directly from its editor*.

When you are satisfied, click Next:

Report Creation Wizard

Choose what information you want to see in your report. The preview pane on the left hand-side shows your choices in a sample report.

Report Structure

Table of Contents
1. Introduction
  1.1 Description
  1.2 Card of model
2. Short model description
  2.1 Diagram 1
  2.2 List of objects
  2.3 List of links
3. Full model description
  3.1 Diagram 1
  3.2 Objects
    3.2.1 List of objects
    3.2.2 Object 1
      3.2.2.1 Card of Object 1
    3.2.3 Object 2
      3.2.3.1 Card of Object 2
  3.3 Links
    3.3.1 List of links
    3.3.2 Link 1
      3.3.2.1 Card of Link 1

☑ Introduction section
  ☑ Introduction text   [Define...]
  ☑ Model properties

☑ Short description section
  ☑ Display diagrams
  ☑ List objects

☑ Full description section
  ☑ Display diagrams
  ☑ List objects before full description
  ☑ Objects detailed description

☐ List objects diagram by diagram

[< Back] [Next >] [Finish] [Cancel] [Help]

4. The Report Structure page allows you to specify the kinds of information that will appear in your report. 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* – specifies an introduction sub-heading, preceded by a page break. Makes available the following sub-options:
  - *Introduction text* - Text paragraph. Click the Define button to open an editor.
  - *Model properties* - Model card
- *Short description section* – specifies a short description section sub-heading, preceded by a page break. Makes available the following sub-options:
  - *Display diagrams* - [not for RQM] Diagram book, Graphics, Diagram description, Diagram annotation
  - *List objects* - List item for each object type in the model
- *Full description section* – specifies a full description section sub-heading, preceded by a page break. Makes available the following sub-options:
  - *Display diagrams* - [not for RQM] Diagram book, Graphics, Diagram description, Diagram annotation
  - *List objects before full description* - Title item and list item for each object type
  - *Objects detailed description* - Book item for each object type
- *List objects diagram by diagram* - [not for RQM] Diagram book - Instead of a flat display of object types, it allows you to display objects sorted by diagram in which they

are displayed. Global objects, such as business rules for example do not belong to a diagram but to the whole model, and therefore are listed under a specific Title book.

When you are satisfied, click Next:
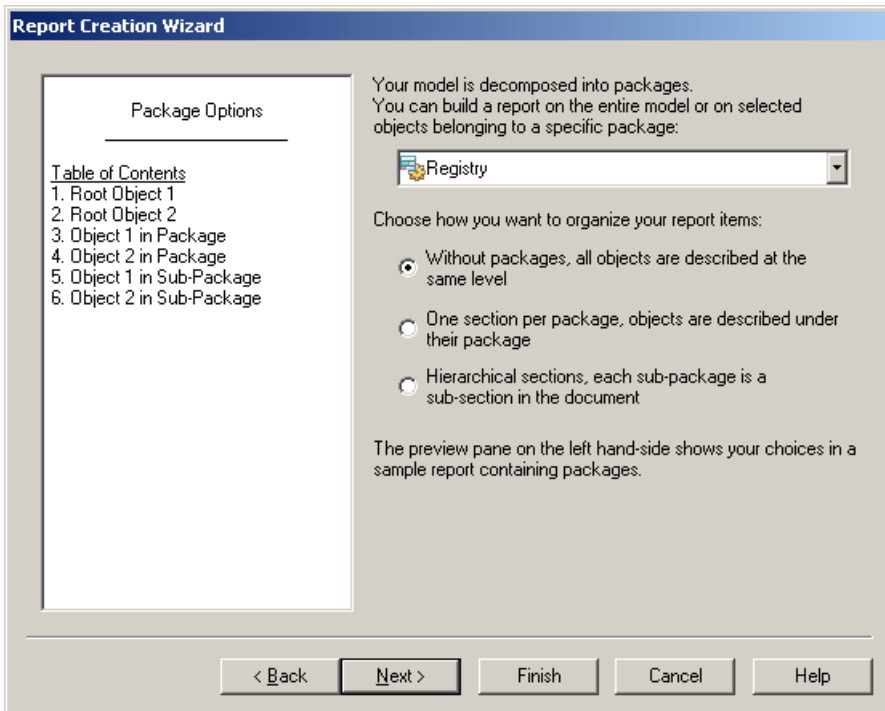
5. The Package Options page is displayed only if your model contains packages, and allows you to control the treatment of packages within the report. The preview pane (on the left hand-side) changes dynamically in order to preview your selection. You can select whether to base the report on the entire model or on a sub-package, and then choose between the following options:

    *   *Without packages, all objects are described at the same level* [not available for HTML or an RQM] – Lists all the objects together on the top level, whether or not they belong to a package.
    *   *One section per package, objects are described under their package* [not available for HTML or an RQM] – Groups objects by the package to which they belong.
    *   *Hierarchical sections, each sub-package is a sub-section in the document* - Groups objects by the package to which they belong, and groups packages beneath their parents to recreate the package hierarchy.

    Note that:

    *   Global objects (such as business rules) are always included, even when you select a package.

• Composite objects (such as activities, or processes in an OOM) are always displayed in an Hierarchy, even if you have selected a different package option.

When you are satisfied, click Next:
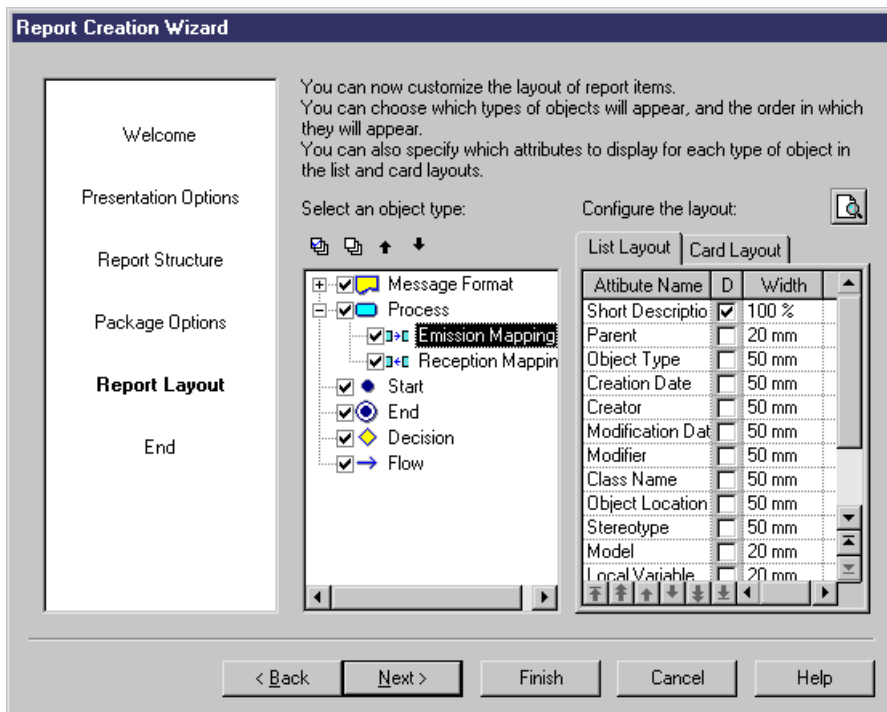
6. The Report layout page allows you to select the types of object to include in the report and to configure the layout of the object lists and cards.

   To specify the inclusion of an object type, select its checkbox in the list. The list includes only those objects that are present in the selected model or package. Click on the plus sign to the left of an object to reveal and select or deselect its sub-objects.

   Stereotypes used as metaclass display in the list of object types, below those of their parent object (see *Reporting on stereotypes as metaclasses* on page 233).

   The Select All and Deselect All tools are available above the list. You must select at least one object type to proceed with the wizard. The Move Up and Move Down tools allow you to promote or demote the position of an object (though child object types cannot be moved beyond the scope of their parents.

7. Select an object type in the object type list, and the click the List Layout or Card Layout tab in the right-hand pane to configure its display. These tabs list the object attributes that can be displayed in the report. The following parameters are available:

- *Attribute Name* – You can use the arrows at the bottom of the list to promote or demote the attribute in the list
- *D[isplayed]* – Select the checkbox to enable the display of the attribute
- *Width* [List Layout only] – Each attribute selected for display will be a column in the list. Use this parameter to specify the width of the column either as a percentage or in millimeters.

8. When you are satisfied, click Next:
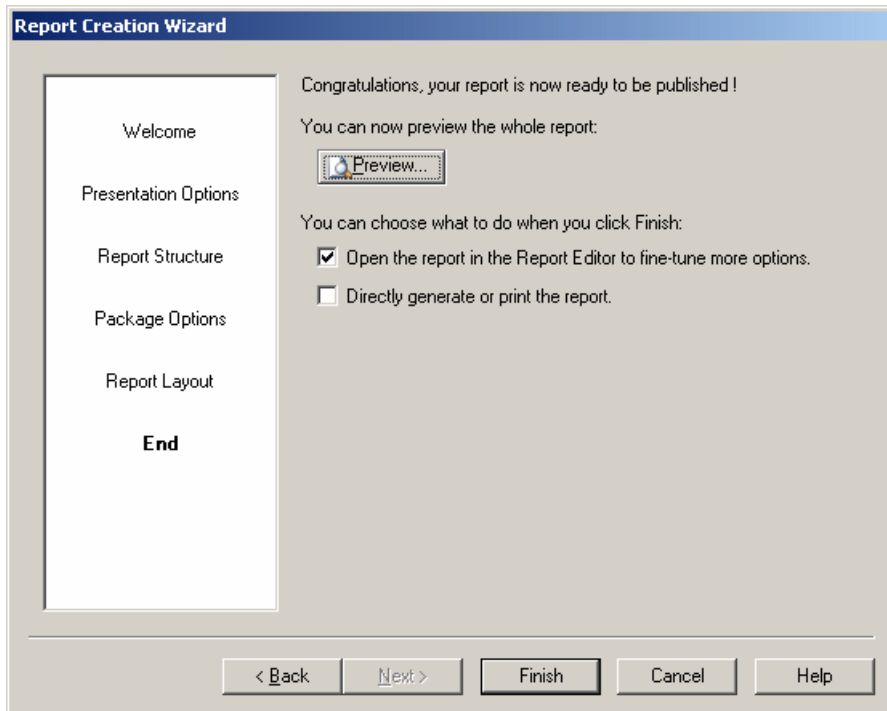
9. The End page allows you to decide what you will do with your newly created report. Click the Preview button to preview your report in:

- your Browser – for HTML reports
- your RTF Editor – for RTF reports
- the PowerDesigner preview window – for reports to be printed directly.

Before clicking Finish to complete the report creation, you can select one or both of the following options:

- *Open the report in the Report Editor to be able to fine-tune more options*
- *Directly generate or print the report*

When you are satisfied, click Finish to exit the wizard and create the report.

Reporting on Stereotypes as Metaclasses

When a stereotype is defined as "Use as metaclass" in a language, DBMS or extension, it inherits the attributes of a standard metaclass.

The report wizard support metaclass stereotypes as follows :

• Lists metaclass stereotypes in the list of object types, below the parent object in the Report Layout page.

• Generates a book and a list separately for each metaclass stereotype, below those of the parent object in the Report Editor.

For example, you create a stereotype "used as metaclass" (called metaclass stereotype) in the process language associated with a BPM, then you associate the metaclass stereotype to a process in the diagram and launch the Report Wizard. The metaclass stereotype is listed in the object type list below the parent object in the Report Layout page. Its associated extended attributes if any, are listed in the List Layout and Card Layout tabs.
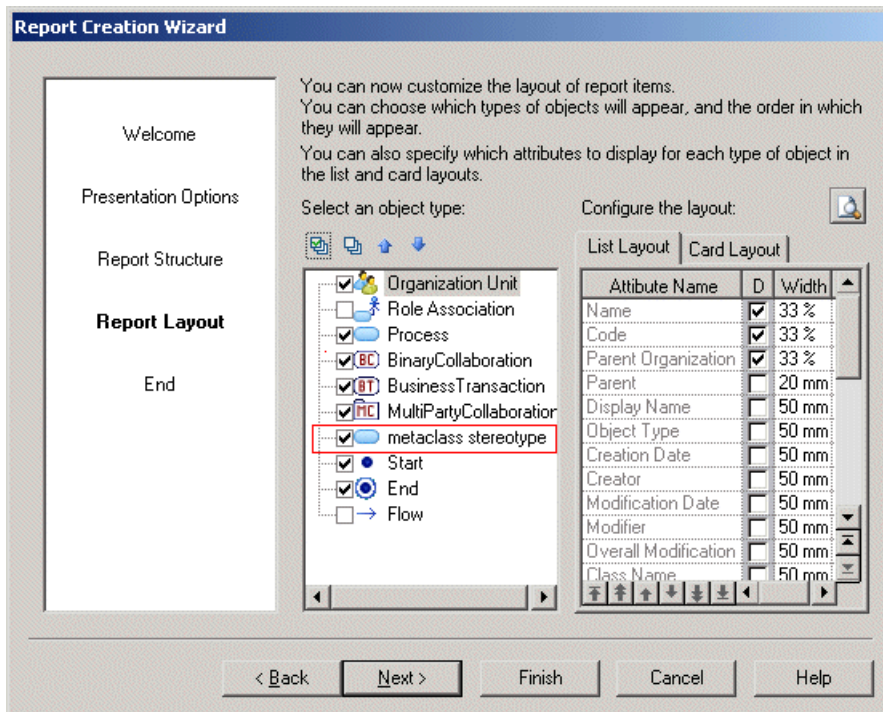
And the generated report shows the list and the book for the metaclass stereotype below those of the parent object in the Report Editor:

```
Section_1
  Table of contents
  Title - Short model description
    List of Diagrams - List of diagrams
    Diagram - Diagram %ITEM%
    List of Organization Units - List of organization units
    List of Processes - List of processes
    List of Processes - List of BinaryCollaboration objects
    List of Processes - List of BusinessTransaction objects
    List of Processes - List of MultiPartyCollaboration objects
    List of Processes - List of metaclass stereotype objects
    List of Starts - List of starts
    List of Ends - List of ends
    Process - Process %ITEM%
  Page break
  Title - Full model description
    List of Diagrams - List of diagrams
    Diagram - Diagram %ITEM%
    Title - List of organization units
    Title - List of processes
    Title - List of BinaryCollaboration objects
    Title - List of BusinessTransaction objects
    Title - List of MultiPartyCollaboration objects
    Title - List of metaclass stereotype objects
    Title - List of starts
    Title - List of ends
```

Creating a Report with the Report Editor

The Report Editor gives you complete control over the content and format of your report. You can create a blank report from the List of Reports, open it in the Report Editor, and build it using Report Items.

1. Select **Report > Reports** to open the List of Reports, and click the **New Report** tool.

```
New Report                                    ×
  Report name:      Report 1
  Language:         English              ▼  □
  Report template:  <None>               ▼  □
                    OK      Cancel      Help
```
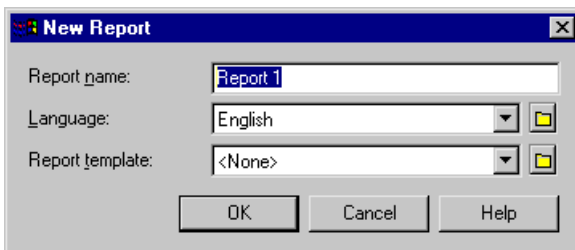
2. Enter a report name, and select a language for the report's titles from the list.
3. [optional] Select a report template from the list to prepopulate the report (see *Report Templates* on page 272).
4. Click **OK** to create the report and open it in the Report Editor (see *The Report Editor* on page 245).

Generating a Model Report

The Generate Report window also allows you to access the standard report templates, or templates or reports that you have previously created, and to generate them in HTML or RTF format, or to print them directly.

1. Select **Report > Generate Report** to open the Generate Report window.
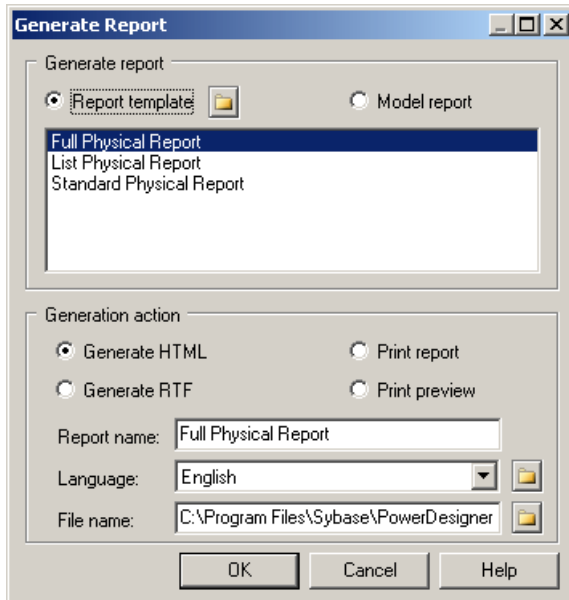
2. Select a Generate report option. You can choose from the following options:

- Report template – PowerDesigner provides standard report templates to select the objects to include in your report and to provide formatting. You can also create your own templates. You can select an alternative folder to look for additional templates by clicking the Browse button to the right of the radio button.
- Model report – lists any reports that you have previously created in the model.

3. Select the appropriate template or report from the list.

4. Select a Generation action. You can choose from the following options:

- Generate HTML – Generates a frameset named after the File name field, together with a folder containing the necessary HTML and GIF files. For example MyReport.html would have an associated folder called MyReport_files.
- Generate RTF - Generates a Rich Text Format file named after the File name field.
- Print report – Opens the Windows Print window to allow you to directly print the report.

- Print preview – opens the print preview window, from which you can choose to print the report, or to generate it as HTML or RTF.

5. [for HTML or RTF only] Specify a report name and file name for the generated file.

6. Specify the language in which to generate the report.

7. Click OK to begin the generation. The Generation action specified above will be performed.

Note: If you want to quickly generate a report that you have already defined, and which is visible in the Browser, right-click its Browser entry, and select Print, **Generate > RTF**, or **Generate > HTML** from the contextual menu.

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.

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* on page 241).

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:
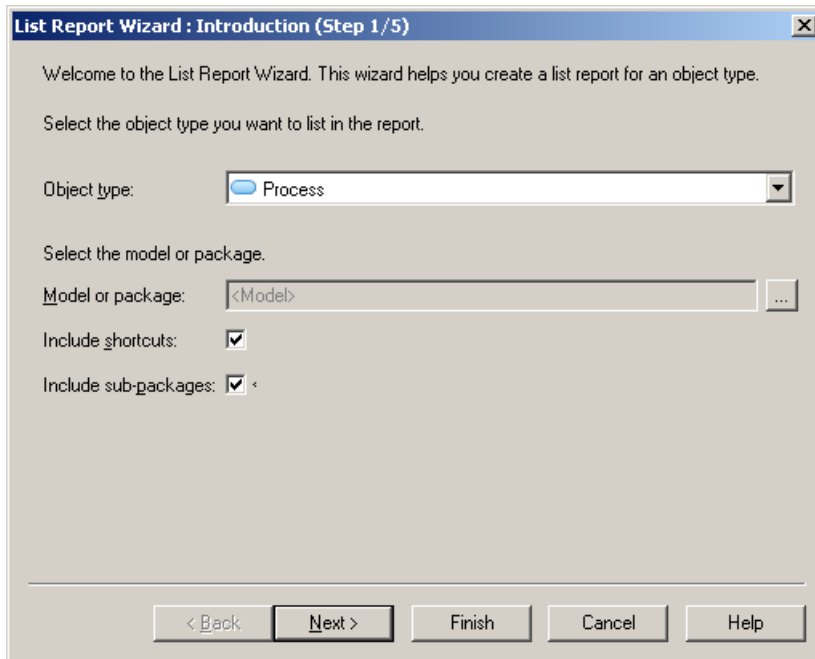
2. Click **Next** to go to the Column Filter page, which allows you to specify which of the object's properties will be included in the list report. Select a property in the left-hand pane to add it to the list in the right-hand pane. You can promote or demote properties in the list using the arrows 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* on page 124). You can specify to sort the list on the values of a property by selecting 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:
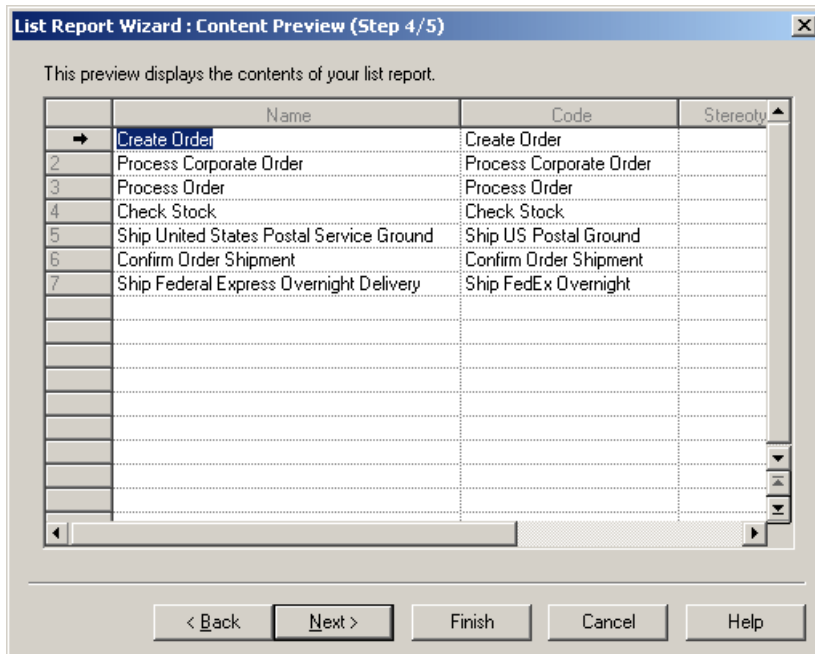
To alter the definition click the **Back** button.

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.

```
List Report Wizard : End (Step 5/5)                                    [X]

   Congratulations, you have created a list report!

   Type a name, a code and a comment for the list report in order to identify and describe it.


   Name:       | Process List Report                                  |  [=]

   Code:       | Process List Report                                  |  [=]

   Comment:    |                                                      |
               |                                                      |
               |                                                      |
               |                                                      |
               |                                                      |
               |                                                      |

   ┌─ ☑ Generate the list report to an external file ──────────────────┐
   │   File format:  ○ CSV      ○ RTF      ○ HTML      ⦿ XML           │
   └───────────────────────────────────────────────────────────────────┘

          [ < Back ]   [ Next > ]   [ Finish ]   [ Cancel ]   [ Help ]
```
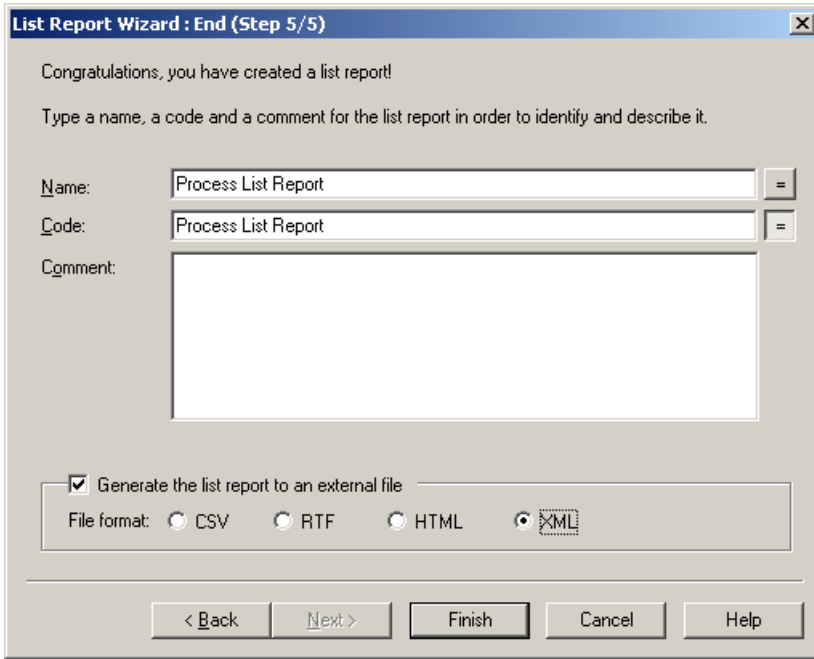
6. Click **Finish** to exit the wizard and create the report.

    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.

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* on page 124). 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* on page 242).
- **Result Sets** - lists the result sets stored for the list report (see *Creating Result Sets* on page 241). Click the **Properties** tool to open a result set for generation.

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.

Generating a List Report

You can generate a list report to a CSV, XLS, RTF, HTML, or XML file.

•   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**.

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* on page 236).

Tool	Description
🖼	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.

Creating a Multi-Model 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.

1. Select **File > New Model** to open the New Model dialog box.
2. Select Multi-Model Report.

   This option may be in the left or right hand pane, depending on the configuration of your New Model dialog (see *Creating a Model* on page 6).
3. Enter a name for the report, specify a language, and select a model from the Model Name list as the basis for the first section of the report.
4. [optional] Select a template (see *Creating a Report with a Report Template* on page 226) from the Report Template list.

   Note: When you use a template in a language different from the one specified for the report, only user-defined items such as Title or Text paragraph will retain the language of the template. Other items will be displayed in the report language.
5. Click OK to create the multi-model report.

   The report opens in the Report Editor, and is added to the Browser:
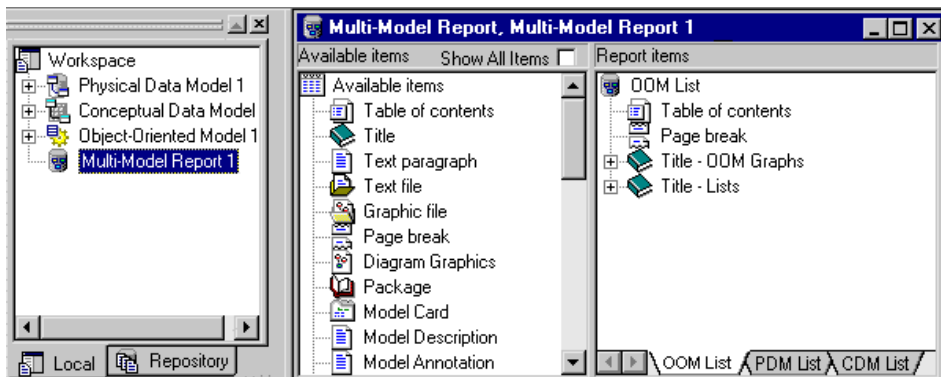
6. Select **File > Save** to save your report.

Note: If you want to quickly generate a multi-model report from an existing one without modifying it, right-click a multi-model report node in the Browser and select Print or **Generate > RTF** or **Generate > HTML**.

Adding a Second Model to a Multi-Model Report

You can create a multi-model report that contains only one model, but in the majority of cases, you will want to add additional models. Each model in the report belongs to a separate *section*, which can be accessed via the tabs at the bottom of the Report Editor window. 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.

1. Open the multi-model report, and select **Report > Report Properties** to open its property sheet.

2. Select the **Sections** tab, and click the **Add a Row** tool.

3. In the **Model** column, select the model you want to add from the list.

4. In the **Sections** column, enter the name that you want the model to have in the Report Editor.

5. Click **OK** to return to the Report Editor (see *The Report Editor* on page 245).

Configuring a Multi-model Report with the Report Wizard

You can use the Report Wizard to configure a multi-model report. The wizard works on a single section at a time, and will delete any content currently in the section.

1. Open the multi-model report, and select the section you want to work on by clicking the appropriate tab at the bottom of the Report Editor window.

2. Click the Report Wizard tool in the Reports toolbar to launch the Report Wizard, and follow the procedure in *Creating a report with the Report Editor* on page 234.

Generating a Multi-model Report

You can generate a multi-model report from within the Report Editor by clicking one of the Generate <format> tools on the Reports toolbar.

Alternatively, you can right-click the multi-model report entry in the Browser, and select **Generate > <format>** from the contextual menu.

The Report Editor

You can edit model reports, multi-model reports, and report templates in the Report Editor. 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.
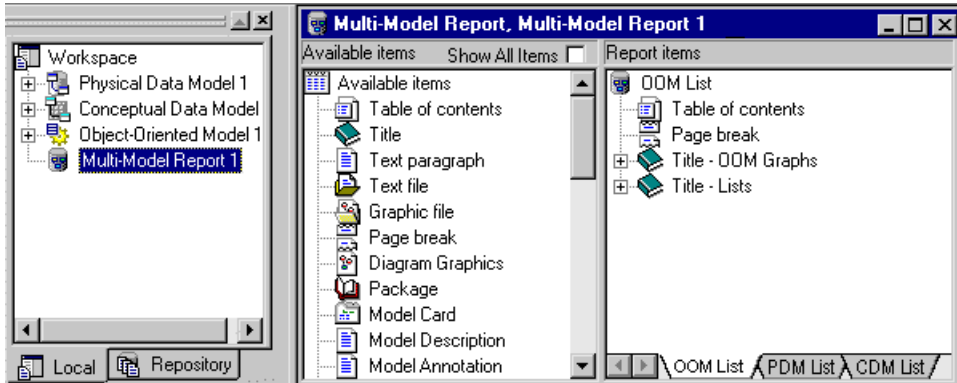
The **Available Items** pane lists only the report items that correspond to objects that exist in the current model. For example, if you have not created any interfaces in your OOM, then the List of Interfaces and Interface book will not be listed. Select **Show All Items** at the top of the pane to 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* on page 226).
	Print Preview - Displays a report print preview (see *Previewing Your Report* on page 247)
	Print - Prints the report.
	Generate RTF - Generates the report as an RTF file.
	Generate HTML - Generates the report as an HTML file.
	Add Item - Adds an item to the Report Items pane.
	Up One Level - Moves item up one level.

Tool	Description
↓	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.

Opening a Report in the Report Editor

You can open and edit any existing report in the Report Editor.

For information about creating a report, see *Creating a Model Report* on page 225.

1. Select **Report > Reports** to open the List of Reports.
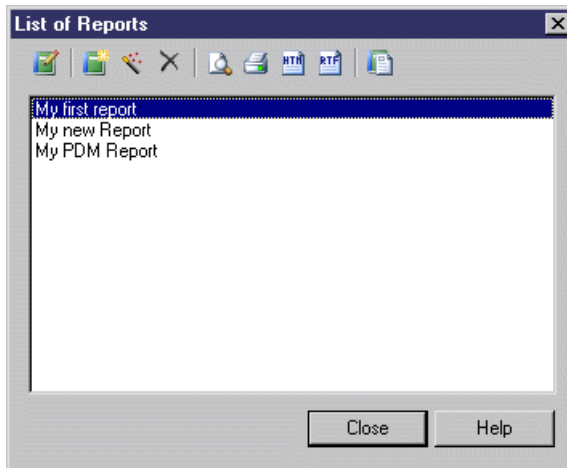
2. Select the report you want to modify and click the Edit Report tool to open it in the Report Editor.

Note: If you want to quickly generate a report from an existing one without modifying it, select a report in the List of Reports then click the Print Report, Generate HTML, or Generate RTF tool.
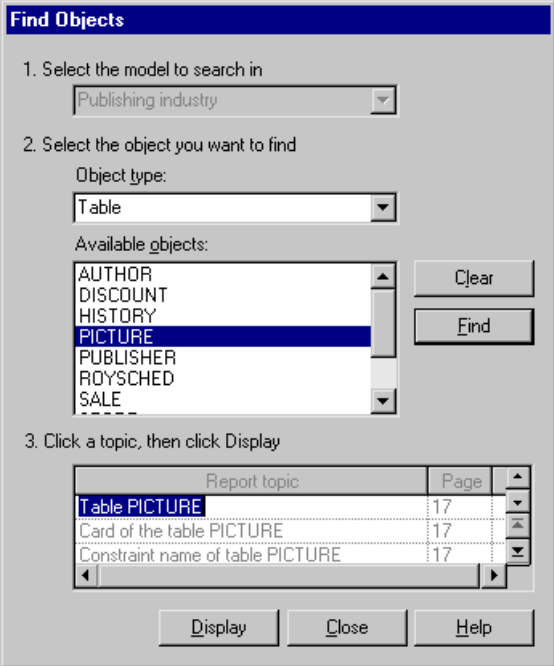
Previewing Your Report

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**.
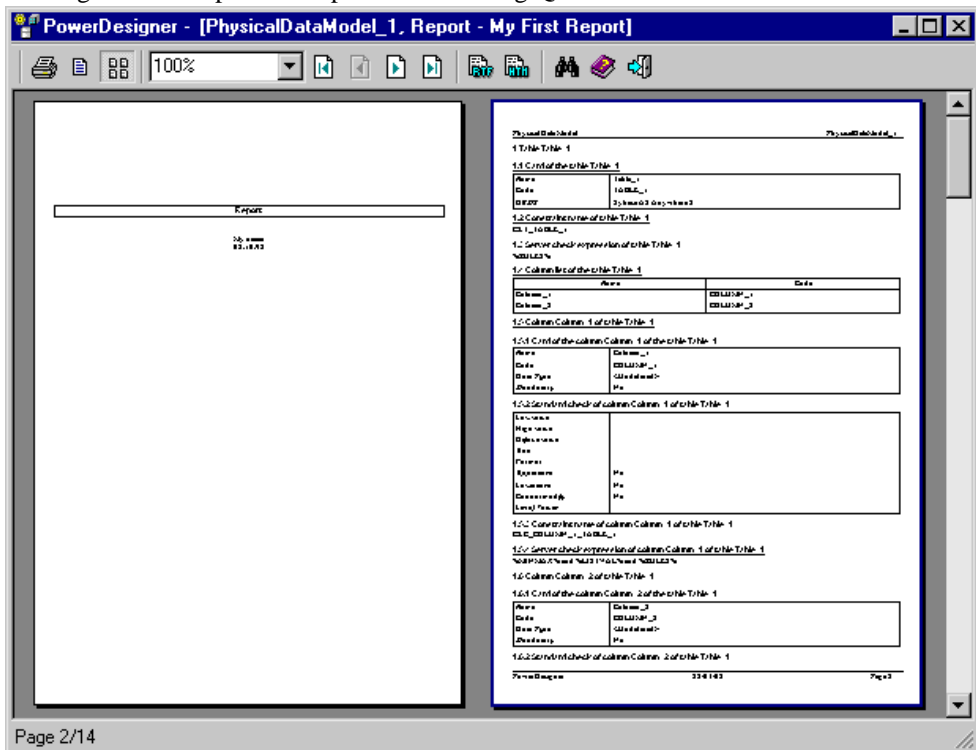
The following tools are available in this window:

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.
HTM RTF	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.

Saving a Model Report

Reports are saved with the model to which they belong.

Note: To reduce the size of your model, you can create its report as a multi-model report, which is saved to a separate file. For more information, see *Creating a Multi-Model Report* on page 243.

Adding Items to a Report

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.
	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 Notes tab of the model or object property sheet • Annotation – contents of the Annotation sub-tab of the Notes 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* on page 261)
	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* on page 260)
	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:
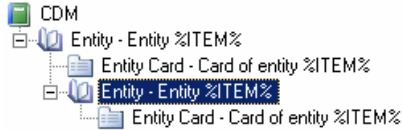
- Package and composite object hierarchy - When you add a package or composite object book item to a report, you can select the Hierarchical Display command in the item contextual menu to display the package or composite object hierarchy in the generated report. You do not need to use this command when 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.

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.

Modifying Items in the Report Items Pane
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.

Command	Description
Show Contents in HTML TOC	[book items] Controls the display of the sub-item titles in the table of contents of HTML reports.
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 up one level in the tree.
Format	Opens the dialog allowing you to modify the formatting of the item (see *Formatting textual report items* on page 256 and *Formatting graphical report items* on page 260).
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* on page 260 and *Controlling the layout of card report items* on page 261).
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* on page 252).
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 *Refining the selection of objects for a report item* on page 254).
Edit Title	Opens the item in a dialog allowing you to edit its title (see *Modifying the title of an item* on page 251).
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* on page 247).

Modifying the Title of an Item

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.

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.

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 Resource Files.*

Modifying the Collection of an Item

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.

Note: If you create extended or calculated collections in your model and use the Report Wizard (see *Creating a Report with the Report Wizard* on page 226) 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:
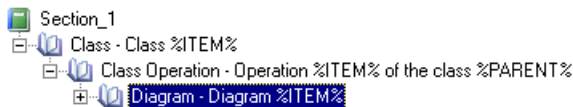
In the following example, in order to manage the "schedule" and "event" concepts, the extended object <<event>> has been created, along with an extended collection of extended objects <<schedule>>. The Extended Object report item is added to the report and filtered on the stereotype Event (see *Refining the Selection of Objects for a Report Item* on page 254) and the List of Extended Objects is added to the Extended Object book, with its collection set to Schedule:
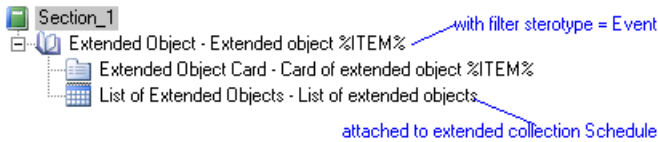
1. Drag and drop the appropriate book or list item under the selected book item in the Report Items pane.

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.

Selecting Objects to Include 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.

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.
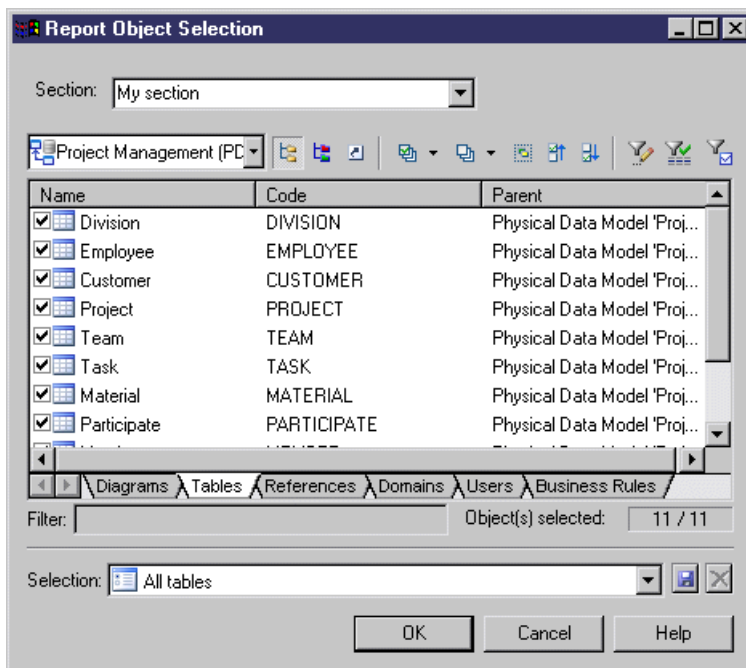
2. Select a report section from the Section list (see *Report Properties* on page 263).

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.

For information about the tools in this dialog, see *Adding an Item from a Selection List* on page 126.

Refining the Selection of Objects for a Report Item

You can refine the selection of objects for individual report items. For example you can have a list of all the tables in your PDM, but you need detailed information about only some of them. You can refine the object selection either by selecting individual objects or by applying a filter. You can also control the ordering of individual objects.

Note: If you have made a selection of objects at the report level (see *Selecting objects to include in the report* on page 253), objects excluded there will not be available for selection at the report item level.

1. Right-click a book or model-dependent list item in the Report Items pane and select Selection (or, for matrix items, right-click the matrix item and select Row Selection or Column Selection) to open the Define Sort and Filter dialog box:
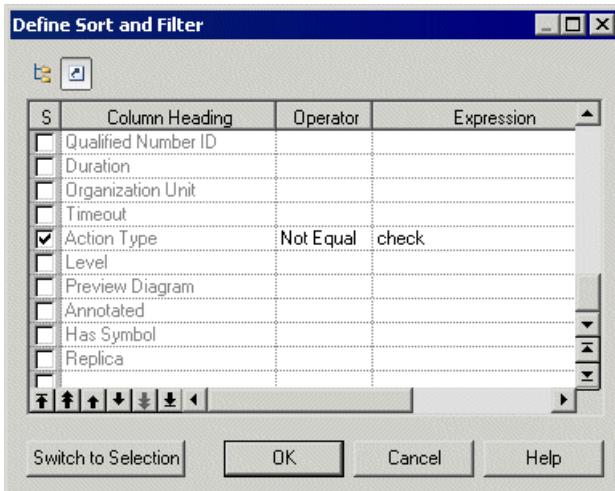
2. [optional] Sort and filter the objects you want to include as follows:

   • Select the S column next to a property to sort by its value
   • Select an operator to apply on the expression value in the Operator column.
   • Enter an expression to filter by in the Expression column, and select the U column to apply the filter to your selection.

For more information about the operators and expression syntax, see *Defining a Filter Expression* on page 124.

- Use the arrow buttons at the bottom of the dialog to reorder properties in the list

3. [optional] If you want to refine your selection by choosing from a list of available objects, click the Switch to Selection button in the lower left corner of the dialog box to display the Select Objects dialog box:

Note: If you click the Switch to Selection button, any information entered in the Define Sort and Filter dialog will be lost.
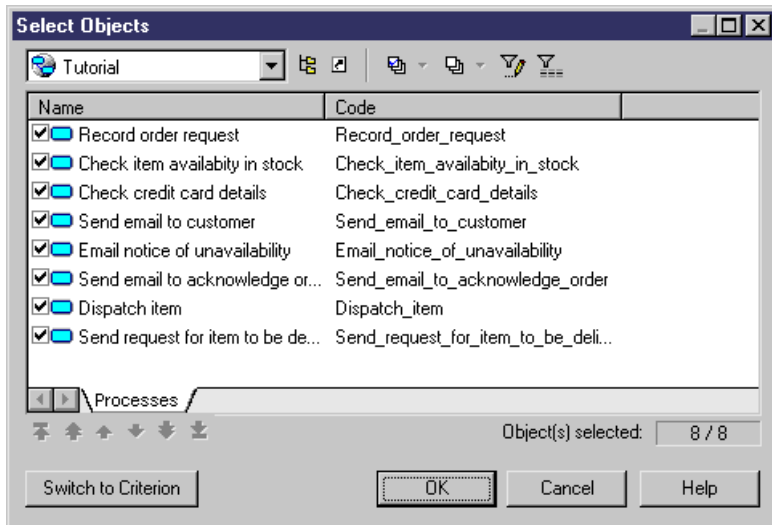
4. Select the objects that you want to include at this place in the report. By default, all the objects are selected. You can reorder the objects in the list by using the arrow buttons at the bottom of the dialog.

Note: If you define a selection of ordered objects on a package for which have selected the Hierarchical Display command (see *Adding items to a report* on page 248), 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.

Formatting Report Items

The formatting of your report is highly customizable. You can set default formats and/or create specific formats for individual report items:

Note: Right-click any item and select Format to access its formatting options.

Specifying Default Fonts for Report Items

You can specify default fonts for report items, report templates and reports. The default fonts are saved in the registry.

1. Select **Report > Change Fonts** from the Report Editor window to open the Change Fonts Properties dialog box:
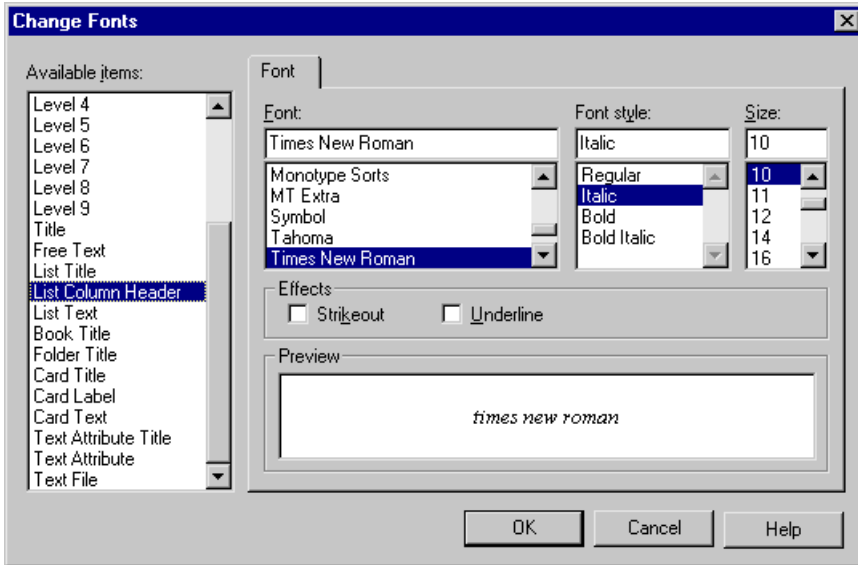
2. Select one or more available items in the left-hand list, and specify a font, style, size, and effects as appropriate.

3. Click OK to return to the Report Editor.

Formatting Textual Report Items

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.
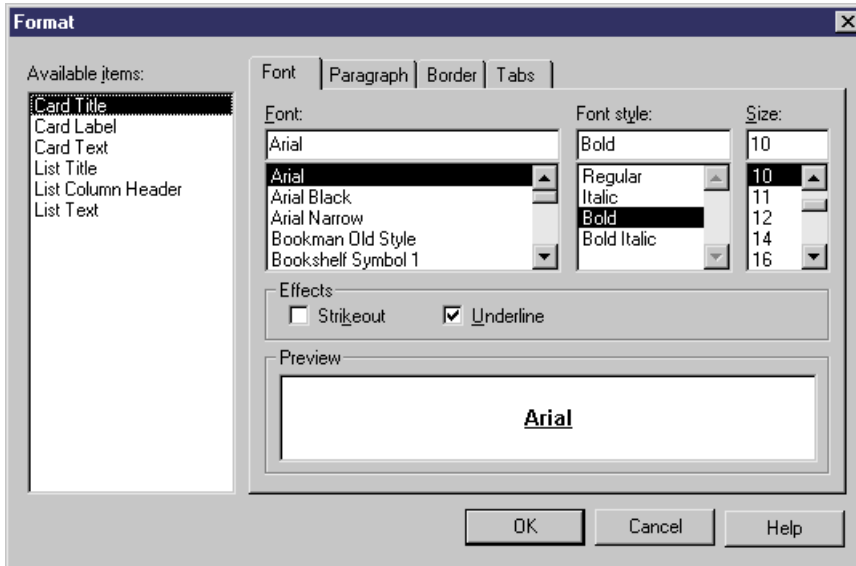
Available Items

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"
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 "

Report Item	Text Selections
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* on page 263), 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 "

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* on page 263.

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* on page 263).

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
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

Editing Textual Report Items
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**.

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.

Formatting Graphical Report Items

You can access the report items Graph Format dialog in any of the following ways:

- 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.
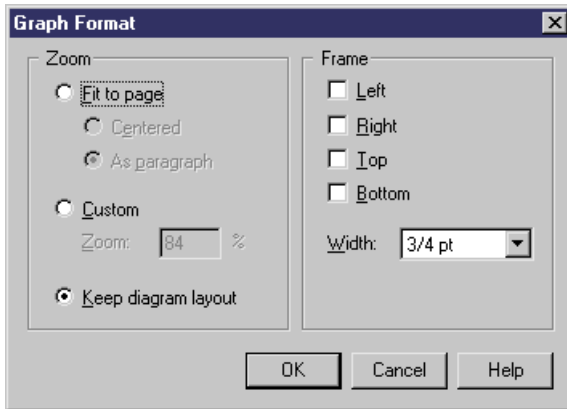
The following options are available in the Graph Format dialog:

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.

Controlling the Layout of List Report Items

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.

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:
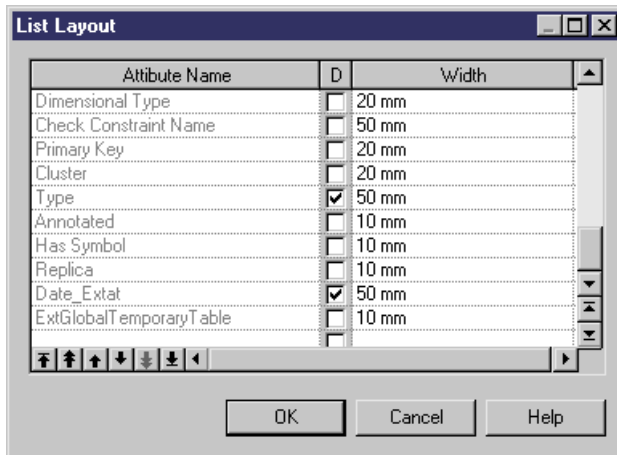
2. Enter any appropriate layout changes as follows:

   • 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.

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.

Controlling the Layout of Card Report Items

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.

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:
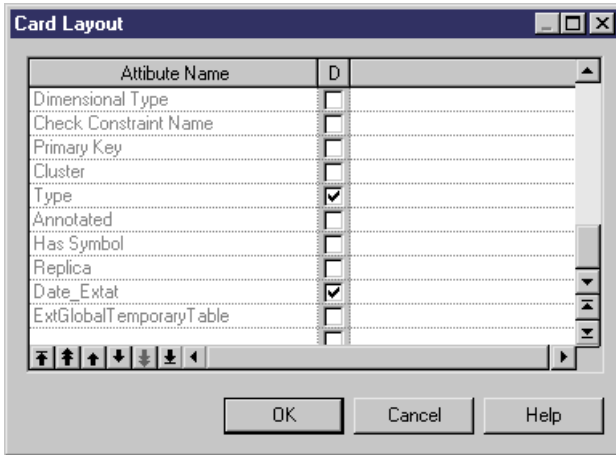
2. Enter any appropriate layout changes as follows:

   - 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.

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.

Suppressing the Titles of Report Items

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:

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	

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.

Modifying Report Section Headers and Footers

The Report includes a default header and footer that you can modify.

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* on page 263.

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:
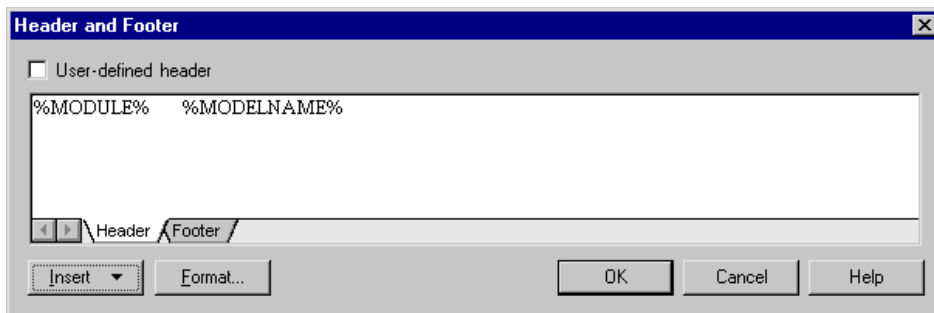
2. Enter any appropriate changes to the header text. Edit the Footer text by clicking on the **Footer** sub-tab. You can use any of the variables available to the RTF template (see *RTF and HTML Report Presentation Templates* on page 269).

   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 Resource Files*.

   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* on page 256) to set tab stop values.

3. Click **OK** to return to your report.

Report Properties

To open the report property sheet, select **Report > Report Properties** from the Report Editor window.

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 Resource 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
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. Defaults to the name of the report.
Author	Specifies the report author. Defaults to user shown on the Version Info tab.
Date	Specifies the date of the report. Defaults to the current date.
Version	Specifies the version number of the report. Defaults to the 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.

Note: For HTML reports, a home page is generated instead of a title page (see below). 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 *RTF and HTML Report Presentation Templates* on page 269) 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* on page 263. 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.

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 *RTF and HTML Report Presentation Templates* on page 269) 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 *RTF and HTML Report Presentation Templates* on page 269) 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 *RTF and HTML Report Presentation Templates* on page 269) 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. The default is 4. • **Width** - the width of the frame as a percentage of the total width of the report window. The default is 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 using the following commands from the contextual menu of book items in the Report Items pane: • **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. You can choose from the following: • 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.

Property	Description
Web browser	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	Inserts a page break before each top level list and card to avoid long HTML pages.

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
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.
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. **Note:** You cannot undo this action.

Note: You can launch the Report Wizard (see *Creating a Report with the Report Wizard* on page 226) 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.

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.

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.

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 Resource Files.

RTF and HTML Report Presentation Templates

PowerDesigner provides a set of predefined RTF and HTML presentation templates to style reports in these formats. You can use these templates and their associated files as a basis to create your own presentation templates. These RTF presentation templates are stored in the Resource Files\RTF Report Templates folder.

The following GTL variables are available to these templates:

Variable	Description
%DATE%	Report date defined in report property sheet.
%TIME%	Report generation time.
%MODELNAME%	Model name of a report section.
%MODELCODE%	Model code of a report section.
%MODULE%	Module name of a report section.
%APPNAME%	Application name (PowerDesigner).
%TITLE%	Report title defined in report property sheet.
%AUTHOR%	Report author defined in report property sheet.
%VERSION%	Report version defined in report property sheet.
%SUMMARY%	Report summary defined in report property sheet.
%COMPANY%	Company name (Sybase, Quest).
%REPORTCONTENT%	[RTF only] Report insertion point in the RTF presentation template. If this variable is not defined, the report is appended at the end of the RTF presentation template.
%NEXT%	[HTML only] URL of the Next page of the home page.
%NEXTLABEL%	[HTML only] Name of the Next page of the home page.

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>.

HTML Presentation Templates

An HTML presentation template file has an .HTMLTPL extension and 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	Defines the background styles for all report pages, except the home page frame and the browser frame page.
.BROWSERBODY	Defines the background styles for the browser page.
.HOMEBODY	Defines the background styles for the home page.
.TEXT	Defines the default font styles for text blocks, like description, annotation, etc.
.TITLE	Defines the default font styles for the title of the home page.
TABLE	Defines the default styles for tables.
TD	Defines the default styles for table cells.
Table.GRID	Defines the styles for object list tables.
TABLE.GRID TD.HEADER	Defines the styles for the header cells of object list tables.
TABLE.GRID TD	Defines the styles for the cells of object list tables of even rows.
TABLE.GRID TD TD2	Defines the styles for the cells of object list tables of odd rows.
Table.FORM	Defines the styles for object card tables.
TABLE.FORM TD.HEADER	Defines the styles for the cell name of object card tables.
TABLE.FORM TD	Defines the styles for the cell value of object card tables.
Table.TEXT	Defines the styles for the tables around the text blocks.
TABLE.TEXT TD	Defines the styles for the cells of tables around the text blocks.

Class names	Description
Table.TITLE	Defines the styles for the tables around the home page title.
TABLE.TITLE TD	Defines the styles for the cells of tables around the home page title.
Table.GRAPHICS	Defines the styles for the tables around graphics.
TABLE.GRAPHICS TD	Defines the styles for the cells of tables around graphics.
H1	Defines the styles for level 1 headings.
H2	Defines the styles for level 2 headings.
H3	Defines the styles for level 3 headings.
HR	Defines the styles for separator lines.
A:LINK	Defines the default colors for non-visited hyperlinks.
A: VISITED	Defines the default colors for visited hyperlinks.
A: HOVER	Defines the default colors for highlighted hyperlinks.
.BROWSER	Defines the default font styles for the browser.
A.BROWSER:LINK	Defines the default colors for non-visited hyperlinks of the browser.
A.BROWSER:VISITED	Defines the default colors for visited hyperlinks of the browser.
A.BROWSER:HOVER	Defines the default colors for highlighted hyperlinks of the browser.
TABLE.NAVGROUP	Defines the styles for the tables around the navigation buttons.
TABLE. NAVGROUP TD	Defines the styles for the cells of tables around the navigation buttons.
.NAVBUTTON	Defines the default font styles for the navigation buttons.
A.NAVBUTTON:LINK	Defines the default styles for non-visited hyperlinks of the navigation buttons.
A.NAVBUTTON:VISITED	Defines the default styles for visited hyperlinks of the navigation buttons.
A.NAVBUTTON:HOVER	Defines the default styles for highlighted hyperlinks of the navigation buttons.

Report Templates

A report template specifies content and formatting for a report section. PowerDesigner
provides a set of standard report templates, and you can create your own. If your report has
more than one section, you can apply the same or different report templates to each section.

Creating a Template from the List of Report Templates

You can create a report template from scratch, or based on an existing template, from the List
of Report Templates.

1. Select **Tools > Resources > Report Templates** to open the List of Report Templates:
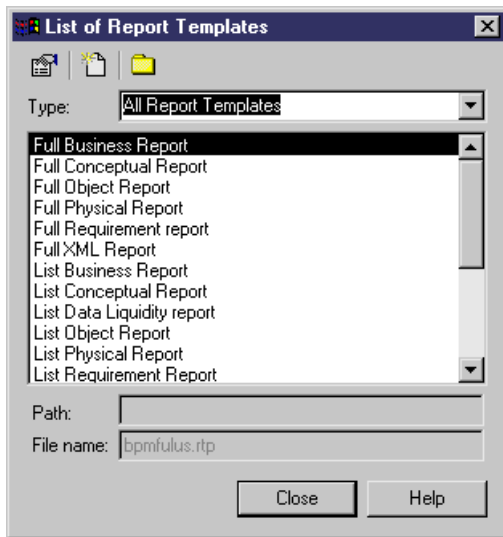
2. Click the **New** tool and enter a name for your template:
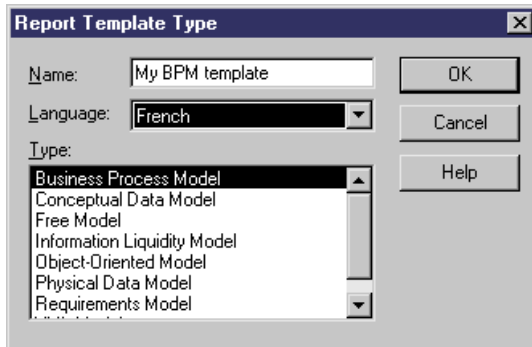
3. Select a language and model type for the template, and then click **OK** to open the empty template in the Report Template Editor.

4. Add report items to the Template Items pane (see *Adding Items to a Report* on page 248) and set any desired formatting (see *Formatting Report Items* on page 255).

5. Select **File > Save** to save the template.

   The template is available for selection when you create a report.

Creating a Template from a Report Section

Once you have created a report section, you can save it as a template for use with other reports.

1. In the Report Editor, click the tab corresponding to the section you want to save as a template.
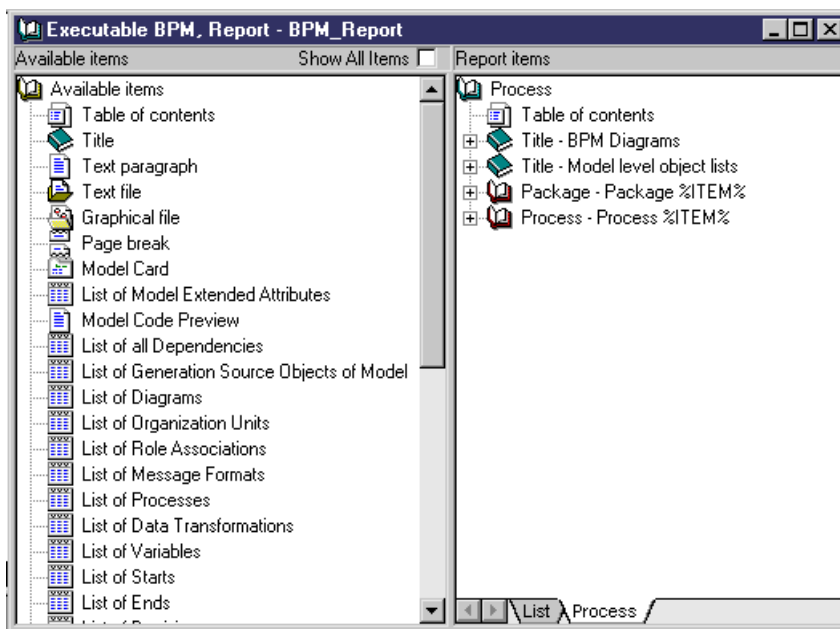
2. Select **Report > Create Template From Section** to open the section in the Report Template Editor.

3. Select **File > Save**, enter a name for the template, and click OK to save it for future use.

   Note: You can rename a template in the Template Items pane by right-clicking the Template node at the root of the tree and select Rename.

Modifying and Saving a Template

You can modify existing templates from the List of Report Templates.

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 Template Editor.

5. Make any appropriate changes to the template and then select **File > Save**.

CHAPTER 7 **Comparing and Merging Models**

PowerDesigner provides powerful tools for comparing models and merging them.

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.

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.
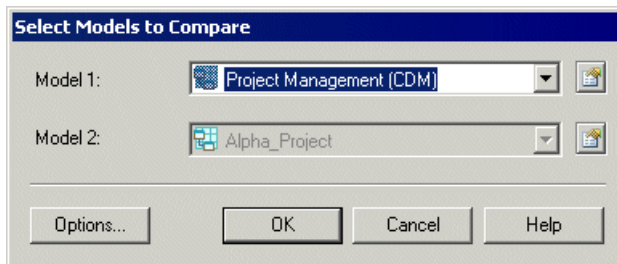
3. [optional] Click the **Options** button to open the Comparison Options window and specify which objects and properties you want to include in the comparison (see *Comparison Options* on page 275).

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 *Analyzing Differences in the Compare Models Window* on page 277). You can apply a filter to simplify the list of differences (see *Filtering Changes in the Compare and Merge Windows* on page 280).

Comparison Options

The Comparison Options dialog box allows you to specify which types of objects you will compare. By default, all objects are selected for comparison.

The dialog box is divided into three panes:

• Metaclasses – top-level objects such as tables, classes, entities, etc.

- Attributes – simple properties associated with the selected metaclass.
- Collections – properties that can contain multiple properties, and which are often sub-objects in their own right.

When you select a metaclass, the lists of attributes and collections change accordingly.

When you merge two models, differences are ignored for metaclasses, attributes and collections not selected for comparison. No changes will be applied to the model to be merged for a metaclass, attribute or collection where the selection checkmark has been cleared in the comparison options lists.

In the following illustration, the Entities collection check box is deselected for comparison:
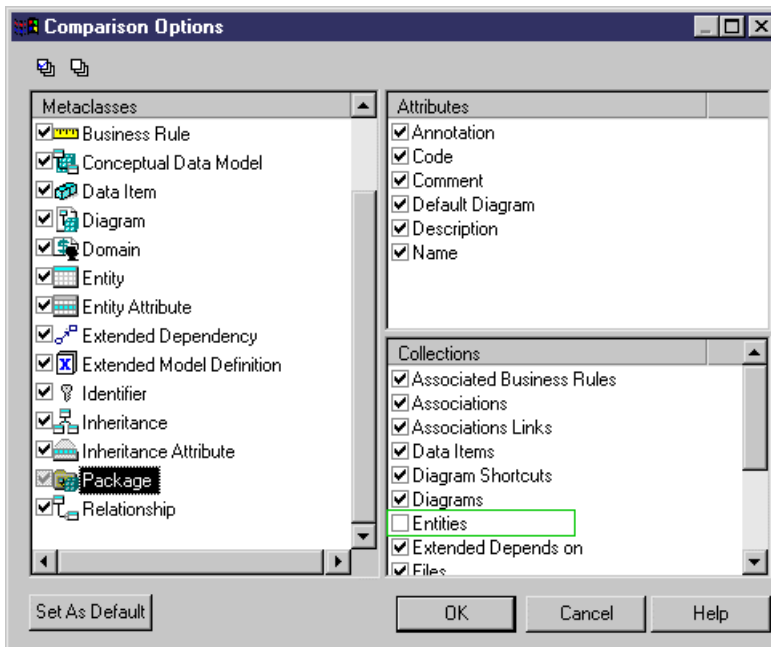
Note: You can set your current selection of comparison options as default by clicking the Set as default button in the Comparison Options dialog box.

You can perform the following option selections:

You can...	To...
Select or deselect a meta-class check box	Select or deselect all its attributes and collections at once
Right-click a metaclass attribute or collection	Select or deselect the attribute or collection for all metaclasses at once from the contextual menu

You can...	To...
Click the Select All or De-select All tools	Select or deselect all metaclasses, attributes and collections at once

You can also select several metaclasses at the same time. Only attributes and collections that are common to all metaclasses are listed on the right hand side. When an attribute or collection property value is different from a selected metaclass to another, the attribute or collection check box is grayed-out to indicate the difference.

When you select a metaclass check box, you do not select the metaclass itself for comparison, but only its attributes and collections. If you want to select or deselect a metaclass for comparison, you have to click a Model and/or Package metaclass to display its corresponding attributes and collections and select or deselect the collection check box that corresponds to the unwanted metaclass in the list of Collections on the bottom right hand side.

Analyzing Differences in the Compare Models Window

The Compare Models window allows you to compare the contents of two models.

Four types of differences between the models can be detected during a comparison:

- *Creation*: When an object is created in one of the models, it is displayed in blue.
- *Deletion*: When an object is deleted from a model, it is displayed as a ghost object in the model tree.
- *Modification*: When the properties of an object are different between models, a red exclamation mark is displayed in the bottom right corner of the object node.
- *Move*: When the object has been moved to a different position, for example, a column moved in the list of columns. In this case a green triangle is displayed in the bottom right corner of the object indicates that there has been a move. The green triangle is displayed both on the ghost object indicating the original position of the item and on the object in its new location.
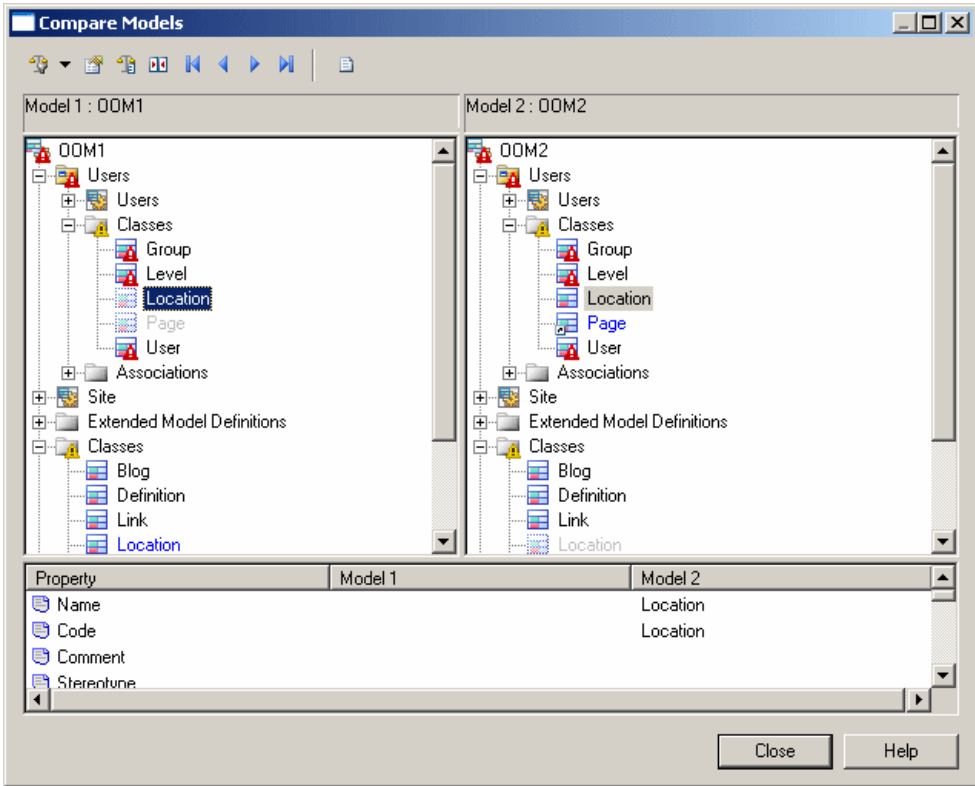
The window is divided in two parts:

- The *object comparison panes* in the upper part of the comparison window display two models of the same type in a tree format and highlight the differences between them.
- The *property comparison pane* in the lower part of the comparison window displays the properties of the nodes selected in the model trees. The property comparison part is divided in three columns, listing the property name, and the values in the two models. The following kinds of differences are possible:

Difference type	Property name column	Model 1 properties column	Model 2 properties column
Object creation	No sign	Object properties displayed	No properties displayed
Object deletion	No sign	No properties displayed	Object properties displayed
Object modification	Diverging property type highlighted by an exclamation point	Divergent object properties displayed	Divergent object properties displayed

Difference type	Property name column	Model 1 properties column	Model 2 properties column
Object move	No sign	Identical object properties displayed	Identical object properties displayed
Object moved and modified	Diverging property type highlighted by an exclamation point	Object properties displayed	Object properties displayed

Comparison Symbols

Symbol	Description
🔲 Customer	Ghost object which does not exist in one of the models
🔲 Customer	Shortcut exists in model
⚠ (yellow)	Sub-objects are different
🔴 (red)	Object exists in both models with different properties
🟢 (green)	Object was moved and modified. This symbol indicates the initial location of the object in the model. This symbol always is displayed besides a ghost object.
🔴 (red)	Object was moved and modified. This symbol indicates the new location of the object in the model.
🟢 (green)	Object was moved

Note: You can open the property sheet of an object (in read-only mode) by double-clicking its node in the tree view. You can also right-click the object node and select Properties from the contextual menu . You can open several property sheets in order to compare the properties of different objects. However, all property sheets will be closed when you close the Comparison window.

Navigating Among Differences
You can navigate among the differences that appear in the object comparison part using the following tools from the comparison toolbar:

Tool	Description
⏮	Goes to the first difference in the tree view
◀	Goes to the previous difference in the tree view

Tool	Description
▶	Goes to the next difference in the tree view
▶❙	Goes to the last difference in the tree view

Note: To expand all nodes at once, you can press the numpad star (*). To collapse all nodes at once, you can press the numpad minus sign (-).

Filtering Changes in the Compare and Merge Windows

You can filter the objects that are shown in the Compare and Merge windows.

1. In the Compare or Merge Models window, click the arrow next to the **Change Filter** tool to display the list of filters:

   - • All objects
     - All changes
     - Only created objects
     - Only deleted objects
     - Only modified objects
     - Only moved objects
     - Only identical objects
     - Only selected objects [Merge Only]
     - Show only conflicts [Merge Only] - 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 [Merge Only] - Hides all differences associated with changes made in the model to be merged as well as all identical objects in order to show only changes made in the model on the left since last generation.
     - Advanced Filters – Opens the Comparison Filters dialog box, which allows you to combine several filters.
       [Merge Only] The following additional Filters on State are available for combining:
       - Show only selected objects - Displays only selected merge actions
       - Show only conflictual objects - Displays objects that have been changed simultaneously by two different users. This filter is only available for merge during consolidation
       - Show only non-preserved objects - Displays objects modified in a generated model but not in the source model. This filter is only available for merge during model generation
       You can display all created and deleted objects in the same tree view, check which objects have changed parent after a move (from one package to another package,

for example), or which objects have been moved within the same collection (when you move attributes within the same entity, for example).

2. Select the appropriate filter to apply it to the Compare or Merge Window. Only objects complying with the filter criteria are displayed.

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.
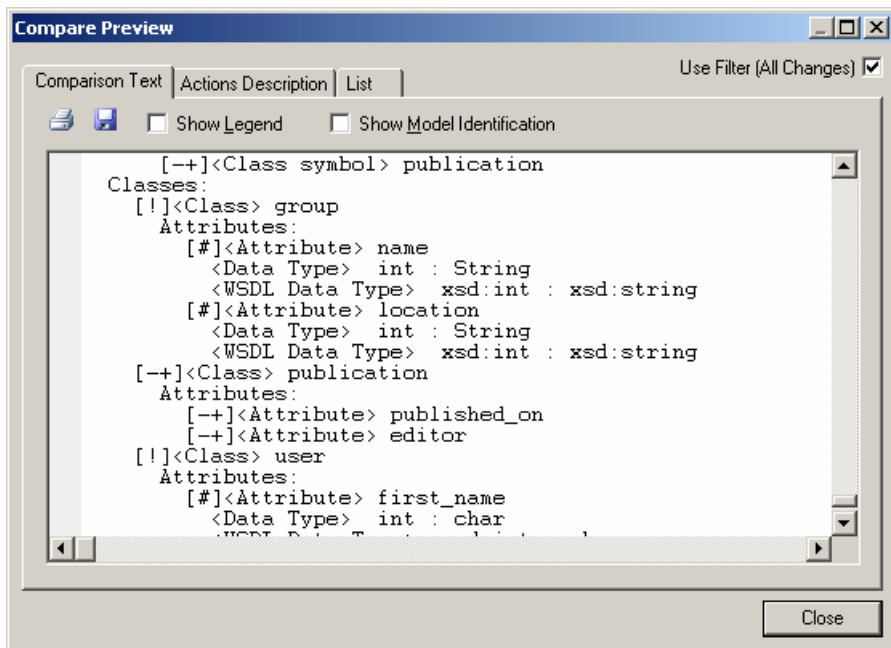
• *Actions Description* – a list of the actions that would need to be taken to render the models the same. This will often be the most readable format as it represents each change in the form of a sentence:
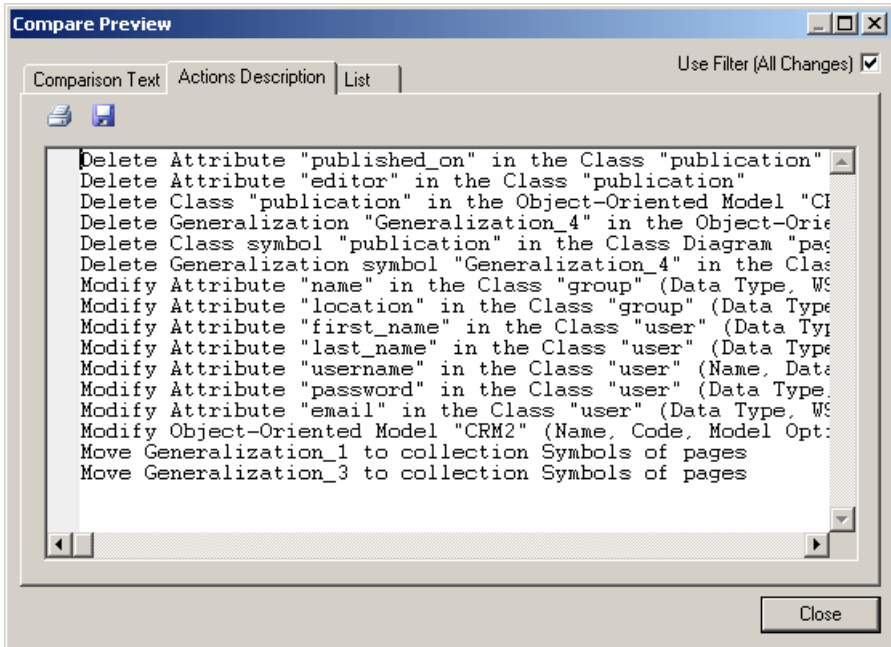
• *List* – a columnar representation of the required actions. This format is the most easily manipulated. You can sort by any of the columns, and export the list in a variety of formats to allow for various kinds of transformation:
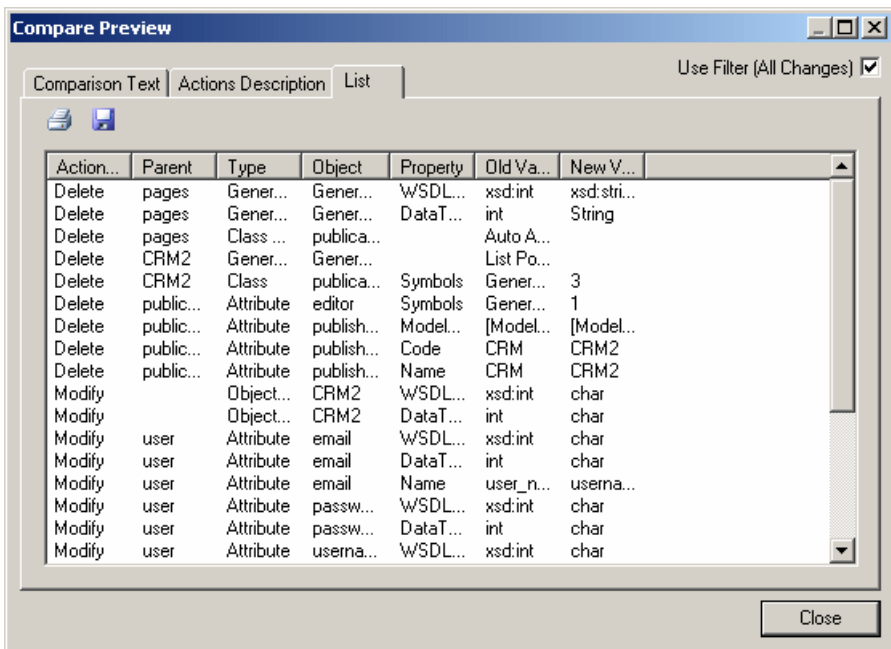
The following tools are available on each of the tabs:

Tool	Description
Use Filter	Filters the changes using the filter specified in the main window (see *Filtering Changes in the Compare and Merge Windows* on page 280). 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
```

Merging Models

You can merge the content of two PowerDesigner models or two resource files of the same type at any time.

1. Select **Tools > Merge Models** to open the Select Models to Merge dialog box.

    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.
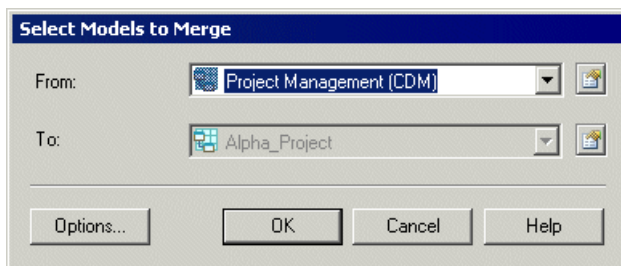
    Note: If you want to merge two PDMs, BPMs, OOMs or XSMs, you have to select two models with the same target DBMS or language

3. [optional] Click the Options button to open the Comparison Options window and specify which objects and properties you want to include in the comparison and possible merging (see *Comparison Options* on page 275).

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 *Analyzing Differences in the Merge Models Window* on page 285). You can apply a filter to simplify the list of differences (see *Filtering the Changes to Merge* on page 280).

5. After selecting the merge actions that you want to perform, click **OK**.

    The Merge Models dialog box closes and the merge begins. Merge messages are displayed in the Output pane. A final message informs you that the models were successfully merged.

Analyzing Differences in the Merge Models Window

The Merge Models window shows you the anticipated result of the merge of your model, with the differences highlighted, and allows you to select or deselect the elements to merge.
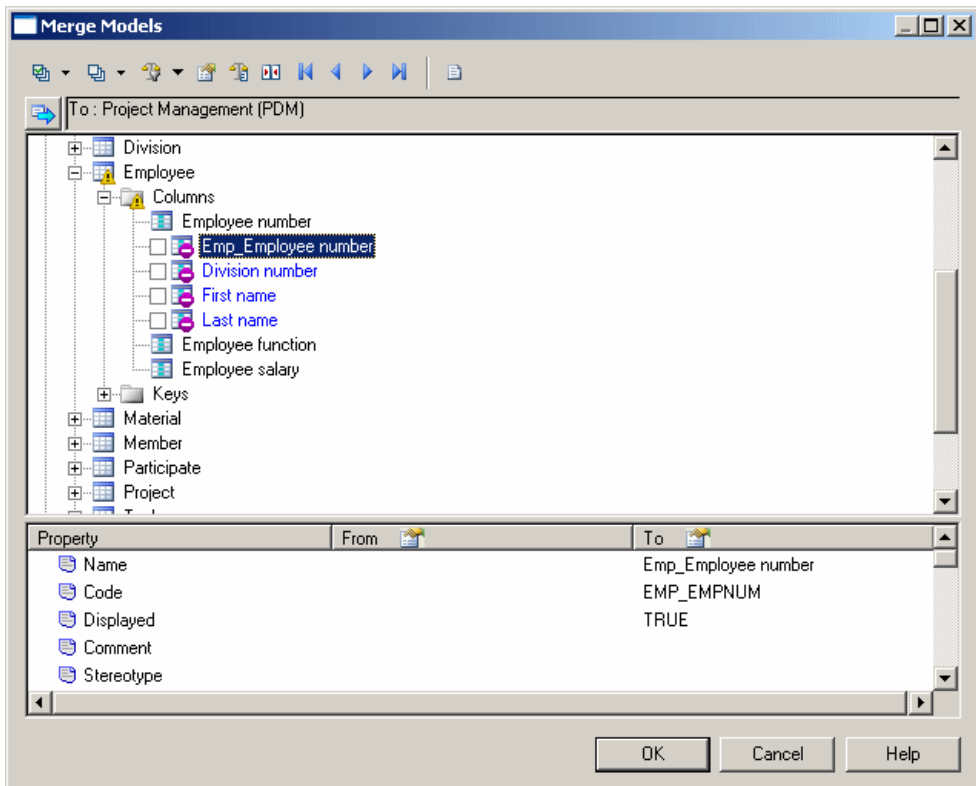
The window is divided in two parts:

- 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 lower part 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.

  Click the **Properties** tools in the **From** or **To** column heading to open the property sheet of the selected object in the source or target model.

You can navigate among the proposed merge actions displayed in the model to be merged using the following tools:

Tool	Description
⏮	Goes to the first difference in the tree view
◀	Goes to the previous difference in the tree view.
▶	Goes to the next difference in the tree view
⏭	Goes to the last difference in the tree view

Note: You can right-click a moved object and select Next or Previous occurrence to go to its next or previous occurrence. To expand all nodes at once, press the numpad star (*). To collapse all nodes at once, you can press the numpad minus sign (-).

For each difference detected during the comparison process, a merge action is proposed for you to synchronize the model to be merged with the model on the left pane. Merge icons are composed of a symbol and a color. A triangle provides an indication that there is a difference between the two models. A circle with a check box, which only appears on the right-hand pane specifies a merge action that will be performed if the check box is selected.

The following table explains the color code:

Color	Description
Red	Modification
Yellow	Indication
Green	Move
Purple	Deletion and shortcut replacement
Blue	Addition

The following table lists the indications and merge action symbols:

Icon	Description
Entities	[yellow] Notification that children were modified
Table_1	[red] Notification that object properties were modified
Entt_5	[green] Notification that object was moved
☑ Dept	Object creation [blue] - Adds object to model to be merged because it exists in model on left pane
☐ Marketing	Object deletion [purple] - Deletes object from model to be merged

Icon	Description
☑ 🖼 Name	Object modification [red] - Updates object definition in model to be merged using object definition in model on left pane
☑ 🖼 Id_number	Object move [green] - Moves object in model to be merged to the same location as in model on left pane or replaces it with shortcut
☑ 🖼 Entt_2	Shortcut merge [purple] - Replaces the shortcut with target object
☐ 🖼 Applet	Shortcut deletion [purple] - Deletes shortcut from target model
☐ 🖼 Exception	Shortcut creation [blue] - Creates shortcut in target model

Selecting Merge Actions in the Model to Be Merged

You can select or deselect merge actions in the pane of the model to be merged, using:

• A category contextual menu
• An object contextual menu

To do so, you right-click an object or a category in the pane of the model to be merged and select the Select or Deselect command to open the sub-menu of the merge actions.

By default, all creation, modification and move actions are selected in the model to be merged, deletion actions are not selected.

Note: For consolidated models, the merge actions selected by default depend on the latest modifications saved in the model. For generated models, if the Preserve modification generation option was selected, they depend on the modifications saved in the model to be merged.

The following selection tools from the merge toolbar:

Tool	Description
🗗 ▾	Select all merge actions. Click the arrow to select all of one kind of merge action.
🗗 ▾	Clear all merge actions. Click the arrow to deselect all of one kind of merge action.

Merging Object Properties
PowerDesigner lets you select individual object properties to merge.

Note: You can double-click a property in the property comparison part to display differences between models. When a line is added the following symbol is displayed >>, when a line is deleted, the following symbol is displayed <<.

1. Select objects with different definitions in the object comparison part of the Merge window.

2. Select or clear the check boxes corresponding to the properties you want to merge in the properties comparison part of the Merge window.

   By default, the diverging properties are selected in the list:

Property	From	To
☐ Name	11	11
☐ Code	11	11
☐ Displayed	TRUE	TRUE
☐ Comment		
☑ Data Type	<undefined>	char(0)
☐ Identity	FALSE	FALSE
☐ With Default Option	FALSE	FALSE
☐ Mandatory	FALSE	FALSE
☐ Check on Table Constraint Name	CKC_11_TABLE_3	CKC_11_TABLE_3
☐ Distinct Values	40	40

Synchronizing Objects Manually

When an object has been renamed or modified out of the PowerDesigner interface, the link with the corresponding object in the compared model is lost, and the comparison interface will display two disconnected objects in the model trees with object creation and deletion actions in each model.

Manual synchronization lets you perform the following operations:

• Synchronize moved or disconnected objects. Use the Parent box to select the parent package corresponding to the package on the left pane.
• De-synchronize objects that share common properties and are naturally related in the comparison process. You can select <NONE> or another object from the Object list in order to break the relation between objects.

The tree view of the model to be merged is automatically updated after you validate a manual synchronization.

1. Select the item you want to synchronize from one of the model trees, and then click the Manually Synchronize Two Objects tool to open the Manual Synchronization dialog box is displayed:
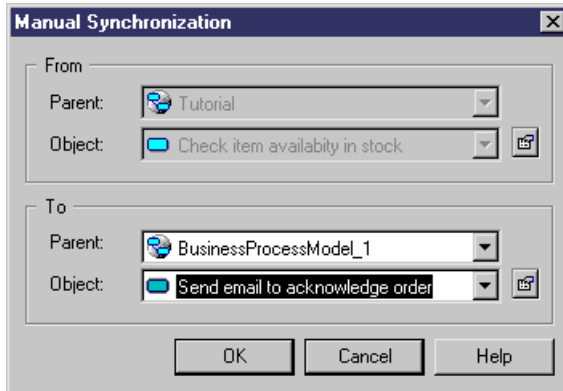
Depending on the merge action proposed, the From or the To box is editable.

2. Select a model or package in the Parent list, and then select the object that you want to synchronize with in the Object list. You can use the Properties tool to better identify object

3. Click OK.

The selected items are synchronized and the tree view of the model to be merged is updated in order to show the impact of the manual synchronization.

When you synchronize objects and linking objects, you obtain the following result:

Object	Within the same package	Between different packages
Object Linking object	Objects on the left pane and parent objects are synchronized	Object moved to package to be merged and removed from package on the left pane

Previewing, Printing, and Saving Merge Actions

You can preview and print the selected merge actions by clicking the Merge Preview tool to open the Merge Preview window.

This dialog is the same as the Compare preview dialog. See *Previewing, printing, and saving comparison results* on page 281.

CHAPTER 8 **The Repository**

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

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

The majority of repository users:

- Browse and search the repository (see *Browsing the Repository* on page 310 and *Chapter 9, The PowerDesigner Portal* on page 331)
- Add and update documents in the repository (see *Checking Documents into the Repository* on page 292)
- Check documents out of the repository to view or edit them (see *Checking Documents Out of the Repository* on page 302), and may also:
  - Create versions to retain a history of the development process (see *Freezing Document Versions* on page 318)
  - Lock document versions to temporarily protect them from modification by other users (see *Locking Document Versions* on page 320)

More advanced users will have additional rights to:

- Create groups of document versions to mark particular milestones in the development process (see *Grouping Document Versions in a Configuration* on page 322)
- Creating branches in the repository version tree (see *Branching Version Trees* on page 325)

Administrators take responsibility for:

- Installing and maintaining the repository (see *Installation Guide > Installing the Repository*)
- Providing access to the repository by managing rights and permissions (see *Chapter 18, Controlling Repository Access* on page 521)
- Providing customization of the PowerDesigner interface through permission, UI, an general profiles (see *Chapter 19, Customizing the PowerDesigner Interface* on page 539)
- Deploying an enterprise glossary and library of reference models and other documents to be shared with all users (see *Chapter 20, Deploying an Enterprise Glossary and Library* on page 555)

- Adding resource files to the repository and managing their sharing (see *Sharing Resource Files via the Library* on page 565)

Checking Documents into the Repository

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

1. Select the documents to check in on the **Local** tab of the Browser, then right-click and select **Check In** to open the:
    - Check In Project window - if you selected a project (see *Checking In a Project* on page 292).
    - Check In Document window - if you selected a single model or file (see *Checking In a Document* on page 293).
    - Check In Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see *Checking In Multiple Documents* on page 295).
2. Select the appropriate parameters (see *Check In Parameters* on page 297), and click **OK** to check in the document.
3. If you are checking in a document that is already stored in the repository, the Merge Changes to Repository Document dialog (see *Resolving Conflicts During Check In* on page 298) may open to let you review the differences between your local document and the version in the repository. When you are satisfied, click **OK** to complete the check in.

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

Checking In a Project

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

You arrive here by right-clicking a Project in the **Local** tab of the Browser, and selecting **Check In**. The dialog is populated with any project documents that have changes. The first time you check in a project, you must check in all its documents; subsequently, only documents with changes are listed and can be deselected if desired.
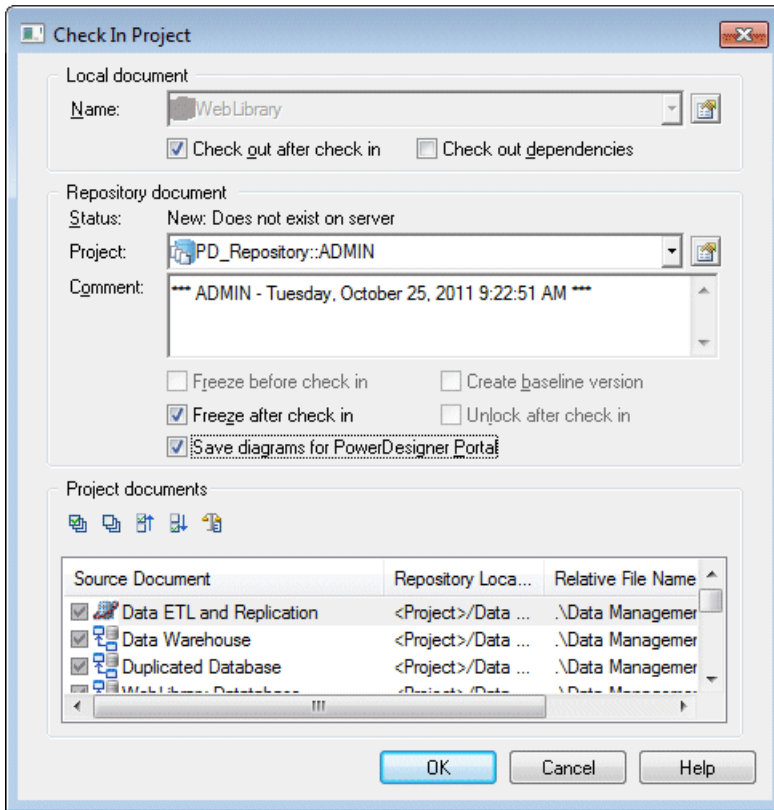
For information about the parameters available in this dialog, see *Check-In Parameters* on page 297.

Checking in a Document

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

You can arrive here by:

*   Right-clicking a model or file in the **Local** tab of the Browser, and selecting **Check In**.
*   Selecting a model or file in the Check In Multiple Documents dialog (see *Checking In Multiple Documents* on page 295), and clicking the **Settings** tool.
*   Clicking the **Add Document** tool in the Check In Multiple Documents window (see *Checking In Multiple Documents* on page 295). You must specify the model or file to check in in the **Local Document** group box.
*   Selecting a resource file, and clicking the **Check In** tool in the a resource file list (see *Sharing Resource Files via the Library* on page 565).

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

- Simplified - Checking in a document for the first time (select a folder to check into, enter a comment, and click **OK**):
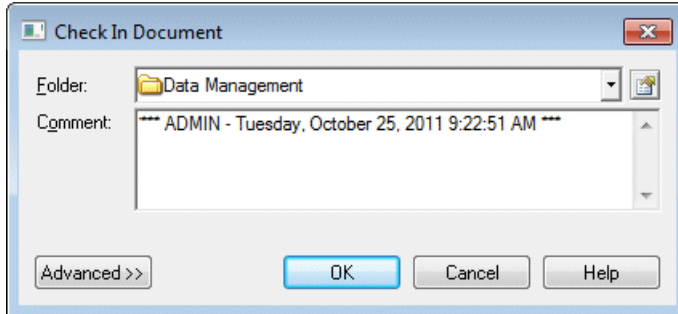
- Simplified - Checking in updates to a document already in the repository (choose whether to review your changes (see *Resolving Conflicts During Check In* on page 298), enter a comment, and click **OK**):
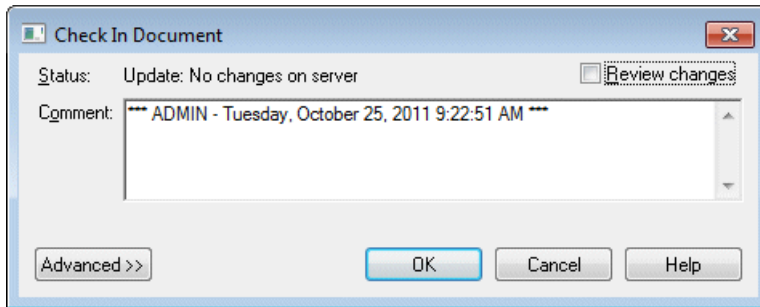
- Advanced - When you click the **Advanced** button on either of the simplified dialogs:
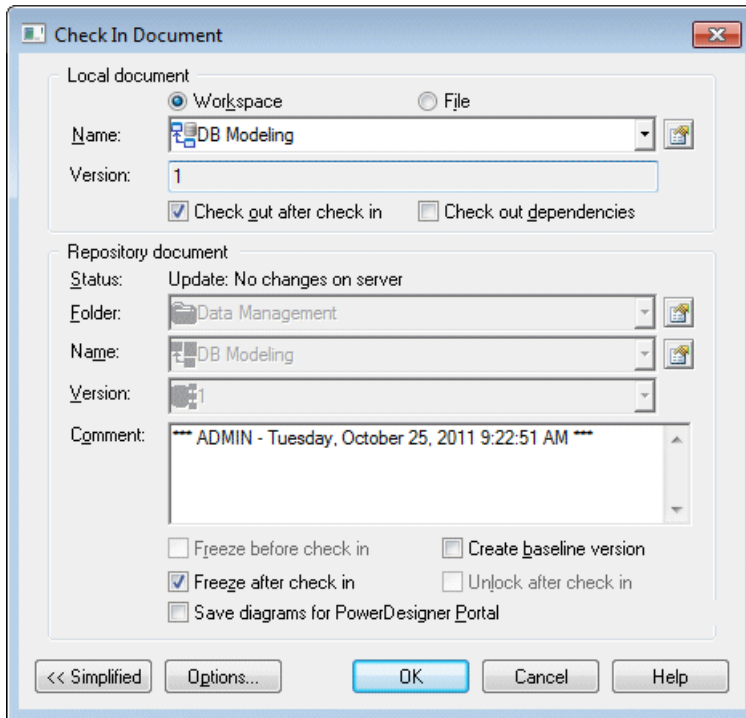
For information about the parameters available in this dialog, see *Check In Parameters* on page 297.

Checking In Multiple Documents

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

You can arrive here by:

* Right-clicking the workspace root or a folder containing multiple documents (or manually selecting multiple documents) in the **Local** tab of the Browser, and selecting **Check In**.
* Selecting **Repository > Check In** (or pressing **Ctrl+Alt+C**).
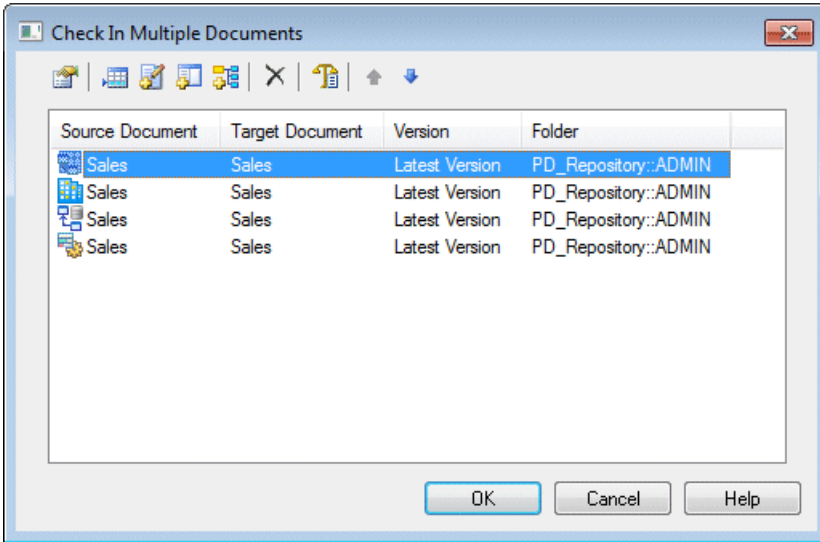
The following tools are available to manage the documents in the list:

Tool	Description
	Settings – Opens the Check In Document window to let you specify check in parameters for the selected document (see *Checking in a Document* on page 293).
	Add Document - Opens the Check In Document window to let you to specify a document to add to the list (see *Checking in a Document* on page 293).
	Add Modified Documents in Workspace - Adds all the documents modified in the workspace during the current session to the list.
	Add All Documents in Workspace - Adds all the documents in the workspace (including closed documents) to the list.
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation or other links to the list. When you check in PowerDesigner models related by shortcuts, the models being referenced are checked in before the models containing the shortcuts.
	Delete – Removes the selected document from the list.
	Check In Options – Opens the Check In Options window to allow you to specify the update mode for changes (see *Resolving Conflicts During Check In* on page 298).
	Move Up – Moves the selected document up in the list.
	Move Down - Moves the selected document down in the list.

Check-In Parameters

These check-in parameters allow you to control details of your check in.

Note: You can set default values for some of these parameters (see *Repository General Options* on page 407). 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.
Version	[not for project – read-only] Specifies the current version of the model or file in the repository, if available.
Check out after check in	[models and resources only] Updates the local version after check in by checking out the repository version to ensure that you have the most up-to-date version where other users may be working concurrently.
Check out dependencies	[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* on page 511).

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.

Parameter	Description
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.
Freeze before check in	Freezes the previous versions of any modified objects in the repository, and creates a new version to accept the changes to be checked in (see *Freezing Document Versions* on page 318).
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 Document Versions* on page 318).
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.
Unlock after check in	[locked documents only] Unlocks the document version after check in (see *Locking Document Versions* on page 320).
Save diagrams for PowerDesigner Portal	[not for resources] Prepares snapshots of the model diagrams that can be viewed in the PowerDesigner Portal (see *Chapter 9, The PowerDesigner Portal* on page 331).

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.

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.
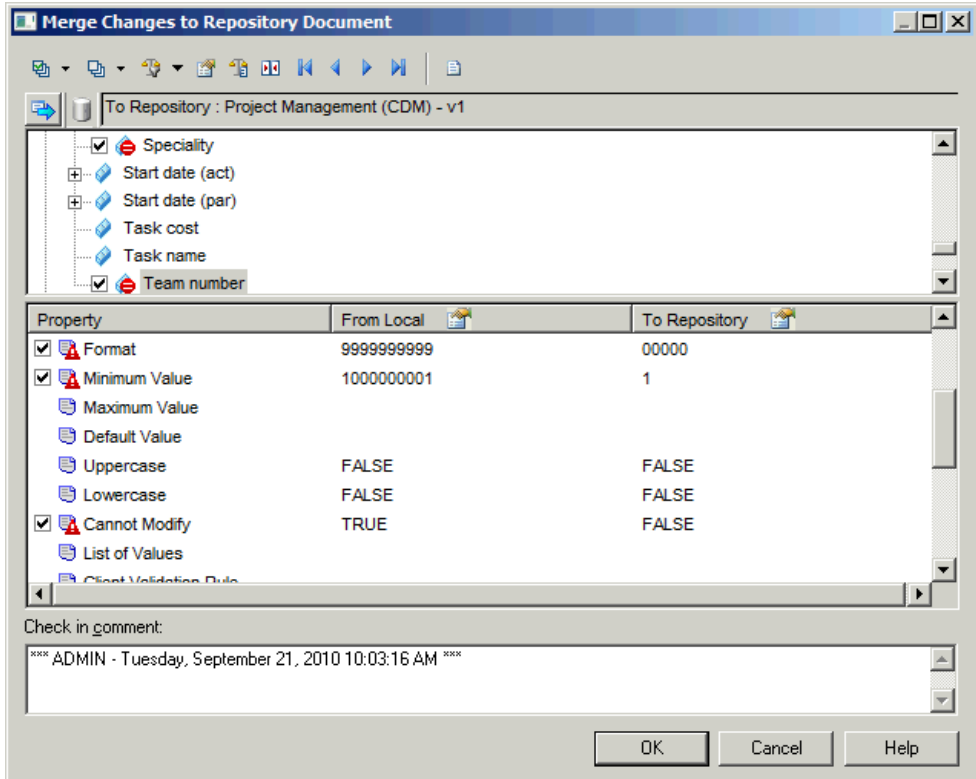
The window is divided in three parts:

- 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* on page 284.

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.

```
┌─ Check In Options ─────────────────────── ✕ ┐
│  ┌─ Update mode ──────────────────────────┐ │
│  │  ○ Interactive                         │ │
│  │  ────────────────────────────────────  │ │
│  │  ● Batch                               │ │
│  │       Resolve conflicts:  ○ Use my changes    │ │
│  │                           ○ Keep repository changes │ │
│  │                           ● Resolve manually  │ │
│  └────────────────────────────────────────┘ │
│        [  OK  ]   [ Cancel ]   [ Help ]       │
└──────────────────────────────────────────────┘
```
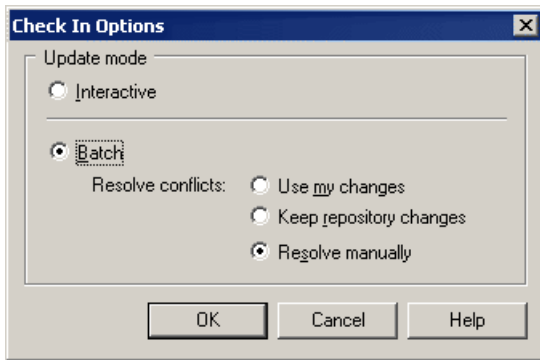
The following modes are available:
- Interactive - [default] Always opens the Merge Changes to Repository Document dialog to let you review and approve or reject each change before check in.
- Batch - Automates the merging of changes and treats conflicts (where the same object has been modified both by you and by another user) using one of the following rules:
    - Use my changes – applies your changes and overwrites changes in the repository version (even if they are to different attributes).
    - Keep repository changes – retains the changes in the repository and rejects your changes (even if they are to different attributes).
    - Resolve manually – opens the Merge Changes to Repository Document dialog to let you review and resolve conflicts.

Checking In Packages

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

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

For information about checking out package, see *Checking out packages* on page 308.

Checking In External Application Documents

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

To be able to update an external application document that does not belong to the current workspace, you have to check the document out from the repository, select the Add to

Workspace check box and then save the workspace. This allows you to update the document the next time you check it in.

Undoing a Check In

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

Since a check in to the repository does not necessarily create a new document version (see *Document Versions* on page 318), such a deletion may not always be possible.

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.
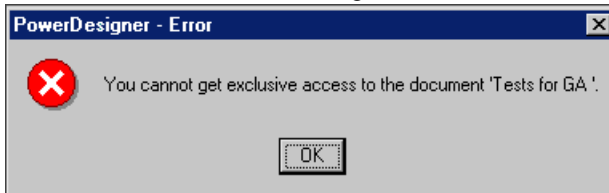
In rare cases, if the connection to the repository is interrupted during an update to a document, the document may remain protected, and you may not be able to update the document in the future until you bypass the protection or cancel the action:

• 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:
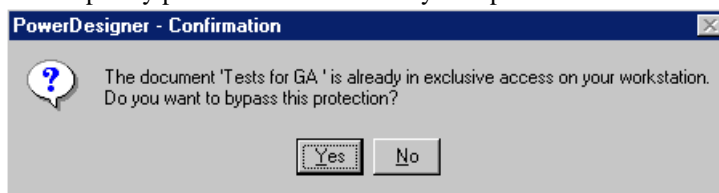
Note: If you have two PowerDesigner sessions running on your workstation and are performing an operation on a document in one session, and want to perform another operation on the same document from the second session, then you should wait for the first operation to complete.

• 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:

PowerDesigner - Confirmation

The document 'Tests for GA ' is already in exclusive access (locked by 'PEGASUS').
Do you want to bypass this protection?

[Yes] [No]
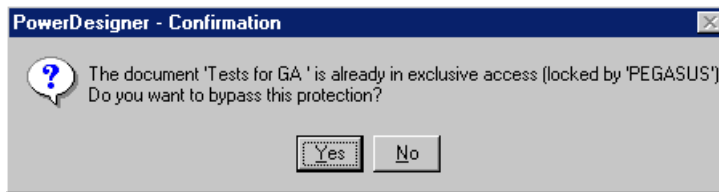
Warning! Bypassing the protection when you are not entitled to may severely damage the document.

Checking Documents Out of the Repository

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

1. To check out a document not currently in your workspace, select the **Repository** tab of the Browser. If you are updating an existing file, select the **Local** tab.

2. Select the documents to check out, then right-click and select **Check Out** to open the:
   - Check Out Project window - if you selected a project (see *Using the Check Out Project Window* on page 302).
   - Check Out Document window - if you selected a single model or file (see *Using the Check Out Document Window* on page 303).
   - Check Out Multiple Documents window - if you selected multiple files, the workspace root, or a folder containing multiple files (see *Using the Check Out Multiple Documents Window* on page 304).

3. Select the appropriate parameters (see *Check Out Parameters* on page 305), and click **OK** to check out the document.

4. If you are checking out a document that is already present in your workspace (and you have selected the **Merge document** option), the Check Out Document window (see *Resolving Conflicts During Check Out* on page 307) will open to allow you to review the differences between the version in the repository and your local version. When you are satisfied, click **OK** to complete the check out.

Checking out a Project

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

You can arrive here by:

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

- Right-clicking a project in the **Local** tab of the Browser, and selecting **Update from Repository**.
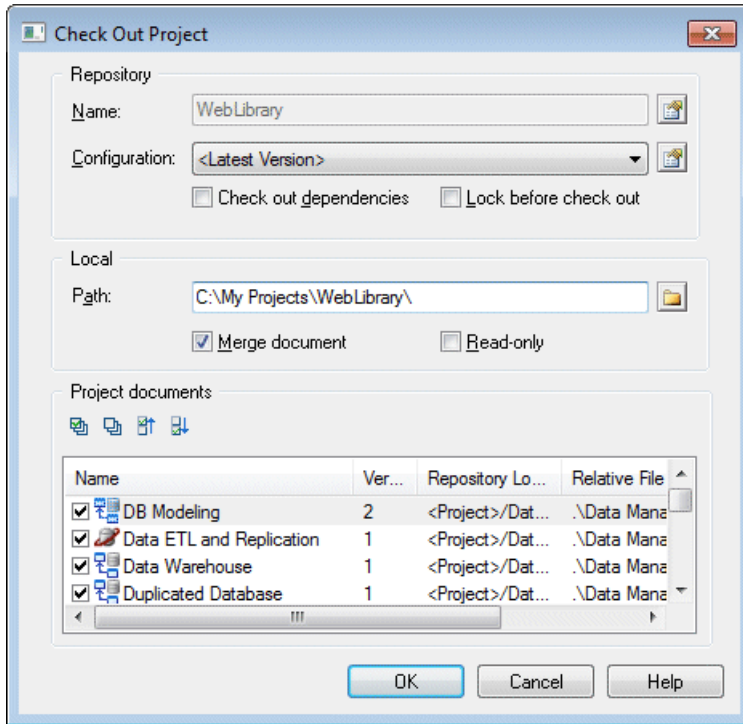
For information about the parameters available in this dialog, see *Check Out Parameters* on page 305.

Checking Out a Document

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

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

- Right-clicking a model or file in the **Repository** tab of the Browser and selecting **Check Out**.
- Right-clicking a model or file in the **Local** tab of the Browser and selecting **Update from Repository**.
- Selecting a model or file in the Check Out Multiple Documents window (see *Checking out Multiple Documents* on page 304) and click the **Settings** tool.
- Clicking the **Add Document** tool in the Check Out Multiple Documents window (see *Checking out Multiple Documents* on page 304). You must specify the model or file to check out in the **Local Document** group box.

- Select a resource file, and click the **Update from Repository** tool in a resource file list.
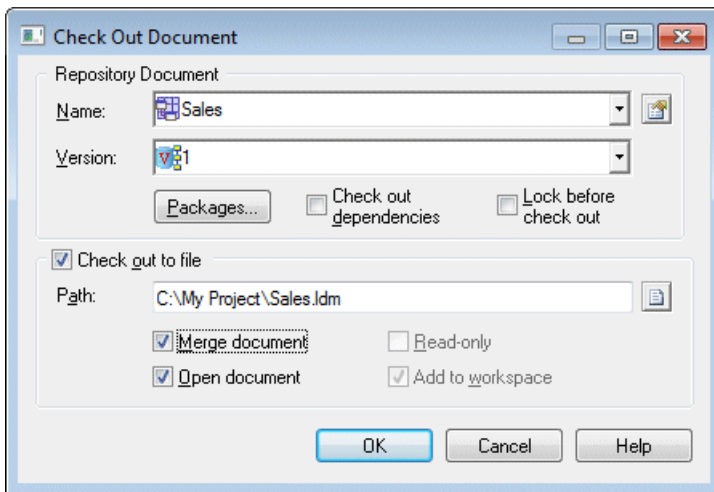
For information about the parameters available in this dialog, see *Check Out Parameters* on page 305.

Checking Out Multiple Documents

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

You can arrive here by:

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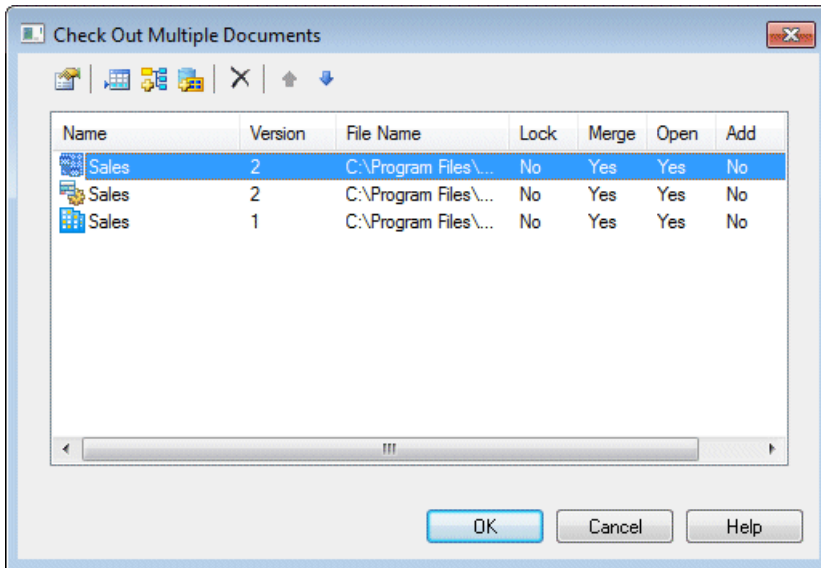
The following tools are available in this window:

Tool	Description
	Settings – Opens the Check Out Document window to let you specify check out parameters for the selected document (see *Checking out a Document* on page 303).
	Add Document Version - Opens the Check Out Document window to let you to specify a document to add to the list (see *Checking out a Document* on page 303).
	Add Related Document Versions – Adds all documents related to the selected document through shortcuts or generation links to the check out list.
	Add Document Versions from a Configuration – Opens a selection window allowing you to add all the document versions included in a configuration to the list (see *Grouping Document Versions in a Configuration* on page 322).
	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.

Checkout Parameters

The checkout parameters allow you to control details of your check out.

Note: You can set default values for some of these parameters (see *Repository General Options* on page 407).

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* on page 322).
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* on page 308).
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* on page 511).
Lock before check out	Locks the selected version of the document before checking it out, to prevent other users from modifying it. This option is available only to users who have the `Lock Versions` right.

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* on page 307), 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.

Parameter	Description
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.

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.
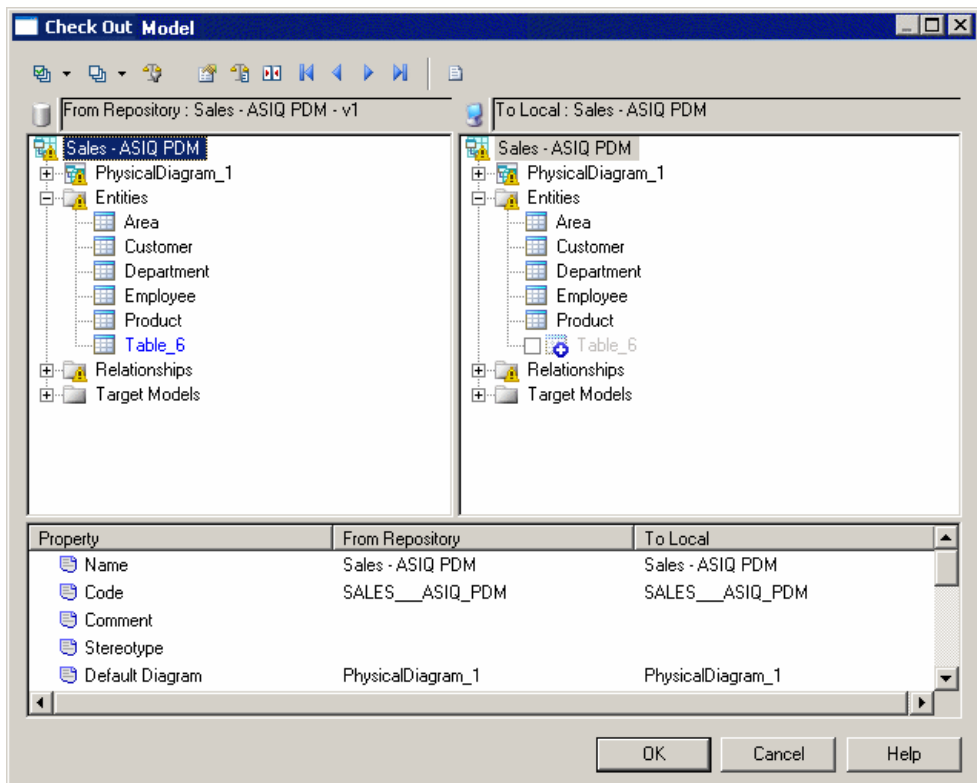
The repository version of the model that you are checking out is displayed in the left pane, and the local version is in the right pane. To filter the tree and show only the changes, click the

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* on page 284.

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	☑ modelingproject
	  ☑ Package_1
	    ☑ Package_2
	      ☑ Package_3
		Package 1, 2, and 3 are automatically selected

Deselect Package 1	☑ modelingproject
	  ☐ Package_1
	    ☐ Package_2
	      ☐ Package_3
		Package 2, and 3 are automatically deselected

Select Package 3	☑ modelingproject
	  ☑ Package_1
	    ☑ Package_2
	      ☑ Package_3
		Package modelingproject, 1, and 2 are automatically selected

Deselect Package 3	☑ modelingproject
	  ☑ Package_1
	    ☑ Package_2
	      ☐ Package_3
		Selection check boxes in package modelingproject, 1, and 2 are grayed

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

Checking the State of a Document

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

Note: You can control the display of status icons with the **Display status icons in Local Browser** general option (see *Repository General Options* on page 407).

In the following example:

• the document "DFD sample" has been changed by you and by another user
• the document "Project Management (PDM)" is up-to-date
• the document "Acme Enterprise Architecture" has been changed and locked by you and changed by another user
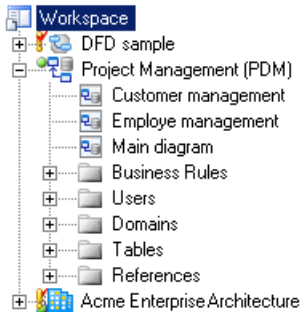
The following table lists the possible states of a document:

Icon	Description
	Up to date (green circle) – Your local version is the same as the one in the repository.
	Changed by you (red checkmark) – Your local version has changes not saved to the repository.
	Changed by another (yellow exclamation point) – The repository version has changes that are not reflected in your local version.
	Up to date and locked by you (green circle and green padlock). For information about locks, see *Document Locks* on page 320.
	Up to date but locked by another (green circle and red padlock).
	Changed by you and changed by another (red checkmark and yellow exclamation point).

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

Browsing the Repository

The PowerDesigner repository can contain various types of node.

Node	Description
Root	[when connection is established] Displays the repository definition name, the user login, and the current branch.
Folder	Used to organize documents in the repository.
Project	Container for models and other files.
Document	Models, multi-model reports, resource files (see *Sharing Resource Files via the Library* on page 565), and external application files such as MS Office files, or graphics.
Package	Packages in models.
Object category	Object categories in models (if the Show Objects check box is selected in the General Options).
Diagram	Diagrams in models (if the Show Objects check box is selected in the General Options).
Object	Objects in models (if the Show Objects check box is selected in the General Options).

Repository Root Properties

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

The **Connection** tab contains the following properties:

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

In addition, the following tabs are available:

- **LDAP** - lists LDAP configuration parameters to enable remote LDAP users to access the repository (see *Controlling Repository Access with LDAP* on page 529).
- **SMTP** - lists parameters for connecting the repository to an SMTP server (see *Specifying an SMTP Server for Notifications* on page 534).
- **Passwords** - lists the requirements of the policy which controls the qualities of user passwords for the repository (see *Defining a Password Policy* on page 533).
- **Permissions** – lists those users and groups with permissions on the entire repository (see *Granting Access Permissions on Repository Items* on page 528).
- **Version Info** – specifies when the repository was created and last modified (see *Version Info Tab* on page 106).

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* on page 40) and library (see *The Library* on page 43).

Repository Folders

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

Note: A PowerDesigner project (see *Chapter 3, Projects and Frameworks* on page 53) acts as both a repository document and a folder. Use the project properties to control permissions for working with the project file itself (project diagram and any framework or dependency matrices) and the folder properties to grant access permissions to the models contained within the project, as you would for a repository folder.

You create a folder by right-clicking the root node, a project, or a folder, selecting **New > Folder**, and entering a name for the folder in the property sheet. Click the **Permissions** tab to add permissions for all appropriate users and groups for the folder and its contents (see *Granting Access Permissions on Repository Items* on page 528).

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

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

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

Repository Document and Object Properties

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

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

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	[not available for folders] The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	[not available for folders] Additional information about the object.
Type	[not available for folders] Type of object: PowerDesigner model or package, other types of documents and object.
Location	[folders, documents, and model objects] Full path of the folder or document. [model packages]: hierarchy of packages in the model.
Status	[not available for folders] Indicates the status of the document, which can be Updatable, Frozen, or Locked.

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

Property	Description
Creation	Displays the full name of the user who created the item in the repository, and the date of creation.
Last modification	Displays the full name of the user who last modified the item, and the date of this modification.

Property	Description
Generated From Origin Object	[model objects only] This box is displayed when the current object has been generated from another object, and displays the name of the origin object. Click the Properties button to the right of this field to open the origin object property sheet.
Version number	[not available for folders] Number of the version.
Version branch	[not available for folders] Name of the branch to which the version belongs.
Comment	[not available for folders] Version comment.

Versions Tab

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

Tool	Description
	Properties - Displays the property sheet of the document
	Check Out - Checks out the selected version. See *Checking Documents Out of the Repository* on page 302.
	Compare - Compares two versions. See *Comparing Models in the Repository* on page 315.
	Freeze/Unfreeze - Freezes or unfreezes the selected version. See *Freezing Document Versions* on page 318.
	Lock/ Unlock - Locks or unlocks the selected version. See *Locking Document Versions* on page 320.
	Delete - Deletes the selected version

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

Other Tabs

The following tabs are also available:

• Notes [models, packages, and model objects only] - contains annotations and descriptions of the object (see *Notes Tab* on page 101)
• Attributes [models, packages, and model objects only] - displays additional properties set for these items.

- Collections [models, packages, and model objects only] - displays a list of collections or related objects (but not sub-objects) for the item. For example, the Collections tab in the property sheet of a CDM entity may display the business rules attached to the entity, but will not display entity attributes.

- Permissions [models, packages, folders and other documents only] – lists those users and groups with permissions on the document. For more information about permissions, see *Granting Access Permissions on Repository Items* on page 528.

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.

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:
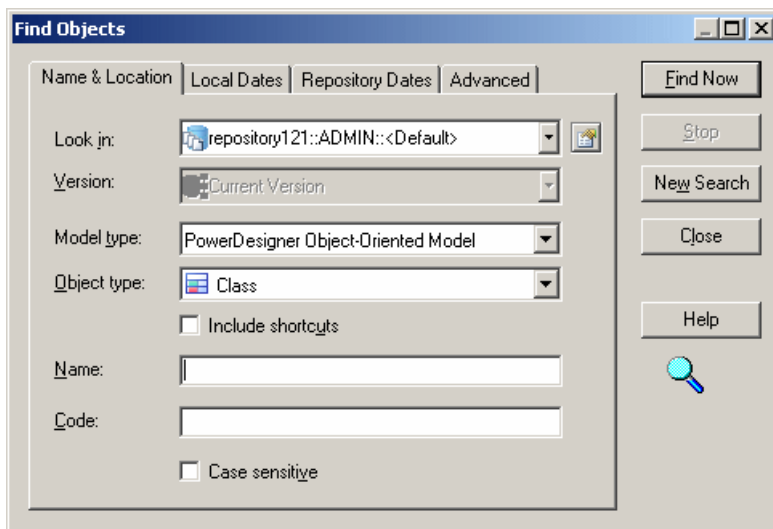
2. Specify the appropriate parameters in the different tabs of the dialog box.

   For detailed information about these tabs, see *Find Parameters* on page 140. 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.
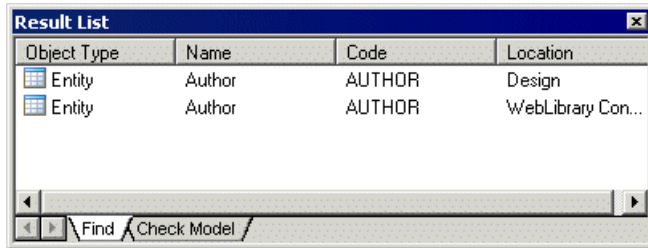
If you right-click an object in the Result list you can select:
- **Properties** - to display the properties of the found objects.
- **Check Out Model** - to check out the model containing the found object.

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.

Warning! You can only compare models of the same type (for example, PDM, OOM, EAM).

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

   The Select Models to Compare window opens:
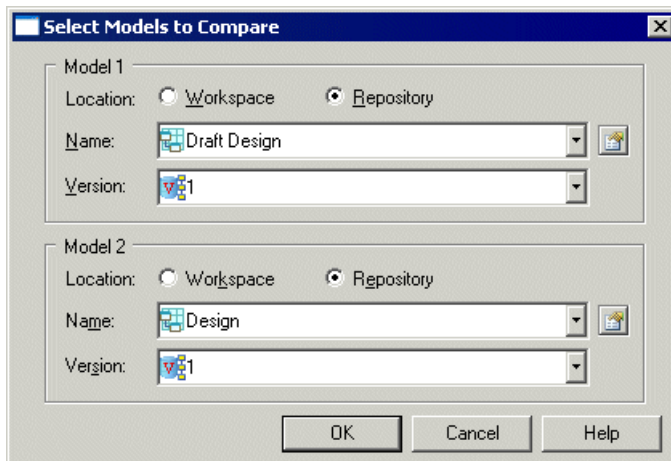
2. For each of Model 1 and Model 2, select:

   - A location to specify whether the model to compare is located in your local workspace or in the repository.

- The name of the model to compare – click the Properties tool to the right of the list to open the property sheet of the selected model. Select the same model in both lists if you want to compare different versions.
- The version of the model to compare – the most recent version is selected by default.

3. Click **OK** to open the Compare Models window:
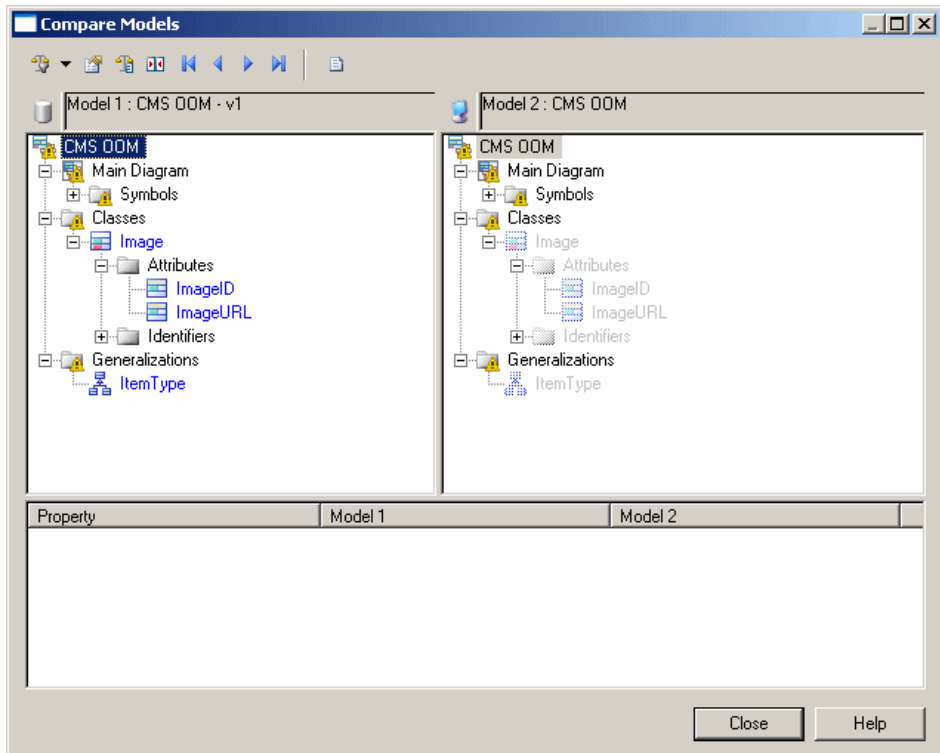
For detailed information about using this dialog, see *Comparing Models* on page 275.

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.
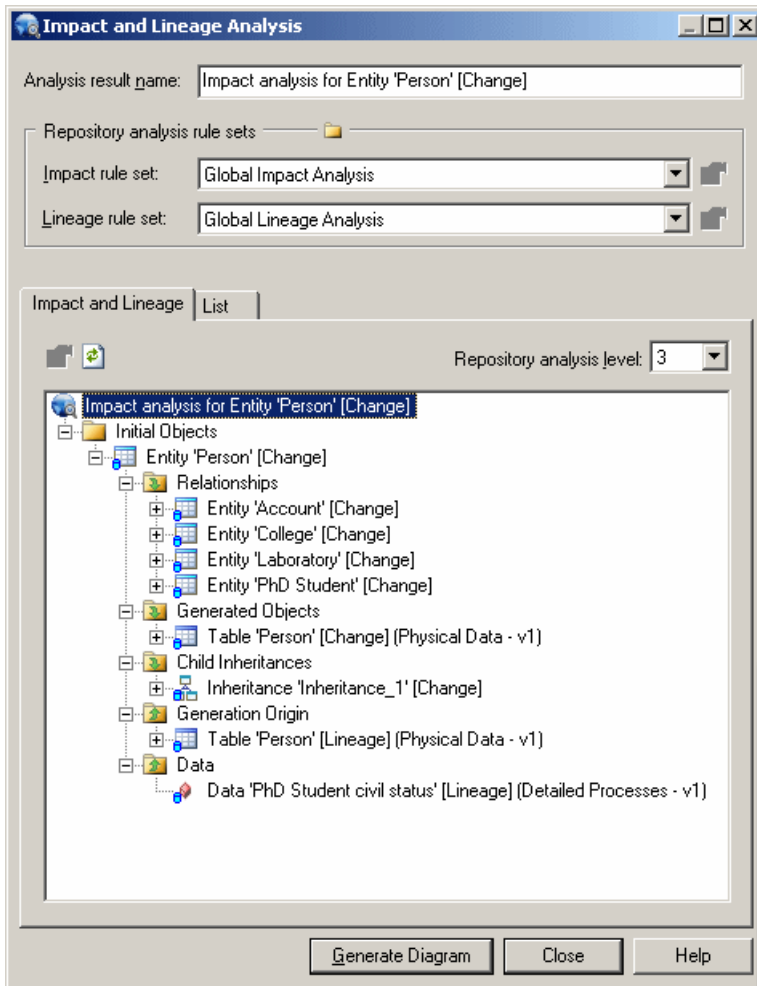
For detailed information about using this dialog, see *Chapter 17, Impact and Lineage Analysis* on page 499. For information about the limitations of repository analyses, see *Repository Rule Set Limitations* on page 518.

Note: By default, impact and lineage analyses launched from the Repository tab use rule sets checked into the Library folder at the root of the repository. To use other rule sets to perform analyses in the repository (see *Checking Rule Sets into the Repository* on page 517), use the **Choose Path** tool above the rule set fields to select the folder containing your rule sets.

Freezing Document Versions

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

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

Note: To freeze a document, you must have the Freeze versions right (see *Specifying User Rights* on page 524) and the appropriate permissions on the document.

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

You can freeze a document:

*   During check in (see *Checking Documents into the Repository* on page 292) by selecting one or both of:
    *   the **Freeze before check in** option - to preserve the state of the model before your changes
    *   the **Freeze after check in** option - to preserve the state of the model including your changes (this option can be selected by default if you select the **Freeze after check in** general option (see *Repository General Options* on page 407.
*   At any time, by right-clicking the document, folder, project, or root node and selecting the **Freeze** command. Performing a freeze on a folder, project, or root node will freeze all its contents.
*   At any time, by clicking the **Freeze** tool on the property sheet of:
    *   a document - on the **Versions** tab
    *   a branch - on the **Members** tab
    *   a configuration - on the **Members** tab

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

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

Unfreezing Document Versions

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

You can unfreeze a frozen document at any time by:

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

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

Deleting Documents and Document Versions

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

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

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

You can delete a frozen document version by:

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

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

Locking Document Versions

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

Locked documents display a lock icon in the repository browser:

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

Note: To lock a document, you must have the Lock versions right (see *Specifying User Rights* on page 524) and the appropriate permissions on the document.

You can lock a document:

- During check out (see *Checking Documents out of the Repository* on page 302) by selecting the **Lock before check out** option.
- At any time, by right-clicking the document, folder, project, or root node and selecting the **Lock** command. Performing a lock on a folder, project, or root node will lock all its contents.
- At any time, by clicking the **Lock** tool on the property sheet of:
  - a document - on the **Versions** tab
  - a branch - on the **Members** tab
  - a configuration - on the **Members** tab
- Using the List of Locks (see *Managing Document Locks in the List of Locks* on page 321).

Unlocking Documents

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

You can unlock a locked document:

- During check in (see *Checking Documents into the Repository* on page 292) by selecting the **Unlock after check in** option.
- At any time, by right-clicking the document, folder, project, or root node and selecting the **Unlock** command. Performing an unlock on a folder, project, or root node will unlock all its contents.

- At any time, by clicking the **Unlock** tool on the property sheet of:
  - a document - on the **Versions** tab
  - a branch - on the **Members** tab
  - a configuration - on the **Members** tab
- Using the List of Locks (see *Managing Document Locks in the List of Locks* on page 321).

Managing Document Locks in the List of Locks

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

The following tools are available on the List of Locks:

Tool	Description
🖼	Settings – Opens the Lock Document dialog for the selected document, which shows details of the lock upon it.
🖼	Add Lock – Opens the Lock Document dialog. Select a document to lock in the Name list and a comment, and click OK to lock it and return to the List of Locks.
🖼	Add Related Document Versions - [models only] Locks any models that the selected model references through external shortcuts. The related models are automatically added to the List of Locks.
🖼	Add Document Versions from a Configuration – Opens the Select Configuration dialog box, which allows you to select a configuration for locking. Document versions contained within the configuration are locked and added to the List of Locks.

Tool	Description
✕	Delete – Unlocks the selected document, and deletes it from the List of Locks. Note that this tool does not delete the document version.

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

Grouping Document Versions in a Configuration

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

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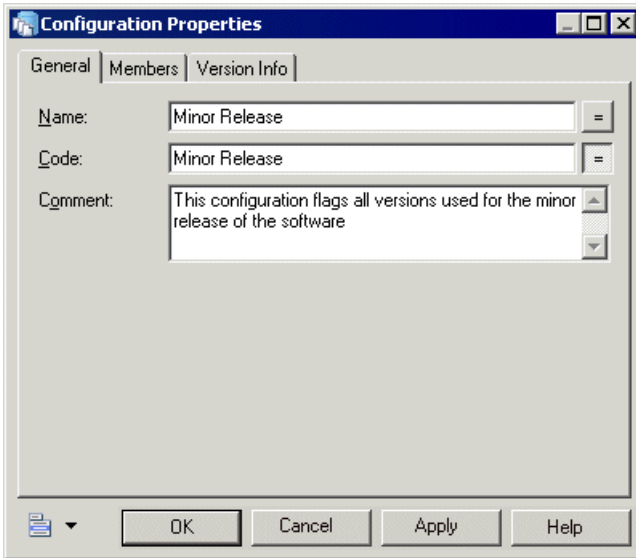
4. Click the **Members** tab to add document versions to the configuration, using the following tools.

Tool	Description
🗐	Properties – Opens the selected repository document property sheet.

Tool	Description
	Add Document Version – Opens the Add Document to Configuration dialog, which allows you to specify the version of a document to add to the configuration.
	Add Related Document Versions – [models only] Adds any models that the selected model references through external shortcuts. The related models are automatically added to the configuration.
	Add Document Versions From a Configuration – Opens the Select Configuration dialog, which lets you select a configuration, whose members will be added to the configuration.
	Check Out – Checks out the selected repository document version (see *Checking Documents out of the Repository* on page 302).
	Freeze – Freezes the selected repository document version (see *Freezing Document Versions* on page 318).
	Unfreeze – Unfreezes the selected repository document version (see *Unfreezing Document Versions* on page 319).
	Lock – Locks the selected repository document version (see *Locking Document Versions* on page 320).
	Unlock – Unlocks the selected repository document version (see *Unlocking Documents* on page 320).
	Delete – Deletes the selected document version from the configuration (see *Deleting Document Versions* on page 319).
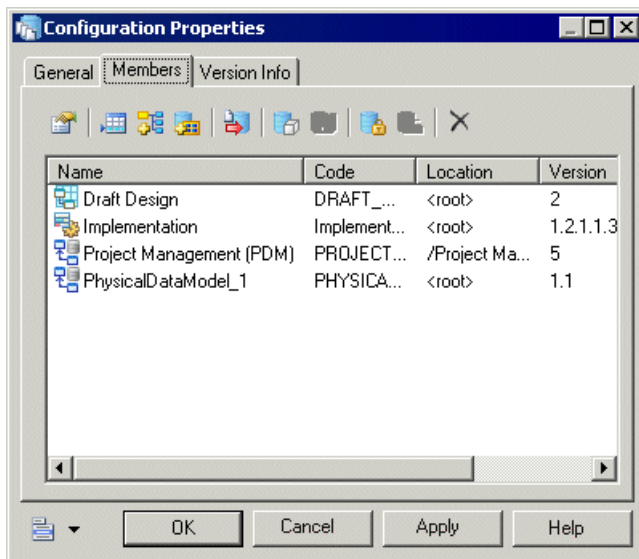
5. Click **OK** to return to the List of Configurations.

In the following example, the versions used for a minor release are included in a configuration called Minor Release:
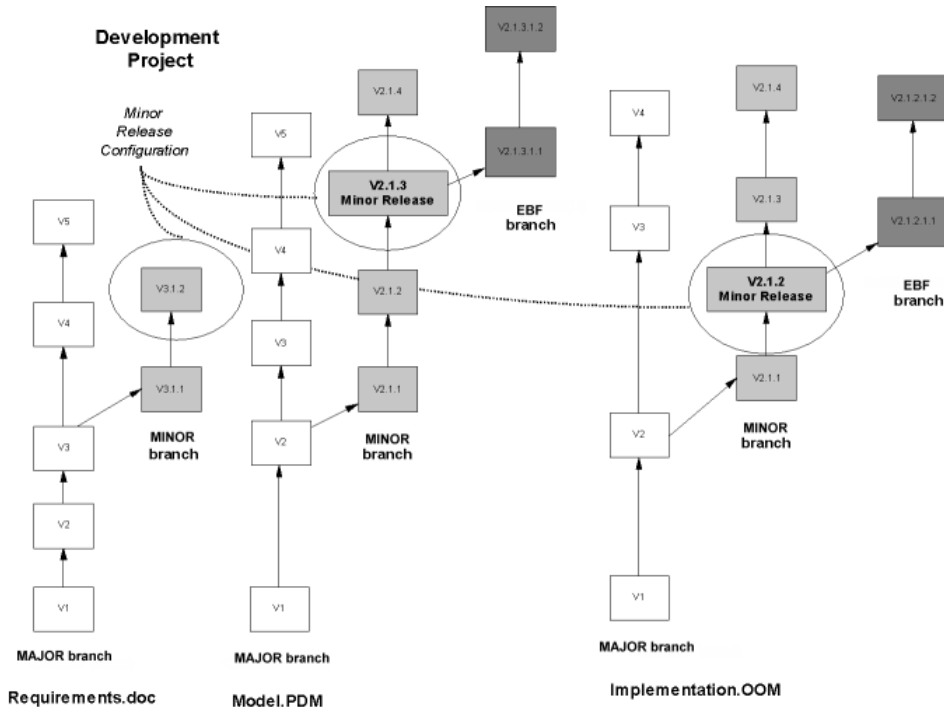
Creating a Configuration from a Project or Folder

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

The property sheet of the new configuration opens. You can modify the default name and add a comment on the **General** tab and modify the document versions it contains on the **Members** tab (see *Grouping Document Versions in a Configuration* on page 322).

Checking Out a Configuration

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

1. Select **Repository > Check Out** to open the Check Out Multiple Documents dialog (see *Using the Check Out Multiple Documents Window* on page 304).

2. Click the **Add Document Versions from a Configuration** tool, select the configuration to check out, and click **OK** to add the document versions it contains to the list.

3. Click **OK** to check out the documents.

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

Branching Version Trees

Users with the `Manage Branches` right can create branches in the repository to support concurrent development of multiple versions of documents. You can use permissions to control access to individual branches and integrate changes made in one branch into another.

You can branch the contents of any folder in the repository. Branch names must be unique for the folder, but may be reused in another folder. In the following example, 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`:
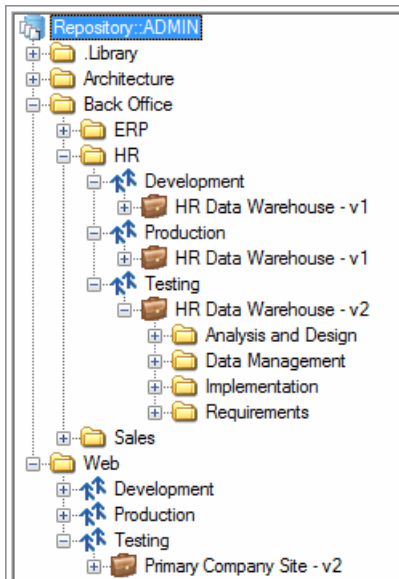
> **Note:** Prior to v16.1, the repository browser only supported displaying a single branch at a time. When you upgrade an existing repository to v16.1, all branches will become, by default, visible for all users. You may wish to simplify viewing the repository by deleting documents and branches that 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 *Installation Guide > Installing the Repository > Upgrading the Repository*).

1. Right-click the folder that you want to branch (and on which you have at least `Write` permission), and select **New > Branch**.

> **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:
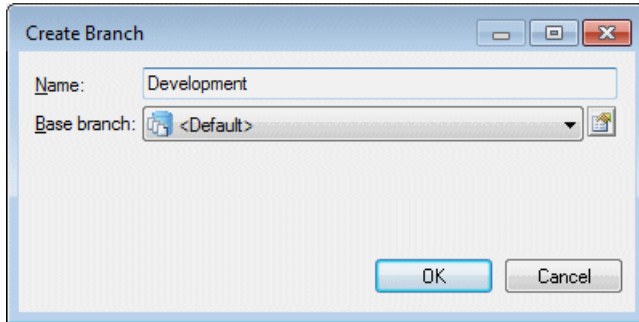
The base branch can be any of the sibling branches (or Default if this is your first branch in this location), and all the contents of the base branch will be copied to the new branch. Alternatively, you can select None to create an empty branch.

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* on page 327).

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**.

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. You can simplify the view of the repository for users and groups by revoking their permissions on branches that do not concern them.

1. Right-click a branch in the repository browser and select **Permissions** to open the branch property sheet at the **Permissions** tab:
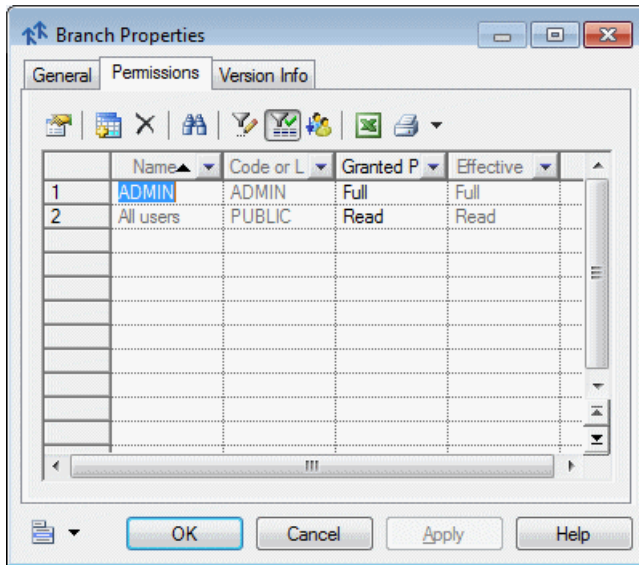
2. If the `All Users` group is present, delete it from the list. Also delete any other users or groups that should not have access to the branch from the list.

3. 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**.

4. Set each of the groups Granted Permission to an appropriate level (see *Granting Access Permissions on Repository Items* on page 528).

5. Click **OK** to complete the changes to the branch permissions.

From now on, users connecting to the repository will only see the branch if they belong to one of the groups listed.

Integrating Changes from One Branch to Another

You can integrate changes made to a model or other document (or to a project or the contents of a folder) in one branch to another branch. You must have at least Write permission on the branch to which you want to integrate the changes.You can select individual changes to integrate on an object-by-object basis, and your choices are retained as defaults for subsequent integrations.

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 target branch to integrate your changes to, and can optionally enter a comment (click the **Advanced** button to access the same options as are available in the Check In Document dialog (see *Check-In Parameters* on page 297):

CHAPTER 8: The Repository
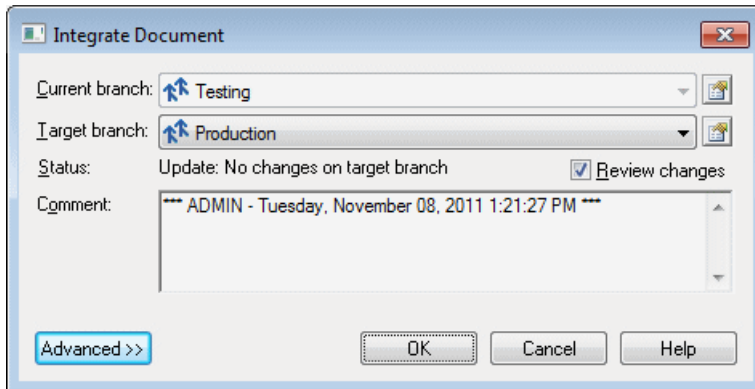
- For a folder or project, you must select a target branch to integrate your changes to and can optionally enter a comment:
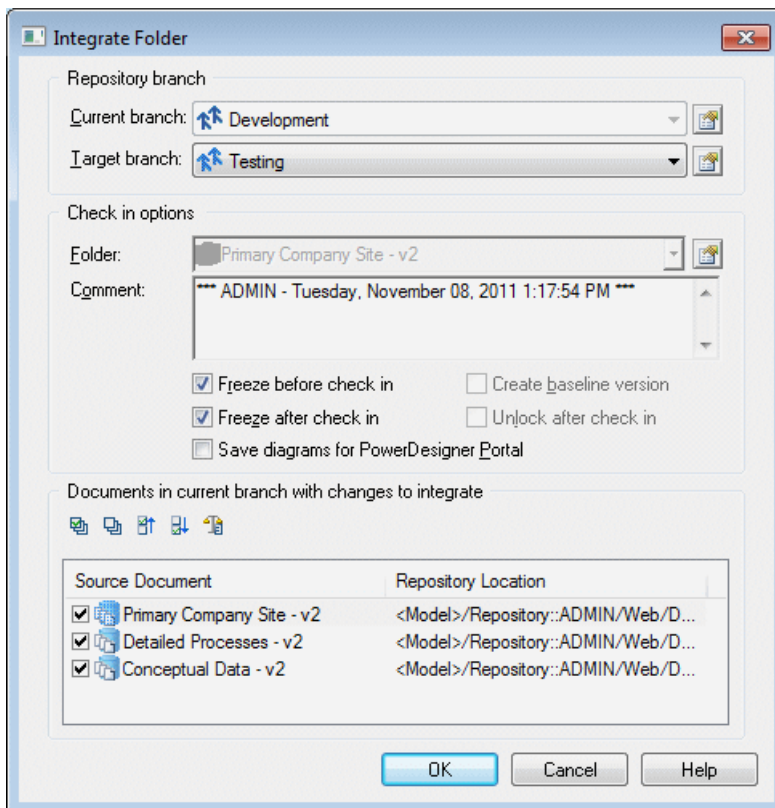
Note: Only documents with changes since the creation of the branch (or the last integration) are listed and you can deselect documents in the list to exclude them from the integration.

2. Click **OK** to begin merging changes from the current branch to the target branch. For PowerDesigner models, a merge dialog (see *Merging Models* on page 284) opens to allow you to review changes to individual objects:
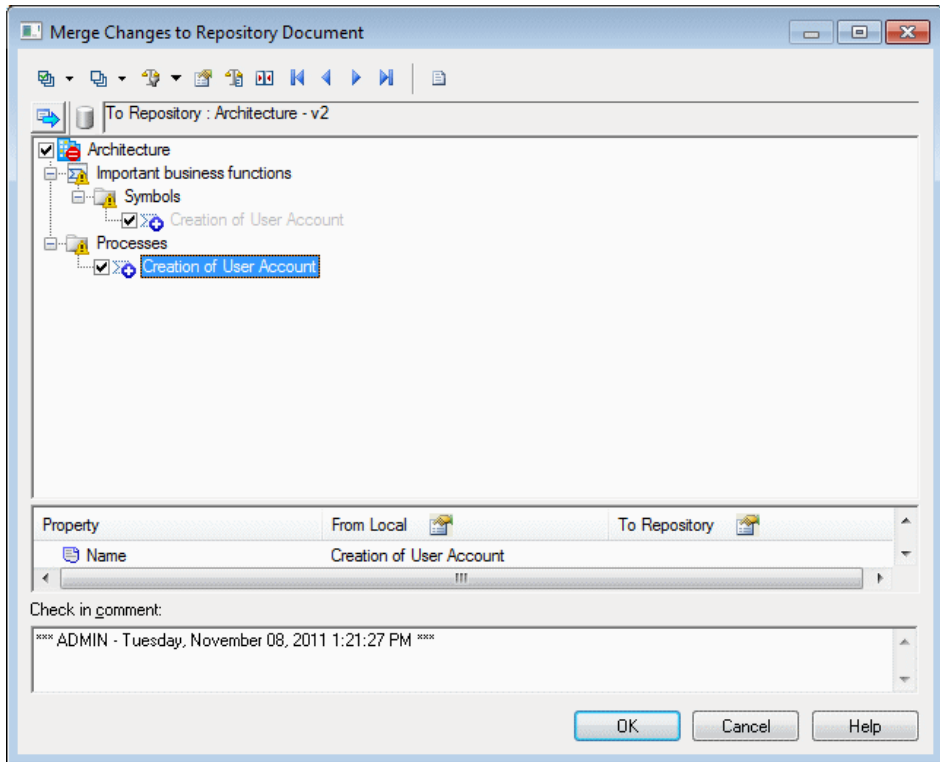
3. Select or reject each change as appropriate and then click **OK** to merge the selected changes from the current branch model to the target branch model.

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.

Deleting a Branch

Administrators and users with the `Manage Branches` right can remove unused branches by right-clicking them and selecting **Delete**.

Before deleting a branch, you must delete all its contents (see *Deleting Documents and Document Versions* on page 319).

Warning! Deleting a branch cannot be undone.

CHAPTER 9   **The PowerDesigner Portal**

The PowerDesigner Portal allows you to view the contents of your PowerDesigner repository in your web browser. You can share your models with a wide audience, including those who are not PowerDesigner users. Users with the appropriate rights can edit object descriptions directly in the Browser.

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

For information about the PowerDesigner repository, see *Chapter 8, The Repository* on page 291.

Logging into the PowerDesigner Portal

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

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

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

Click **Show profile information** to display details of the selected profile or **Advanced options** to manually enter a host name and port instead of using a profile.

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

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

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

PowerDesigner Portal Repository Explorer

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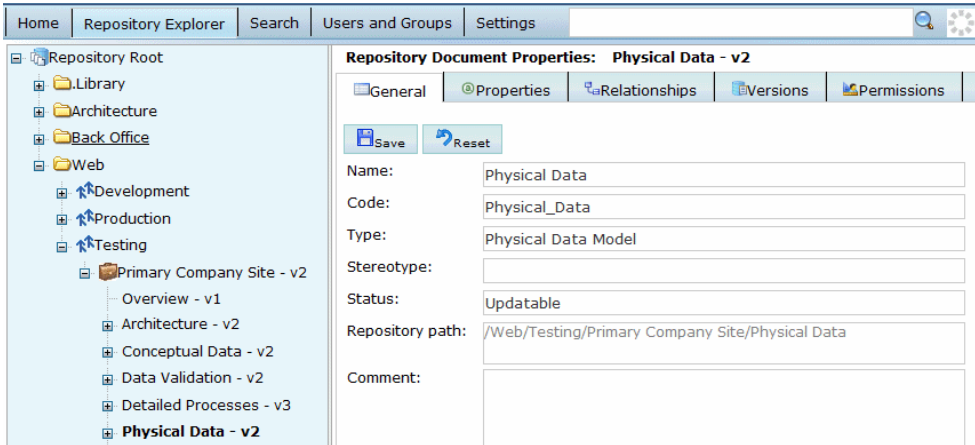
The explorer can contain the following types of node.

- Root - Displays the current branch.
- Folders - Subdivide and organize the documents in the repository.
- Branches - Allow concurrent development of multiple versions of documents.
- Projects – Group related models and other documents.
- Documents – Projects, models, multi-model reports, and external application files such as MS Office files, or graphic files.
- Packages - Subdivide and organize model objects in models.
- Object categories - Organize model objects in models.
- Diagrams - You can click on object symbols in the diagrams to go to the relevant object.

• Model objects

Repository Document and Object Properties

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

The following tabs are available:

• **General** - displays basic information about the object. Users with the appropriate rights can edit the object comment, description, and annotation. For shortcuts, click the entry in the **Target object** field to open the object that the shortcut references.
• **Properties** - [models, packages, and model objects] displays the properties of the object.
• **Diagram** - [diagrams] displays the diagram. Click on symbols in the diagram to go to the related object.
• *Sub-objects* - [objects] displays a list of this type of sub-object (for example, table columns, keys, or indexes, or class attributes or operations).
• **Relationships** - [objects] displays lists of the other objects with which the object is related.
• **Versions** - displays the various versions of the document stored in the repository, along with information about the current version of the document, its creation and last modification. For external application files, click the icon in the **Download** column to download the file.
• **Analysis** - displays the impact of a change to the object (impact analysis) and identifies the objects that take part in the definition of the object (lineage analysis) (see *Impact and Lineage Analysis in the PowerDesigner Portal* on page 335).
• **Permissions** - [models, packages and other documents] lists those users and groups with permissions on the document (see *Adding User or Group Permissions for a Repository Item* on page 333).
• **Compare** - [models and objects] allows you to compare different versions of models and objects (see *Comparing Model and Object Versions* on page 334).
• **Subscriptions** - [projects, folders, and models] allows you to subscribe to email notifications of changes to the object (see *Subscribing to Change Notifications* on page 334).

Adding User or Group Permissions for a Repository Item

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

Note: These permissions can also be controlled through PowerDesigner (see *Granting Access Permissions on Repository Items* on page 528).

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

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

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

4. [optional – for projects, folders, models] Select **Cascade permission changes to folder contents** to apply your changes to children of the node.

5. Click **Save** to save your changes and return to the **Permissions** tab.

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

Subscribing to Change Notifications

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

1. Navigate to the project, model, or folder for which you want to subscribe in the Repository Explorer, and then click on the **Subscriptions** tab.

2. Select the Email checkbox against your name in the **Notification Method** column, and then click **Save**.

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

Comparing Model and Object Versions

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

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

2. Select the branches and versions numbers of the two versions of the model or object that you want to compare.

3. [optional] Click **Expand All** to open all the nodes.

4. [optional] Click any of the child nodes to view changes to their properties.

For detailed information about working with this feature, see *Comparing Models* on page 275.

Impact and Lineage Analysis in the PowerDesigner Portal

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 selecting it in the **Repository Explorer** and clicking the **Analysis** tab.

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 **Analysis level** list.
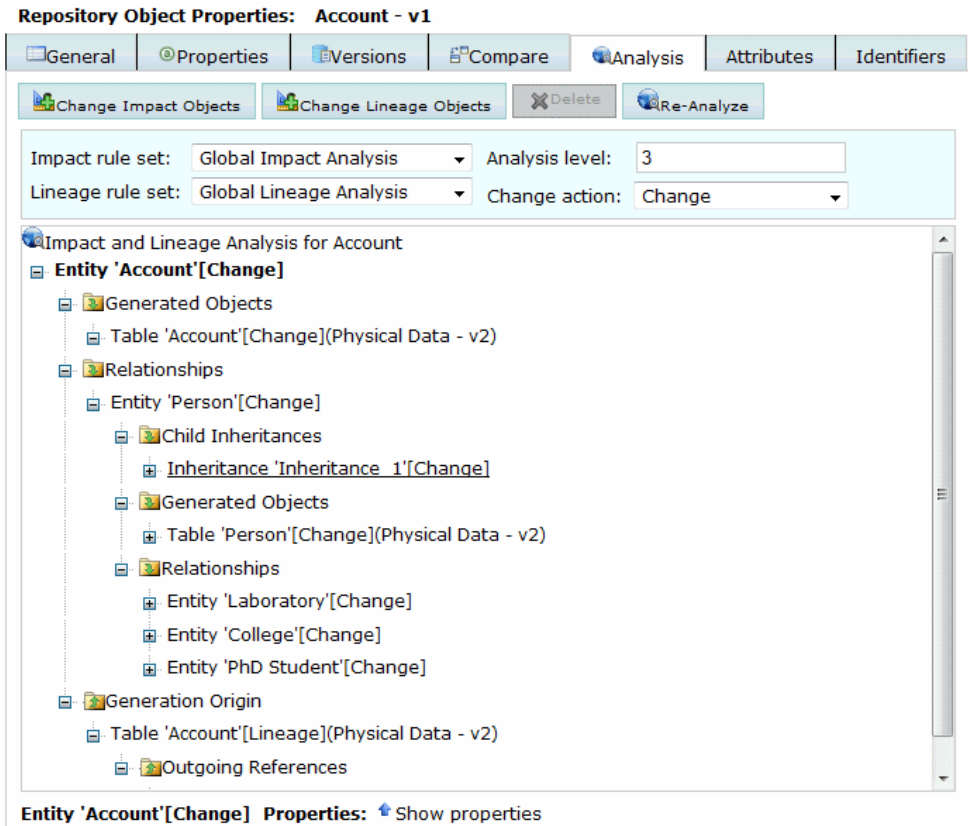
Note: By default, impact and lineage analyses launched from the PowerDesigner Portal use rule sets checked into the **Library** folder at the root of the repository.

For detailed information about using this feature, see *Chapter 17, Impact and Lineage Analysis* on page 499.

Editing Object Comments and Descriptions

Users with the `Edit PowerDesigner Portal Objects` repository right can edit the **Comment**, **Description**, and **Annotation** properties of objects directly in the PowerDesigner Portal.

Note: For information about repository rights, see *Granting Rights to Users and Groups* on page 342.

1. Navigate to the object whose properties you want to edit in the Repository Explorer and select the **General** tab.

2. Make any appropriate modifications directly in the text-only **Comment** field and the RTF **Description** or **Annotation** fields.

   The RTF fields provide integrated toolbars to allow you to modify the font (Arial, Comic Sans, Courier New, Tahoma, Times New Roman, and Verdana), font size, and appearance, to change justification and add bulleted lists, images (BMP, JPEG, non-animated GIF, and PNG), tables, and links.

3. Click the **Save** button to save your changes.

   Warning! If you navigate to another object without clicking **Save**, all of your changes will be lost.

PowerDesigner Portal Search

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

1. Enter your search string in the search field.

The following special operators are supported:

Example	Returns
`sales`	Objects containing the text `sales`.
`sales AND account` (or `+sales +account`)	Objects containing the text `sales` and `account`.

Example	Returns
`sales AND NOT account` (or `+sales -account`)	Objects containing the text `sales` but not `account`.
`sales OR account` (or `sales account`)	Objects containing the text `sales` or `account`.
`sales~`	Objects containing text like `sales`.
`"sales account"`	Objects containing the exact string `sales account`.
`sal?s`	Objects containing text such as `sales` or `salts` (? represents a single wild card character).
`sal*s`	Objects containing text such as `sales` or `salaries` (* represents one or more wild card characters).

2. Click the **Quick search** button to display your results in the **Search** tab:

Click on a result line to display the object's property sheet.

Advanced PowerDesigner Portal Search

You can perform searches with more complicated and multiple criteria from the **Search** tab. To use a saved search, click its **Run** button.
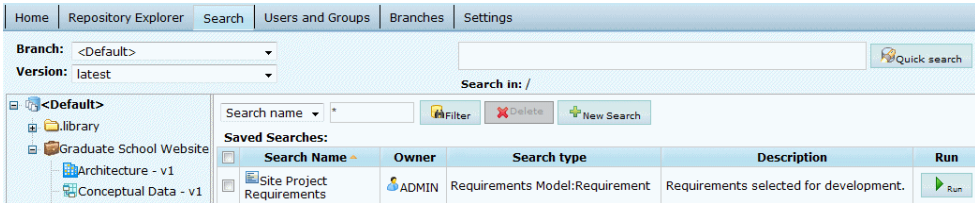
1. Click the **Search** tab and then click **New Search**.
2. [optional] Select a node in the Repository Explorer tree such as a branch, project, or model to restrict the scope of your search. By default, you search the entire repository.

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

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

   In this example, I select `Physical Data Model` and `Table`:
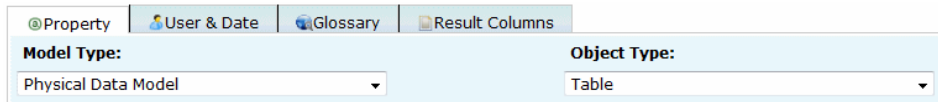
4. Select a property, operator and value to search against. Depending on the property that you select, the choice of operator may be restricted and/or values may be suggested. Note that the case sensitivity of your searches depends on your repository DBMS.

   When using the `Contains` operator, the following special characters allow you to use basic regular expressions for text searches:
   - * - none to any number of characters
     For example: W* finds "Work" and "Washington"
   - ? – exactly one character
     For example: ????ER finds "Writer" and "Seller" but not "inner"
   - \ - escapes *, ?, or \
   - true/false - Boolean value (True is when the check box is selected in the interface)
5. Click the **Add** button to add the complete search clause to the **Search clause** table.

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

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

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

   In this example, I refine my search to look only for tables that are estimated to have more than 100 records:

7. [optional] Click the **User & Date** tab and specify additional search clauses based on user names and creation and modification dates.

8. [optional] Click the **Glossary** tab and specify additional search clauses based on glossary objects (terms, related terms, synonyms, and keywords) with which objects may be associated.

9. [optional] Click the **Result Columns** tab and select the columns that you want to display on the result page. By default, only the object type and name are shown.

10. [optional] Select the **Include Shortcuts** check box to include shortcuts to objects in your search results.

11. Click the **Search** button to start your search.

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

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

    •   Select either pdf or xls and click the **Report** button to generate an Excel or PDF version of your results.
    •   Select an attribute, enter a value in the search box at the top of the screen and click the **Search within results** button to further refine your search.
    •   Click the **Back** button and:
        •   Modify your search string.
        •   Enter a Saved search name and description and then click the **Save** button to make your search available for reuse on the Search page.

PowerDesigner Portal Users and Groups

Repository users with the `Manage users` right can view, create, modify, and deactivate users and groups, and manipulate their assignment to groups via the PowerDesigner Portal.
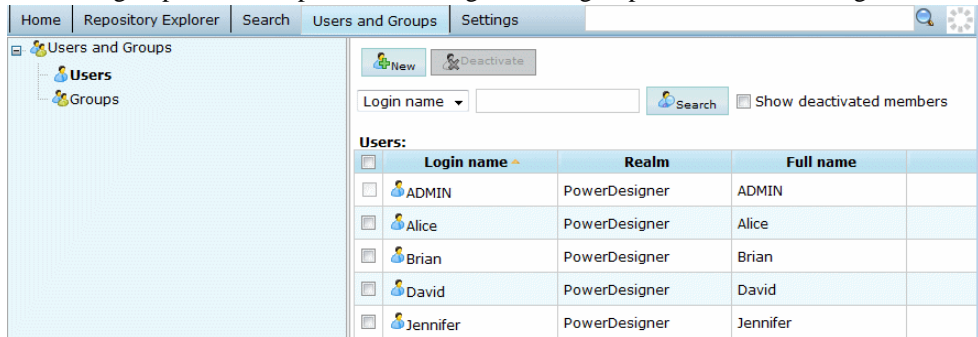
Users and groups can be created here or through PowerDesigner (see *Chapter 18, Controlling Repository Access* on page 521).

Creating a Repository User

You can create users on the **Users and Groups** tab.

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

2. Click **New**, select to create a PowerDesigner or external user, and enter a login name.

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

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

4. [optional] Click the **Parent Groups** tab and add the user to any appropriate groups (see *Adding a Member to a Group from the Member's Property Sheet* on page 343).

5. [optional] Click the **Rights** tab and assign the user any appropriate rights (see *Granting Rights to Users and Groups* on page 342).

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

6. Click **Save** to create the user and return to the list of users.

Creating a Repository Group

You can create groups on the **Users and Groups** tab.

1. Click the **Users and Groups** tab and click on **Groups** on the left of the screen to display the list of groups.
2. Click **New**, select to create a local or remote group, and enter a group and full name.
3. [optional] Enter an email address and/or a description for the group.
4. [optional] Click the **Members** tab and add any appropriate users to the group (see *Adding a Member to a Group from the Group's Property Sheet* on page 343)
5. [optional] Click the **Rights** tab and assign the group any appropriate rights (see *Granting Rights to Users and Groups* on page 342)

   Note: Rights granted to a group are inherited by its members. If a user connects to the repository without access rights, he cannot see any document in the browser.

6. Click **Save** to create the group and return to the list of groups.

Granting Rights to Users and Groups

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

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

1. Click the **Users and Groups** tab, select Users or Groups on the left of the screen, and click the appropriate entry in the list to go to its property sheet.
2. Click the **Rights** tab and select the check boxes corresponding to the rights you want to assign. The following rights are available:

   - `Connect` - To connect to the repository.
   - `Freeze` - To freeze document versions.
   - `Lock` - To lock document versions.
   - `Edit PowerDesigner Portal Objects` - To edit objects in a Web browser (see *Editing Object Comments and Descriptions* on page 336).
   - `Manage branches` - To create, modify, and delete branches.
   - `Manage configurations` - To create, modify, and delete configurations.
   - `Manage all objects` - To create, check in, check out, unlock, unfreeze, define permissions, and delete any document version.

     Note: This right is typically granted to data administrators, and implicitly includes `Full` permission on all repository documents to allow them to check in, freeze, lock, and delete any document.

- `Manage users` - To create, modify and delete repository users and groups, grant rights to users and groups, and add users or groups to a group.
- `Manage repository` - To create, upgrade, and delete the repository database.

3. Click **Save** to save your changes and return to the list of users or groups.

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

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

1. Click the **Users and Groups** tab, select **Users** or **Groups** on the left of the screen, and click the appropriate entry in the list to go to its property sheet.

2. Click the **Groups** tab to display the list of available groups and the list of groups to which the user or group belongs:

3. Select one or more groups from the Available groups list, and then click the right-pointing arrow to move them to the Parent groups list.

4. Click **Save** to save your changes and return to the list of users or groups.

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

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

1. Click the **Users and Groups** tab, select **Groups** on the left of the screen, and click the appropriate entry in the list to go to its property sheet.

2. Click the **Members** tab to display the list of available members and the list of child members:
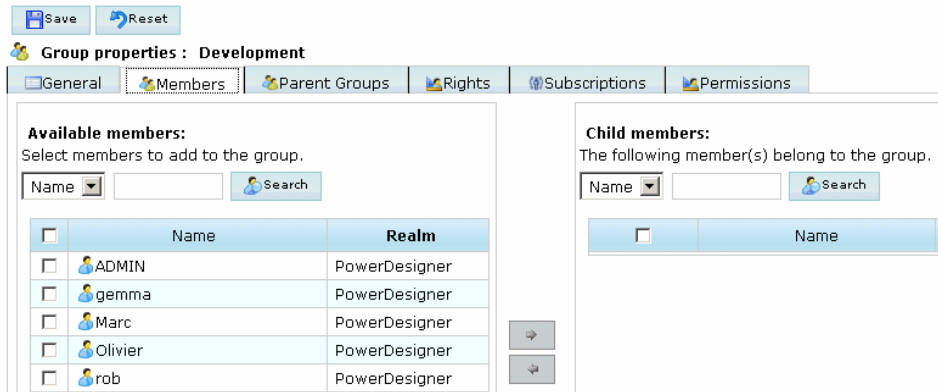
3. Select one or more members from the Available members list, and then click the right-pointing arrow to add them and move them to the Child members list.

4. Click **Save** to save your changes and return to the list of groups.

Deactivating a User

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

1. Click the **Users and Groups** tab, and select **Users** on the left of the screen to display the list of users.

2. Select a user in the list, click the **Deactivate** button, and then confirm the deletion by clicking **OK**.

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

Deleting a Group

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

1. Click the **Users and Groups** tab, and select **Groups** on the left of the screen to display the list of groups.

2. Select a group in the list, click the **Delete** button, and then confirm the deletion by clicking **OK**.

   The group is removed from the list. Deleting a group has the following effects:
   • The users who belonged to the group lose the group rights
   • All permissions on documents granted by the group and subscriptions via the group are canceled

- The group is removed from the groups to which it belonged
- The group is no longer displayed in the list of groups

PowerDesigner Portal Settings

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

PowerDesigner Portal Client Settings

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

Logging
The following settings allow you to control logging:

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

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

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

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

Theme

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

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

Creating a New Color Theme

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

1. Go to *install_dir*\Tomcat\webapps\cmr\theme and make a copy of one of the existing color theme directories to use as the basis of your new theme.
2. Rename the copied directory to the name that you want to see in the Color theme list on the Settings page.
3. Go to *new_theme_dir*\images\css-images\dgm to view all the component images that are used to construct the PowerDesigner Portal site.
4. Open each of the images in turn in your image editor, make appropriate, consistent changes to their color schemes, and then save them back into the folder.
5. Go to *new_theme_dir*\css and open cmr.css in your text editor to view the css code that controls the look of other elements of the PowerDesigner Portal site.
6. Make appropriate changes to the background-color and color statements and then save your changes.
7. Navigate to the PowerDesigner Portal site in your Web browser and click the Settings tab. Click **Refresh** to ensure that your changes are taken into account by the server.
8. Click **Settings > Client settings > Theme** and select your new color theme from the list.

9. [optional] Select an image to use as your site logo and enter a new logo title for your site.

10. Click **Save** to apply your changes.

PowerDesigner Portal Server Settings

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

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

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

LDAP
The following LDAP settings are available and can be set here or through PowerDesigner (see *Controlling Repository Access with LDAP* on page 529). You must restart the PowerDesigner Portal for changes to take effect:

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

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

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

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

Logging
For information about the Logging settings, see *PowerDesigner Portal Client Settings* on page 345.

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

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

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

Search

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

Setting	Description
Index Information	Specifies the last time the search index was rebuilt, the next time it will be built, its location, size, and the number of attributes indexed.
Index rebuild interval	Specifies the interval between rebuilds of the search index in minutes. By default, the index is rebuilt every 120 minutes. Click the **Rebuild Index** button to perform an immediate rebuild.

License

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

Setting	Description
Host	Specifies the name of the SySAM license server that serves licenses to Power-Designer Portal users.
Port	Specifies the port to use to access the license server.

Security

The following password policy settings are available and can be set here or through PowerDesigner (see *Defining a Password Policy* on page 533). You must restart the PowerDesigner Portal for changes to take effect:

Setting	Description
Password length	Specifies the minimum and maximum permitted length of passwords. This option cannot be disabled. The minimum 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.

Setting	Description
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.

Notification
The following change notification settings are available and can be set here or through PowerDesigner (see *Specifying an SMTP Server for Notifications* on page 534):

Setting	Description
SMTP host	Specifies the host name of the SMTP server used to send mail.
SMTP port	Specifies the port number of the SMTP server used to send mail.
Sender's email address	Specifies the email address from which to send 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 instruct PowerDesigner to use Secure Password Authentication (SPA).

In addition, you can control the content of notification emails by selecting items in the **Mail template** list and overriding the default text associated with them.

Enabling SSL

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

Using the Default Tomcat Server

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

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

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

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

Using an Existing SSL Implementation

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

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

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

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

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

Using an Existing Client Keystore

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

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

PowerDesigner Portal Server Status Settings

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

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

Troubleshooting the PowerDesigner Portal

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

Unicode Encoding with Sybase SQL Anywhere V10

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

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

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

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

Increasing Tomcat Memory

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

CHAPTER 10   **The PowerDesigner Plugin for Eclipse**

During your installation of PowerDesigner, you can choose to install a plugin to allow you to use PowerDesigner within your Eclipse environment. The PowerDesigner plugin for Eclipse is available for Eclipse v3.2 to v3.6.

The PowerDesigner plugin for Eclipse has the following limitations:

* You cannot use the Copy/Paste/Rename features available in the Eclipse contextual menu for PowerDesigner resources.
* In the standalone PowerDesigner application, you can modify the source code of a class or interface from its property sheet Preview tab. This feature is not available in the plugin for Eclipse.
* You cannot export an Eclipse project containing PowerDesigner resources. If you want to export a project, you must deselect any PowerDesigner resources on the File system page of the Export wizard or the export will fail.

Note: The PowerDesigner plugin for Sybase Workspace, which is called Enterprise Modeling, includes all the features of the standard PowerDesigner plugin for Eclipse. When modeling within the Sybase Workspace environment, we recommend that you use the Enterprise Modeling perspective.

Activating the PowerDesigner Perspective

Click the title bar of a view to make it active. (The title bar turns blue). The name of the active perspective is shown in the title bar of the window and its icon is pushed-in in the shortcut bar. We recommend you use the PowerDesigner perspective.

1. Click the Open a Perspective icon in the shortcut bar and select Other, or select **Window > Open Perspective > Other**, to open the Select Perspective dialog box:
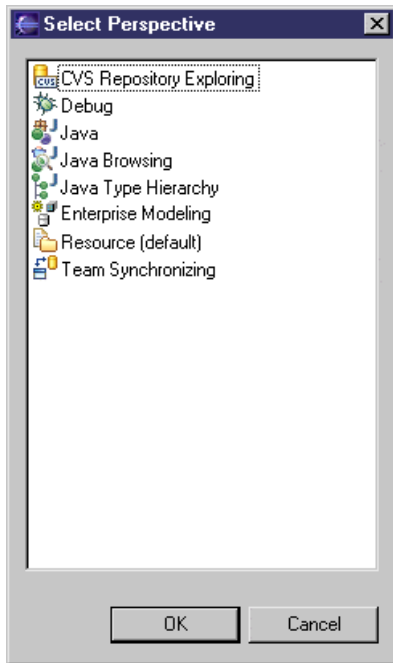
2. Select PowerDesigner in the list of perspectives, and then click **OK** to display the PowerDesigner perspective in the workbench window. The PowerDesigner perspective contains the following components:

   - *Model Explorer* – Equivalent to the standard PowerDesigner Browser. Allows you to manage the objects you use to perform a modeling task. It displays your models and the objects belonging to them in a tree view, and allows you to rapidly navigate between them. The Model Explorer also has a tab that gives you access to a PowerDesigner repository, where you can store all your models and associated files.
   - *Navigator* - displays a tree view of all the resource files attached to Eclipse projects open in the workbench window. These resource files can be model files, diagram files, source code files, specification files, or any type of file. You can use the Navigator to open models, create new projects and models, or even open object property sheets.
   - *Editor Area* – Equivalent to the standard PowerDesigner canvas. The primary pane that displays your present model diagram or report outline.
   - *Modeling Output* - shows the progress of any PowerDesigner process, such as checking a model or generating or reverse engineering a database.
   - *Modeling Result List* - displays the results of a search or a model check.

Note: Once you have activated the PowerDesigner perspective, an icon to switch to it is displayed in the shortcut bar. You can use the **Window** menu to open, customize, and reset this and other perspectives.

Manually Activating PowerDesigner Toolbars

Specific toolbars are available in the PowerDesigner perspective. If they do not appear, you can activate them manually.

1. Select **Window > Customize Perspective** or right-click the toolbar section and select **Customize Perspective** in the contextual menu, to open the Customize Perspective dialog box.

2. Select the **Shortcuts** tab, and:

    - Select `New` in the **Submenus** list, then select `PowerDesigner` in the **Shortcut Categories** list, and select `Model` and `Modeling Project` in the **Shortcuts** list.
    - Select `Open Perspective` in the **Submenus** list, and select `PowerDesigner` in the **Shortcuts** list.
    - Select `Show View` in the **Submenus** list, then select `PowerDesigner` in the **Shortcut Categories** list, and select `Model Explorer`, `Modeling Output`, `Modeling Result List`, and `Tool Palette` in the **Shortcuts** list.

3. Select the **Toolbar Visibility** tab, and select `PowerDesigner Diagram`, `PowerDesigner Standard` and `PowerDesigner View` in the **Tool Bar Structure** list.

4. Click OK.

    The PowerDesigner toolbars that are common to all PowerDesigner modules appear. Additional, module-specific toolbars will appear automatically when you open or create a model.

Setting PowerDesigner Eclipse Preferences

The PowerDesigner plugin for Eclipse provides options to customize your environment.

1. In Eclipse, select **Window > Preferences** to open the Preferences dialog.

2. Select **Sybase, Inc > PowerDesigner** or **Enterprise Modeling** and set preferences as appropriate:

Preference	Description
Close diagram editors on exit	By default, when you restart Eclipse, any diagrams that were previously open will be automatically reloaded. Select this option to prevent reloading.
Flag diagram editor as dirty if parent model requires save	Select this option if you want to be reminded that a model has been modified before you save the changes.
Automatically merge models on code-to-model synchronization	Select this option if you want the merge process to be silent during reverse engineering.

3. Click **OK**.

   For information about the standard PowerDesigner options and preferences, see *Chapter 12, Customizing Your Modeling Environment* on page 389.

Creating a Modeling Project in Eclipse

You can create a modeling project in Eclipse to group together all your models and other resources.

1. Select **File > New > Project** to open the New Project window:
2. Select **PowerDesigner > Modeling Project** in the list of Wizards and click Next to open the PowerDesigner New Project window.
3. Specify the project name and location, and select the type of PowerDesigner project to create. PowerDesigner project templates allow you to create projects that are already populated with models and/or that contain matrices that help you to follow various modeling frameworks, such as FEAF.
4. Click OK to create the project.

For detailed information about working with projects, see *Chapter 3, Projects and Frameworks* on page 53.

Creating a Model in Eclipse

A model is the basic work unit in PowerDesigner. Every model is contained within a project, and contains at least one diagram and any number of other objects. Though a model may be split into packages for organizational reasons or may contain several diagrams, it remains the fundamental basis for your modeling work.

1. Select **File > New > Model** to open the New Model dialog box.
2. Select a model type and specify whether you want to create an empty model or one based on a template, then click **Next** to go to the Destination Container page.
3. Select a project or project folder in which to create the model. Use the buttons to create a project (see *Creating a Modeling Project in Eclipse* on page 358) or folder if necessary, and then click **Next** to go to the Model Definitions page.
4. Enter a name for your model and (for BPM, PDM, OOM, and XSM) select the target of the model. Depending on the type of model, you may also need to specify the first diagram to create.
5. Click **Next** to go to the Extensions page, and select any extension files that you want to attach to the new model.

6. Click **Finish** to create the model in your project in the Model Explorer, and open its default diagram in the editor area.

Importing an Existing Model into Eclipse

To open an existing model you have to import it into Eclipse.

1. Select **File > Import** to open the Import dialog box.
2. Select Model in the list and click Next.
3. Type a model filename in the corresponding box or click the Browse button to select a model from a selection dialog box.
4. Select the Create a linked resource check box if you want to associate the model to the current project without copying the file to the project location.
5. Select an existing project or create a new project where to open the existing model and click Finish.

   The model default diagram is displayed in the Editor area.

Generating an Eclipse Java Project from an OOM

You can model the structure of your Java application (including EJBs, Servlets, and JSPs) in a PowerDesigner OOM and then generate an Eclipse Java Project in which you will complete the implementation of the classes, and from which you will compile, package, deploy, and debug the application.

You can use the PowerDesigner plugin for Eclipse to enable round-trip engineering for Java development:

- Perform high level analysis and design using PowerDesigner
- Design and create Java components in PowerDesigner
- Generate an Eclipse Java project containing the following files:
  - A `.project` file - that defines the name of the project and the build commands
  - A `.classpath` file - that defines the source directory, the binary directory and the list of libraries
  - Source files and other files
  - A `build.xml` for Ant build script - to specify the necessary build tasks and libraries.
- Finish the implementation of Java classes within the generated Eclipse project
- Compile, package, deploy and debug the application
- Reverse engineer the final Java code to synchronize the PowerDesigner model

1. Select **Tools > General Options** and click the Variables category, then add the following variables if they are not already present:

- ECLIPSE_HOME - your Eclipse home directory
- J2EE_HOME - your J2EE SDK home directory
- JWSDP_HOME - [optional] your Java Web Service Developer Pack home directory

2. [optional] Preview the Eclipse `.project`, `.classpath` and Ant `build.xml` files by right-clicking the model node in the Browser and selecting Properties. Each of the files is available on its own sub-tab on the **Preview** tab of the property sheet.

3. Select **Language > Generate Java Code** to open the Generation dialog.

4. Enter the project folder where you want to generate the files in the **Directory** field. The name of this folder will be used as the project name.

5. Verify that Eclipse is selected on the **Targets** tab.

6. [optional] Click the **Selection** tab and specify the packages, classes and interfaces that you want to generate. By default, all objects are generated.

7. Click the **Options** tab and review the generation options, including those that control the Eclipse version to generate for and whether to overwrite existing `.project` and `.classpath` files.

8. [optional] Click the **Generated Files** tab and specify which files will be generated. By default, all files are generated

9. Click **OK** to generate the Eclipse Java project.

If you are generating a new project from the PowerDesigner plugin for Eclipse, the project opens automatically.

When generating code for an existing project, you must refresh the project by right-clicking the project in the Eclipse Package Explorer and selecting **Refresh**.

If you have generated the Eclipse project from outside Eclipse, you will need to import the project into Eclipse by selecting **File > Import**.

Synchronizing an OOM with Its Java Source

You can synchronize an Object Oriented Model (OOM) with its Java source code, so that each time you modify the model, a source code file is automatically created or updated, and when you modify and save a source code file, the changes appear in the model. Synchronization is only available for Java OOMs.

You can activate synchronization from the Model Explorer or Navigator. A model is synchronized per *package*.

Note: You can only synchronize each package with one set of source files and each set of source files with one package. If you try to synchronize a second model with a package, its check box is unavailable.

1. In the Model Explorer, right-click the model and select **Synchronize**, then select the **Synchronized** check box:
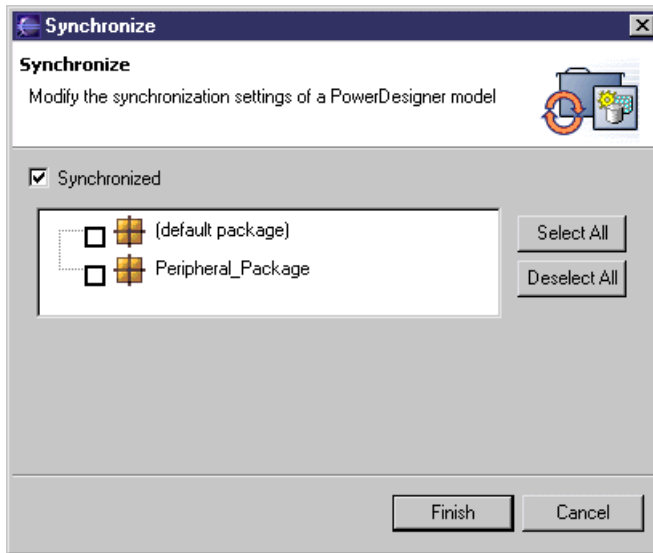
Note: You can, alternatively, enable synchronization from the Navigator by right-clicking the model, selecting **Properties**, selecting **PowerDesigner** in the left pane and the **Synchronized** check box in the PowerDesigner page.

2. Select the packages you want to synchronize with their source code. The default package regroups all the objects at the root of the model.

3. Click **Finish**.

   In the Navigator, the model bears a synchronized label, each synchronized package bears a synchronized symbol overlay, and a Java file (.java) and compiled file (.class) appear for each synchronized object:
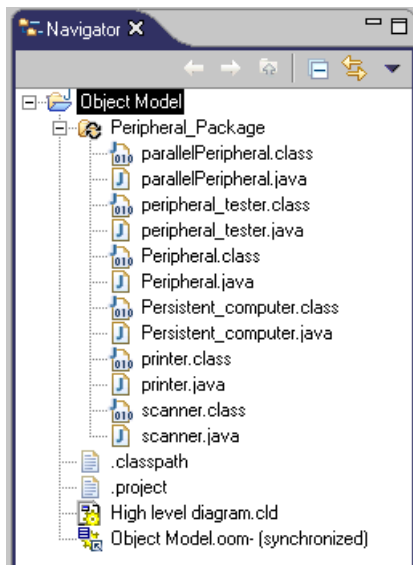
You can double-click a Java or compiled file, to display its source code in the Editor view.

Example: Modifying Attributes in a Synchronized Model

This example is developed from the example synchronized OOM, `demo.oom`, and demonstrates the creation and deletion of attributes in a synchronized model.

Note: If you change generation templates, you can regenerate source code manually by right-clicking a class, interface, synchronized package, or model, in the Model Explorer or in a class diagram, and selecting **Refresh Source Code**.

1. Open the property sheet of the `printer` class, click the **Attributes** tab, and add a `usbPort` attribute with a boolean data type and public visibility.

2. Click **OK** to return to the class diagram, where the usbPort attribute is displayed in the printer symbol:

3. Right-click the printer symbol or the printer item in the Model Explorer and select **Edit Source Code** to open the source code file of the printer class in a pane directly under the diagram:

```
printer.java ✕
package Peripheral_Package;

import java.util.*;

/** @pdOid d89d6951-81d1-11d5-ad8a-0008c752dcf8 */
public class printer extends parallelPeripheral
{
    /** @pdOid d89d693a-81d1-11d5-ad8a-0008c752dcf8 */
    public boolean laser;
    /** @pdOid d89d693b-81d1-11d5-ad8a-0008c752dcf8 */
    public int printSpeed;
    /** @pdOid 5938ae5d-3a45-4a11-9017-20fbf8216601 */
    public boolean usbPort;
```

Note that the usbPort attribute is displayed in the synchronized source code.

4. Delete the code concerning the usbPort attribute.

```
printer.java ✕
package Peripheral_Package;

import java.util.*;

/** @pdOid d89d6951-81d1-11d5-ad8a-0008c752dcf8 */
public class printer extends parallelPeripheral
{
    /** @pdOid d89d693a-81d1-11d5-ad8a-0008c752dcf8 */
    public boolean laser;
    /** @pdOid d89d693b-81d1-11d5-ad8a-0008c752dcf8 */
    public int printSpeed;
    /** @pdOid 5938ae5d-3a45-4a11-9017-20fbf8216601 */
    public boolean usbPort;
```
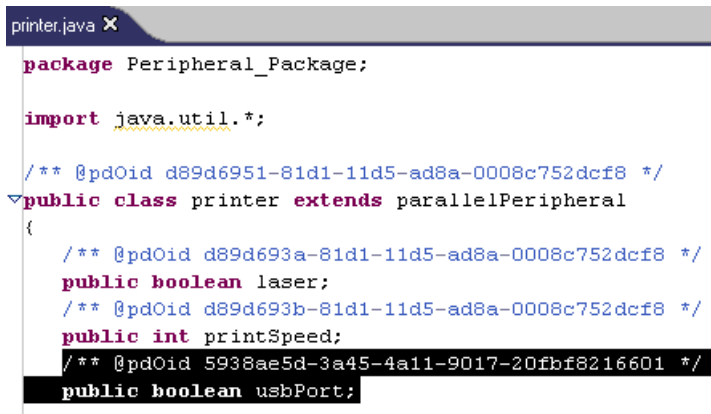
5. Select **File > Save** in the Eclipse menu bar.

The usbPort attribute disappears from the printer symbol because the printer class in the OOM has been synchronized with its source code.

Deactivating Synchronization

If you decide to deactivate synchronization, the model and its source code remain as they were after the last synchronization.

You can deactivate synchronization:

- In the Model Explorer, by right-clicking the model and selecting **Synchronize**, unselecting **Synchronized**, and clicking **Finish**.
- In the Navigator, by right-clicking the model and selecting **Properties**, unselecting **Synchronized** in the PowerDesigner page, and clicking **OK**.
- In the Navigator, right-clicking a synchronized package and selecting **Unsynchronize**.

Navigating Between an OOM and Its Java Source Code

Once you have synchronized an OOM with its Java source code, you can find its diagram symbols and open its property sheet from the Model Explorer, the Navigator, or source code files.

To locate an object symbol in a diagram:

- In the Navigator, right-click an object Java file and select **Find In Diagram**.
- In Outline view, select an object to automatically find in diagram.

  Note: To open the Outline view, select **Window > Show View > Outline**.

- Double-click the name of an object in a source code file in order to select it, then right-click the object name and select **Find In Diagram**.

To open an object property sheet:

- In the Navigator, right-click an object Java file and select **Model Object Properties**.
- Double-click the name of an object in a source code file in order to select it, then right-click the object name and select **Model Object Properties**.

CHAPTER 11 **The PowerDesigner Add-In for Visual Studio**

During your installation of PowerDesigner, you can choose to install a plugin to allow you to use PowerDesigner within your Visual Studio environment. The PowerDesigner add-in for Microsoft Visual Studio and Team Foundation is available for Visual Studio 2005, 2008, and 2010.

The Visual Studio development is highly configurable, with many different explorers and other windows that can be moved all over the screen and be docked and tabbed together. Consequently, the screen below shows only one possible configuration:
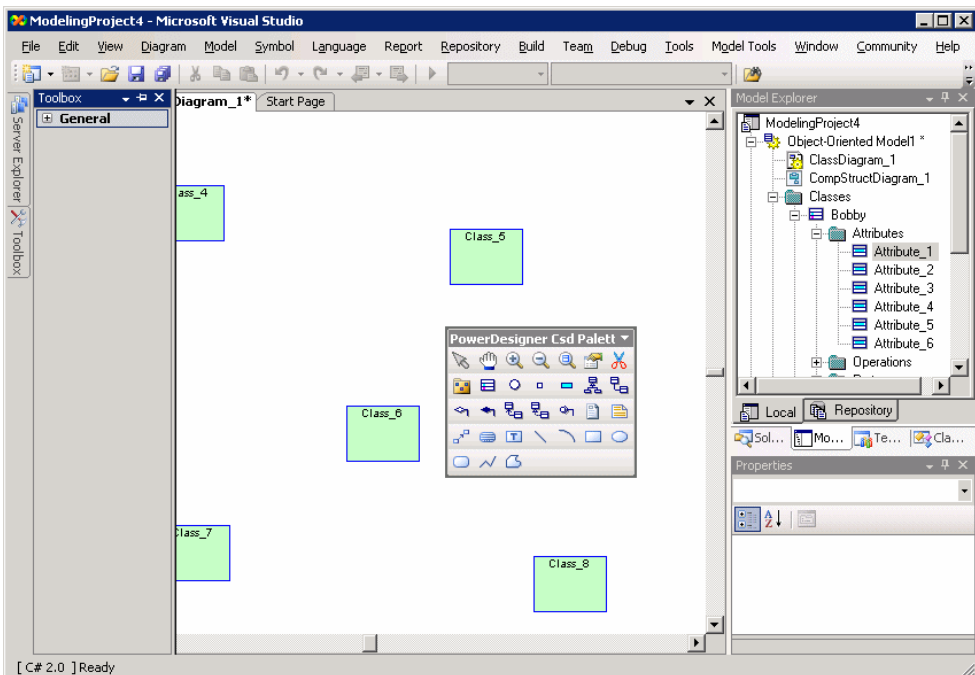
The main screen areas displayed here are as follows:

• The Toolbox, like the PowerDesigner toolbox, is contains tools to build your diagrams.
• The Document Window is equivalent to the PowerDesigner canvas, and is the space where you build your model diagrams.
• The Model Explorer is equivalent to the PowerDesigner Browser, and lists all the models in your project along with all their model objects displayed in a tree view.

- The Properties window is equivalent to a PowerDesigner property sheet, and lists the properties of the currently selected object. Standard PowerDesigner property sheets are also available.

The following PowerDesigner-specific menus are available in the Visual Studio menu bar:

- Diagram [View in the standard PowerDesigner interface] – tools for manipulating model diagrams
- Model – lists of model objects
- Symbol – tools for manipulating diagram symbols
- Language/Database etc. – depending on the type of model currently open in the document window, this menu will change to provide tools for manipulating the subject of your model, including code generation and reverse-engineering
- Repository – tools for working with the PowerDesigner model repository
- Model Tools [Tools in the standard PowerDesigner interface] – tools for manipulating the model

Other PowerDesigner menu functions are integrated into the standard Visual Studio menus.

Creating a Modeling Project in Visual Studio

You can create a modeling project in its own standalone solution, or add a modeling project to an existing solution.

Note: The creation of a PowerDesigner modeling project requires the presence of Microsoft .NET Framework v3.5.

1. [optional] Open an existing solution to which you want to add the modeling project.
2. Select **File > New > Project** to open the New Project window:
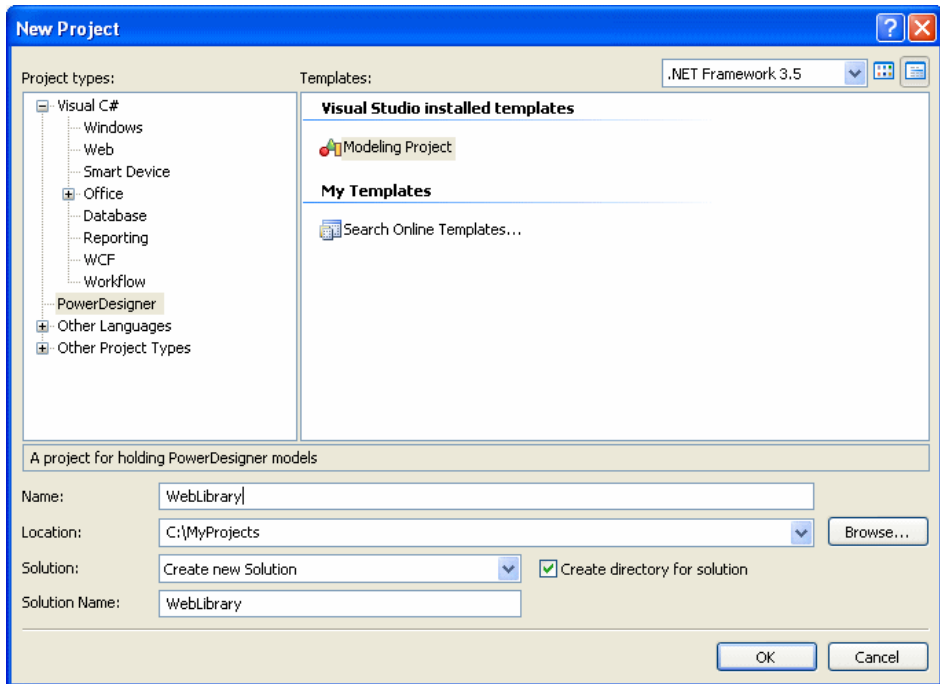
3. Select PowerDesigner in the Project types pane, and then Modeling Project in the Templates pane.

4. Enter a name for the project, and specify a location for its files.

5. Select one of the following options in the Solution field:

   • Add to Solution – to add the project to an existing solution

   • Create new Solution – to create a new solution.

6. Click OK to go to the PowerDesigner New Project window:
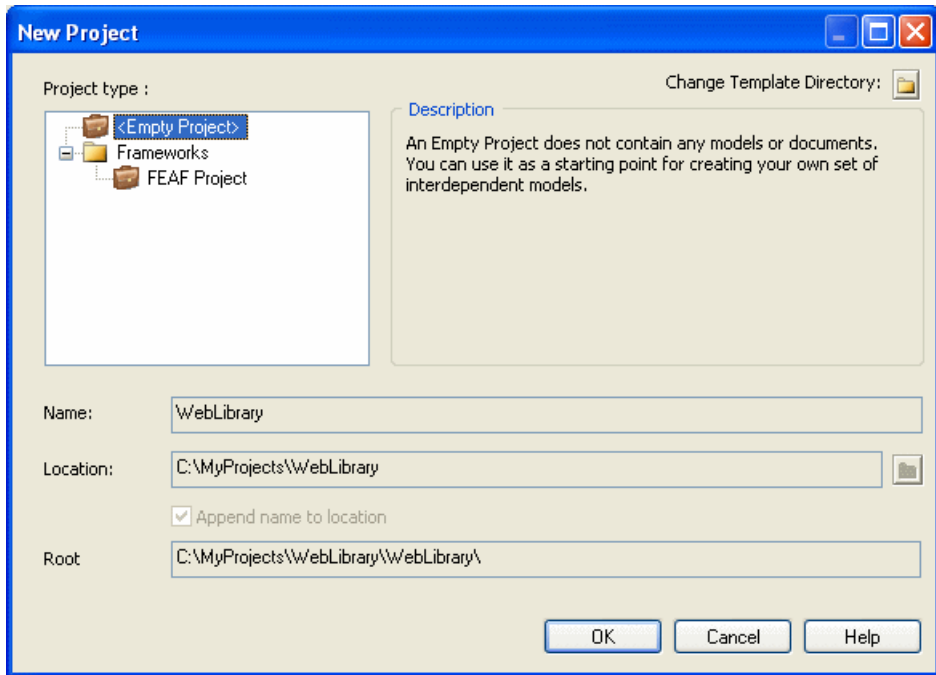
7. Select the type of PowerDesigner project to create. PowerDesigner project templates allow you to create projects that are already populated with models and/or that contain matrices that help you to follow various modeling frameworks, such as FEAF.

8. Confirm the name, location, and root, and click OK to create the project.

For detailed information about working with projects, see *Chapter 3, Projects and Frameworks* on page 53.

Creating a Model in Visual Studio

You can create any of the model types supported by PowerDesigner from within Visual Studio. The following procedure focuses on how to create an object-oriented model (OOM).

1. Select a project in the Solution Explorer.

2. Select **File > New > File** to open the New File window:
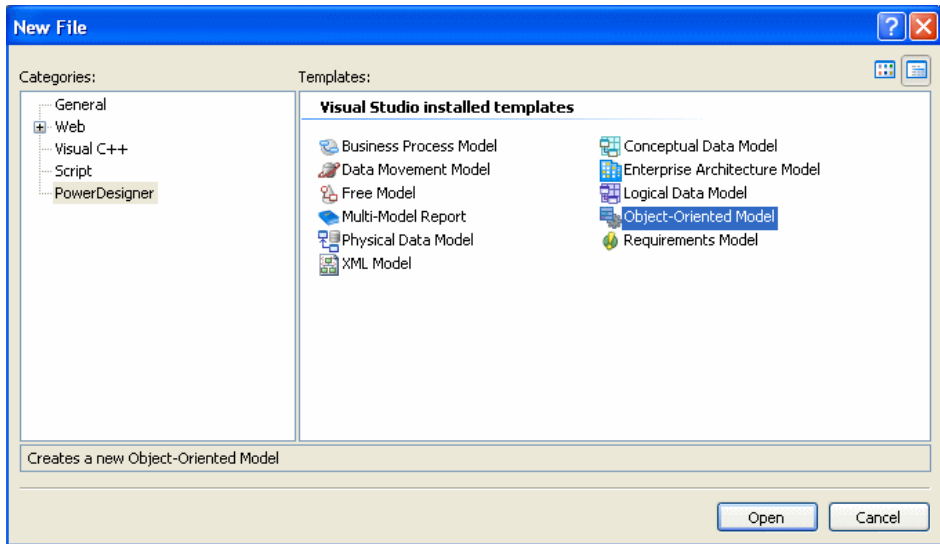
3. Select PowerDesigner in the Categories pane, and the type of model that you want to create in the Templates pane.

4. Click OK to open the New Model window:
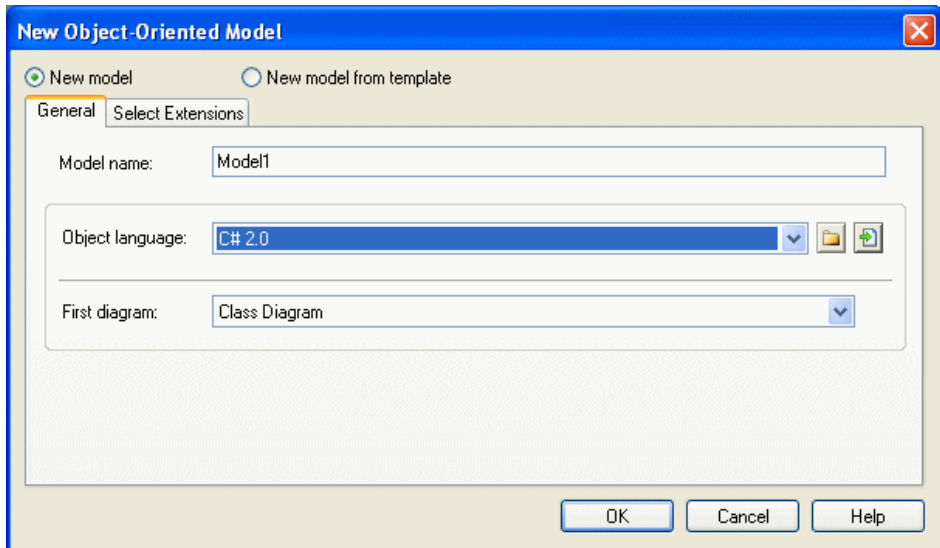
5. Select one of the following radio buttons:

- New model – Creates a new, standard, model.
- New model from template – Creates a model from a model template, which can contain pre-configured options, preferences, extensions, and objects. For more information, see *Model Templates* on page 13.

6. Enter a model name. The code of the model, which is used for script or code generation, is derived from this name according to the model naming conventions.

7. Select an Object language from the list.

   Object languages are defined in dedicated XML files (with a .XOL extension), which are provided as part of your PowerDesigner installation in the "\Resource Files\Object Languages" directory, and contain all the syntax and specifications for each target language.

   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.

8. Select the type of the first diagram in the First Diagram list. The first type of diagram selected remains in memory, and is the default for the next time when you create a new OOM.

   You can create as many diagrams as you need in the same OOM. They are sorted alphabetically in the Model Explorer, except the diagram specified here, which is always the first in the list.

9. [optional] Click the Select Extensions tab, and select one or more extensions to complement the selected object language (see *Attaching Extensions At Model Creation Time* on page 9).

10. Click OK to create the OOM.

Working with an RQM and Visual Studio Team System

A PowerDesigner Requirements Model (RQM) is used to create a hierarchy of project requirements that must be fulfilled during the development of the project.

You can export RQM requirements to one or more Visual Studio Team System projects and then create task work items to structure how the requirements will be fulfilled. Doing so, permits you to link the power and simplicity of a PowerDesigner hierarchical RQM to your use of the Visual Studio Team System.

The RQM displays the following toolbar when you are connected to a Team System server:

Icon	Description
	Import Requirements – imports requirement work items from a Team System project to an RQM. See *Importing Team Project Work Items to an RQM* on page 377.
	Export Requirements – exports requirements from an RQM as work items to a Team System project. See *Exporting RQM Requirements to a Team Project* on page 373.

Icon	Description
	Update Requirements – updates the content (for example, description, status) of requirements already exported to a Team System project. See *Updating Requirement Work Items Linked to an RQM* on page 379.

For detailed information about developing and using an RQM, see *Requirements Modeling*.

Exporting RQM Requirements to a Team Project

Once you have created a requirements model you can export your requirements to a Team project as work items. The exported requirements retain a memory of their place in the requirements hierarchy:
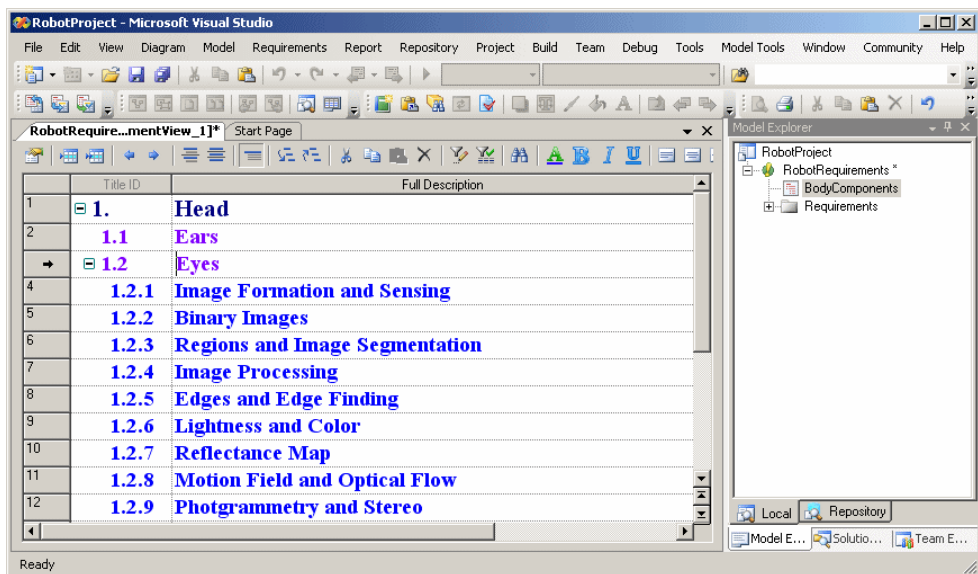
To begin an export, you must be connected to a Team System server and have created a Team project to receive your requirements.

You can export requirements from a single RQM to many Team projects by running the export wizard multiple times, but you cannot export the same individual requirement to more than one project.

1. Review the Team Explorer pane to ensure that you are connected to a Team System server, and that you have created an project to receive your requirements.
2. Select **Team > Requirements > Export** (or click the Export Requirements tool) to open the Export Requirements wizard.
3. The Team Project screen allow you to specify to which project you will export your requirements and, optionally, an Area and/or Iteration, if these have been defined for the project.
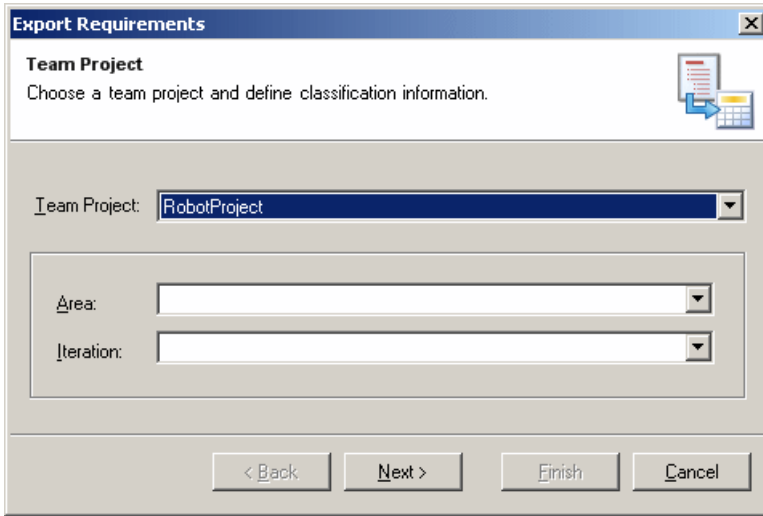
Note that, although you cannot select multiple projects to export to on this screen, you can relaunch the wizard as many times as necessary to export your requirements to a variety of projects.

4. Click Next to continue. The Requirements Selection screen allows you to specify which requirements you want to export. All requirements are shown except those that have already been exported to another project, or to another area or iteration of the present project.
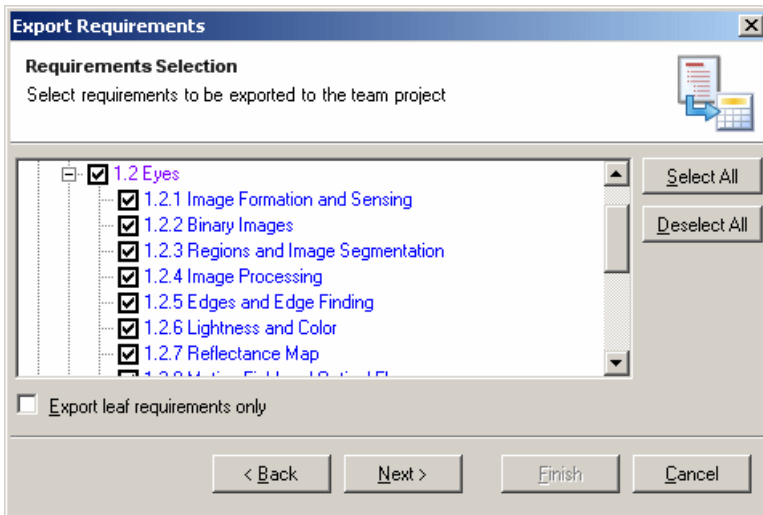
Specify a requirement to export by selecting its checkbox. If you select the Export leaf requirements only checkbox, then only those requirements without children will be exported.

5. Click Next to continue. The Work Item Type screen allows you to specify the type of work item to which your RQM requirements will be exported. You can select an existing type or create a new one.
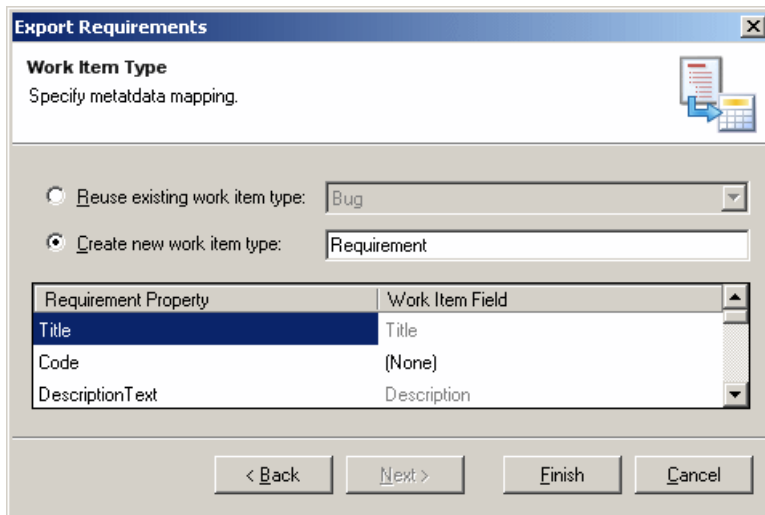
This screen also allows you to specify how the properties of the RQM requirements will be mapped to the fields of the designated work item type. Certain fields, such as the Title and Description are hard-coded and cannot be changed. You can choose a Work Item Field for the other properties by clicking on the entry, and then selecting the appropriate field (or Add, in the case of a new work item type) from the list.

If you select to create a new work item type, you must create a new field for each of the Requirement Properties you want to export by clicking in the Work Item Field column and selecting Add from the menu, and specifying a Name for the field.

Note that any mappings set at this stage cannot be changed for subsequent exports, although you can specify additional property-to-field mappings.

6. Click Finish to begin the export. When it is complete, you can view the work items that you have created by accessing one of the standard queries in the Team Explorer, or writing your own.

Linking Work Items to Requirements

Once your requirements have been exported into Team project work items, you will need to create associated work items (generally of type Task), in order specify how they will be fulfilled. You may link multiple Tasks and other work items to your requirements and, in general, treat them like any other work items.

1. Display your requirement work items by using the All Work Items team query, or by writing your own. In the window below, I have written a query called RequirementsWorkItems to display only the requirements:
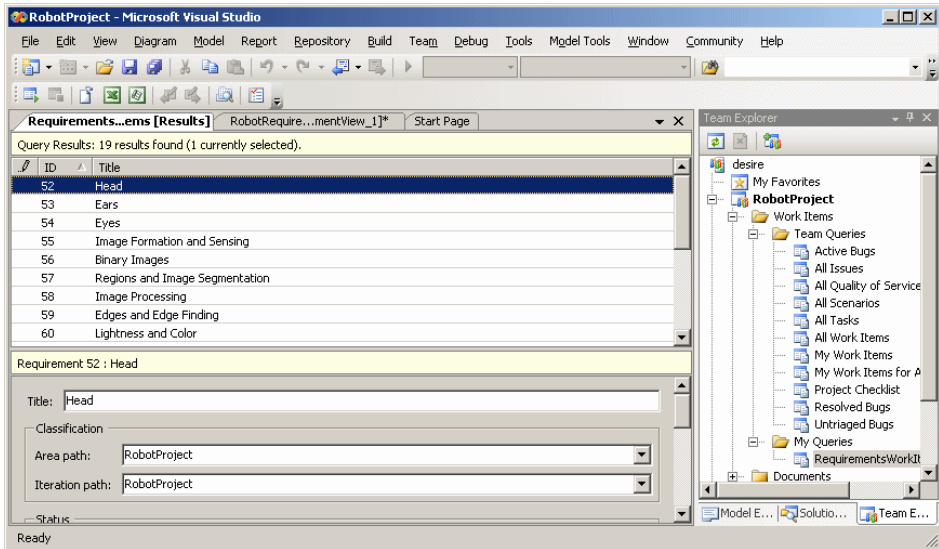
2. Right-click a requirement in the list, and select Add Related Work **Item > Task** (or another work item type) to create the new work item:
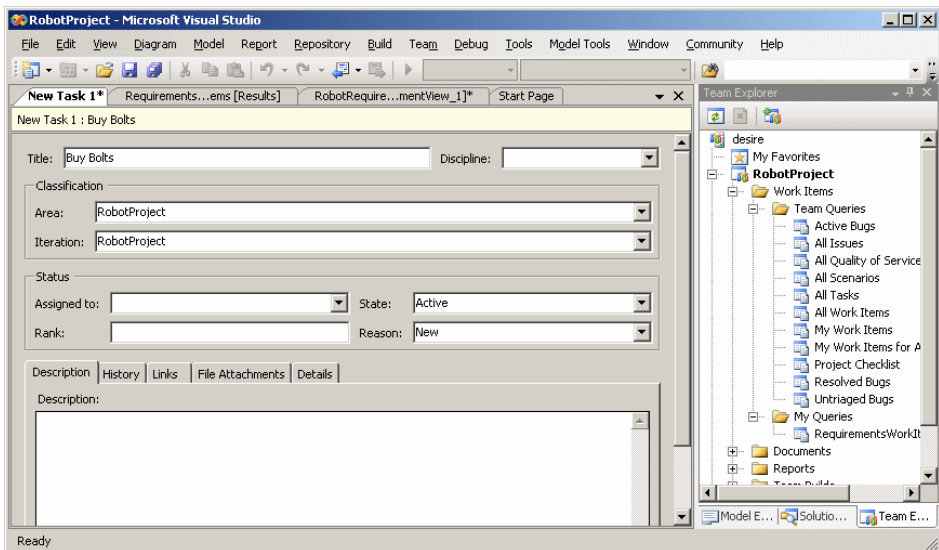
3. Complete whichever fields you deem necessary in order to specify the nature of the work item. If you click on the Links sub-tab, you will see that it is linked to the original requirement.
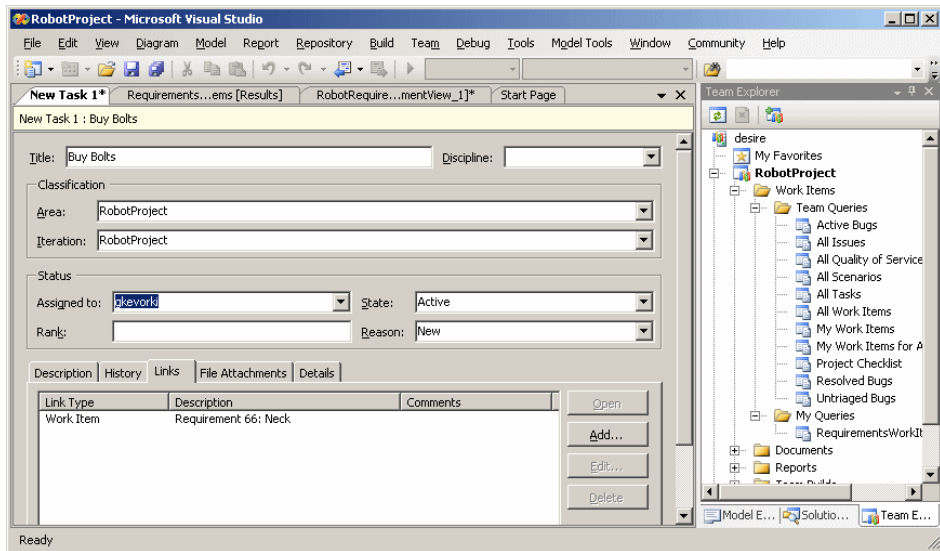
For more information about the Team system, see your Microsoft documentation.

Importing Team Project Work Items to an RQM

We recommend that you begin by developing an RQM, export your requirements to a Team project, and then add tasks to fulfill the requirements. However, there may be a situation where you develop requirements in the Team project, and then wish to import them to a new or existing RQM.

1. Review the Team Explorer pane to ensure that you are connected to a Team System server, and that the project from which you want to import is available.

2. Open a new or existing RQM in Visual Studio, and select **Team > Requirements > Import** (or click the Import Requirements tool) to open the Import Work Items Wizard.

3. The Team Project screen allow you to specify from which project you will import your work items and, optionally, an Area and/or Iteration, if these have been defined for the project.
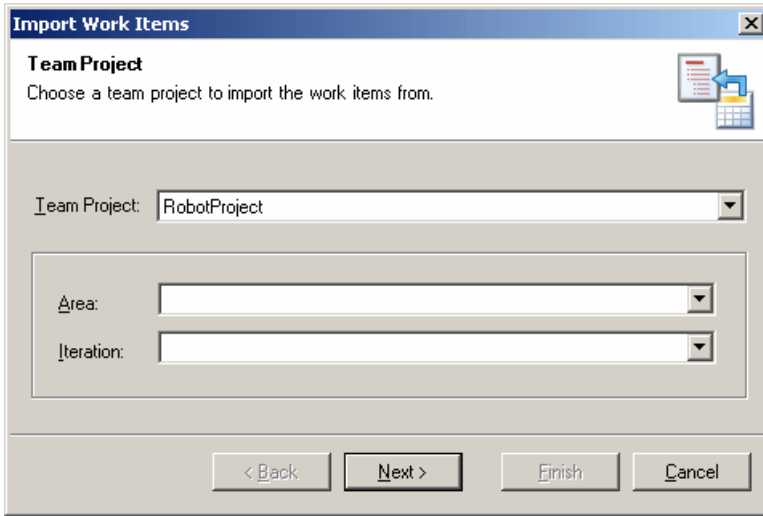
4. Click Next to continue. The Work Item Type screen allows you to specify the type of work item from which your RQM requirements will be imported. Note that the type may not be changeable if you have previously specified a mapping during an export for this project.
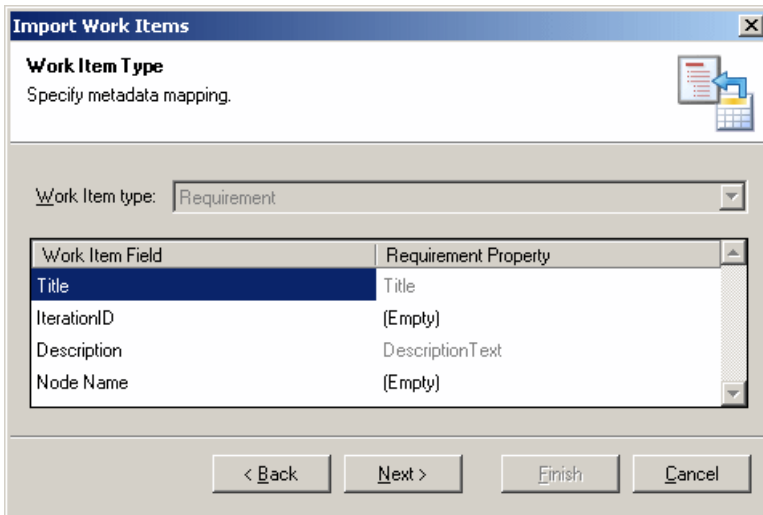
This screen also allows you to specify how the fields of the designated work item type will be mapped to the properties of the RQM requirements. Note that certain fields, such as the Title and Description, cannot be changed. You can choose a Requirement Property for the other fields by clicking on the entry, and then selecting the appropriate property from the list.

Note that any mappings set at this stage cannot be changed for subsequent imports, although you can specify additional field-to-property mappings.

5. Click Next to continue. The Work Items screen allows you to specify which requirements you want to import. Work Items appear greyed and cannot be selected if they are already linked to requirements in the current or another requirements model.
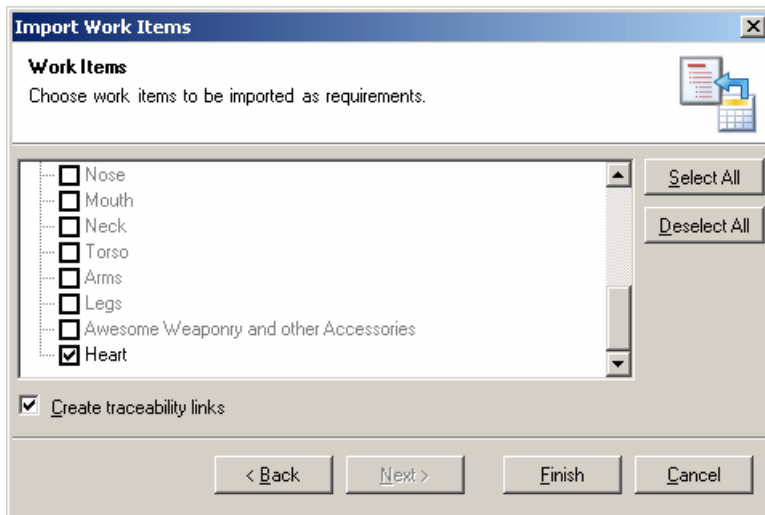
Select the Create traceability links checkbox if you want to retain links between the requirement work items in your Team project and the RQM. Note that traceability links can only be created for work items that are not already linked with another RQM.

6. Click Finish to begin the import. Any hierarchy defined between the work items will also be preserved.

Updating Requirement Work Items Linked to an RQM

Once you have exported your requirements from an RQM to a Team System project, it is possible that you will continue to edit the details of these items (by adding or editing a detailed description, changing the status, or any of the other available properties). To cascade these edits to your requirements work items held with a Team System project, simply select **Team > Requirements > Update** (or click the Update Work Items tool).

The Update tool refreshes all the requirement work items exported to any number of Team System projects. Note that it does not export any new requirements (this requires that you use the Export tool) and that it does not update any changes made in requirement work items to their counterparts in the RQM.

Navigating Between RQMs and Team Projects

PowerDesigner provides various methods for navigating between an RQM and a Team System project

When working in the Model Explorer, you can, at any time, double-click the file named after your Team server and project (auto-generated in the Files folder):
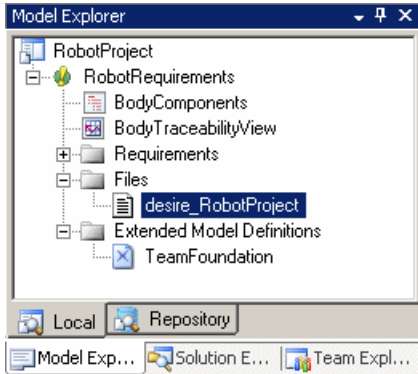
Doing so will navigate to the linked Team Project in the Team Explorer:
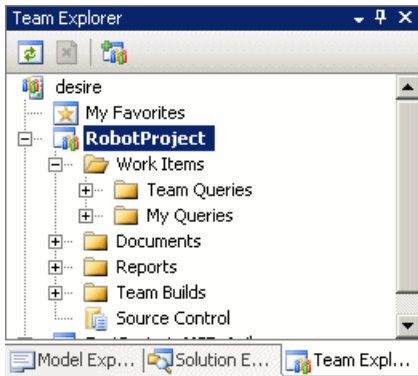
To navigate to a particular requirement work item (linked via a traceability link to an RQM requirement), right-click the requirement in an RQM document view, and select Team **Foundation > Work Item Properties** from the contextual menu:
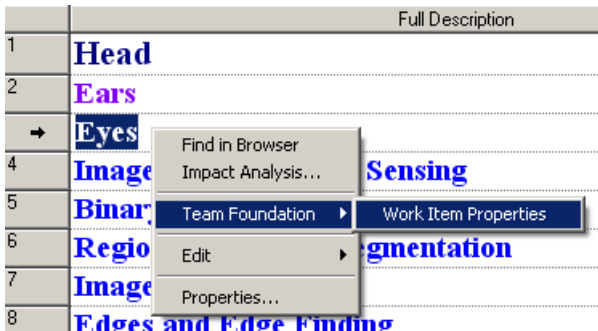
Creating a Traceability Links Matrix to Track Requirement Work Items

A Traceability Links Matrix View can list all the traceability links between your requirements and work items.

1. Select **Requirements > Create a Traceability Matrix View**.

2. In the Matrix View toolbar, click the Change Traceability Matrix Type tool, and then select the External files radio button in the dialog box.

3. Click OK. The Matrix View will display a column for each of the Team System projects to which you have exported requirements, with a check mark in the cell of each requirement that is linked to a work item by a traceability link:
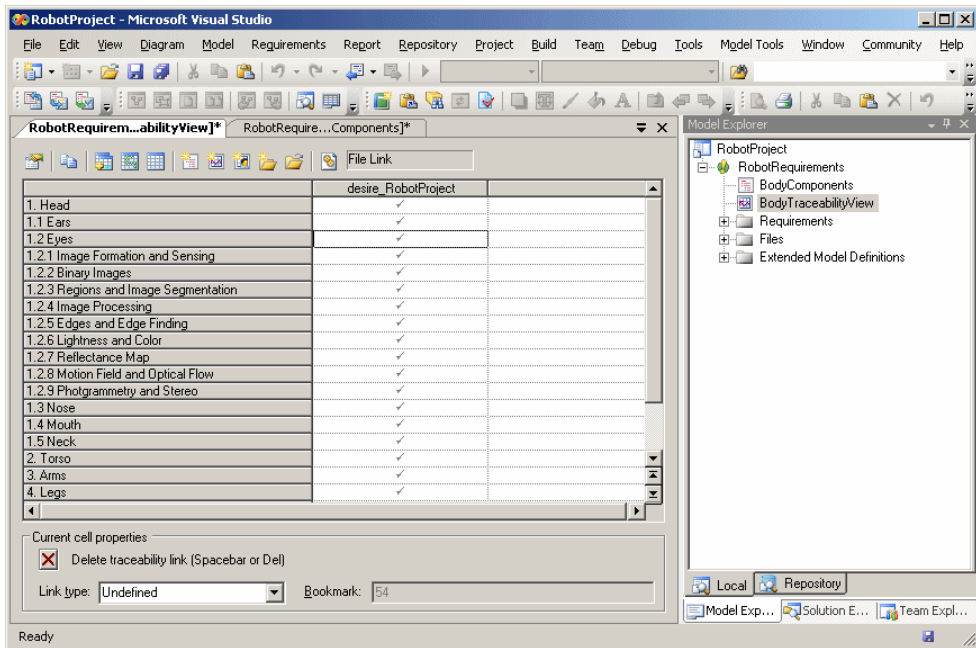
For more information about the Traceability Matrix View, see *Requirements Modeling*.

You can also view the links to work items by opening the property sheet of a requirement and clicking the Traceability Links tab. Select the link and then click the Properties tool to go to the associated work item:

Model-Code Synchronization

By using code synchronization with PowerDesigner genlets, you can realize aspects of the model-driven development process to improve productivity through the automation of code generation. The PowerDesigner add-in for Visual Studio provides the capability for instantaneous bidirectional updates of model objects and actual code, including automating the transformations and merging of generated files.

You configure synchronization initially through the Synchronization Wizard, and then continue with development, with changes automatically cascading to the code. Code synchronization is available for all PowerDesigner diagrams and models supporting generic generation, and thus could be used with, for example, a state diagram, or a BPM (Business Process Model).

1. Create an empty Visual C# or other code project, and then add your model project to the solution.

2. Select the model in the Solution Explorer, and then click the **Synchronize** tool in the toolbar (or select **Synchronize** from the contextual menu) to open the Synchronize dialog at the Code Generation page:

This page allows you to specify file types that will be synchronized for forward generation (i.e. changes made to model objects will be cascaded down immediately to the code files). File types are organized by generation targets (C# 2.0 and WSDL for .NET in the screenshot above), and defined in the resource file or extension for the target.

3. Select the file types that you want to synchronize for forward generation.

4. [optional] Select a file type and click the **Details** button to open the Generation Details dialog (see *Custom Generation Post-Processing with Genlets* on page 385)

5. Click **Next** to move to the Reverse-engineering page of the Synchronization dialog:
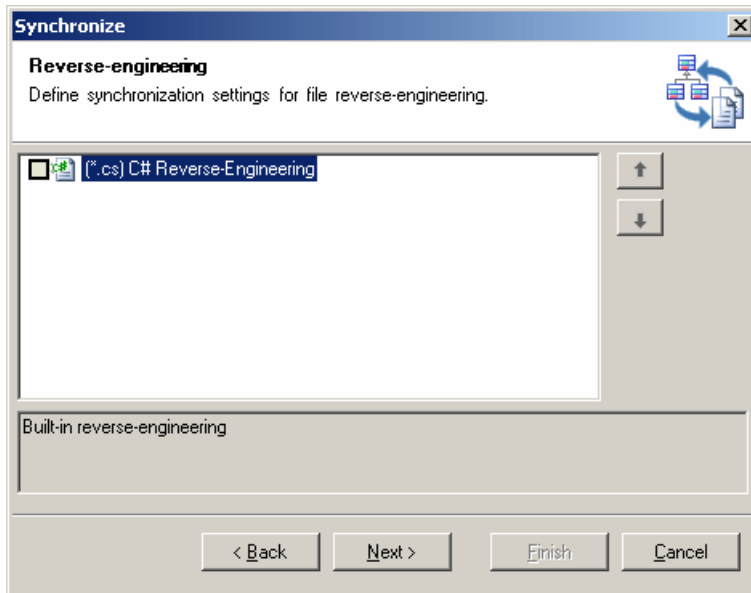
This page, which is available only for C# and Visual Basic projects only) allows you to specify file types that will be synchronized for reverse-engineering (i.e. changes made in the code files will be cascaded up immediately to the model objects).

6. Select the file types that you want to synchronize for reverse-engineering, and then click **Next** to move to the Files page of the Synchronization dialog:
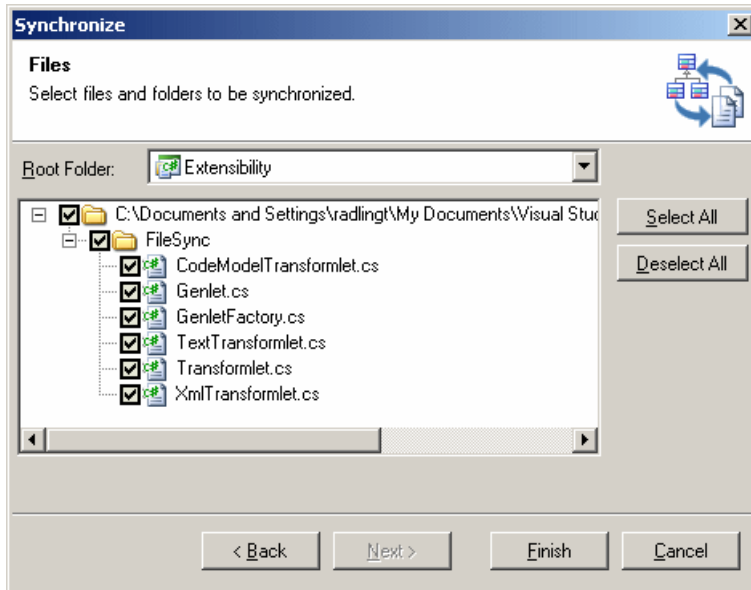
This page allows you to specify the specific files that will be synchronized, and where they will be generated. By default:

- If the model is located directly in a solution folder (outside of a project) or in a modeling project, the files are generated at the root of the solution, and project files are added as projects in the Solution Explorer.
- If the model belongs to any other project type, the files are generated by default at the root of the project, or in another folder within the project specified in the Root Folder field.

If there is the possibility of a conflict because both model object and file already exist, the file will not be selected to generate by default, and its checkbox will carry a small red cross. To force synchronization and specify who will have priority, click the checkbox and select **Overwrite file** or **Overwrite model** from the contextual menu.

7. Select the files that you want to synchronize and click **Finish**.

The wizard generates and marks as synchronized all the specified files, and a record of the synchronization state of all the files in the solution is stored in a file with the name of the project or solution and the extension `.pdsync`.

Note: Once you have synchronized your model and files, you can disable or enable synchronization on a file-by-file basis by selecting a file in the Solution Explorer. If the file is currently synchronized, the **Synchronize** tool is depressed in the Solution Explorer toolbar. Click the tool to change the state of the file. If the file was not previously synchronized, the Merge Model window will open to enable you to review the changes that synchronization will make.

Custom Generation Post-Processing with Genlets

Model-code synchronization provides a powerful method for visualizing your code and providing a strong link between your source files and model objects. PowerDesigner genlets enable you to additionally automate post-generation transformations and handle complex merging of changes to your files.

Genlets are small .NET classes that can be automatically invoked by the synchronization process to perform transformations on the generated files.

There are two kinds of genlets:

- Transformlets – perform a transformation on a generated file. They allow you to define reusable cross-target policies to be applied to generated code. For example, you could use transformlets to add a standard header and footer to every file or to ensure that every static field name begins with an underscore
  You can chain multiple transformlets together to perform various steps in a transformation.
- Mergelets – are of use whenever a given file is being regenerated and the target content may change independently of the model. This may be the case with an XML document in which some sections are generated from a model and others are modified by another system or where standard code skeletons are generated and custom code is added inside the code editor.

In the case of source code, mergelets can be used to maximize readability by preserving order and formatting over multiple generation cycles, by comparing and merging elements at the code model level addresses this issue.

Although especially useful for languages that do not offer partial class support (facilitating the separation of generated and user code), code model mergelets may nonetheless be of use with the VB and C# programming languages. Only one mergelet may be used for each generation.

PowerDesigner provides C# templates for three forms of transformlets and mergelets:

- Text genlets– transform any text file
- Xml genlets - manipulate XML data via the System.Xml classes
- Code-model genlets - use VisualStudio parsers and the CodeModel API to traverse and transform source files

Creating a Genlet

You create genlets in a C# or other .NET projects.

1. Right-click your project in the Solution Explorer and select **Add > New Item**.
2. In the Add New Item dialog, expand the PowerDesigner category and select the File Synclets item.
3. Choose a genlet type, specify a name, and then click OK to create the item. The editor opens to display the genlet template:

```csharp
using System;
using System.Text;

using Sybase.PowerDesigner.VisualStudio.Extensibility.FileSync;

namespace Genlets
{
    /// <summary>
    ///
    /// </summary>
    [Genlet(DisplayName = "TextTransformlet1", Description = "")]
    class TextTransformlet1 : TextTransformlet
    {
        /// <summary>Implements the text transform</summary>
        /// <param name="text">input text</param>
        /// <returns>transformed text</returns>
        protected override string TextTransform(string text)
        {
            // TODO: implement your text transform here
            throw new NotImplementedException();
        }
    }
}
```

4. Enter your transformation code and save the file:

```
using System;
using System.Text;

using Sybase.PowerDesigner.VisualStudio.Extensibility.FileSync;

namespace MyGenlets
{
    /// <summary>
    ///
    /// </summary>
    [Genlet(DisplayName = "Header Transformlet",
        Description = "Adds a default header ")]
    class TextTransformlet1 : TextTransformlet
    {
        /// <summary>Implements the text transform</summary>
        /// <param name="text">input text</param>
        /// <returns>transformed text</returns>
        protected override string TextTransform(string text)
        {
            string header = "// My Header\r\n";
            return header + text;
        }
    }
}
```

5. Right-click your model in the Solution Explorer and select Synchronize, to open the Synchronization wizard (see *Code Synchronization* on page 382).

6. On the Code Generation page, select a file type (for example Source), and click the Details button to open the Generation Details dialog:
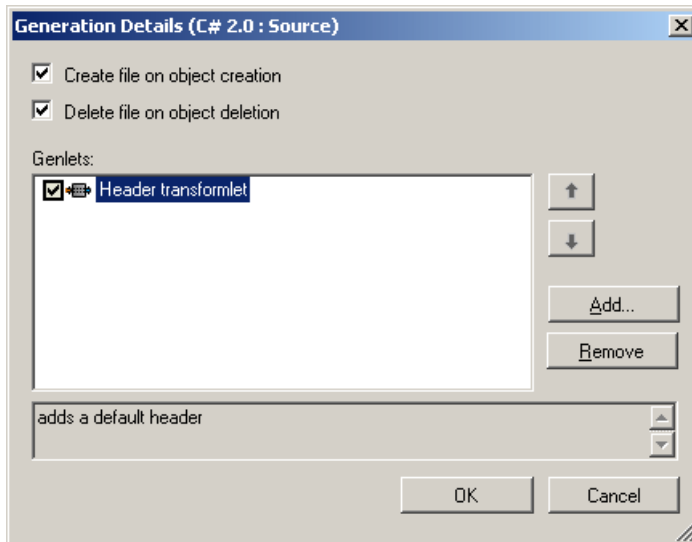
This dialog allows you to specify whether to synchronize object and file creation and deletion, and also to add any genlets.

7. Click the Add button to open the Add Genlet dialog:
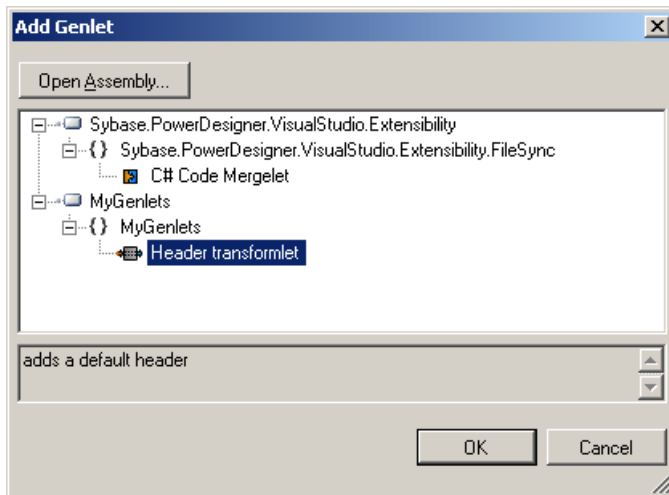
Select the genlet to add (or click the Open Assembly button to choose additional assemblies from which to select your genlets) and click OK to return to the Generation Details dialog, and then click OK again to return to the Synchronize dialog.

8. [optional] Select a transformlet and use the Up and Down arrows to change its position in the chain of genlets. Use the Remove button to remove the selected genlet.

9. Once you have completed the Synchronization wizard and generated your files, the transformation specified will have been applied to all the files of the specified type.

Note that changes to genlets are not automatically updated in synchronized files. When you change a genlet, you must then click the Refresh tool in order to force a synchronization.

CHAPTER 12   **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.

Note: Administrators can prepare customizations to the PowerDesigner interface and push them to users (see *Chapter 19, Customizing the PowerDesigner Interface* on page 539).

User Profiles

User profiles help you to standardize the look and feel of your models and to support standards. Profiles allow you to group options and preferences together for sharing and reuse across your organization. Various profiles are provided with PowerDesigner, and you can create your own.

The 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* on page 211).
*   Model Options – [model only] to control naming conventions, 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* on page 395).
*   Check Model options – [model only] to control which checks are applied, and which are errors, warnings, etc (see *Checking Models* on page 148).
*   Other options – [registry only] such as the layout of toolbars and windows (see *Organizing Views* on page 409), favorite property tabs (see *Customizing a Property Sheet* on page 99), default columns for object lists (see *Customizing Object List Columns and Filtering Lists* on page 120), etc.

Note: An administrator can prepare customized preference profiles, which are made available to users on their first connection to the repository (see *Controlling Defaults for Options and Preferences* on page 552).

User profiles can contain any number of default preferences and options, the application of which is cumulative. Thus, if you:

1. Apply a user profile that directs that table symbols in the physical data model are drawn by default in red, and that view symbols are drawn in blue (via display preferences)
2. Apply a second user profile that directs that table symbols are drawn in green.

The result will be that table symbols are drawn in green, and view symbols are drawn in blue. You can, of course, override these default preferences and options locally in the model.

User profiles are PowerDesigner resource files, and so can be opened and edited in the resource editor, but in general you will create and edit them by extracting preferences defined in a model, or from your registry.

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.

Note: To return to a default user profile prepared for you by an administrator(see *Controlling Defaults for Options and Preferences* on page 552), select the `General Preferences` entry in the Personal Profiles category. To return to your default UI profile, select the `UI Preferences` entry.

1. Select **Tools > Apply User Profile** to open the Apply User Profile dialog:
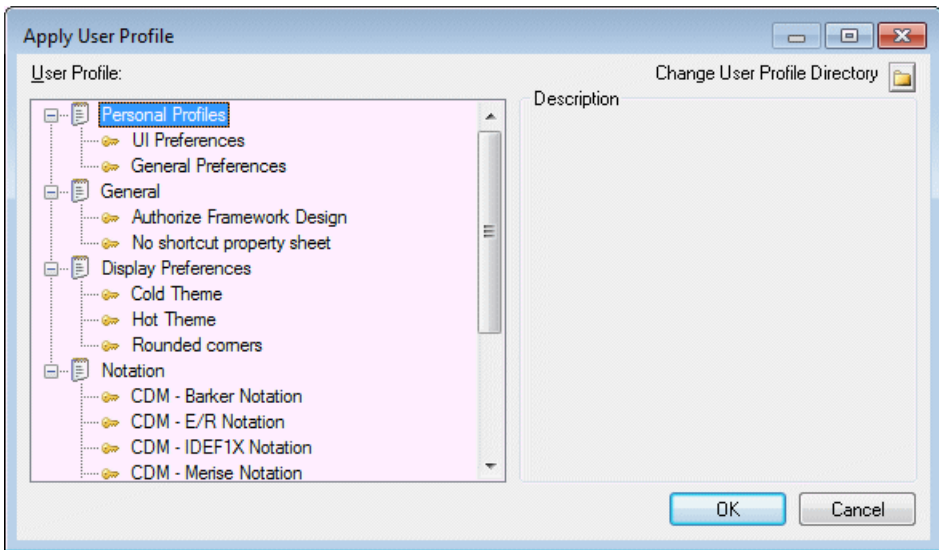
2. [optional] Click the **Change User Profile Directory** button, and browse to a folder where you have saved user profiles. By default they are saved in the *install_dir/* `Resource Files/Profiles` directory.

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.

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.

1. Select **Tools > Resources > User Profiles** to open the List of User Profiles:
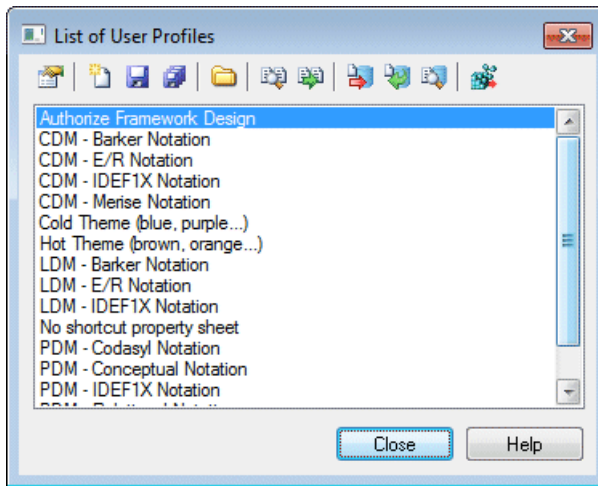
The following tools are available above the list:

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.
	Merge – Updates the selected profile by merging preferences from another profile, a model, or the registry (see *Updating a User Profile* on page 392).

Tool	Description
	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* on page 565.
	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* on page 390).

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* on page 393).

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.

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.

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.
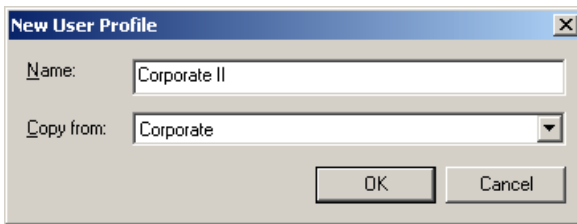
2. In the From field, select a source from which to copy from. You can choose between:

- 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* on page 393). 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 *Chapter 7, Comparing and Merging Models* on page 275.

4. When you are satisfied, click OK to update the profile and return to the list.

Copying Preferences from a Model

When you choose to create or update a user profile by copying preferences from a model, you open the Model for User Profile dialog:

For more information, see *Creating a user profile* on page 391 or *Updating a user profile* on page 392.
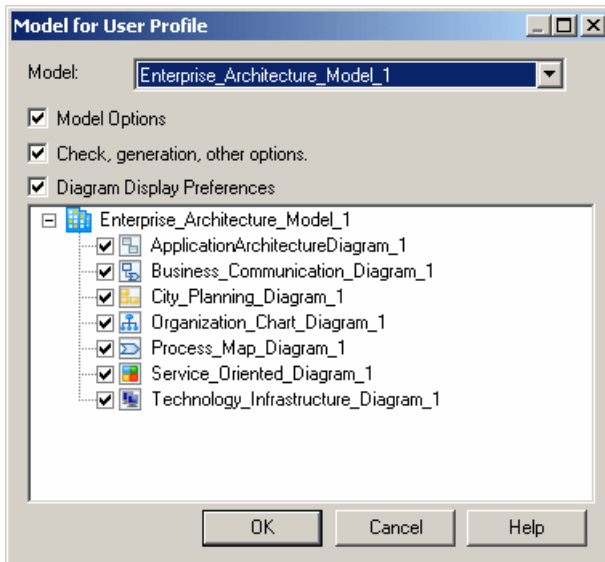
You can to specify the following options and preferences to capture:

Option	Description
Model	Specifies the open model from which you will capture preferences.

Option	Description
Model Options	Captures the model options from the specified model.
Check, generation, and other options	Captures the check model, generation, and other options from the specified model.
Diagram Display Preferences	Captures the display preferences from the diagrams that you select in the tree below. Note that, to avoid conflicts, you can only select one instance of each diagram type. Thus, if your chosen model is an OOM, which contains two class diagrams, selecting one class diagram will deselect the other.

Reviewing and Editing User Profiles in the Resource Editor

You can open any user profile in the Resource Editor to review and edit it. 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:
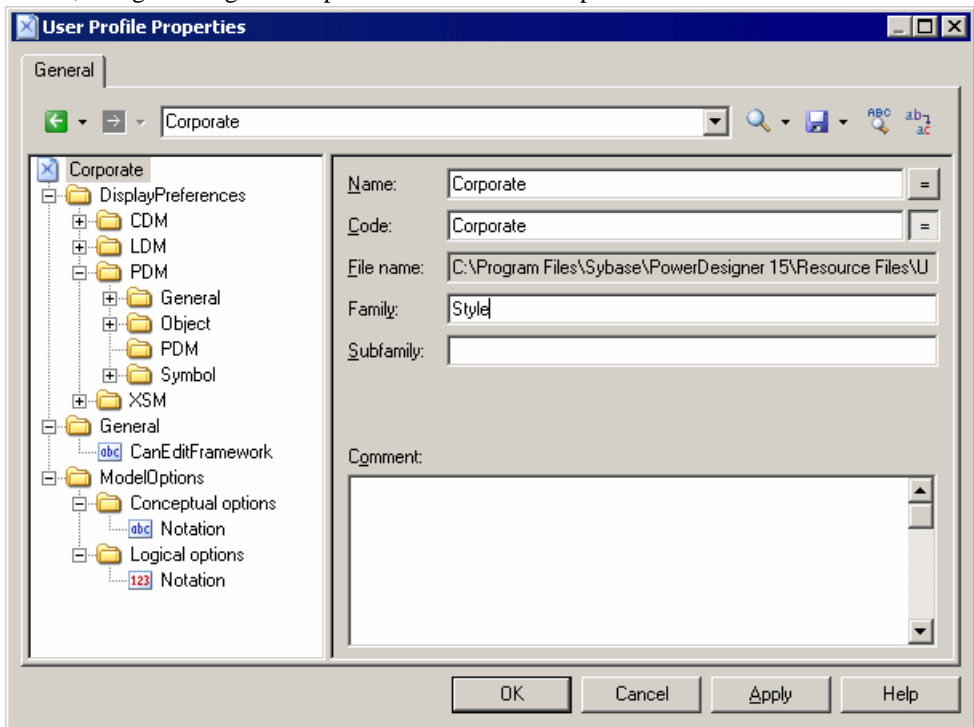
The profile root node has the following properties:

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.

Property	Description
Family	Specifies the family in which the profile will appear in the Apply User Profile window (see *Applying a user profile* on page 390) and if it is added to the Power-Designer installer image. If no value is entered, the profile is added to the General family.
Subfamily	Specifies a subfamily within the profile family.

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.

To delete an item, simply right-click it and select Remove. You can always make additions or adjustments to the values of user profiles by extracting them from models or other profiles.

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*.

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 Power-Designer 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 postpone reviewing the recovered models until the next time you open PowerDesigner.
	Note: This option is not supported for the PowerDesigner plugins for Eclipse and Visual Studio.
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* on page 134.
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, the Show Welcome page option is deselected.

Option	Description
Browser	**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 is created in the new location. • **Create replica (Alt+Shift)** – A replica of the object is created in the new location. The modifier keys given after the option are valid no matter what the default behavior.
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	**Natural Sort** - 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.

Dialog Box General Options

To set general options for dialog boxes, select **Tools > General Options**, and select the Dialog category in the left hand 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 cancel a change by using the Undo feature. If this option is disabled, you must click Apply or OK to commit your changes.

Option	Description
Name to Code mirroring	Automatically keeps the code of an object synchronized with its name via the application of any naming conventions (see *Naming Conventions* on page 107). To disable Name to Code mirroring on an individual object, click to deselect the equal sign button to the right of the code field.
	This option is independent of (but required by) the **Enable name/code conversions** model option, which permits more complex transformations on codes via controls the application of conversion scripts (see *Name and Code Conversion Scripts* on page 110).

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* on page 453.

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.

Specifying Text Editors

When you are using PowerDesigner, you will need to edit such things as descriptions and annotations for objects, or generated scripts like SQL in the PDM or Java in the OOM

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.

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.

Defining Environment Variables

PowerDesigner defines a certain number of environment variables for use used in generation commands and with GTL. You can edit these variables and add your own.

1. Select **Tools > General Options**, and click the **Variables** category in the left-hand pane.

Variable	Description	Default
CMD	Windows command interpreter	command.com or cmd.exe
HOME	Variable defining the default home directory	—

Variable	Description	Default
J2EEVERIF	Batch program for verifying if the deployment jar for an EJB is correct	verifier.bat
JAR	Command for archiving java files	jar.exe
JAVA	Command for running JAVA programs	java.exe
JAVAC	Command for compiling JAVA source files	javac.exe
JAVADOC	Command for defining JAVA doc comments	javadoc.exe

2. Click in the row of an existing variable in order to edit its value, or click the **Add a Row** tool to create a new variable and enter its value.

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

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%`.

Defining Named Paths

Named paths allow you to handle links between models and resource files in environments where users may install PowerDesigner in different locations or have customized their install. PowerDesigner initializes a number of predefined named paths, each of which is preceded by an underscore to point to various file locations that depend on the installation directory.

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.

Note: An administrator may push named paths to users in a general preferences profile (see *Controlling Defaults for Options and Preferences* on page 552).

The following paths are initialized by default:

Name	Path
_DBMS	Folder where DBMS definition files are stored.
_EXAMPLES	Folder where demo examples are stored.

Name	Path
_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.
_MODELCATE-GORY	Folder where model category files are stored (see *Customizing the New Model Dialog* on page 406).
_OBJLANG	Folder where object language definition files are stored.
_PRCSLANG	Folder where process language definition files are stored.
_RTPLANG	Folder where report language definition files are stored.
_RULESET	Folder where impact and lineage rule set files are stored (see *Chapter 17, Impact and Lineage Analysis* on page 499).
_SHARED	Folder where the glossary and library model files are stored (see *The Glossary* on page 40 and *The Library* on page 43).
_SMARTWIN-FORM	Folder where smart winform files are stored.
_WINFORM	Folder where winform files are stored.
_XEM	Folder where extension files are stored.
_XMLLANG	Folder where XML language definition files are stored.

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

Defining Default Interface Fonts

You can modify the default font proposed for:

- User interface (lists)
- Code editor (SQL preview)
- RTF editor (description, annotation)

1. Select **Tools > General Options**, and then click the Fonts category.

2. Specify the appropriate font format options.
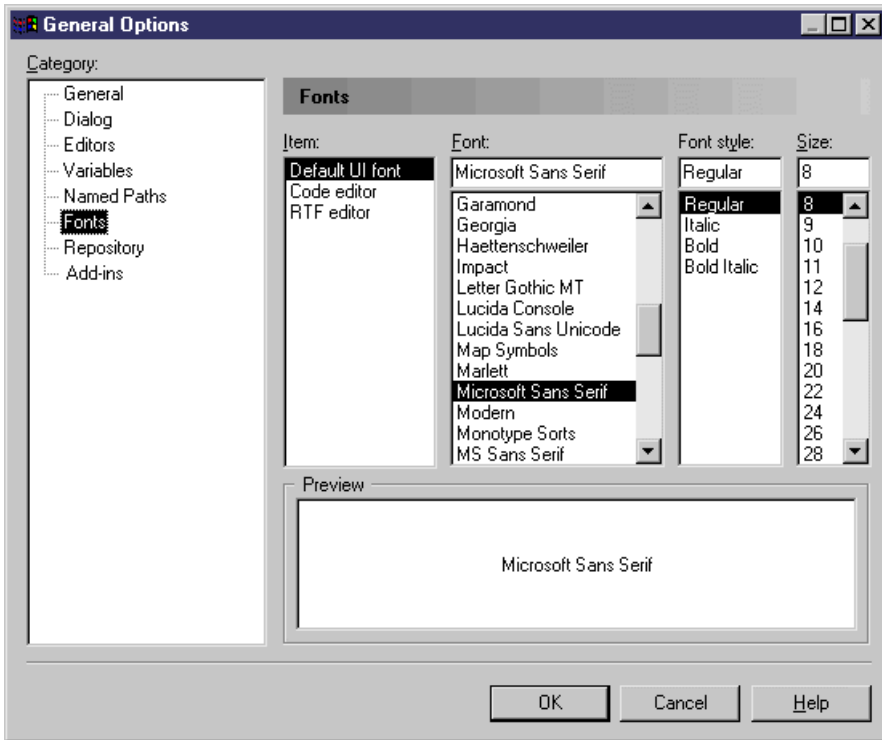
3. Click OK.

Managing Add-Ins

An add-in is a module that adds a specific feature or service to PowerDesigner. PowerDesigner provides several add-ins by default.

Add-in	Description
Microsoft Analysis Services	Imports multidimensional data from MS SQL Server into a PDM. Microsoft Analysis Services must be installed on the current machine.

Add-in	Description
Microsoft Word Import Export	Imports and exports MS Word documents to and from requirements models.
.Net Reverse Engineering	Reverse engineers VB .Net and C# source files.
Spell Checker	Uses the MS Word spell checker for PowerDesigner object names, codes, comments, descriptions, and annotations
XML Validation	Checks that an XML document conforms to the current schema in the XML model. This add-in requires MSXML 4.0

You enable an add-in by selecting its checkbox.
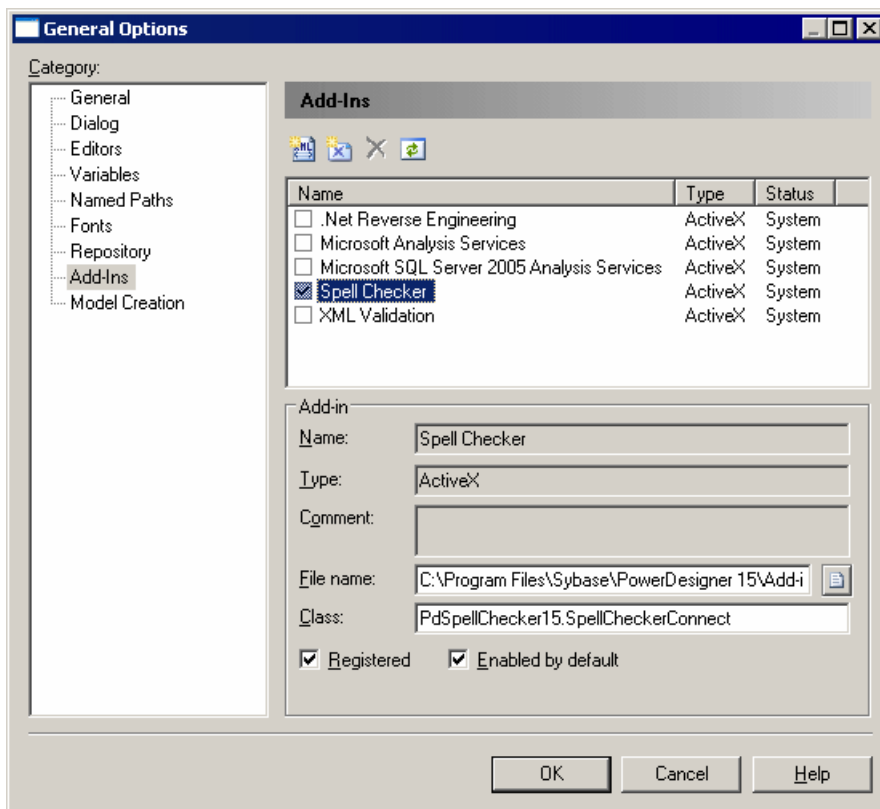
Add-ins installed with PowerDesigner are called *system* add-ins, and are declared in a Local Machine Registry key. If you modify system add-in properties, you can click the Reset Values For System Add-In tool to recover add-in parameters as they are registered in the Local Machine key.

You can create your own add-ins. User-defined add-ins are called *user* add-ins, they are declared in a Current User Registry key.

.Net Reverse Engineering
In order to use the .Net binary reverse, you need to register the application reverseapp.exe.

.Net Framework 1.1 must be installed. You must use the regasm.exe program located in the Windows directory under the Microsoft.NET\Framework folder. The command line is the following:

```
regasm /codebase reverseapp.exe
```

The .net Assemblies command is displayed in the **File > Reverse engineer** menu. The executable can also be used separately from PowerDesigner.

```
ReverseApp -c|-v [-l ] [-r ] [-g]
```

-l: followed by library file name, this option can be repeated several times.

-r: recursively forces to reverse engineer parameter type and return type.

-g: does not display reverse engineering dialog box and directly proceeds with reverse engineering.

-c: reverse engineers C# model.

-v: reverse engineers VB.Net model.

ReverseApp will retrieve the namespaces, classes, and other objects defined in these assemblies and create a corresponding Object-Oriented Model.

Declaring Your Own Add-In
You can create your own XML or ActiveX add-ins by declaring and enabling them in the **Add-Ins** page of the General Options dialog.

1. Select **Tools > General Options**, and then click the **Add-Ins** category.
2. Click the **Add New XML Add-In** or **Add New ActiveX Add-In** tool.

   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. An XML add-in uses a declarative program with a language linked to an .EXE file or a VB script.
3. Enter the name of your add-in in the **Name** box and click the **Select File** tool to the right of the File name field to browse to the .xml, .dll, or .exe file.
4. [ActiveX only] Enter the name of the ActiveX implementation class in the **Class** field.
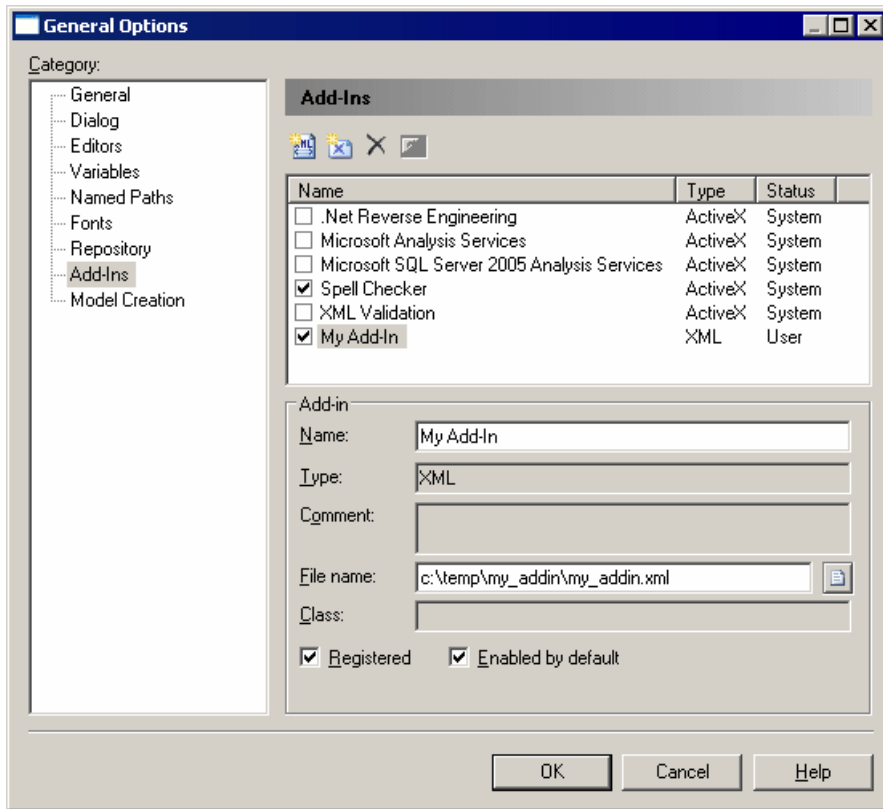
5. Click **OK**.

For more information about the add-in creation procedure, see *Customizing and Extending PowerDesigner > Scripting PowerDesigner > Customizing PowerDesigner Menus Using Add-Ins*.

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.

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.

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.

Customizing the New Model Dialog

You can control the format of the New Model dialog by hiding some of the methods of model creation.

To set general options for model creation, select **Tools > General Options**, and click the **Model Creation** category in the left hand 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* on page 539.
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* on page 13.

For information about using the New Model dialog, see *Creating a Model* on page 6.

Repository General Options

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

Connection

The following options are available in the Connection group box:

Option	Description
Auto-connect	Automatically opens the last connected repository when you start a Power-Designer session.
Default bytes per char	Specifies the default number of bytes per character used by the repository database. This value is used to initialize the Bytes per char parameter in Direct Repository Definition dialog (see *Setting up PowerDesigner for Direct Access* on page 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 *Checking the State of a Document* on page 309).

Check Out

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

Option	Description
Merge document	Specifies that the Merge document check box in the Check Out window is selected by default.

| Option	Description
Add to workspace	Specifies that the Add to workspace check box in the Check Out window is selected by default.
Check out dependencies	Specifies that the Check out dependencies check box in the Check Out dialog box is selected by default.
Automatic resources update	Enables the automatic checking out and updating of shared resource files (see *Sharing Resource Files via the Library* on page 565).

Check in

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

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

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.
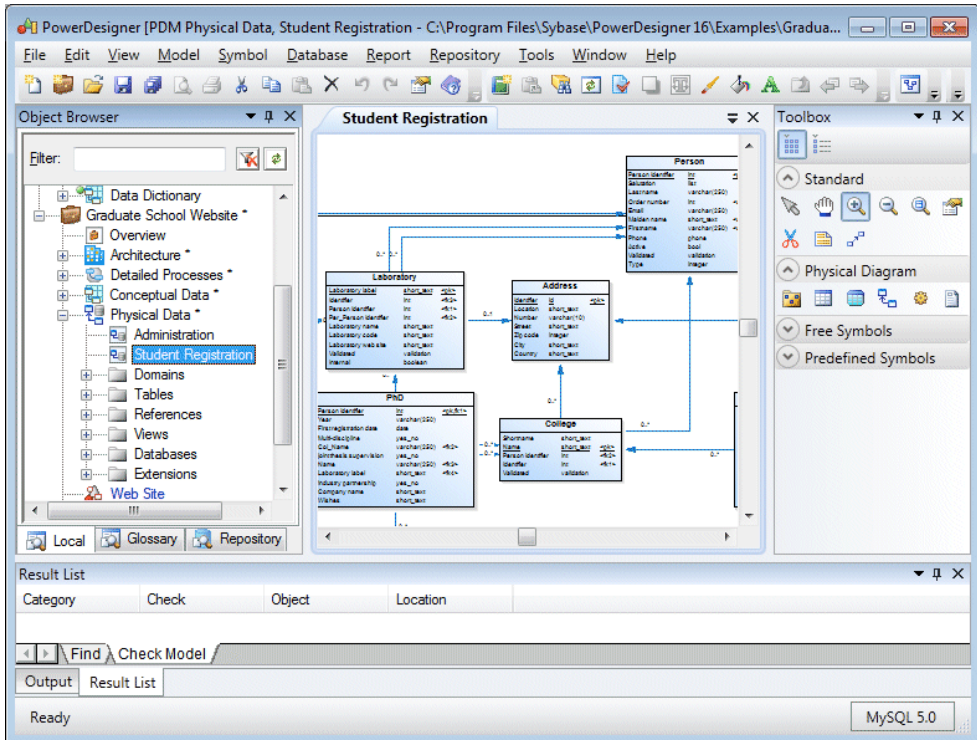
To move a view, click on its title bar or upper or lower tab and drag and drop it onto the docking control that appears as you hover over a view. You can perform the following kinds of movement in this way:

• 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.

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***.

Note: The Browser is a single view with up to three sub-views (Local, Glossary, and Repository). These sub-views cannot be moved independently.

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* on page 411).

Note: The Toolbox (see *Creating Objects from the Toolbox* on page 95) 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.

Customizing Menus, Toolbars, and Toolboxes

To customize PowerDesigner menus, toolbars, and toolboxes, select **Tools > Customize Menus and Tools**.

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* on page 551).

1. Select **Tools > Customize Menus and Tools**. The dialog opens to the **Menus** tab, initialized to your current view (here, the PDM Physical Diagram):
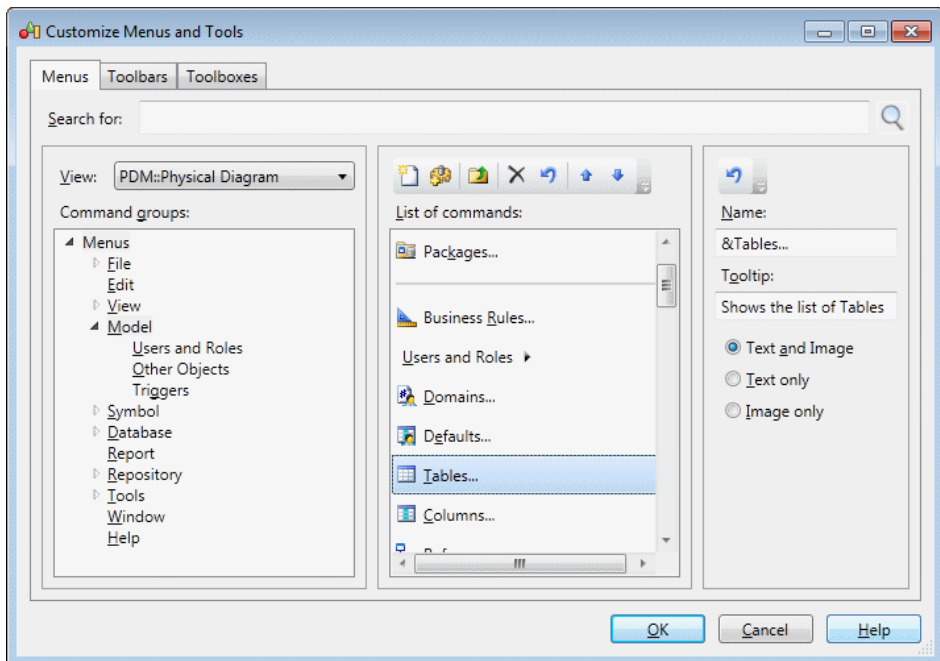
Note: Search for any command available in the present context by entering it in the **Search for** field and clicking **Find**.

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.

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* on page 413). 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.
	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**.

To ol	Description
🔼	**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* on page 410) or **Toolboxes** (see *Creating Objects from the Toolbox* on page 95) tab to customize these UI elements.

7. Click **OK** to save your changes and apply them to your environment.

PowerDesigner Resource Files

The 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) – 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* on page 415).

- *Report templates* (.rtp) - specify the structure of a report. Editable within the Report Template Editor (see *Report Templates* on page 272).
- *Report language files* (.xrl) – translate the headings and other standard text in a report (see *Customizing and Extending PowerDesigner > Translating Reports with Report Language Resource Files*).
- *Impact and lineage analysis rule sets* (.rul) - specify the rules defined for generating impact and lineage analyses (see *Working with Analysis Rule Sets* on page 512).
- *Object permission profiles* (.ppf) - customize the PowerDesigner interface to hide models, objects, and properties (see *Controlling the Availability of Models, Objects, and Properties* on page 546).
- *User profiles* (.upf) - store preferences for model options, general options, display preferences, etc (see *User Profiles* on page 389).
- *Model category sets* (.mcc) - customize the New Model dialog to guide model creation (see *Customizing the New Model Dialog* on page 539).
- *Conversion tables* (.csv) - define conversions between the name and code of an object (see *Creating a Conversion Table* on page 113).

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* on page 415).
	Save - Saves the selected resource file.
	Save All - Saves all the resource files in the list.
	Path - Specifies the directories that contain the resource files to populate this list and other places in the PowerDesigner interface where resources of this type can be selected. By default, only the directory containing the resource files delivered as part of the PowerDesigner installation is specified, but you can add as many additional directories as necessary. The root of the library belonging to your most recent repository connection is implicitly included at the head of the list (see *Sharing Resource Files via the Library* on page 565) **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 sought for resource is used.
	Compare - Selects two resource files for comparison.
	Merge - Selects two resource files for merging.

Tool	Description
	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* on page 565.
	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.

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.

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.

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.

Warning! Since each resource file has a unique id, you should only copy resource files within PowerDesigner, and not in Windows Explorer.

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.

Note: You can create an extension file directly in your model from the List of Extensions. For more information, see *Creating an Extension File* on page 418.

Extension Files

Extensions files (`*.xem`) provide means for customizing and extending PowerDesigner metaclasses, parameters and generation. You can use extensions to define additional

---

properties for existing object types or entirely new kinds of objects, to modify the PowerDesigner interface (reorganizing and adding property sheet tabs, Toolbox tools and menu items), and to define additional generation targets and options.

PowerDesigner provides a number of predefined extension files and you can also create your own. 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* - a UML extension mechanism, which is used for extending a metamodel for a particular target. Profiles are used in PowerDesigner for adding additional metadata to objects and creating new kinds of links between them, sub-dividing object types (via stereotypes and criteria), customizing symbols, menus, and forms, and modifying generation output. For example:
  - The Java 5.0 object language resource file - extends the Component metaclass via several levels of criteria to model various forms of EJBs.
  - The BPEL4WS 1.1 process language resource file - extends the Event metaclass through stereotypes to model Compensation, Fault, and Timer events.
  - The MSSQLSRV2005 DBMS resource file - uses stereotyped extended objects in order to model aggregates, assemblies, and other SQL Server-specific objects.

You can extend the metamodel in the following ways:

- Add or sub-classify new kinds of objects:
  - Metaclasses – drawn from the metamodel as a basis for extension.
  - Stereotypes [for metaclasses and stereotypes only] – to sub-classify objects.
  - Criteria – to evaluate conditions to sub-classify objects.
  - Extended objects, sub-objects, and links – to create new kinds of objects.
- Provide new ways of viewing connections between objects:
  - Dependency matrices – to show connections between two types of objects.
  - Extended collections and compositions – to enable manual linking between objects.
  - Calculated collections – to automate linking between objects.
- Add new properties to objects and display them:
  - Extended attributes – to provide extra metadata.
  - Forms – to display custom property tabs or dialog boxes.
  - Custom symbols – to help you visually distinguish objects.
- Add constraints and validation rules to objects:
  - Custom checks – to provide data testing.
  - Event handlers – to invoke methods when triggered by an event.
- Execute commands on objects:

- Methods – to be invoked by other profile extensions such as menus and form buttons (written in VBScript).
- Menus [for metaclasses and stereotypes only] – to customize PowerDesigner menus.
- Generate objects in new ways:
  - Templates and generated files – to customize generation.
  - Transformations and Transformation profiles – to automate changes to objects at generation or on demand.
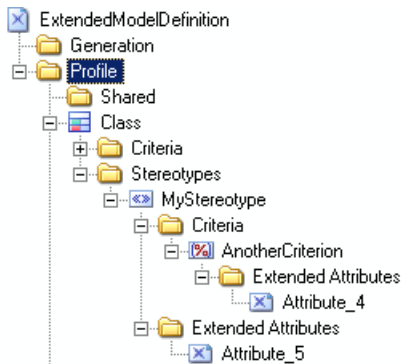
You can review and edit the profile in a resource file by opening it in the Resource Editor and expanding the top-level Profile category. You can add extensions to a metaclass (a type of object, such as Class in an OOM or Table in a PDM), or to a stereotype or criterion, which has previously been defined on a metaclass:
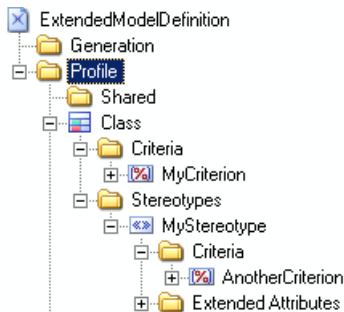
In the example above:

- Class is a metaclass. Metaclasses are drawn from the PowerDesigner metamodel, and always appear at the top level, directly beneath the Profile category
- MyCriterion is a criterion that refines the Class metaclass. Those classes that meet the criterion can be presented and processed differently from other classes.
- MyStereotype is a stereotype that refines the Class metaclass. Those classes that bear the MyStereotype stereotype can be presented and processed differently from other classes.
- AnotherCriterion is a criterion that refines further those classes that bear the MyStereotype stereotype. Classes bearing the stereotype AND meeting the criterion can be presented and processed differently from those that merely bear the stereotype.

Extensions are inherited, so that any extensions made to a metaclass are available to its stereotyped children, and those that are subject to criteria.

```
☒ ExtendedModelDefinition
  └─📁 Generation
  ⊟─📁 Profile
        ├─📁 Shared
        ⊟─🖽 Class
              ⊞─📁 Criteria
              ⊟─📁 Stereotypes
                    ⊟─«»» MyStereotype
                          ⊟─📁 Criteria
                                ⊟─🔀 AnotherCriterion
                                      ⊟─📁 Extended Attributes
                                            └─☒ Attribute_4
                          ⊟─📁 Extended Attributes
                                └─☒ Attribute_5
```

Thus, in the example above, classes that bear the MyStereotype stereotype have available the Attribute_5 extended attribute, and those that bear this stereotype AND meet AnotherCriterion have Attribute_4 and Attribute_5 available.

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.

Creating an Extension File

You can create an extension file from the list of extension files or directly embedded in your model.

Note: For information about creating an extension file from the list of extension files, see *Creating and Copying Resource Files* on page 415.

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* on page 420.

Attaching Extensions to a Model

Extensions can be in independent *.xem files that are attached to models or can be embedded in model files. Independent extension files can be referenced by multiple models, and any

---

changes made to such a file are shared by all models that attach it. Changes made to extensions embedded in a model file affect only that model.

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.

You can attach an extension file (.xem) to your model at the creation of the model by clicking the **Select Extensions** button on the New Model dialog. You can subsequently attach an extension file to your model at any time from the List of Extensions.

1. Select **Model > Extensions** to open the List of Extensions.
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.
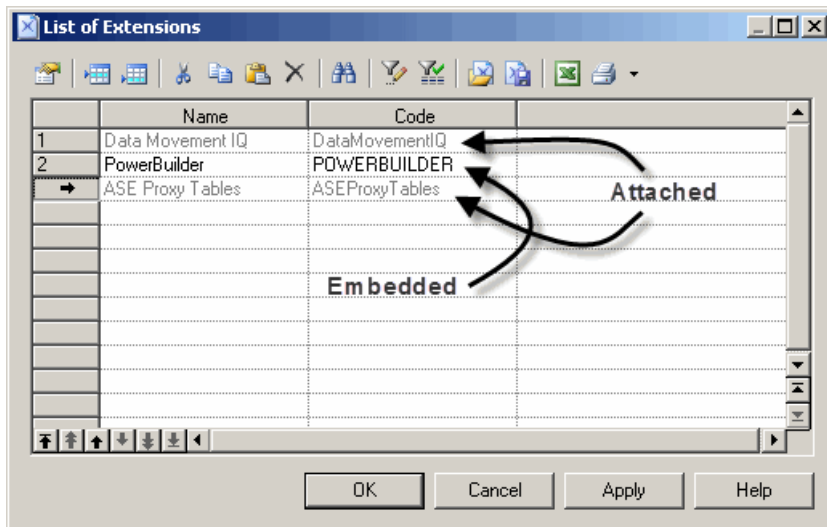
Extension files listed in grey are attached to the model, while those in black are embedded in the model.

> **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.

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.

1. Select **Model > Extensions** to open the List of Extensions.
2. Select an extension in the list.
3. Click the **Export an Extension** tool.
4. Type a name and select a directory for the extension file.
5. Click Save.

   The extension is saved in a library directory where it can be shared with other models.

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.
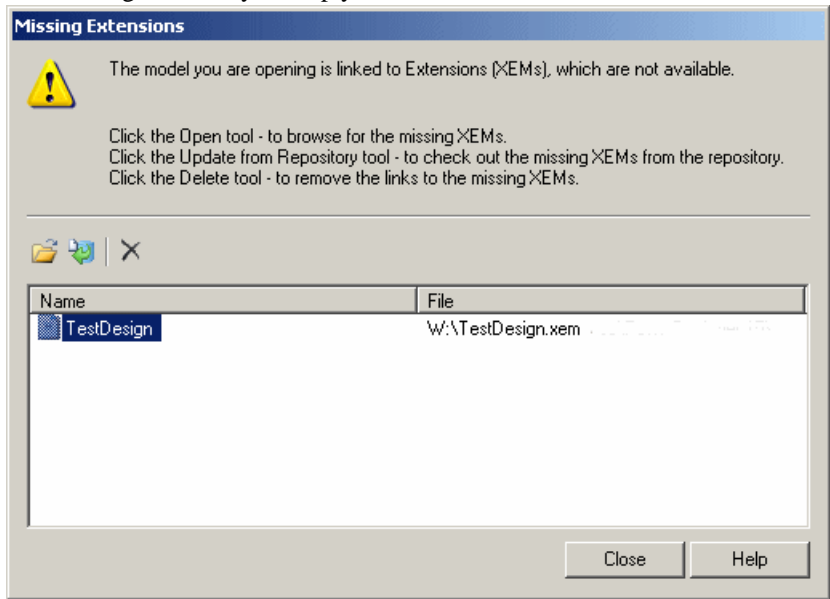
The following tools are available in this dialog:

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.

Shortcut Keys

PowerDesigner provides many shortcut keys to accelerate your workflow.

Toolbox Shortcuts

Shortcut	Action
Double-click Pointer tool / Ctrl+A	Select all symbols in diagram.
Double-click Delete tool	Delete all 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 and Shift+DEL	Delete object (with and without confirmation).
Alt+ENTER	Open object property sheet.

Shortcut	Action
Ctrl+K	Paste as shortcut.
Ctrl+H	[on selected link] Force link to horizontal.
Ctrl+L	[on selected link] Force link to vertical.
Ctrl+Page Up and Ctrl+Page Down	Bring symbol to front and send symbol to back.
Ctrl+Home and Ctrl+End	Align selection on vertical and 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+Resize	Resize object while keeping same center point.
Ctrl+Shift+Resize	Resize object while keeping same center point and proportions.
F2 or Ctrl+Right-click on Name	Rename object.

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.

Shortcut	Action
F9	Previous view.
F10	Display used pages.
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+Double-click / Ctrl++ (plus)	[on package or other composite symbol] Open subdiagram.
Ctrl+Tab	Cycle through property sheet tabs.
Ctrl+Page Up / Ctrl+Page Down	Cycle through property sheet tab subtabs.
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.
Ctrl+F6 / Ctrl+Shift+F6	Cycle between property sheets.
Ctrl+U	Go to parent diagram.
Shift+F2	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.

PART II

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.

CHAPTER 13   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.
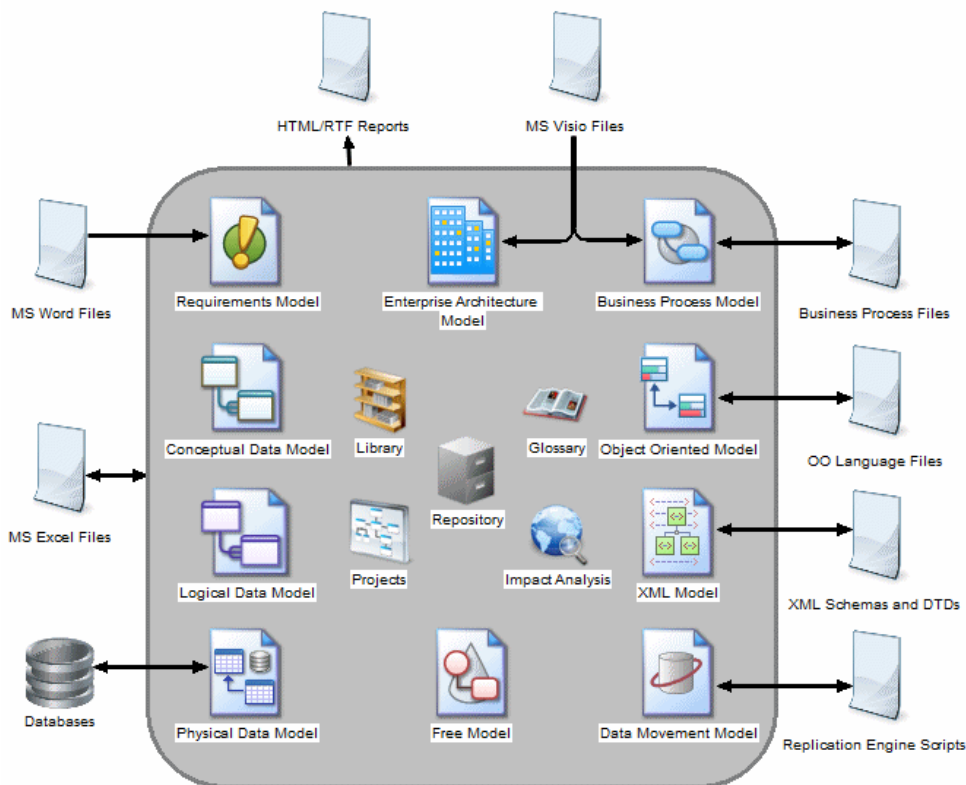
A single data item such as a product name can be implicated in multiple systems, which are modeled in a number of interdependent models. It may also be represented at various levels of abstraction in conceptual, logical, and physical data models, or in different contexts such as object-oriented, XML, 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.

Note: When working with multiple connected models, we recommend that you group your models together into a project (see *Chapter 3, Projects and Frameworks* on page 53).

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* on page 433).
    * Object Generation - allows you to define the generation of objects from one model to another with subsequent synchronization on demand (see *Generating Model Objects* on page 444).
* *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* on page 453) 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:
    * RQM Traceability Links - link requirements with design objects that are intended to satisfy them (see).
    * EAM Object Export/Import - link enterprise architecture objects with the design objects that will implement them (see).
    * BPM Data Export/Import - link data objects in the BPM with objects in other models (see).
    * Extended Attributes and Collections - allow you to extend the PowerDesigner metamodel to define new kinds of links between objects (see *Extending Objects* on page 155).
    * Object Mappings - specify connections between certain predefined types of objects (see *Chapter 16, Object Mappings* on page 469).
    * Related Diagrams - specify connections between objects and other diagrams in which they are not present (see *Specifying Diagrams as Related Diagrams* on page 178).
    * Extended Dependencies - specify connections between any objects without restrictions (see *Creating Extended Dependencies* on page 429).
* *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* on page 459).

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* on page 55).
- 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* on page 186).
- 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* on page 446).
- 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* on page 476).
- Multi-Model Report - provides a configurable report in HTML or RTF format documenting your models and the links between them (see *Creating a Multi-Model Report* on page 243).
- Impact and Lineage Analysis - provides a configurable analysis of the objects that impact and/or are impacted by the selected objects (see *Chapter 17, Impact and Lineage Analysis* on page 499).
- Object Lists - can include shortcuts to objects of the selected type if the **Include Shortcuts** tool is selected (see *Object Lists* on page 115).
- Object property sheets (see *Object Properties* on page 97) - provide information on various forms of links on the following tabs:
    - Related Diagrams (see *Specifying Diagrams as Related Diagrams* on page 178)
    - Traceability Links (see *Creating Traceability Links* on page 429)
    - Mappings - where available (see *Chapter 16, Object Mappings* on page 469)
    - Requirements - if enabled (see *Requirements Tab* on page 105)
    - Dependencies - lists the objects that depend on the object (see *Dependencies Tab* on page 106)

Creating Traceability Links

You can specify any kind of connection between PowerDesigner objects through traceability links. These links can be created between objects of any type of model. They are used for documentation purposes only, and are not interpreted or checked by PowerDesigner.

You create an traceability link between objects in the same diagram by selecting the **Link/Traceability Link** tool in the Toolbox. Click inside the influent object symbol and, while

continuing to hold down the mouse button, drag the cursor and release it on the dependent object symbol.

In the following example, the CDM entities `School` and `Work` do not have any direct calculable connection, but `Work` is shown as being dependent on `School` through a traceability link:

You can refine the meaning of traceability links by applying stereotypes to them to identify the kind of dependency represented. You can enter a stereotype name by hand in the **Stereotype** column of the **Traceability Links** tab of the dependent object, or select a value from the list if you have previously defined stereotypes in an extension (see *Extension Files* on page 415).

Creating Traceability Links Between Objects from Different Models

You can create traceability links between any objects (including objects in different models) from the dependent object's property sheet. The model that contains the influent object must be open in the workspace.

1. Open the property sheet of the object, and click the **Traceability Links** tab.
2. Click the **Add Objects** tool to open the Add Objects dialog box.
3. Select a model from among the models open in the workspace from the Model box, (optionally) a package from the list, and an object from one of the sub- tabs and then click **OK** to add a traceability link to it.

   The traceability link you have created is listed in the List of Traceability Links (available from **Model > Traceability Links**) and, if the objects are in the same diagram, as an oriented link symbol.

4. [optional] Enter a type to further identify the traceability link you have just created in the **Link Type** column. You can specify link types for selection (see *Creating New Types of Traceability Links* on page 430).
5. Click **OK** to close the property sheet and return to the diagram.

Creating New Types of Traceability Links

You can enter a type directly into the **Link Type** column, or populate the list with link types for re-use.

1. Open the **Traceability Links** tab of any object.
2. Click the **Types and Grouping** tool and select **New Link Type**.
3. Enter a **Name** for the link type and, optionally, a **Comment** to explain its purpose. The name will appear in the list in the **Link Type** column.
4. [optional] If you want to view the new type in the PowerDesigner Resource Editor, select the **Open Resource Editor** checkbox.

All extensions are stored in PowerDesigner resource files. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files*.

5. Click **OK** to exit the dialog and return to the **Traceability Links** tab.

The new type is available for selection in the **Link Type** column.

Grouping Traceability Links by Object or Link Type

You can display the links on an object's **Traceability Links** tab as a single list or grouped by object type or link type by clicking the **Types and Grouping** tool and selecting the appropriate grouping command.

When you select **Group by Object Type** or **Group by Link Type**, the different types of objects or links are displayed on separate sub-tabs. To add a link to a new object type or with a new link type, click the plus sign on the leftmost sub-tab.

CHAPTER 14   **Generating Models and Model Objects**

PowerDesigner provides a number of predefined model-to-model generation paths along with a mechanism for generating any model object into any other kind of object in another type of model.

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.

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

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 window* on page 434.

3. [optional – PDM-PDM generation only] Click the **DBMS Preserve Options** tab and set any appropriate options. For more details, see *Model Generation Options window DBMS Preserve Options tab* on page 435.

4. [optional] Click the **Detail** tab and set any appropriate options. For more information, see *Model Generation Options window Detail tab* on page 435.

5. [optional] Click the **Target Models** tab and specify the target models for any generated shortcuts. For more information, see *Model Generation Options window Target Models tab* on page 439.

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* on page 126.

7. Click **OK** to begin generation.

Model Generation Options Window

The Model Generation Options window 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 target for the model to be generated. Select one of the radio buttons to: - Share and link to the target definition - Copy the target definition to the model to allow it to be modified without affecting other models
Name	Specifies the name for the model to be generated.
Code	Specifies the code for the model to be generated.

Click the **Configure Model Options** button to open the Model Options dialog for the model to be generated, in which you can specify naming conventions (see *Naming Conventions* on page 107) and other model options. For detailed information about the model options available for each type of model, see the appropriate modeling guide.

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 target of the model to be updated.
Preserve modifications	Allows you to manually compare and merge the source model with the model to be generated in the Merge Models window (see *Chapter 7, Comparing and Merging Models* on page 275).
	If this option is not selected, all the existing objects in the target model will be automatically replaced by those in the generated model.

Model Generation Options Window DBMS Preserve Options Tab
This tab is only available when you generate a PDM from another PDM, and allows you to choose to preserve or lose, the values associated with various database objects.

- Physical options - for relevant objects
- Triggers and stored procedures – Note that, if the target belongs to a different DBMS family, triggers are automatically rebuilt.
- Database objects – non-standard objects. If the source DBMS does not support an object its option is not available. Only objects supported by the target DBMS will be generated.
- Extended attributes

Model Generation Options Window 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**. For more information, see *Checking a Model* on 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.
Generate Mappings	Defines the source model as the data source of the generated model and creates mappings between all source and target objects. This option is dependent on the Save Generation Dependencies option.

Option	Description
Rebuild Triggers	[CDM-PDM and OOM-PDM only] Instructs PowerDesigner to build triggers with preservation option, after the PDM generation. Rebuild takes place after merge if you are updating an existing PDM
Convert Names into Codes	Instructs PowerDesigner to generate target model object codes from source model object names using the appropriate conversion script (see *Name and Code Conversion Scripts* on page 110). 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 an OOM Java class attribute code may be `customerName` whereas the PDM table column that you want to generate must be called `CUSTOMER_NAME`. By selecting this option and specifying the appropriate conversion scripts in the model options available from the **General** tab, you can generate your columns with appropriate codes.
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.
Enable transformations	Enables pre- and post-generation transformations. When you click this button, the **Pre-generation** tab is displayed if the source model contains transformations. You can select the transformations to execute before generation. The **Extensions** tab also is displayed for you to select extension files to attach to the generated model. These files may contain post-generation transformations, in this case, the **Post-Generation** tab is displayed to let you select the transformations you want to be executed in the generated model. If the generation is an update, and the generated model contains extensions with post-generation transformations, the **Post-generation** tab is automatically displayed as soon as you click the Enable Transformations button.
Skip single root element	[XSM-PDM only] Specifies to ignore a single root element and to treat its immediate child elements as multiple roots, which will be generated as tables.
Generate columns/ attributes as elements	[PDM-XSM only], Specifies that PDM table columns are generated as child elements (instead of attributes) in the XML model. You can then set attributes to these child elements.
Class Prefix	Prefix for a class. It can help identify a class in a model

PDM Generation Options
The following additional options are available when generating a PDM:

Option	Description
Table: Table Prefix	Specifies a prefix for the names of generated tables.

Option	Description
Table: ID column type	[XSM-PDM only] Specifies the type to use for ID columns, and which will be used to generate `xs:ID` types. Default: `numeric`
Index: PK index names	Specifies the naming convention for primary keys. You can use the following variables in this and the Key index fields: • %TABLE% - Generated code of the table. This is the table code that is generated in the database. It may or may not be truncated if the code contains characters not supported by the DBMS • %TNAME%, %TCODE% - Table name and code • %TLABL% - Table comment Default: %TABLE%_PK
Index: AK index names	Specifies the naming convention for alternate keys. You can use the following variables in this field: • %REFR% - Generated code of the reference • %REFNAME%, %REFRCODE% - Reference name and code • %PARENT% - Generated code of the parent table • %PNAME%, %PCODE% - Parent table name and code • %CHILD% - Generated code of the child • %CNAME%, %CCODE% - Child table name and code • %PQUALIFIER% - Parent table qualifier • %CQUALIFIER% - Child table qualifier 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 Default: %TABLE%_AK
Index: FK index names	Specifies the naming convention for foreign keys, by default %REFR%_FK
Index: FK threshold	Specifies the minimum number of estimated occurrences of an entity necessary to create an index on a foreign key. The estimated number of occurrences can be specified in the entity property sheet. If this property is not set, foreign key indexes are generated by default.
Reference: Update Rule	Specifies the default update constraint for referential integrity.

Option	Description
Reference: Delete Rule	Specifies the default delete constraint for referential integrity.
Reference: 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 entity that generated the parent table, this variable proceeds from the conceptual environment. If no role is defined on the relationship or association, %PROLE% takes the content of %PARENT% to avoid generating a column with no name The following example checks the %PROLE% value; if it is equal to the parent name (which is the replacement value) then the template "%.3:PARENT%_%COLUMN%" will be used, otherwise template "%PROLE%" will be used because the user has entered a parent role for the relationship: `[%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. For more information about PowerDesigner variables, see *Customizing and Extending PowerDesigner > DBMS Definition Files*.
Reference: Use template	Controls when the FK column name template will be used. You can choose between the following radio buttons: • Always use template • Only use template 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`).

Model Generation Options Window 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. 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.

For more information, see *Generating Shortcuts and Replicas* on page 466).

Applying Model Transformations

Transformations are used to perform standard modifications to your model objects. You can apply transformations either on demand or during generation.

For information about creating transformations, see *Customizing and Extending PowerDesigner > Extension Files > Transformations and Transformations Profiles (Profile)*.

Applying Transformations During Generation

Transformation profiles can be used during model generation. Pre-generation transformations are applied to the source model, while post-generation transformations are applied to the target model.

During inter-model generation, it is impossible to execute both lists of transformations of the same transformation profile, because the current model is the source of generation but not the target as defined in the following example.
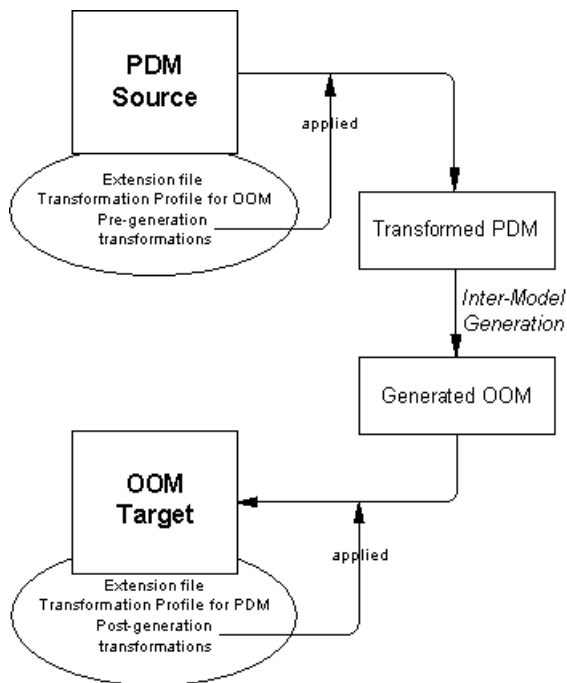
When you generate a model to the same type of model, you can execute both lists of transformations of the transformation profile provided you select the same extension in the source and in the target model.
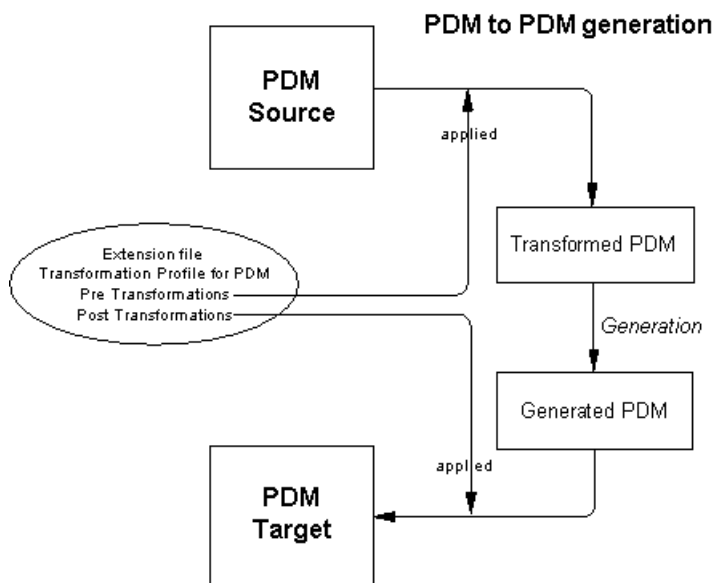
1. Select **Tools > Generate Model** to open the Model Generation Options dialog.

2. Click the *Details* tab, and click the **Enable Transformations** button to display the **Extensions**, **Pre-generation**, and **Post-generation** tabs.

3. Click the **Extensions** tab to select the extensions in which you have defined your transformations.

4. Click the **Pre-generation** tab and select profiles and transformations to be applied before generation. If you deselect a profile checkbox, none of its transformations will be executed. You can drag and drop profiles to modify transformation execution order. During generation, transformations are executed in the following order:

   • The order of profiles in the pre and post generation pages
   • The order of transformations in the profile itself
   • The order in which objects are treated in the model, that is to say beginning at the model level and recursing into sub-packages
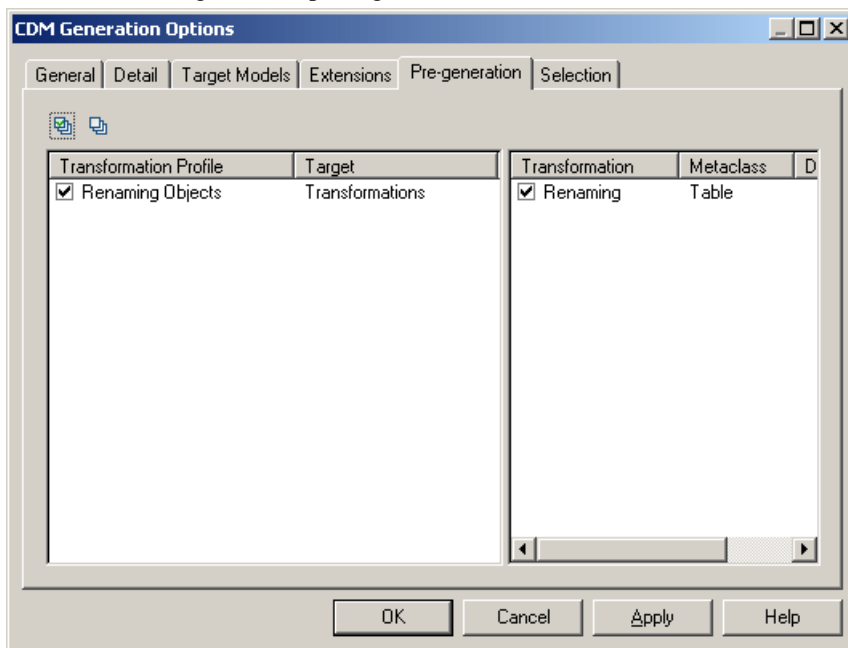
5. Click the **Post-generation** tab and select profiles and transformations to be applied after generation.

6. Click **OK** to begin the generation.

Applying Transformations on Demand

Transformations can be applied on demand in 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, if you are working in an OOM, you can create a transformation that converts all analysis classes with the <<control>> stereotype into components in order to add an implementation layer to your model.

Note: You can only invoke post-generation transformations on demand.

You can apply transformations through a command added to a main or contextual menu (see *Customizing and Extending PowerDesigner > Extension Files > Menus (Profile)*) or by opening the Apply Transformations window from the **Tools** menu.

1. Add one or more extensions containing post-generation transformations to your model.
2. Select **Tools > Apply Transformations** to open the Apply Transformations window.
3. Select transformations profiles and transformations on the Transformations tab.
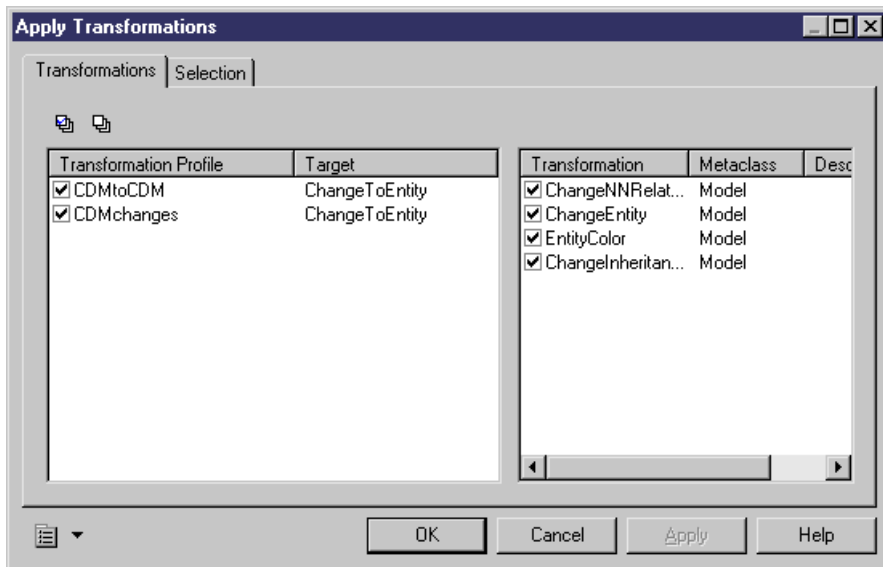
4. [optional] Click the Selection tab and deselect any objects that you want to exclude from the transformation.
5. Click OK to apply the transformations.

Data Type Conversion

PowerDesigner converts from the data types of the source model to those of the target model using its standard conceptual types (which are also used in the CDM). You can review the conversions that will be made by accessing the relevant resource file.

1. Select **Tools > Resources >** *Type* to open the appropriate list of resource files.

2. Select a resource file in the list, and click the Properties tool to open it in the Resource Editor.

3. Expand the following categories:

   • For DBMS resource files: **Script > DataType**
   • For other resource files: **Settings > DataType**

4. Review the entries in the DataType category, each of which is described in its Comment field.
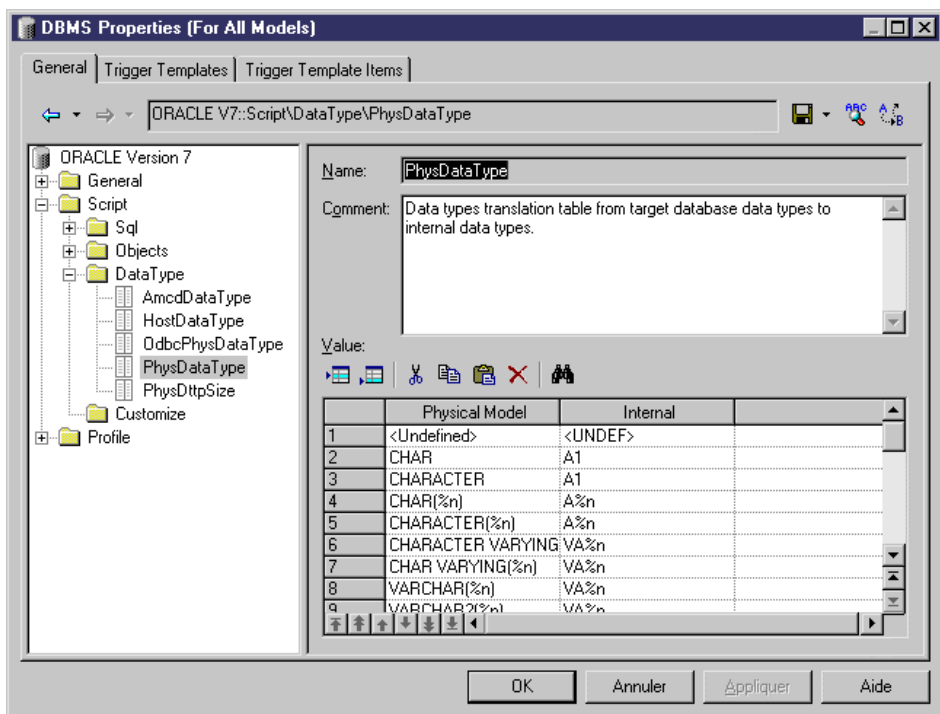
For more information on data types, see *Customizing and Extending PowerDesigner > DBMS Definition Files > Script/Data Type Category* and *Customizing and Extending PowerDesigner > Object, Process, and XML Language Definition Files > Settings Category: Object Language*.

Note: An extension file is provided to let you customize data mappings during a PDM-PDM generation, including on a column-by-column basis (see *Data Modeling > Working with Data*

Models > Generating Other Models from a Data Model > Generating Other Models from a PDM > Specifying Custom Data Type Mappings).

Generating Model Objects

You can extend the inter-model generation capabilities provided by PowerDesigner by defining your own 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.

Before you can generate model objects, you must define the details of the generation (see *Defining an Object Generation* on page 444)

Note: This procedure deals with generating a single type of objects to another model. For information about generating your entire model to another model, see *Generating a Model* on page 433).

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 an Object Generation* on page 444).
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 Window* on page 434).
3. [optional] Click the **Detail** tab and set any appropriate options (see *Model Generation Options Window Detail Tab* on page 435).
4. [optional] Click the **Target Models** tab and specify the target models for any generated shortcuts (see *Model Generation Options Window Target Models Tab* on page 439).
5. [optional] Click the **Selection** tab and select or deselect objects to generate (see *Adding an Item from a Selection List* on page 126).
6. Click **OK** to begin generation.

Defining an Object Generation

You can define as many object generations as you need. The generations are stored as objects in your model and are available as commands under the **Tools** menu.

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.
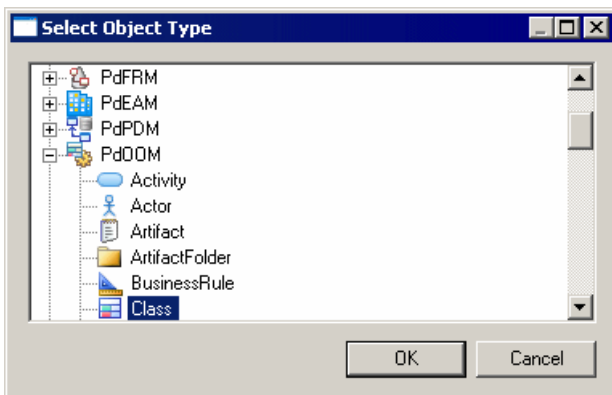
You can choose any object (including extensions) from any PowerDesigner model type, but some choices will be more appropriate than others.

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.
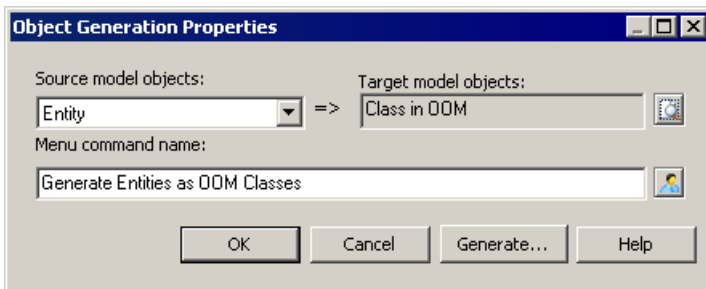
5. Click **Generate** to launch the generation immediately or **OK** to save the generation definition for use later.

Once you have defined an object generation, you can launch it at any time by selecting **Tools > Generate Objects >** *Menu command name* .

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.

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.

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.

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

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.
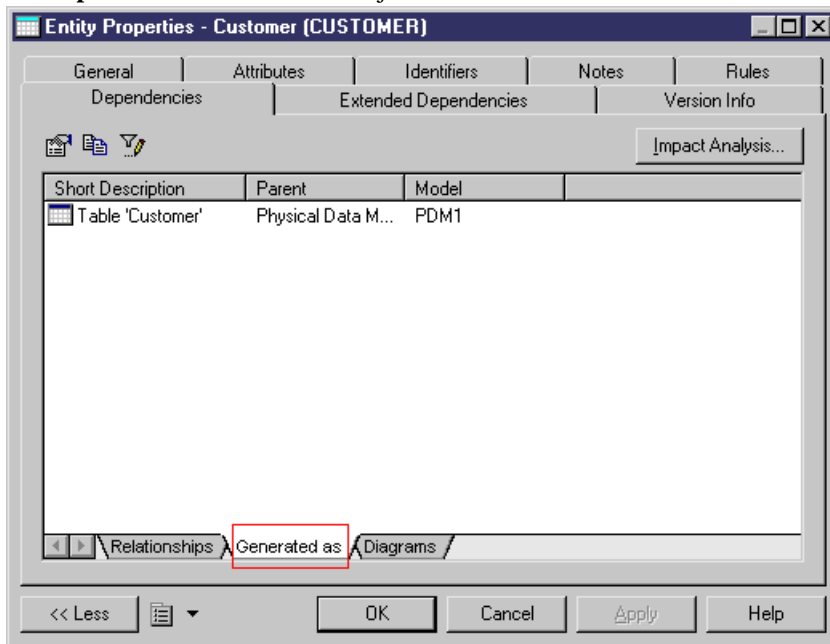
In the Impact Analysis dialog box, this link does not appear by default, you have to click the *Extract Dependencies* tool to display the generation links:
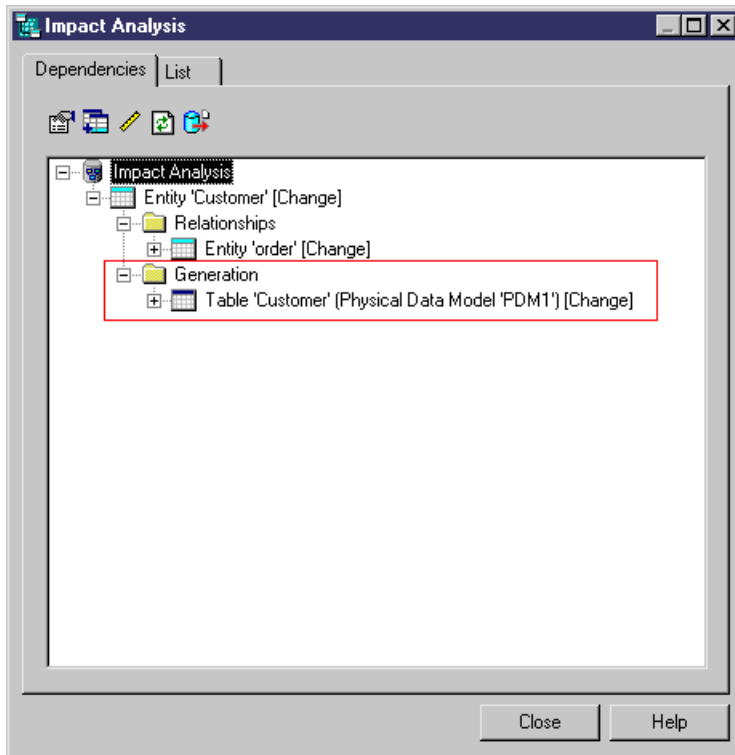
CHAPTER 15 **Shortcuts and Replicas**

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.

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.

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:
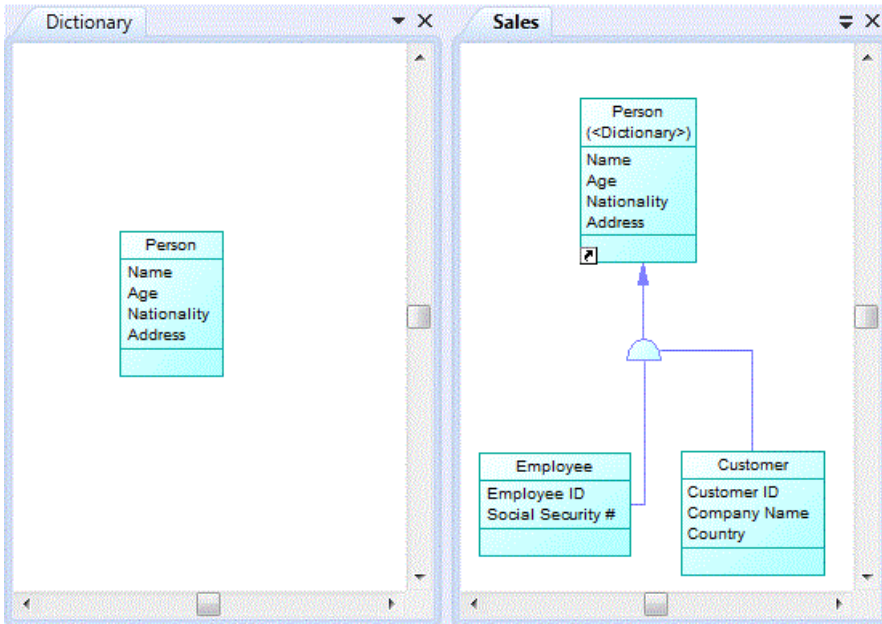
- *Automatic updates* - When the target object changes, the updates cascade automatically (if both the referencing and target models are open in the workspace) to all the shortcuts and replicas.

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, Links	Diagrams, Links, Packages
RQM	Traceability link, user allocation	-
EAM	-	-
BPM	Decisions, synchronizations	Starts, ends, synchronizations
CDM/LDM	Associations, Inheritances	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

Module	No Shortcuts For...	No Replicas For...
XSM	Internal shortcuts not supported for any object. External shortcuts not supported for imports, includes, redefines, annotations	-

Note: You cannot 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. 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.

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 *Dragging and Dropping Objects* on page 130.

- 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 *Copying and Pasting Objects* on page 132.

- List of Shortcuts - Select **Model > Shortcuts** to open the List of Shortcuts and click the Add Shortcuts tool:
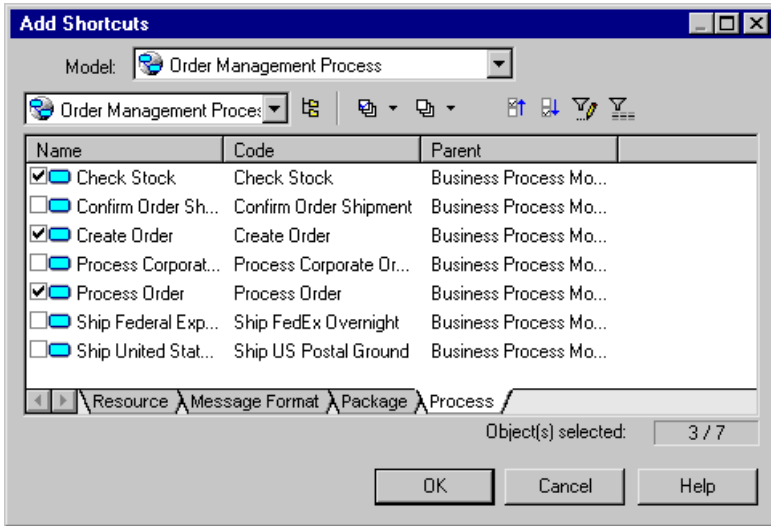
Select the target model and package in the upper part of the dialog and objects in the sub-tabs in the lower part of the dialog, and then click **OK** to add them as shortcuts to the current model.

For more information about the tools in this dialog, see *Adding an Item from a Selection List* on page 126.

Shortcuts in the Browser and shortcut symbols in diagrams both display the shortcut arrow overlay on their bottom left corners.
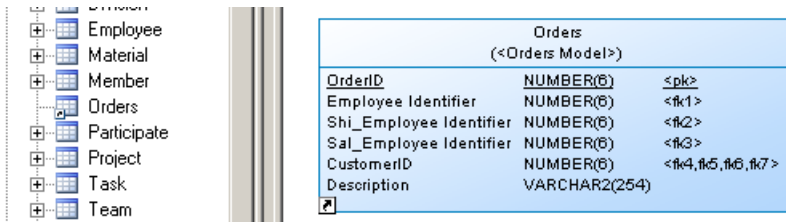
If you select two linked objects in a model and create shortcuts from them, a shortcut of the link between them will also be created. If, after having created shortcuts from two objects, you subsequently create a link between them in the target model, you can update the referencing diagram to display 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:
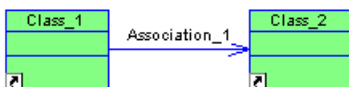
The following rules restrict the use of shortcuts:

- 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.
- You cannot create links between two shortcuts if the link implies a parent/child hierarchy (for example, you cannot create a reference link between two table shortcuts in a PDM).
- You cannot create a link between an entity and the shortcut of an inheritance.

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* on page 137.

Shortcut Properties

To open the property sheet of a shortcut and view or edit its properties, double-click its Browser entry or diagram symbol. The information that you will see depends on your settings for the **Internal Shortcut** and **External Shortcut** general options and the **External Shortcut Properties** model option

To view the target object property sheet in place of the shortcut, click **Tools > General Options > Dialog** and select **Target object** in **Internal Shortcut** and **External Shortcut**. To maximize the information on external shortcut property sheets and symbols, click **Tools > Model Options > Dialog** and select **All** in **External Shortcut Properties**.

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.

The following properties are available on the **General** tab of a shortcut property sheet or on the **Shortcut** tab of a target object property sheet:

Property	Description
Target type	Specifies the type of the target model and the type of object which the shortcut references.
Name	Specifies the name of the target object. Click the **Properties** tool to the right of this field to view or edit its properties.
	Click the **Change Target Object** at the bottom of the dialog to open a dialog to allow you to choose a different target object (which must be of the same type, and cannot already be referenced by another shortcut in the same package).
Code	Specifies the code of the target object.
Target model	Specifies the name of the model to which the target object belongs. Click the **Properties** tool to the right of this field to view or edit its properties.
Target package	Specifies the name of the model package to which the target object belongs. Click the **Properties** tool to the right of this field to view or edit its properties.
Shortcut type	Specifies whether the shortcut is external (from another model) or internal (from another package in the present model).
Status	Specifies 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	[external shortcut only] Specifies that the shortcut should be generated when generating another model from the referencing model (see *Chapter 14, Generating Models and Model Objects* on page 433).
Generated as	[external shortcut only] Specifies how the shortcut will be generated (see *Generating Shortcuts and Replicas* on page 466). You can choose between: • Shortcut - the shortcut is generated as a shortcut and retains its link to its target object • Object – the shortcut is generated as an independent object and is no longer linked to its target object

A shortcut property sheet also includes the following tabs:

- **Dependencies** - Lists the objects with connections to the shortcut (see *Viewing Shortcut and Replica Dependencies* on page 464)
- **Version Info** - Specifies the shortcut owner, modification and creation details

Viewing Shortcuts

Shortcuts can be identified in the referencing model in the Browser and diagrams by the shortcut arrow overlay on their bottom left corners.

In the following example, the Orders table is a shortcut. It appears in the Browser with the other tables that belong to the model. Both the Browser entry and the diagram symbol display the shortcut arrow overlay:
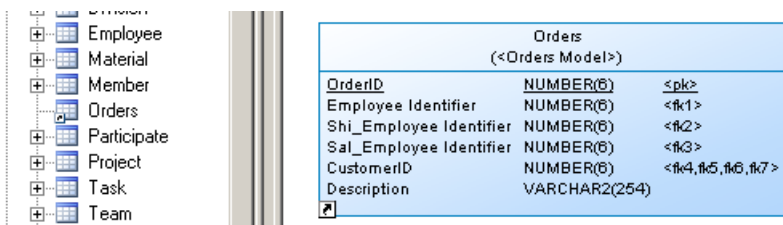
For external shortcuts, if you have selected to display all properties (see *Shortcut Properties* on page 455) or if the target model is open in the workspace, the properties that appear on its symbol are controlled by the standard display preferences for that type of object.

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:

Prefer- ence	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:
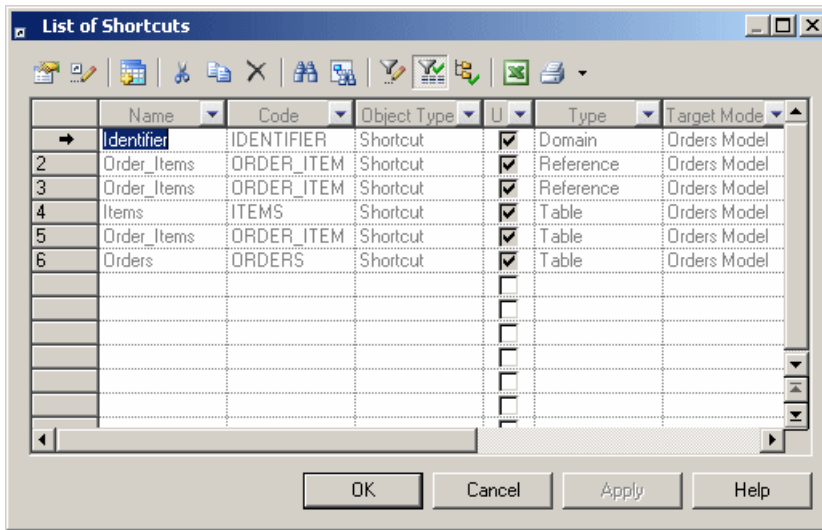
The **U**[sed] column is checked when the shortcut has a symbol in a diagram or if it is referenced by at least one other object. If this column is not checked, the shortcut is not used in the model and you can safely delete it.

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:
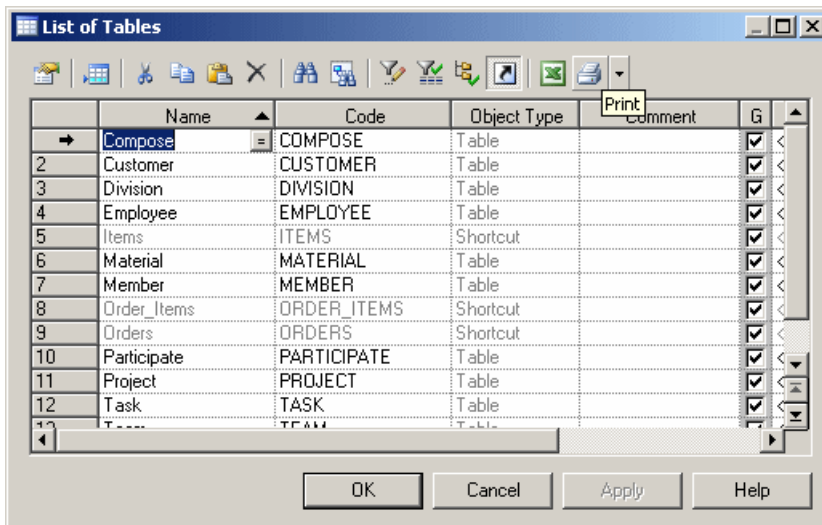
For information about working with these lists, see *Object Lists* on page 115.

Creating Replicas

You can create replicas to target objects from another package in the current model, or from another model open in the workspace through drag and drop, or from the Replicate Objects dialog.

- 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 replica (or onto a model or package in the Browser). Release the right mouse button, and select **Replicate Here**.

  You can also create replicas of objects by left-clicking and dragging and dropping them while holding down **Shift**+**Alt**.

  For more information about dragging and dropping objects, including how to change the default behavior, see *Dragging and Dropping Objects* on page 130.

- Replicate Objects dialog - Select **Edit > Replicate Objects** to open the Replicate Objects dialog:
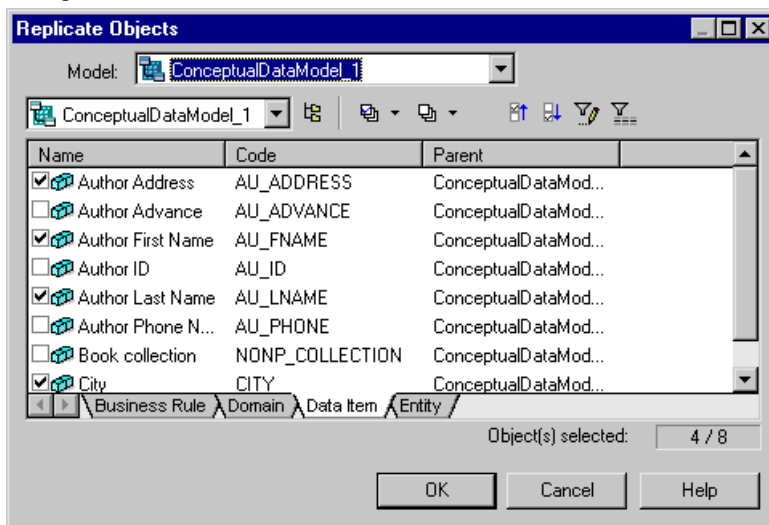
  Select the target model and package in the upper part of the dialog and objects in the sub-tabs in the lower part of the dialog, and then click **OK** to add them as replicas to the current model.

  For more information about the tools in this dialog, see *Adding an Item from a Selection List* on page 126.

Sub-objects (such as table columns or class attributes and operations) are automatically replicated when you replicate their parent object. You can replicate sub-objects individually through drag and drop only. You could, for example, create library models to group sub-objects that you regularly use, ready for replication.

Name uniqueness rules are respected when you replicate objects, so that replicas are renamed upon creation in the referencing model when objects with the same name already exist.

Desynchronizing Replica Properties

By default, replicas have read-only property sheets appropriate to their object type. To desynchronize one or more properties and make them editable, select the **Version Info** tab, click the **Replication Properties** button to open the replication property sheet, select the **Attributes** or **Collections** tab, and deselect the properties that you want to override in the replica.

Note: To break the replica's link to its target and turn it into an independent object, click the **Delete Replication** button. To make multiple replicas into independent objects, select **Model > Replications** to open the List of Replications, and delete the relevant replications.

The **General** tab of the replication property sheet contains the following properties:

Property	Description
Original Object Model	Specifies the name of the model to which the target object belongs. Click the **Properties** tool to the right of this field to view or edit its properties.
Full Name	Specifies the full path to the target object. Click the **Properties** tool to the right of this field to view or edit its properties.
Type	Specifies the type of the target object.
Status	Specifies the status of the target 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* on page 466).

A replication definition also includes the following tabs:

• Attributes - Lists the attributes of the replicated object. Deselect the check box of an attribute to make it editable for the replica.
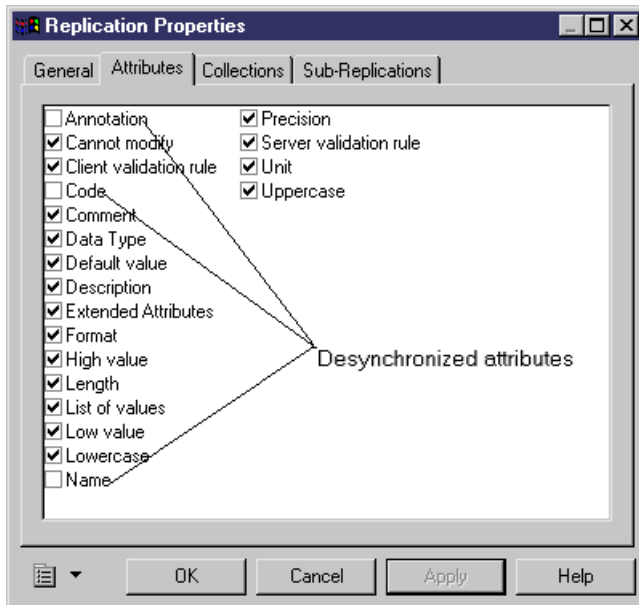
- Collections - Lists the collections of the replicated object. Deselect the check box of a collection to make it editable for the replica.
- Sub-Replications - Lists the sub-objects replicated as part of the object replication. Click the **Properties** tool to open the property sheet of the sub-replication in order to make editable one or more of its properties.

Viewing Replicas

Replicas look identical to normal objects in the Browser and diagrams, but all or part of their property sheets are grayed out and uneditable. Replications, which maintain the link between

the replica and the original object, are not accessible from the Browser or diagrams, but can be viewed by selecting **Model > Replications** to open the List of Replications.

	Original Obje▲ ▼	Replica Object ▼	Original Obje ▼	Original Model ▼
6	Identifier	Identifier	Domain	Project Management (PDM)
7	Material	Material	Table	Project Management (PDM)
8	Member	Member	Table	Project Management (PDM)
9	Money	Money	Domain	Project Management (PDM)
10	Name	Name	Domain	Project Management (PDM)
11	Phone	Phone	Domain	Project Management (PDM)
12	Project	Project	Table	Project Management (PDM)
13	Short description	Short description	Domain	Project Management (PDM)
14	Task	Task	Table	Project Management (PDM)
15	Task date control	Task date control	Business Rule	Project Management (PDM)
16	Team	Team	Table	Project Management (PDM)
17	Used	Used	Table	Project Management (PDM)

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* on page 459), but you can access the property sheets of replications and, from there, the property sheets of replicas and their target or original objects (see *Desynchronizing Replica Properties* on page 460).

If you delete a replication from the list, the associated replica will lose its link to the original 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:
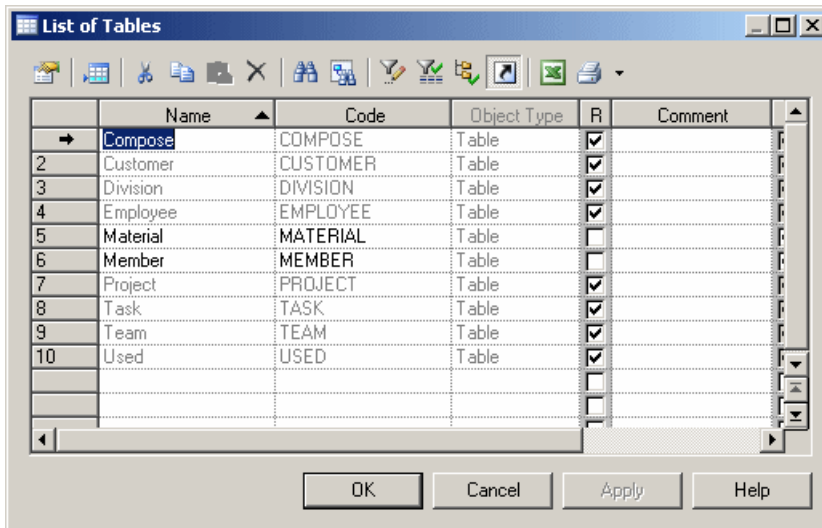
If this column is not visible, click the **Customize Columns and Filter** tool, and select the `Replica` attribute to add it to the list.

Any property that is synchronized (and thus not editable) is grayed.

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 *Chapter 7, Comparing and Merging Models* on page 275.

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.

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.

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.
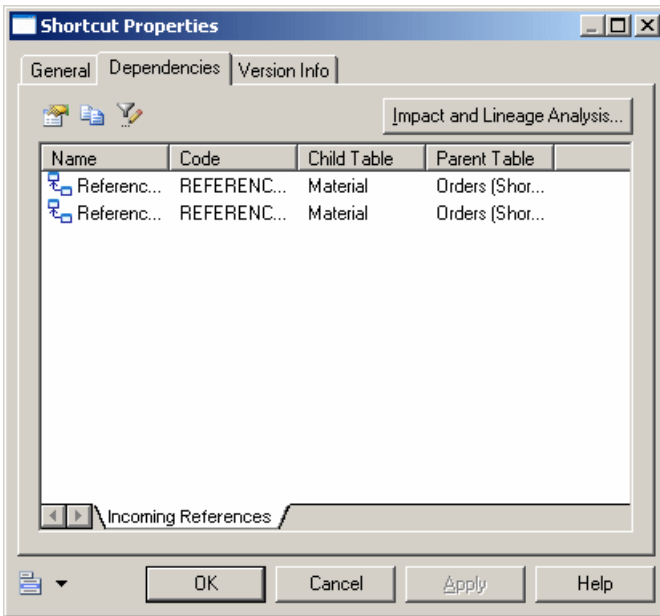
The **Shortcuts** and **Replicas** subtabs on an object property sheet **Dependencies** tab list all the shortcuts and replicas that have been created from the object in all the models open in your workspace:
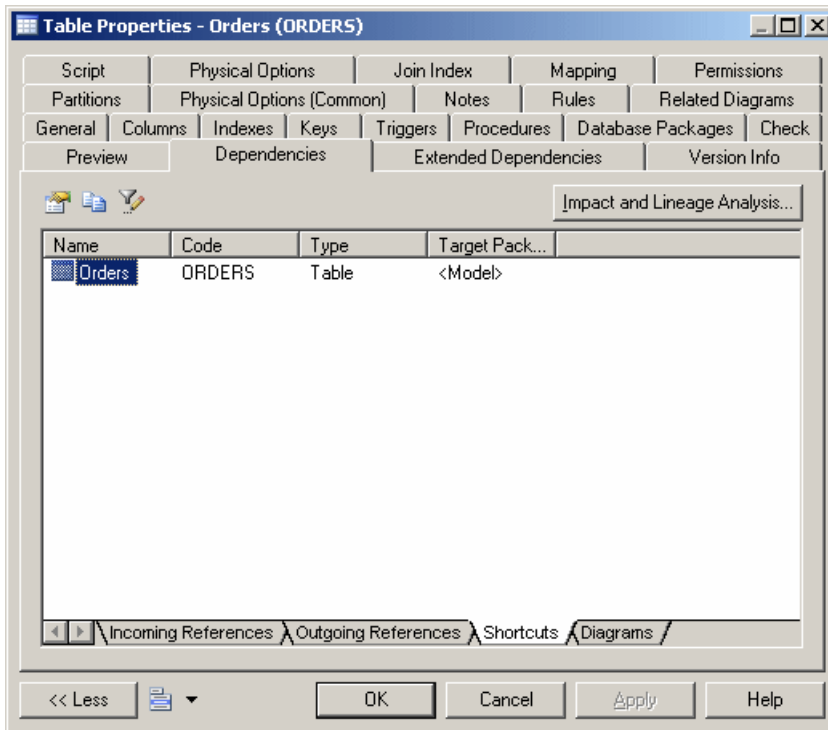
Working with Target Models

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 depend upon

objects in target models. To view all the target models on which your model depends, select
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* on page 115.

Note: Deleting a target model will 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.

Generating Shortcuts and Replicas

If you generate another model from your model, you can choose not to generate your shortcuts
and replicas. If you generate your shortcuts, you can choose to generate them as shortcuts or to
transform them into independent objects. If you intend to generate to another type of model
(for instance, a PDM from an OOM) and want to retain your shortcuts and replicas, then you
should first generate the target models from which they were created, so that your generated
model will have appropriate target objects to synchronize them with.

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* on page 455). For replicas, you must ensure that the **Generate** option on the associated replication property sheet is selected (see *Desynchronizing Replica Properties* on page 460). If this option is not selected, the replica will not be generated.

2. Generate OOM1 to PDM1 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 Window Target Models Tab* on page 439) 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.

CHAPTER 16   **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
XSM					X

Note: Mappings are also used to define data replications in the DMM. For more information, see *Data Movement 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 *Creating Mappings from the Mapping Editor* on page 476)
- on the Mapping tab of the property sheet of the target object (see *Creating Mappings from an Object's Property Sheet* on page 495)
- During model generation, by selecting the **Generate Mappings** option in the Model Generation Options window (see *Generating Models* on page 433).

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

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

LDM Source Objects	CDM Target Objects	LDM Target Objects	PDM Target Objects	OOM Target Objects
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

OOM Source Objects	CDM Target Objects	LDM Target Objects	OOM Target Objects	XSM Target Objects
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 mapings from OOM or PDM source objects.

Object to Relational (O/R) Mapping

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.

Furthermore, object codes in an OOM for a given object language are often different than codes used in a relational database. This requires that you modify object codes after generating an OOM into a PDM or a PDM into an OOM to be compliant with the object language.

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:
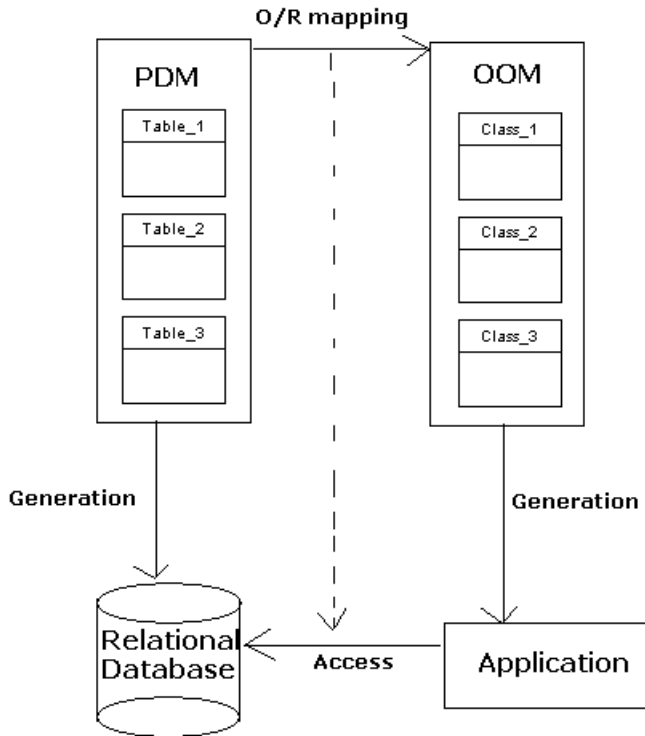
Object persistence implies to store and extract objects in a relational database.

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:
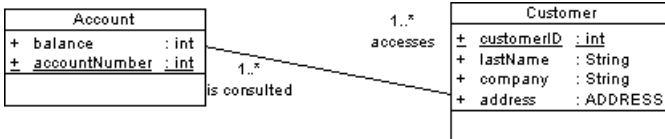
In a relational database, this association is mapped to the following associative table:
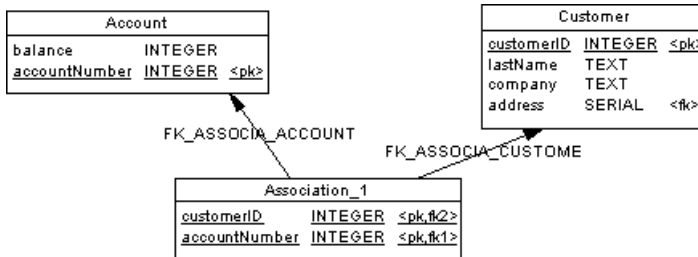
SQL Queries

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
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

Mappings between Operational, Data Warehouse, and OLAP Databases

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.

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.
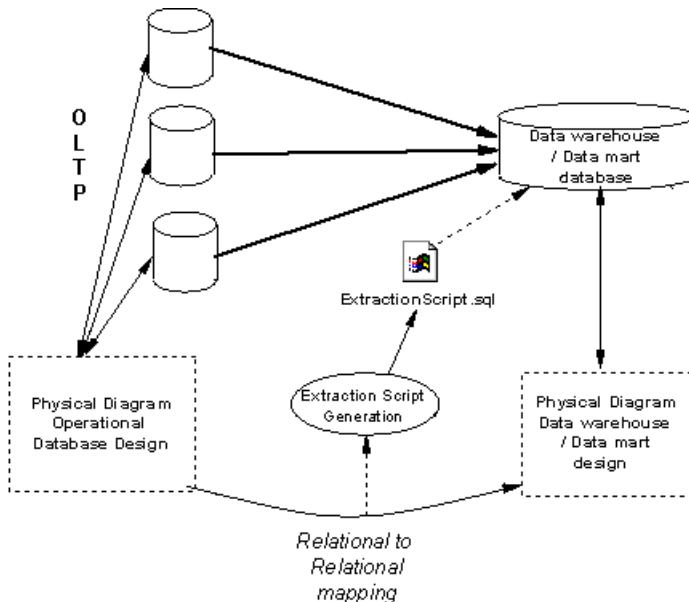
You can map physical tables (including those of type dimension or fact) to cube dimensions or cube measures in OLAP databases, and use these mappings to generate cube data in text files to be loaded by OLAP engines. When you use the Rebuild Cubes command to create cubes and dimensions from fact and dimension tables, mappings between source tables and OLAP objects are automatically created.
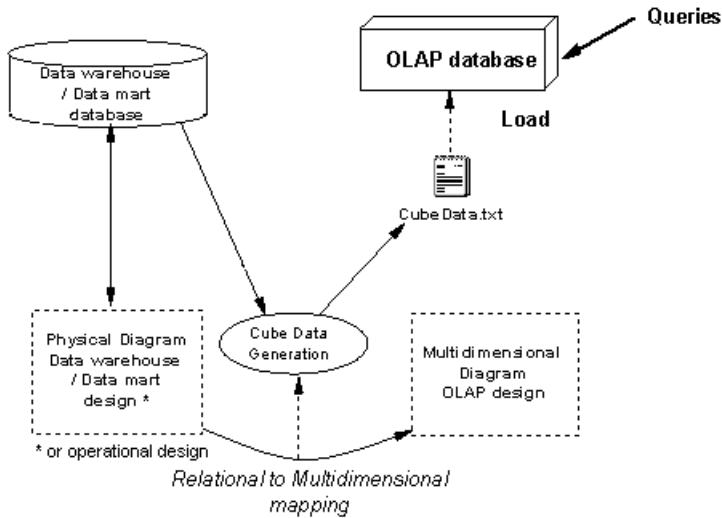
Once the source tables or views are identified, you can define mappings between attributes or measures and table columns.

The **Select** sub-tab displays the entire SQL statement used to select data in the data source. This statement is automatically generated. The Generate Cube Data feature uses this SQL statement to fill the text files used to populate cubes in an OLAP database.

Note: For more information about generating extraction scripts, rebuilding cubes, and generating cube data, see *Data Modeling > Building Data Models > Multidimensional Diagrams*.

The Mapping Editor

The Mapping Editor provides a global view of all the mappings to objects in your model, allowing you to quickly identify mapped and unmapped objects.

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 with the Data Source Creation Wizard (see *Creating a Data Source* on page 491).
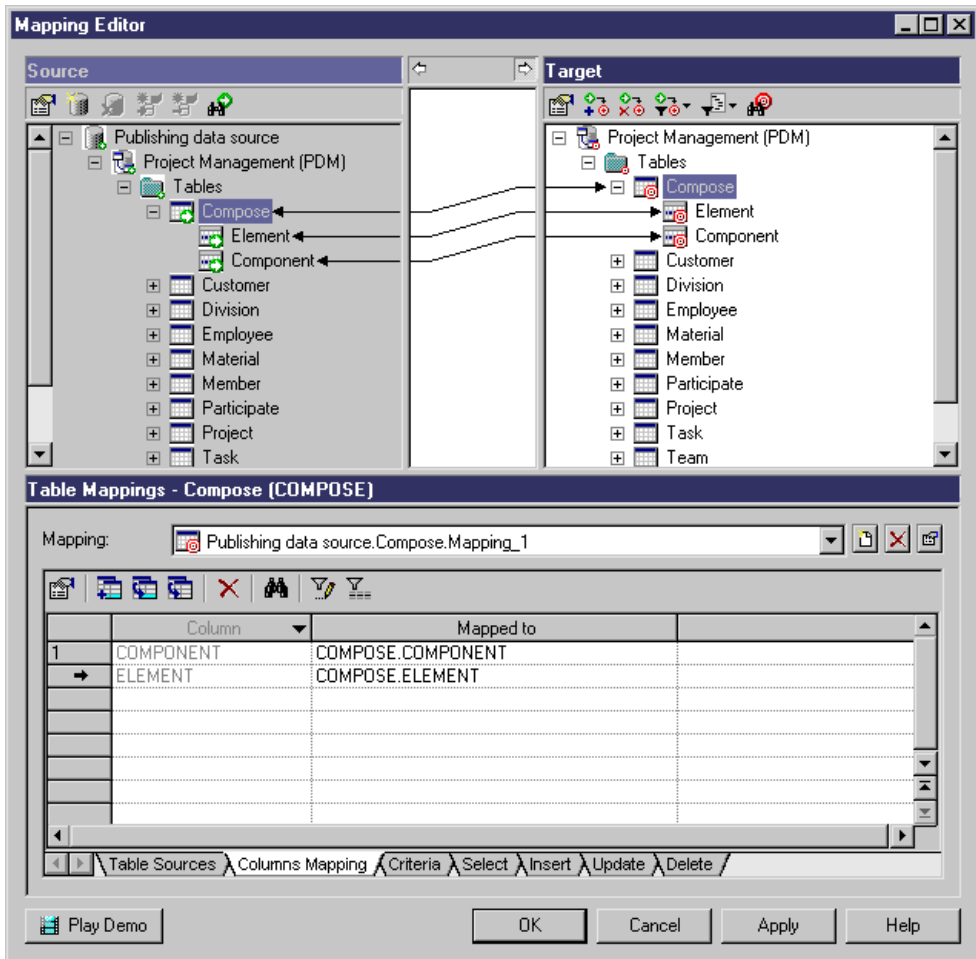
The Mapping Editor window is divided into three panes:

- The Source pane – has a gray background and is, by default, on the left of the window. For more information, see *Mapping Editor Source pane* on page 488.
- The Target pane – has a white background and is, by default, on the right of the window. The model from which you launch the Mapping Editor is displayed in the Target pane, and the mappings are saved in this model. For more information, see *Mapping Editor Target pane* on page 488.
- The Mappings pane - can contain a mapping list or the same content as the content of the Mappings tab of an object's property sheet. If you define a mapping for an object using the Mapping Editor, the Mappings tab is automatically updated in the object's property sheet. For more information, see *Mapping Editor Mappings pane* on page 489.

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.

Mapping Links and Symbols
When a mapping is created, a non-editable link line is drawn between the mapped objects. The arrow specifies the direction of the data transfer. When an arrow is displayed on both extremities of the link, it means the data can be extracted from the source to target and vice versa.

A link extends from the source pane to the target pane and passes through an area between the two panes. If you click a link:

- In the Source pane – you open the property sheet of the source object.
- In the Target pane – you open the property sheet of the target object.
- In the middle area - you display the mapping in the Mappings pane.

A small symbol is displayed in the bottom right-hand corner of the icons and also on their respective parent hierarchy icon to ease readability:

Pane	Mapped object icon	Parent hierarchy icon
Source (green)		
Target (red)		

When you define a mapping for an object (a table, for example) a mapping is automatically defined for its sub-objects (columns, for example) when their name matches the name of the source sub-objects.

Creating Mappings in the Mapping Editor

You can create a mapping in the Mapping Editor by drag and drop or using the **Create Mapping** tool or contextual command. To generate default mappings between source and target objects that have the same name, right-click a data source and select **Generate Default Mappings**.

Note: When dragging and dropping, you can either create a mapping or reuse and complement an existing mapping. When using the **Create Mapping** tools or contextual command, you can create multiple mappings for the same target object. Each mapping object is added to the Mapping list and you can access its property sheet.

When objects in the Source and Target pane are mapped, any sub-objects that share the same name are automatically mapped as well. You can force the mapping with the **Generate Mapping** tool in the **Sub-Object** sub-tab of the **Mappings** pane.

Note: You can multi-select object icons in the Source pane by holding down the SHIFT key while clicking, and map them simultaneously to a target object.

Creating a Mapping by Dragging and Dropping

You can create a mapping by dragging one or more objects from the Source pane and dropping them onto an object in the Target pane.

Drag one or more objects from the Source pane and drop it onto an object in the Target pane.

A visual link is drawn between the objects and a small symbol is displayed on the source and the target object icons. The mapping details display in the Mappings pane.

Note: If a mapping is already defined for an object icon, the drag and drop feature reuse the existing mapping to complement it.

The pointer becomes a barred circle when it is over an impossible drop position.

When a source object in a multi-selection cannot be dropped in the selected place in the Target pane, the whole selection is rejected.

The result of a drag and drop depends on the objects being dragged and where you drop them:

- **Object > Object** - The objects are mapped together, along with any sub-objects that have the same name and code. The mapping is displayed in the Mapping list in the Mappings pane. The source object is in the Sources sub-tab and the mapping of its sub-objects displays in the Sub-Object Mapping sub-tab.
- **Sub-object > sub-object** – [where the sub-objects were not mapped because their names are different, and you have to "force" the mapping] The objects are mapped together and this mapping is added to the list of attributes mappings in the Sub-Object Mapping sub-tab.
- **Sub-object owned by a different object > sub-object** - The objects are mapped together and this mapping is added to the list of attribute mappings in the Sub-Object Mapping sub-tab.
- **Sub-object or object > folder** – The object (and its sub-objects if any) are created in the appropriate folder in the target model and are mapped to their respective source objects. Objects are displayed as new mappings in the Mapping list, and sub-objects are added to the list of attribute mappings in the Sub-Object mapping sub-tab.
- **Sub-object owned by a different object > folder** [XML target only] – The object owing the sub-object is mapped to the target object (and displays in the Sources sub-tab), and the sub-object is mapped to a new created sub-object in the target model, and added to the list of attributes mappings in the Sub-Object mapping sub-tab.

[XML Specifics] As an XML element can correspond to a simple type containing only an elementary value (for example <Name>Bill</Name>), you can directly map an element to a class attribute, an element attribute or to a table column using the drag and drop feature.

Creating a Mapping with the Create Mapping Tool

You can create a mapping using the Create Mapping tool on the Mapping Editor toolbar.

1. Select one or more object icons in the Source pane and select an object icon in the Target pane.

2. Click the Create Mapping tool in the Target pane toolbar.

3. [If one of the selected source objects is ambiguous (a folder for example)] A selection dialog opens. Select an appropriate object from the list, and then click OK.

   A visual link is drawn between the objects and a small symbol is displayed on the source and the target object icons. The mapping details display in the Mappings pane.

In the following example, a mapping is defined between the tables EUROPEAN_SALES in Model_2 and ASIAN_SALES in Model_3 and the table TOTAL_SALES in Model_1. Each table has a column called TOTAL AMOUNT. You can define column TOTAL AMOUNT in TOTAL_SALES as the sum of total amounts in EUROPEAN_SALES and ASIAN_SALES:
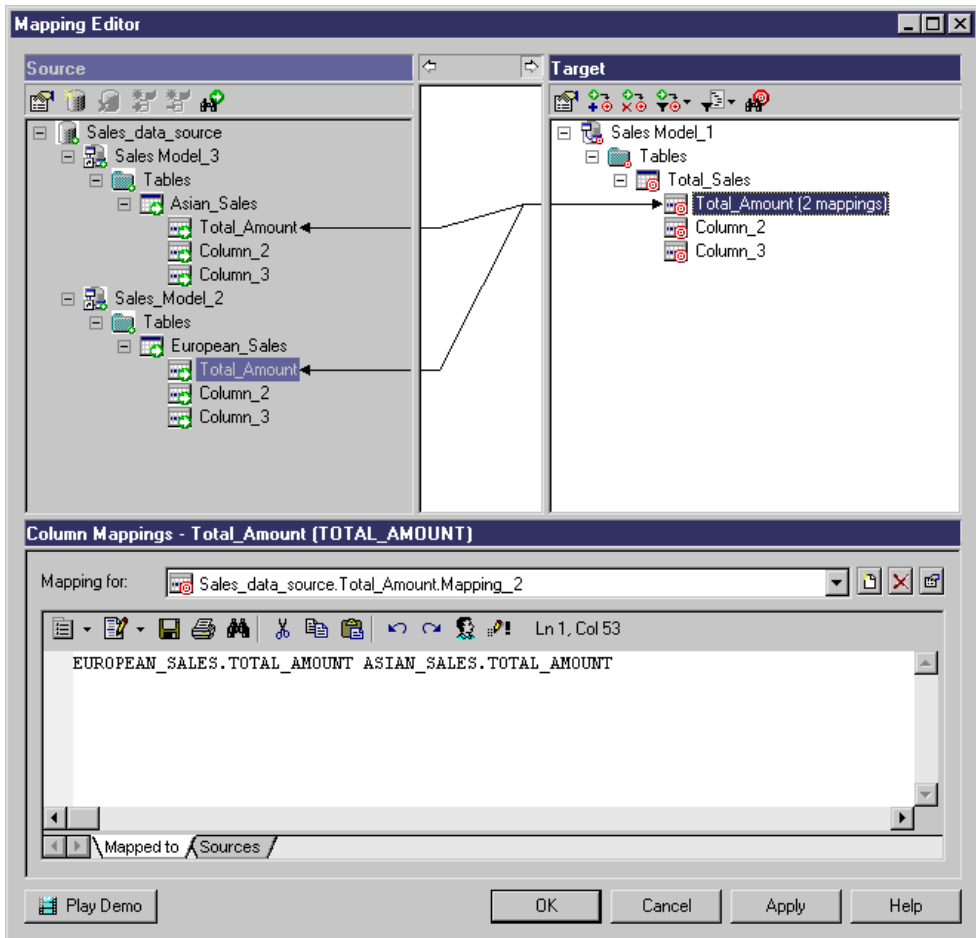
Creating a Mapping with the Create Mapping Tool in the Mappings Pane

You can create a mapping using the Create Mapping tool from the Mappings pane.

1. Select an object in the Target pane .

   or

   [if the target object is a sub-object] Select one or more objects in the Source pane and then select an object in the Target pane .

2. Click the Create Mapping tool in the Mappings pane.

   A visual link is drawn between the objects and a small symbol is displayed on the source and the target object icons. The mapping details display in the Mappings pane.

If the target object is an object owing sub-objects, a selection dialog box is displayed to let you select one or more sources for the target. Then they are displayed in the *Object* Sources sub-tab

of the Mappings pane. Besides, sub-objects that match are automatically mapped and are displayed in the *Sub-Objects* Mapping sub-tab.

If the target object is a sub-object, you also need to select one or more object icons in the Source pane. The mapping expression is displayed in the *Mapped to* sub-tab that you can modify. You can also add other source objects using the Add Sources tool in the *Sources* sub-tab.

Creating a Mapping Using an Object's Contextual Menu

You can create a mapping using the Create Mapping command from the contextual menu of an object icon either in the Target or in the Source pane.

Note that this command is only available for object to object and sub-object to sub-object mappings.

1. Select an object icon in the Target pane and select one or more object icons in the Source pane.

   or

   Select one or more object icons in the Source pane and select an object icon in the Target pane.

2. Right-click the object icon in the Source or in the Target pane and select Create Mapping.

   A visual link is drawn between the objects and a small symbol is displayed on the source and the target object icons. The mapping details display in the Mappings pane.

Mapping Examples

This topic contains different mapping examples.

Simple Mapping

A simple mapping can be designed as follows. The CITY table and its columns in the Source pane are mapped to the CITY table and its columns in the Target pane. Note that columns mapped automatically:
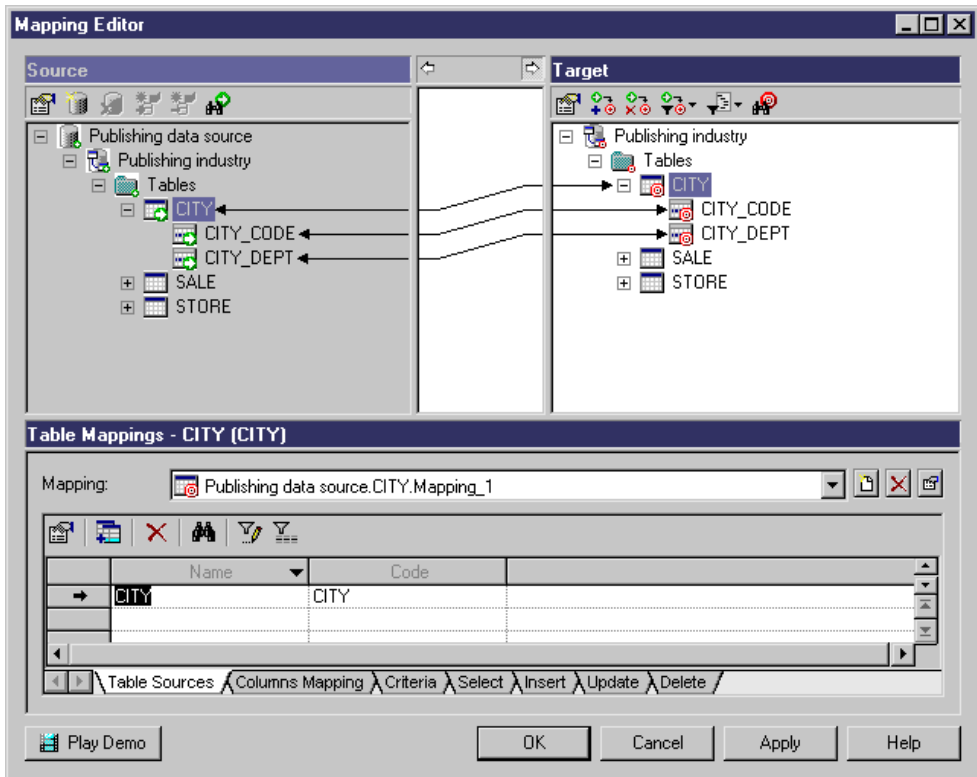
Multiple Sources Mapping
A mapping with multiple sources can be designed as follows. The CITY table in the Target pane has two sources (CITY and SALE tables):
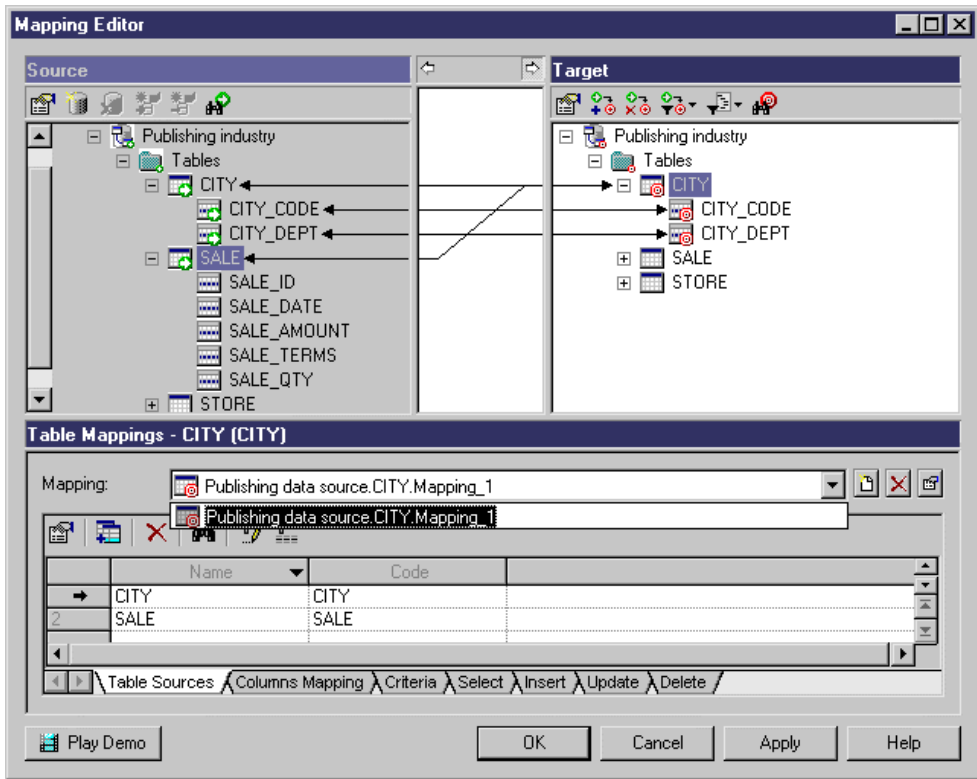
Different Mappings for the Same Target
A same target object can have different mappings. The SALE table in the Target pane is mapped to the SALE table (Mapping_1) and to the STORE table (Mapping_2) in the Source pane:
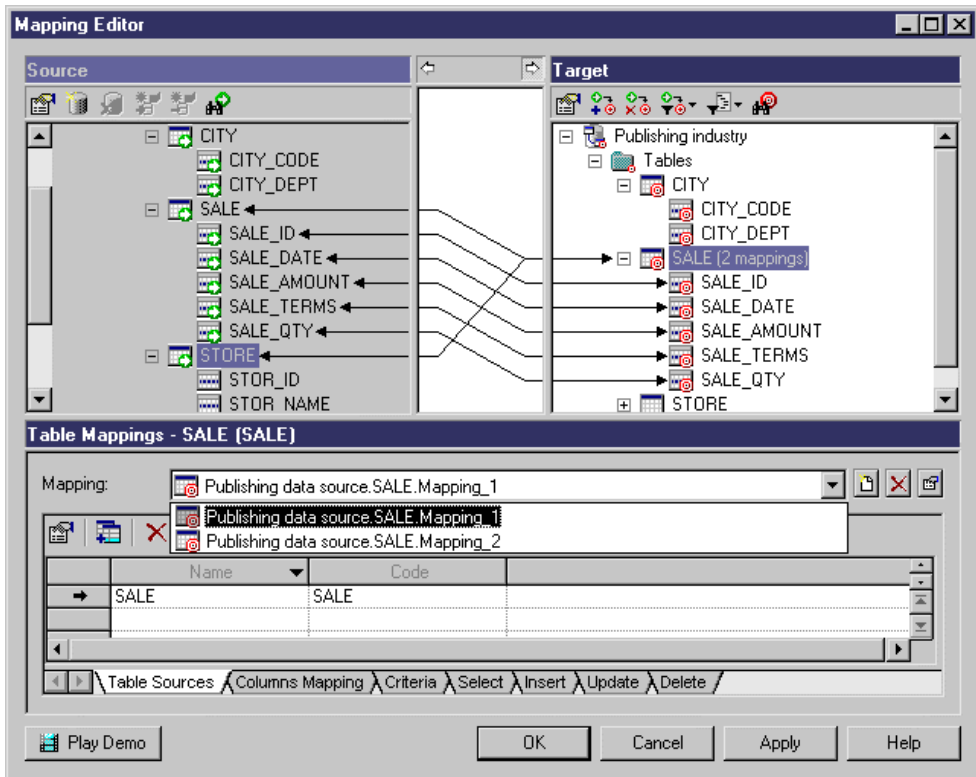
Creating Forward and Reverse Mappings

You primarily create mappings from one or more source models to the target model. These *forward mappings* define how to populate the objects of the current model with those of the data source model.

In forward mapping mode, the Source pane is on the left side of the window, the target pane is on the right side, and the right-hand arrow in the middle area is depressed:

You can also create *reverse mappings, which* define how to populate the objects of the source models with those of the current model. To do this, click the left-hand arrow in the middle area. What was originally the source model is renamed to the target:

When you create a mapping in this mode, PowerDesigner defines a default reverse mapping (bi-directional link) when the access type of the data source (see *Creating a data source* on page 491) is Read/Write or Write-only.

You can modify the default reverse mapping by switching Source and Target panes and edit the Insert, Update and Delete sub-tabs in the Mappings pane. These tabs summarize the reverse mapping for a given mapping. Mappings are always owned by the model on the right-hand side.

Depending on whether you are editing the source model mappings or the target model mappings, the following types of links display. Solid lines indicate links that can be edited in the current mode (forward or reverse), while dotted lines indicate links that cannot be edited in the current mode:

Edit...mappings	Type of link	Description
Current model	◄————————►	Forward mapping with potential reverse mapping (bi-directional link).
	————————►	Forward mapping only.
	◄- - - - - - - - - - -	Reverse mapping with no forward mapping defined.
Data source models	◄————————►	Reverse mapping with potential forward mapping (bi-directional link).
	◄————————	Reverse mapping only.
	- - - - - - - - - - -►	Forward mapping with no reverse mapping defined.

Reverse mappings can be required when, for example, data has several sources and you want to define how the data in the current model objects is inserted, updated or deleted in the data source model objects.

Modifying the Default Mapping Syntax
The Object Expression Editor dialog box allows you to build a textual or an XPATH expression from a list of objects.

The XPath expression allows you to locate a node (an element with its ramifications) in the hierarchical tree structure of an XML document.

When an XML mapping has several sources, the XPATH expression is built using the concatenation of the XPATH expression for each source object separated by a comma.

The Object Expression Editor dialog box is divided into specific panes containing the information shown below:

Information	Pane location
Objects types	Upper left part of the dialog box
Available objects	Upper right part of the dialog box
Expression script textbox	Lower part of the dialog box

The list of available objects depends on the selected object type. You can double-click individual objects from the list of available objects for insertion in the expression script textbox at the cursor last position.

When you commit your changes in the Expression script textbox, the object expression is updated in the "Mapped to" box of the *object* mapping property sheet.

1. Click the expression script textbox where you want to insert the textual or XPATH script.
2. Select an object type from the upper left part of the dialog box.

   For example, select Elements to display the list of available elements. The list of available objects of this type is displayed in the upper right part of the dialog box.
3. Double-click the available object that you want to add to the script.

   The item is added to the expression script.
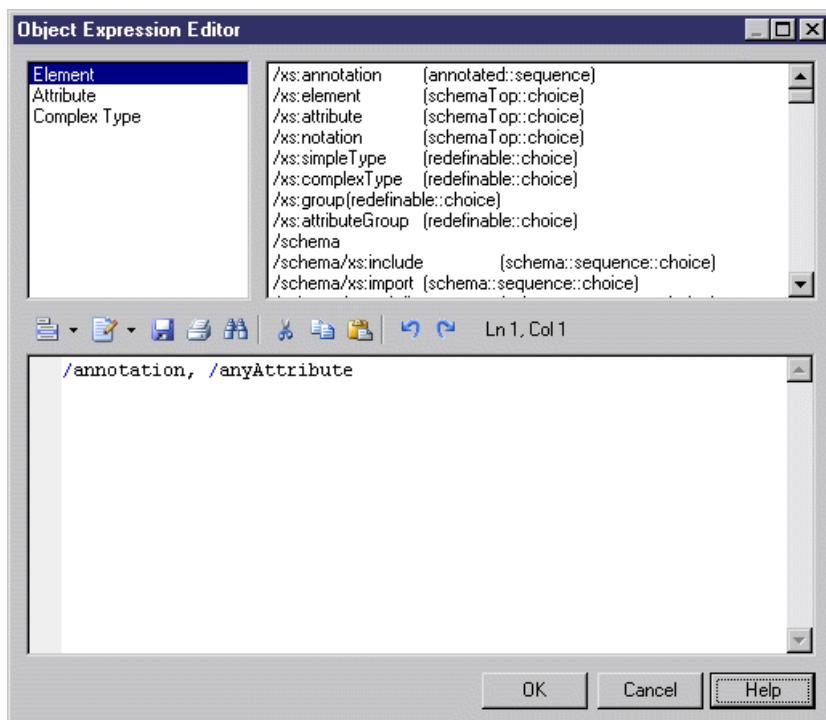
4. Click OK.

Mapping Editor Source Pane

The Source pane displays all the data sources and their source models showing all the objects that can be mapped. You must create one or more data sources and declare models in the list of data source models to define where (in which database or model) data should be extracted to be transferred to the model in the Target pane.

A data source can contain several models, you can select the source models among a list of models open in the current workspace.

The Source pane has a gray background, and is generally on the left in the Mapping Editor, except with reverse mapping editing (see *The Mapping Editor* on page 476).

The following tools are available in this pane:

Tool	Description
	Properties - Opens the property sheet of the selected source object.
	Create Data Source - Launches the Data Source Creation Wizard that allows you to name a data source, specify an access type, a model type and select source models. See *Creating a Data Source* on page 491.
	Delete Data Source - Deletes the selected data source. Related mappings, if any, are automatically deleted.
	Note: Select a source object and press the **Del** key (or right-click and select **Delete Mapping**) to delete the mapping with its source object. To delete a source object, select it and press **Shift**+**Del** (or right-click and select **Delete**).
	Add Model to Data Source - Adds one or more source models to an existing data source from a selection dialog box.
	Remove Model from Data Source - Removes the selected source model from the data source. Related mappings, if any, are automatically deleted.
	Find Source Object - Finds an object in the Source pane and highlights it.

Mapping Editor Target Pane

The Target pane displays the model where data is extracted.

The target pane is generally on the right in the Mapping Editor, except with reverse mapping editing (see *Creating Forward and Reverse Mappings* on page 485).

The following tools are available in this pane:

Tool	Description
	Properties - Opens the property sheet of the selected target object.
	Create Mapping - Creates a mapping between selected source and target objects. The mapping is materialized by a link and the mapping details appear in the Mappings pane. The tool behavior depends on the two selected objects: • Folder, model or sub-object mapped to object - A selection dialog lets you select an appropriate object in the source model. • Object mapped to object or sub-object mapped to sub-object - Objects are mapped together along with sub-objects with matching names and codes. • Object mapped to sub-object - A selection dialog lets you select an appropriate sub-object from the selected object in the source model.
	Delete Mappings - Deletes all the mappings for the selected target object. Alternately select a target object and press the **Del** key (or right-click and select **Delete Mapping**) to delete the mapping with its source object. To delete a target object, select it and press **Shift**+**Del** (or right-click and select **Delete**).
	Filter Mappings - Filters mappings to show: • All mappings • Only mappings of the selected object • Only mappings of the selected object and its sub-objects
	Filter Objects - Filters objects to show: • All objects • Only objects with mappings • Only objects without mappings
	Find Target Object - Finds an object in the Target pane and highlights it.

Mapping Editor Mappings Pane

When you create a mapping, it is displayed in the Mapping list in the lower part of the Mapping Editor dialog box, called the Mappings pane.

The Mappings pane content changes automatically depending on the object you select in the Target pane:

Selected object	Mappings pane displays...
Folder, model or package	A summary of the mappings it contains.

Selected object	Mappings pane displays...
Object	The same content as the Mapping tab in the object property sheet: Sources of the mapped object, mappings of its sub-objects if any, criteria, and SQL queries to execute the mapping if any.
Sub-object	The same content as the object mapping property sheet ("Mapped to" expression and the sources of the sub-object, which are PDM, OOM, XSM or CDM source objects for the target object).

You can create several different mappings for a single target object. Also, the same target object can have several sources. In the following illustration the *SALE* class has two different mappings *Mapping_5* and *Mapping_6*. Also *Mapping-6* has two sources: *CITY* and *SALE*.
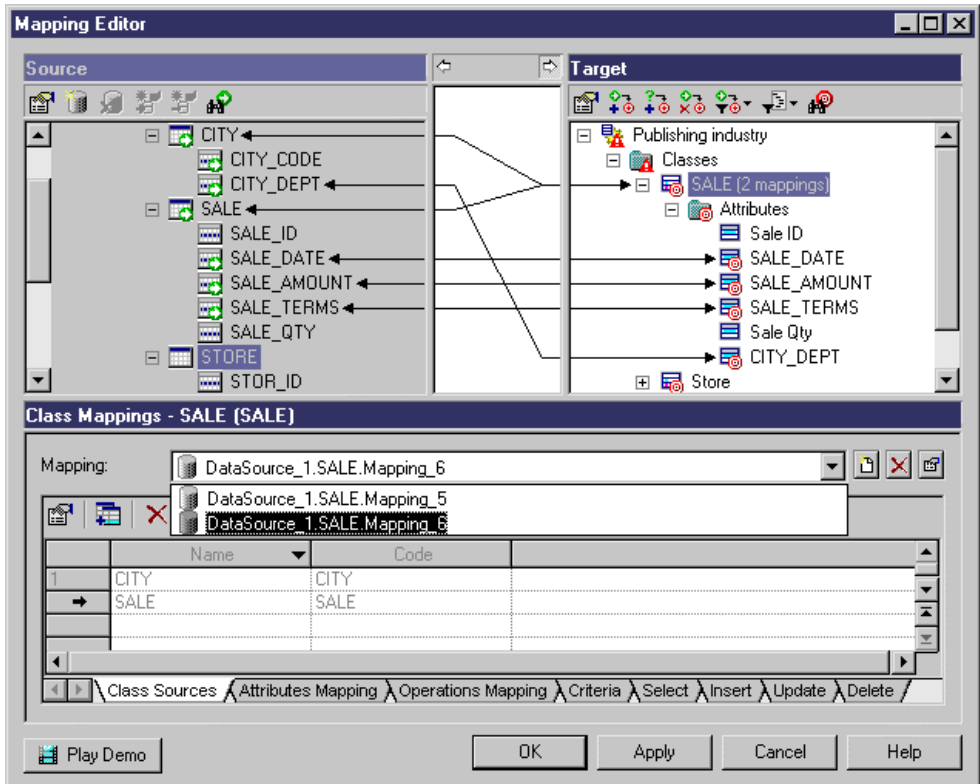
All the mappings for a selected object in the Target pane are displayed in the Mapping list located in the lower part of the Mapping Editor dialog box. You can select one and modify it using the different queries sub-tabs displayed in the lower part of the Mappings pane, or by clicking the Properties tool to the right of the Mapping list to open its property sheet (see *Object Mappings Properties* on page 493).

The following tools help you manage mappings for a data source in the Mappings pane:

Tool	Description
	Create Mapping - Creates a mapping for a data source that you can further define using the Mapped to sub-tab or queries sub-tabs located at the bottom part of the dialog box (PDM data source only). By default, you are prompted to create a data source if no data source exist for the model when you click the Create Mapping tool. Then, you have to declare one or more source models in order to select source objects from a selection list. If sub-objects match, they are automatically mapped. Each time you click the Create Mapping tool for the current object, it creates a new mapping for the same data source if only one data source exists. Otherwise you have to select one from the list of available data sources to proceed.
	Delete Current Mapping - Deletes the current mapping for the data source. The link that materializes the mapping between the Source and the Target panes is also deleted.
	Properties - Opens the property sheet of the current mapping.

Use Case Scenario

You have a target object but you do not know its sources yet. You create a mapping using the Create Mapping tool. The mapping object can be compared to a container for the target object and its source objects. A selection dialog box opens to let you select one or more sources for your target object that is then displayed in the *Object* Sources sub-tab in the Mappings pane.

Then you have to select the sub-objects (source and target) to map using the sub-object column for the target and the Mapped to column for the source in the *Sub-Object* mapping sub-tab.

Then you can configure SQL queries using the Criteria, Select, Insert, Update and Delete sub-tab, as a mapping consists in performing modifications in the data source using SQL.

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.

1. On the Data Source Identification page, enter a name for the data source, and then specify an access type. The selected type controls which types of queries (for PDM data sources) are available in the Mappings pane, and the existence of forward and reverse 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 (Source > Target) mappings to a read-only data source, and such sources are not displayed during reverse mapping (Target > Source). For more information about forward and reverse mappings, see *Creating forward and reverse mappings* on page 485.

- Write-Only - Enables Criteria, Insert, Update and Delete queries. You can only create reverse mappings to a write-only data source, and such sources are not displayed during forward mapping

2. Select the type of the model to use as the data source.
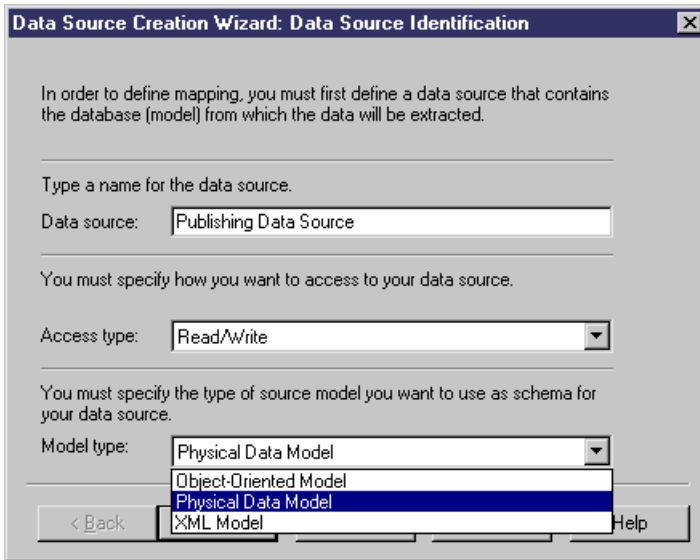
3. Click **Next** to go to the Source Model Selection page.
4. Select one or more models from the list of open models in the workspace, and then:

- 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 create additional data sources in the Mapping Editor, click the **Create Data Source** tool. To add additional models to a data source, click the **Add Model to Data Source** tool.

Data Source Properties

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	Type of model(s) being used as data source. See *Chapter 16, Object Mappings* on page 469
Access type	Controls which types of queries (for PDM data sources) are available in the Mappings property sheet or in the Mappings pane of the Mapping Editor, and also controls the existence of forward and reverse mappings. For more information about forward and reverse mappings, see *Creating a data source* on page 491.
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* on 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.

Object Mappings Properties

Click the **Properties** button to the right of the **Mapping** field on the **Mapping** tab of an object property sheet or in the Mappings pane of the Mapping Editor to open the mapping property sheet. The tabs available for a particular mapping depend on the objects being mapped.

The **General** tab contains the following properties:

Property	Description
Data Source	Name of the data source where data is extracted.
Target	Name of the target object that owns the mapping.
Parent	[Sub-object only] Specifies the name of the parent object.
Name	Specifies the name of the mapping.
Code	Specifies the technical name of the mapping.
Comment	Descriptive comment for the mapping.
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 editor (see *Modifying the Default Mapping Syntax* on page 486). To 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.

Tool	Description
	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*).

Creating Mappings from an Object's Property Sheet

You can create mappings individually for a given object from the **Mapping** tab of its property sheet.

The Mapping tab is equivalent to the Mappings pane in the Mapping Editor (see *The Mapping Editor* on page 476).

1. Open the property sheet of the object and click the **Mapping** tab.

2. Click the **Create Mapping** tool to the right of the **Mapping** list to create a 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* on page 491).
   - 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 name and code of the object are displayed in the Object Sources sub-tab and a mapping is created in the Mapping list.
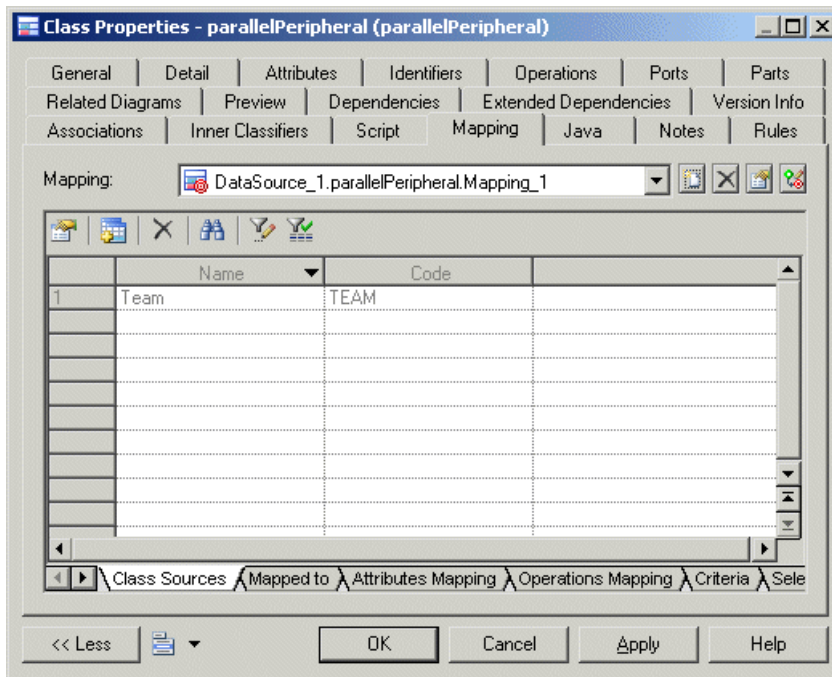
4. [optional] Click the **Add Objects** tool to select more source objects to map with the object.

5. Click the **Sub-Object Mapping** sub-tab and add or remove mappings between sub-objects (see *Object Mappings Properties* on page 493)

   To automatically generate mappings for sub-objects that share a name, click the **Generate Mapping** tool.

6. [optional] Review the other mapping tabs (see *Object Mappings Properties* on page 493) and edit them as appropriate.

7. [optional] Click the **Launch Mapping Editor** tool to review your mappings in the Mapping Editor.

Modifying the Default Mapping of a Sub-object

You can modify the default mapping created for a sub-object. The Attribute Mappings property sheet can be used to fine-tune the mapping between a sub-object in the target object and sub-objects in data source objects.

In the "*Mapped to*" box, you can see the attribute expression. By default, sub-objects used in this expression are prefixed by their parent object. You can customize the content of the "Mapped to" box by inserting comments manually. You can also click the Ellipsis button and use the SQL Editor (for PDM data source) or the Object Expression Editor (for other data sources) to modify the default mapping syntax. To recover the default sub-object expression, click the User-defined tool.

In the *Sources* tab, it is possible to select several sub-objects and map them with the target sub-object. To do so, you have to use the Add Sources tool to select sub-objects from the list of sub-objects belonging to the source objects mapped to the target object. When you add sub-objects from the Sources tab, and you have not modified the sub-object expression, the content of the "Mapped to" box in the General tab is updated.

1. In the Mapping tab, select the Attributes Mapping sub-tab to display the corresponding tab.

2. Select a sub-object in the list and click the Properties tool to display the Attribute Mappings Properties dialog box.

3. Click the Sources tab and click the Add Sources tool to select source sub-objects from a selection dialog box and click OK.

    The sub-objects appear in the Sources list.

4. Click the General tab to display the corresponding tab. The selected sub-objects appear in the "Mapped to" box.
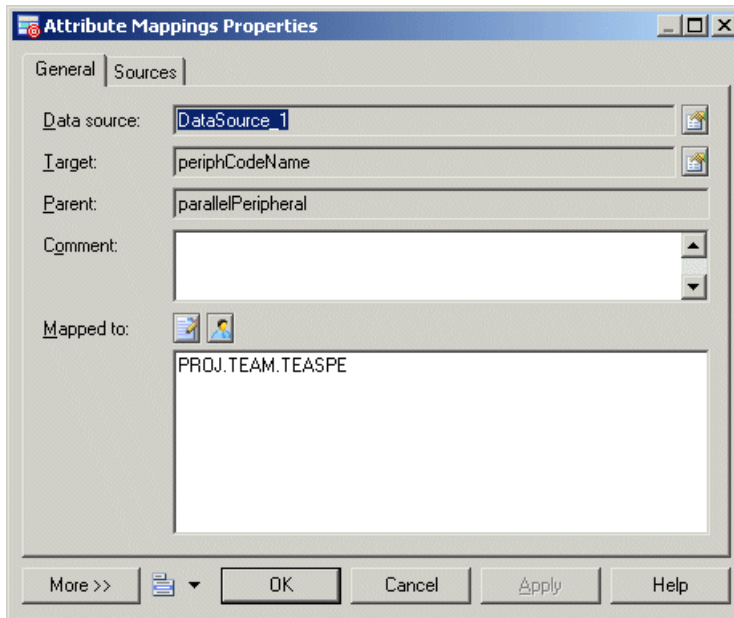
5. Click OK.

CHAPTER 17   **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.

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.

   Note: This feature is also available in the repository (see *Impact and Lineage Analysis in the Repository* on page 316) and the PowerDesigner Portal (see *Impact and Lineage Analysis in the PowerDesigner Portal* on page 335)

   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):
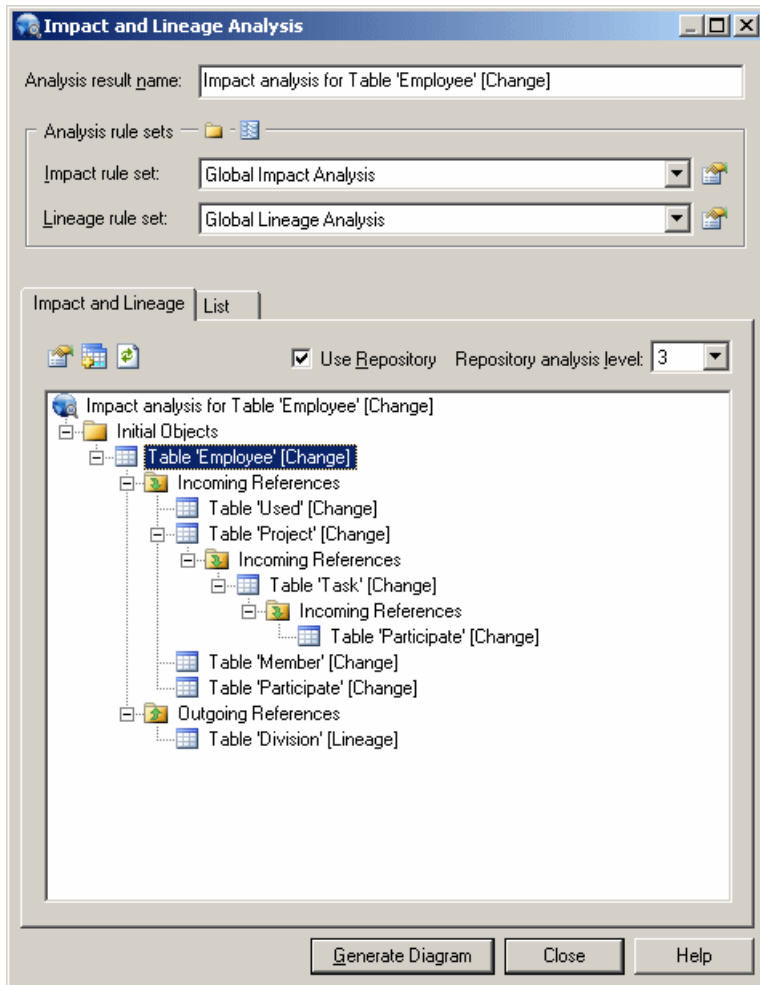
2. [optional] Select **Use Repository** and specify a **Repository analysis level** to search for additional impacted or lineage objects in the repository. If you are not currently connected to the repository, the **Connect** dialog opens to allow you to do so.

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.
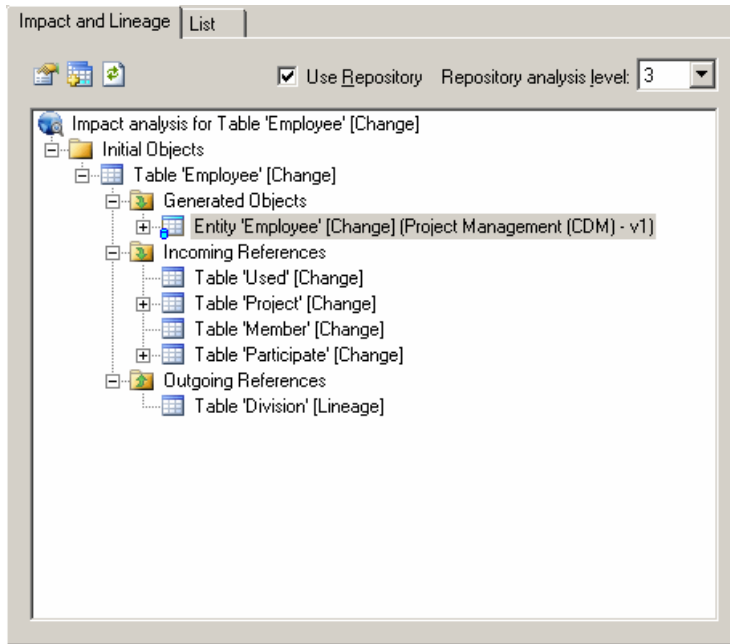
3. [optional] Enter a name for your analysis result, which is then used as the name of any generated model.

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.

   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 *Working with Analysis Rule Sets* on page 512).

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:
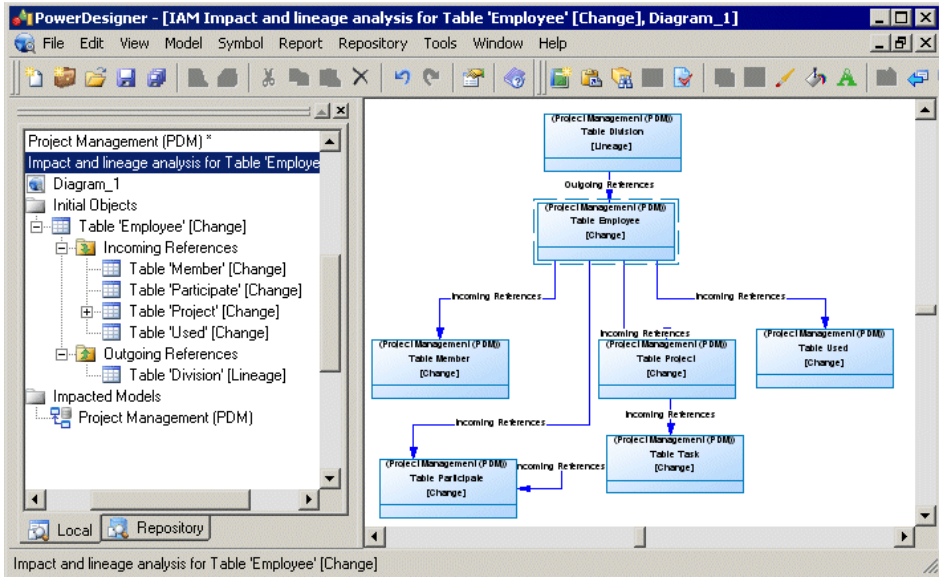
Reviewing and Modifying an Analysis

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).

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.
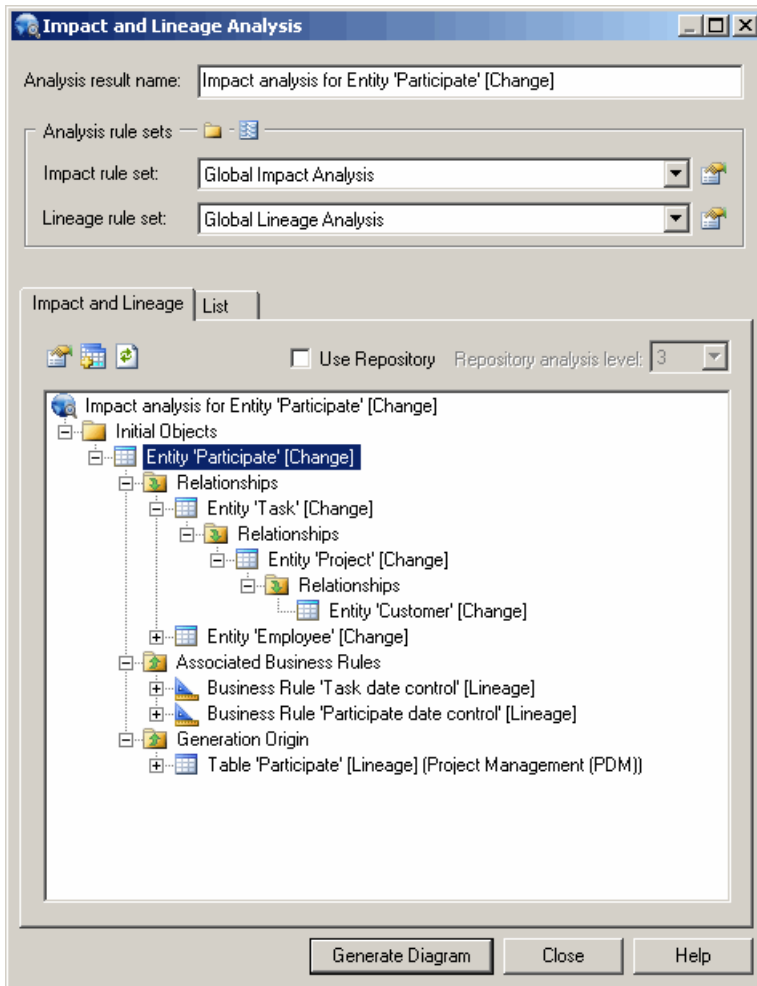
The following tools are available on this tab:

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.

> **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* on page 506).

- 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.

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.
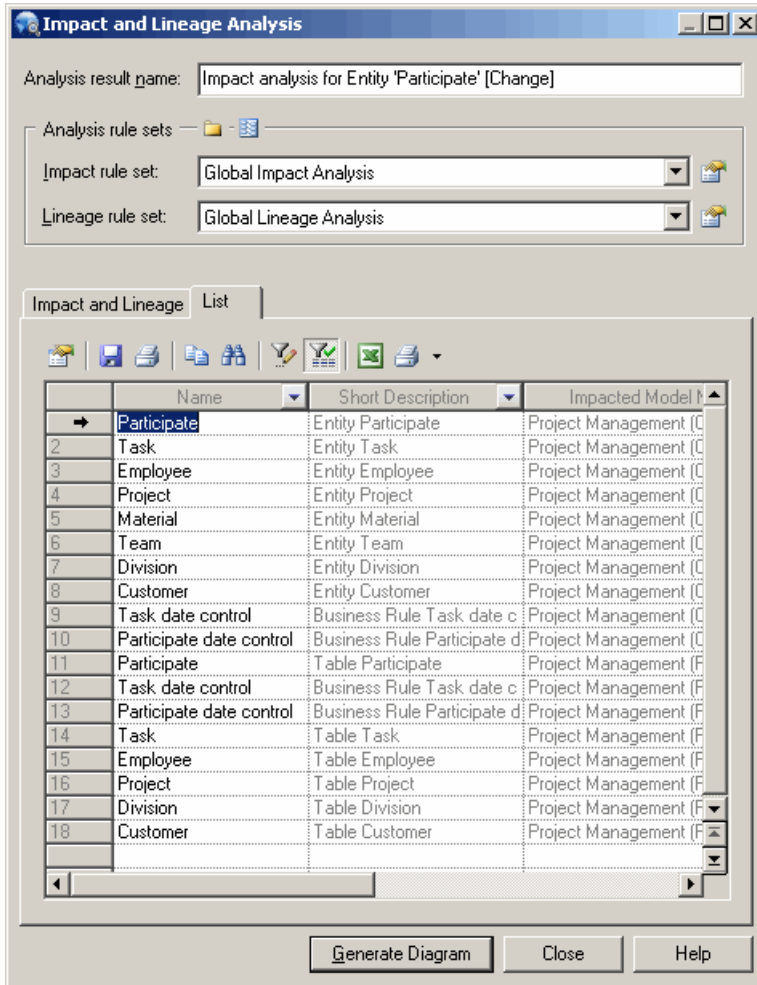
The following tools are available on the preview **List** tab:

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.

Tool	Description
🖨	Print – Prints the list.
📋	Copy – Copies the list in RTF or CSV format.
🔍	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* on page 120).
🔽	Enable/Disable Filter - Enables or disables the filter on the list.

Controlling the Display of Object Collections

You cannarrow or widen the analysis by changing the collections of impact and lineage objects that appear.

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:
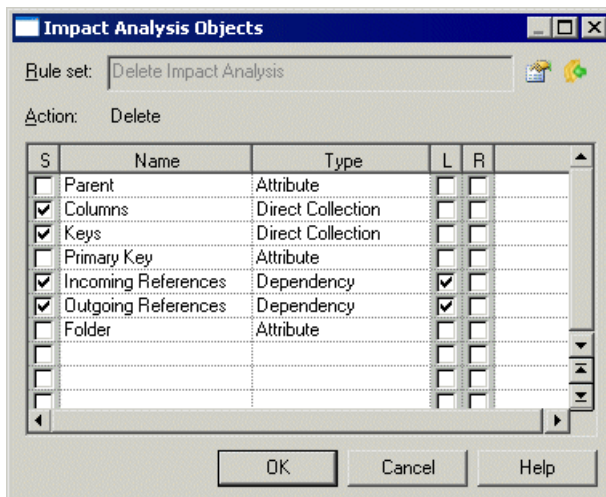
2. Select or unselect the following check boxes to control the display of collections:

   • **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 *Working with Analysis Rule Sets* on page 512).
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.

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.
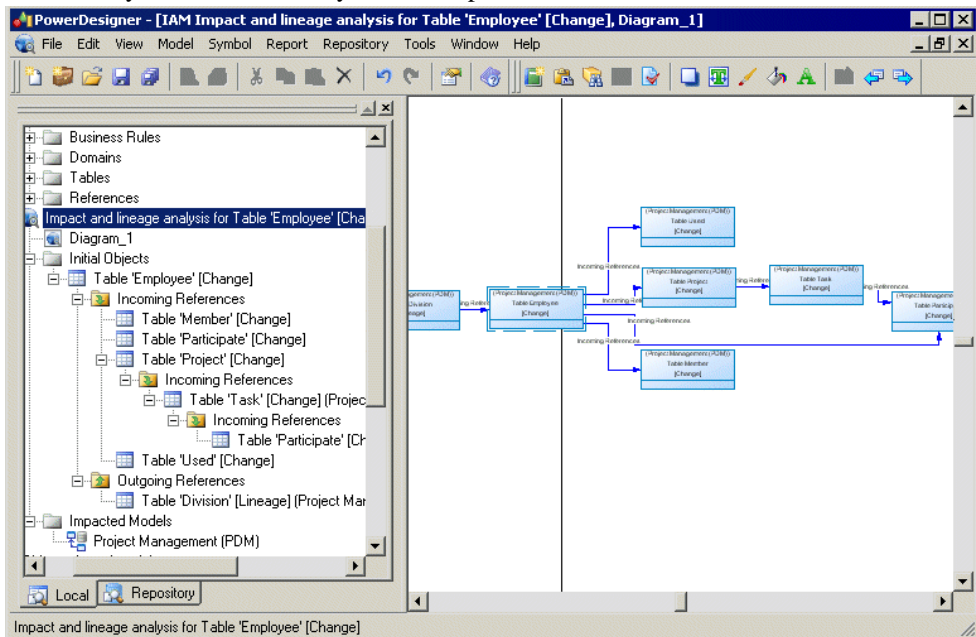
In the above example, the `Employee` table shows influencing (lineage) objects on the left and dependent (impacted) objects on the right. Links between the objects represent the analysis rules that propagate actions on objects.

The following types of objects are available in an IAM:

Symbol	Description
(DesignUseCaseTutorial) Use Case Create Use Cases [Delete]	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**.
(Project Management (PDM)) Table Division [Lineage]	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**.

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**.

  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* on page 506).

- 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**.

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.

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.

Preference	Description
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* on page 217.

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.

For detailed information about comparing models, see *Chapter 7, Comparing and Merging Models* on page 275.

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.

You can print the analysis in any of the following ways:

- [from Preview] Click the **List** tab (see *Viewing the Preview as a List* on page 505) and select the **Print** tool.
- [from the IAM] Select **File > Print** to print the diagram (see *Printing Diagrams* on page 179).
- [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* on page 236).

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* on page 148.

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* on page 305) 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:
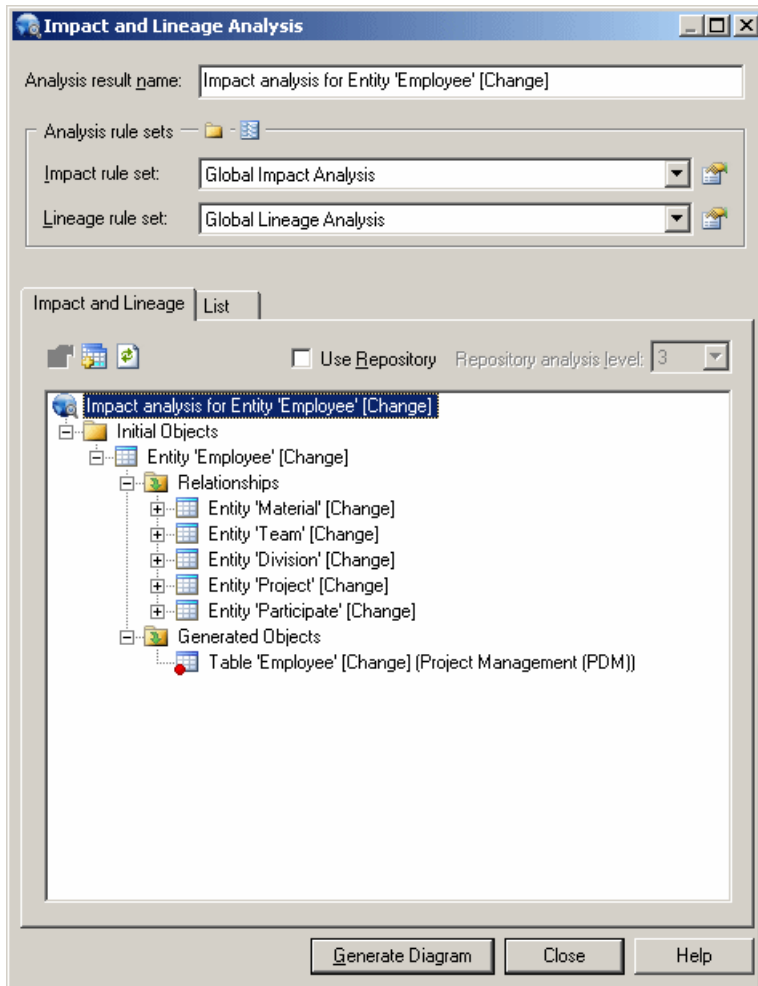
Working with Analysis Rule Sets

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:

```
    Table 'Employee' [Delete]
    └─ Columns
        ├─ Column 'Employee.Employee number' [Delete]
        ├─ Column 'Employee.Emp_Employee number' [Delete]
        ├─ Column 'Employee.Division number' [Delete]
        ├─ Column 'Employee.First name' [Delete]
        ├─ Column 'Employee.Last name' [Delete]
        ├─ Column 'Employee.Employee function' [Delete]
        └─ Column 'Employee.Employee salary' [Delete]
    └─ Keys
        └─ Key 'Employee.Idtf_2' [Delete]
    └─ Indexes
        ├─ Index 'Employee.BELONGS_TO_FK' [Delete]
        ├─ Index 'Employee.CHIEF_FK' [Delete]
        └─ Index 'Employee.EMPLOYEE_PK' [Delete]
    └─ Incoming References
    └─ Outgoing References
```

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

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:
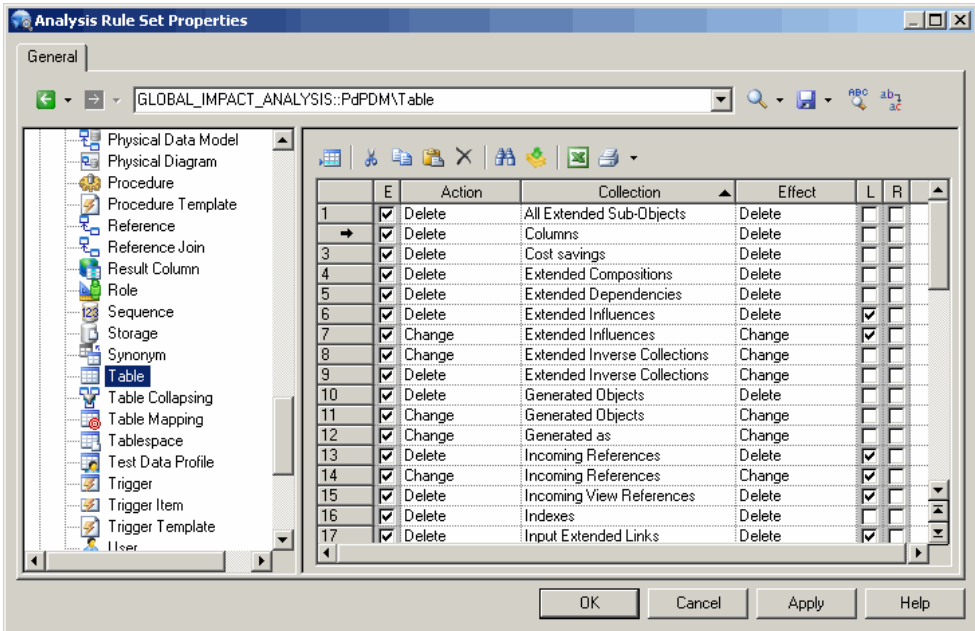
The following properties are available for each analysis rule:

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* on page 515). 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* on page 515). 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).

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.

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.

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.

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.

1. In the Resource Editor, open an impact analysis rule set (see *Working with Analysis Rule Sets* on page 512).

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.
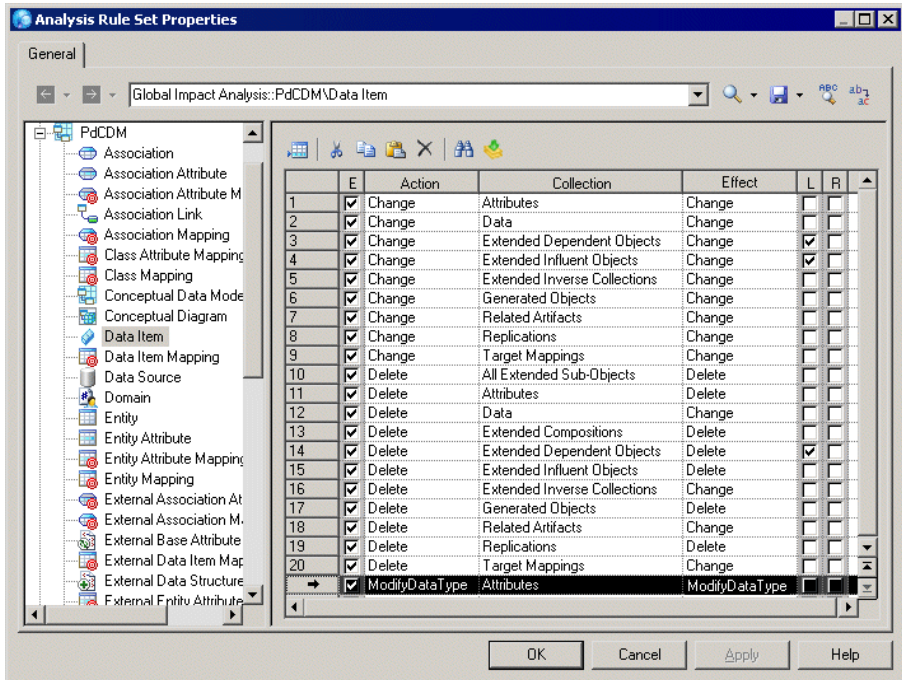
4. Click **OK** to save the rule set and close the Resource Editor.

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:
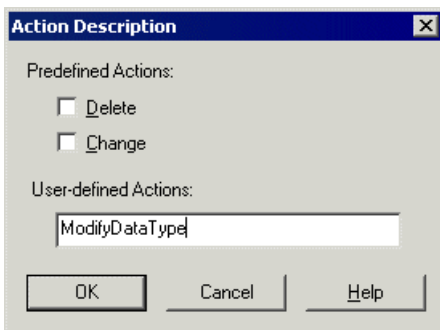
You can enter several actions in a single word using semicolons as separators.

7. Click **OK** to close the dialog box.

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:
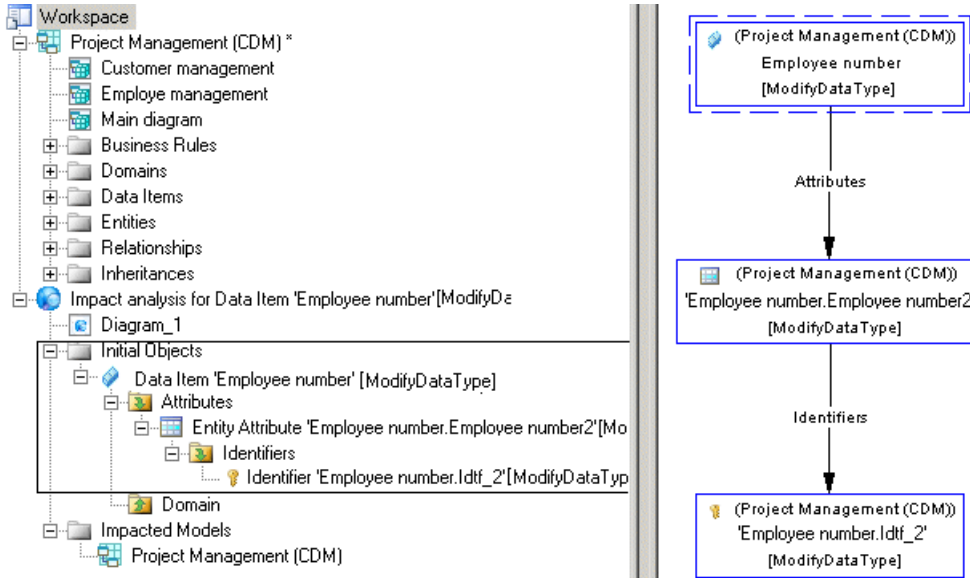
Checking Rule Sets into the Repository

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.

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*.

1. Select **Tools > Resources > Impact and Lineage Analysis Rule Sets** and create your custom rule set (see *Creating an Analysis Rule Set* on page 515).

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* on page 297), 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* on page 316).

Note: All impact and lineage analyses launched from your local workspace, even those that use the repository, are performed with your local rule sets.

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.

PART III

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*.

CHAPTER 18   **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 rights, permissions, and profiles.

By default, the repository has only one user, ADMIN, with all available rights on everything in the repository. This administrator is responsible for creating other users and groups. When you create a user, she is automacially granted the Connect right (see *Granting Rights to Users and Groups* on page 524) and assigned to the PUBLIC group which has, by default, Read permission (see *Granting Access Permissions on Repository Items* on page 528) on the repository root.

The following examples provide some suggestions for the rights and permissions you will want to grant to your development team.

Example 1: Granting Rights to Groups and Users
The following rights are granted to the groups and users who will work on a modeling project:

Groups or Users	Rights
Repository Administrator	Connect, Manage All Documents, Manage Users, and Manage Repository (to manage the repository and users)
Data Administrator	Connect, Manage All Documents, Manage Users, and Manage Repository (to manage repository data)
Project leader	Connect, Freeze Versions, Lock Versions, Manage Branches, and Manage Configurations
Developers	Connect, Freeze Versions, and Lock Versions
Designers	Connect, Freeze Versions, and Lock Versions

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

Example 2: Granting Permissions on Folders and Documents
The Y2K folder includes two sub-folders, Data and Specs. The Data sub-folder contains two models, Firstdraft (PDM) and Classes (OOM). The Specs sub-folder contains Overview.doc.
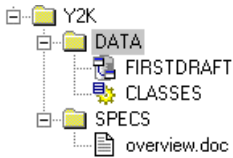
The following permissions are assigned to the groups and users working on the project:

Folder	Project leader (user)	Development leader (user)	Developer (group)	Design leader (user)	Designer (group)
Y2K	Full	Read	Read	Read	Read
Data	Full	Write	Read	Read	Read
Documents in Data	Full	Read	Write	Read	Read
Specs	Full	Read	Read	Write	Read
Documents in Specs	Full	Read	Read	Read	Write

Example 3: Granting Permissions on Packages
The model FIRSTDRAFT is divided into 2 packages, ANALYSIS and IMPLEMENTATION, which correspond to the different tasks the group of developers has to perform. Each of these packages is, in turn divided into two sub-packages, and one developer has responibility for each sub-package.
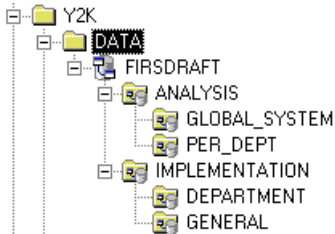
The following permissions are assigned to users:

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

Package	Dev. leader	Dev.1	Dev.2	Dev.3	Dev.4
Department	Full	Read	Read	Write	Read
General	Full	Read	Read	Read	Write

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.

When you create the PowerDesigner repository, only the user ADMIN is created by default, with all available rights on everything in the repository. This administrator is responsible for creating other users and groups as necessary either in the repository or through an external LDAP server (see *Controlling Repository Access with LDAP* on page 529) and granting permissions on the contents of the repository (see *Granting Access Permissions on Repository Items* on page 528).

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

1. Select **Repository > Administration > Users** to open the List of Users.
2. Click the **Add a Row** tool, and type a name in the **Full Name** column.
3. Click the **Properties** button to open the property sheet of the user, and enter a login name, email address, and any other appropriate properties.

   The following properties are on the **General** tab:

Property	Description
Full Name	Specifies the real name of the user.
Login Name	Specifies the name used for logging in to the repository.
Comment	Specifies any additional information about the user.
Email	Specifies the email address of the user. If you have specified an SMTP server (see *Specifying an SMTP Server for Notifications* on page 534) this address will be used to send the password.
External	Specifies that authentication of the user is handled by an LDAP server (see *Controlling Repository Access with LDAP* on page 529). Selecting this check box disables the **Full name** and **Email** fields.

Property	Description
Status	Specifies the status of the user, which can be one of: • Active - the user can access the repository. • Blocked - the user has violated one of the rules of the password policy (see *Defining a Password Policy* on page 533), and cannot access the repository until an administrator unblocks her account (see *Unblocking Blocked Users* on page 526). • Inactive - the user is no longer active in the repository (see *Deactivating Users* on page 525) and may not connect. The user remains in the List of Users and can be reactivated at any time.

4. If you have not enabled delivery of passwords by email, click the **Reset Password** button and then select the **Show** checkbox to display the password and note it for transmission to the user.

5. [optional] Click the **Groups** tab and add the user to any appropriate groups (see *Adding Users and Groups to a Group* on page 527).

   By default, all users are added to the Public group, from which they inherit the Read permission on all the contents of the repository (see *Granting Access Permissions on Repository Items* on page 528).

6. [optional] Click the **Rights** tab and specify any appropriate rights for the user (see *Granting Rights to Users and Groups* on page 524). By default, all users are granted the Connect right, which entitles them to connect to the repository.

7. [optional] Click the **Profiles** tab and associate the user with appropriate object permissions, UI, and general preferences profiles (see *Applying Profiles to Users and Groups* on page 552).

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* on page 533). When they first log in they will be required to change the temporary password.

Granting Rights to Users and Groups

A new user has only the Connect right assigned by default and belongs only to the PUBLIC group, which has no rights. In order for the user to be able to do anything, the repository administrator must grant rights to her either directly or by adding her to other groups.

For examples of combining rights to define roles, see *Chapter 18, Controlling Repository Access* on page 521.

Note: The rights a user or group receive when they become members of a group (see *Adding Users and Groups to a Group* on page 527) are cumulative. For example, a user with the Manage Branches right will not lose this right when he becomes a member of a group that has only the Freeze Versions and Lock Versions rights.

1. Select **Repository > Administration > Users** (or **Repository > Administration > Groups**), select the appropriate user or group in the list, and click the **Properties** tool.

2. Click the **Rights** tab and select the check boxes corresponding to the rights you want to assign. The following rights are available:
   - `Connect` - To connect to the repository.
   - `Freeze Versions` - To freeze document versions (see *Freezing Document Versions* on page 318).
   - `Lock Versions` - To lock document versions (see *Locking Document Versions* on page 320).
   - `Edit PowerDesigner Portal Objects` - To edit objects in a Web browser (see *Editing Object Comments and Descriptions* on page 336).
   - `Manage Branches` - To create branches (see *Branching Version Trees* on page 325).
   - `Manage Configurations` - To create, modify, and delete configurations (see *Grouping Document Versions in a Configuration* on page 322).
   - `Manage All Documents` - To perform any action on any document version.

     Note: This right is typically granted to data administrators, and implicitly includes `Full` permission on all repository documents to allow them to check in, freeze, lock, and delete any document.

   - `Manage Users` - To create, modify and delete repository users and groups, grant rights to users and groups, and add users or groups to a group.
   - `Manage Repository` - To create, upgrade, and delete the repository database.

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

Note: User rights are associated with document permissions (see *Granting Access Permissions on Repository Items* on page 528) to define the actions a user can effectively perform on a document. Some user rights are automatically implied by permissions:
- A user with the `Write` permission on a repository document automatically has the `Lock Versions` and `Freeze Versions` rights for it.
- A user with the `Full` permission on a document can unlock or unfreeze a version he did not lock or freeze.

Deactivating Users

The repository administrator or a user with the Manage Users right can deactivate users. An inactive user cannot connect to the repository, but the information about his checkins and other repository actions remains available to other users. An inactive user can be reactivated at any time.

Warning! A user cannot deactivate himself, even if he has the Manage Users right.

1. Select **Repository > Administration > Users** to display the List of Users dialog box.

2. Select the user in the list and then select the [D]eactivate check box.

   The user remains in the list and can be reactivated at any time by deselecting the [D]eactivate check box. While he is inactive, the user may not connect to the repository.

Unblocking Blocked Users

The repository administrator or a user with the Manage Users right can unblock users blocked for password policy violations.

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

2. Select the user in the list and click the **Properties** tool to open his property sheet.

3. Click the **Reset Password** button.

4. If you have not enabled delivery of passwords by email (see *Specifying an SMTP Server for Notifications* on page 534), select the **Show** checkbox to display the password and note it for transmission to the user.

5. Click **OK** to assign the temporary password to 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* on page 533). When they first log in they will be required to change the temporary password.

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 rights and permissions and the use of profiles. 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.

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

- Administrators, [ADMN] - has all available rights.
- All users [PUBLIC] - has the Connect right and Read permission on the repository root (see *Granting Access Permissions on Repository Items* on page 528). All users belong to this group and can thus connect to the repository and check out any document.
- External users [EXTERNAL] - has the Connect right. Remote LDAP users (see *Controlling Repository Access with LDAP* on page 529) are automatically added to this group when they connect for the first time.

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

2. Click the **Add a Row** tool and enter a name for the group.

3. Click the **Properties** tool to open the property sheet of the new group. The following properties are available on the **General** tab:

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.

4. Click the **Members** tab and add any appropriate users and groups to the group (see *Adding Users and Groups to a Group* on page 527).

5. Click the **Parent Groups** tab and add the group to any appropriate groups (see *Adding Users and Groups to a Group* on page 527).

6. Click the **Rights** tab and specify any appropriate rights for the group (see *Granting Rights to Users and Groups* on page 524). By default, groups do not have any rights.

7. 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* on page 552).

8. Click **OK** to complete the creation of the group.

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.

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.

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.

Deleting a Group

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

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 rights 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.

Granting Access Permissions on Repository Items

The repository administrator or a user with `Full` permission on a document can grant permissions on it from the **Permissions** tab of its repository property sheet. You can grant permissions on the repository root, folders, PowerDesigner models and model packages, and external application files, but not on individual model diagrams or objects.

In order for a user to check documents out of and into the repository, she must have 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).
  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).

Note: Object permissions should be viewed in conjunction with the rights granted to users or groups (see *Granting Rights to Users and Groups* on page 524). For examples of granting rights and permissions, see *Chapter 18, Controlling Repository Access* on page 521.

1. Right-click the item in the Repository Browser, and select **Permissions** to open the item's property sheet at the **Permissions** tab.

2. Click the **Add** tool 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. The following permissions are available:

   - `List` - To view the item in the browser and in search results, and display read-only property sheets.

     Note: Users who do not have even this permission on a document or folder cannot even see it in the repository Browser.

   - `Read` - To have all the List permissions, and also compare documents, and check the document (or folder contents) out from the repository.

   - `Write` - To have all the Read permissions, and also check in, freeze, and lock document versions.

   - `Full` - To have all the Write permissions, and also manage permissions granted to users or groups and remove locks on documents.

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 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 a you grant write permissions to the `Major Project` folder, to the `Development Team 2` group, then they will not automatically be granted write access on its 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.

   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.

Controlling Repository Access with LDAP

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

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

PowerDesigner's repository LDAP integration provides only authentication. Authorization is still managed via the permissions set within the repository environment. Initially, members of the `External users` group have only the Connect right granted, and members of Public have read access to everything in the repository. The administrator will grant other rights and permissions as appropriate (see *Granting Access Permissions on Repository Items* on page 528).

To retain finer control of write permission on repository documents and to have everything in place before LDAP users connect to the repository, the administrator can manually create accounts and assign permissions for them before they connect (see *Pre-Configuring LDAP User Permissions* on page 533).

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

General LDAP Parameters

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

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

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

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

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

Role Parameters

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

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

Pre-Configuring LDAP User Permissions

If LDAP-authenticated users connect to the repository without any preparation, they will automatically create repository accounts, but may not have appropriate permissions. You can manually create accounts for LDAP-authorized users and assign permissions for them, so that everything is in place for them before they connect for the first time.

1. Select **Repository > Administration > Users** to open the List of Users.
2. Click the **Add a Row** tool, and enter an appropriate name for the user in the **Full Name** column.
3. Enter the user's LDAP login in the **Login Name** column, and select the checkbox in the **X** column to specify that he is an externally-validated user.
4. Click the **Properties** tool to open the property sheet of the user and then click the **Groups** tab to add him to any appropriate groups (see *Adding Users and Groups to a Group* on page 527).

Defining a Password Policy

The repository administrator is responsible for defining a password policy to ensure that passwords are sufficiently secure and changed at appropriate intervals. To modify the

password policy, connect to the repository and select **Repository > Administration > Password Policy**.

The password policy allows you to set various parameters for password format and duration of validity:

Setting	Description
Password length	Specifies the minimum and maximum permitted length of passwords. This option cannot be disabled. The minimum 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.

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.

Specifying an SMTP Server for Notifications

The repository administrator can automate the sending of passwords to users by specifying an SMTP server for PowerDesigner to use. To specify an SMTP server, connect to the repository and select **Repository > Administration > SMTP**.

If an SMTP server is not specified, the administrator must distribute passwords manually. The SMTP server can also be used to send change notifications from the PowerDesigner Portal (see *Subscribing to Change Notifications* on page 334).

The following settings are available for configuring PowerDesigner's connection to your SMTP server:

Setting	Description
SMTP host	Specifies the host name of the SMTP server used to send mail.
SMTP port	Specifies the port number of the SMTP server used to send mail.
Sender's email address	Specifies the email address from which to send 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 instruct PowerDesigner to use Secure Password Authentication (SPA).

Auditing Repository Activities

Users with the Manage All Documents right can use the List of Activities to audit operations performed on repository documents, analyze user behavior patterns, and highlight activity sequences. Activities are actions that modify repository documents, such as check in, freezing, and deleting.

1. Press Ctrl+Alt+V or select **Repository > Activities** to open the List of Activities.
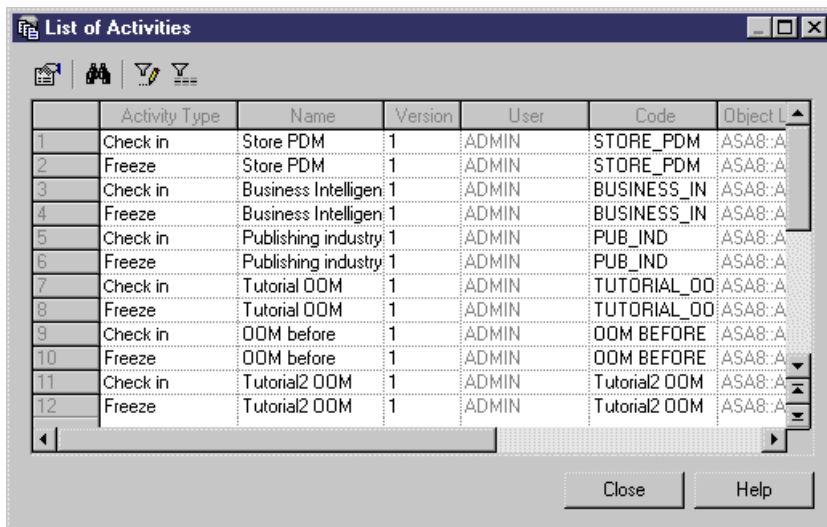
The list displays all the operations on documents stored in the repository. :

2. [optional] Clicking the **Customize Columns and Filter** tool to customize the list. The following columns can be useful for monitoring repository activities:

Column	Description
Activity Type	Specifies the type of action performed on the document, such as check in, freeze or delete. Note that actions, such as lock or unlock, that do not modify the document are not reported in the list of activities.
Name/Code	Specify the name and code of the repository document affected.
Location	Folder in which the document is stored.
Comment	Specifies the comment entered to account for the activity.
Version	Specifies the document version number.
Activity Date	Specifies the date when operation was performed.
User	Specifies the user who performed operation.

3. [optional] 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.

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.

Note: You cannot execute SQL queries when connecting to the repository via the repository proxy.

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.

Obtaining Emergency Access to the Repository

In the event that no administrator is able to log in to a running repository, it is possible to create an emergency administrator account to regain access.

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 > Installing the Repository Proxy*).
2. Launch the Repository Proxy Configuration utility (`pdproxyconf16.exe`).

> **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. Within 15 minutes, 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* on page 523).

7. Select **Repository > Disconnect** to disconnect the emergency user from the repository.

> **Note:** The emergency user is deleted after disconnection.

CHAPTER 19   **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.

Customizing the New Model Dialog

You can create model templates with predefined targets, extensions, and first diagrams to help guide users through model creation. You can organize these templates in categories and share them with your team, supplementing or even replacing the default New Model dialog with these categories.

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. 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.

You can deploy model category sets to users (see *Sharing Resource Files via the Library* on page 565) to ensure that everyone on your team is creating models using the same centralized targets and extensions checked out automatically from the repository.

In the following example, the Information category is selected and provides eight templates to choose from:
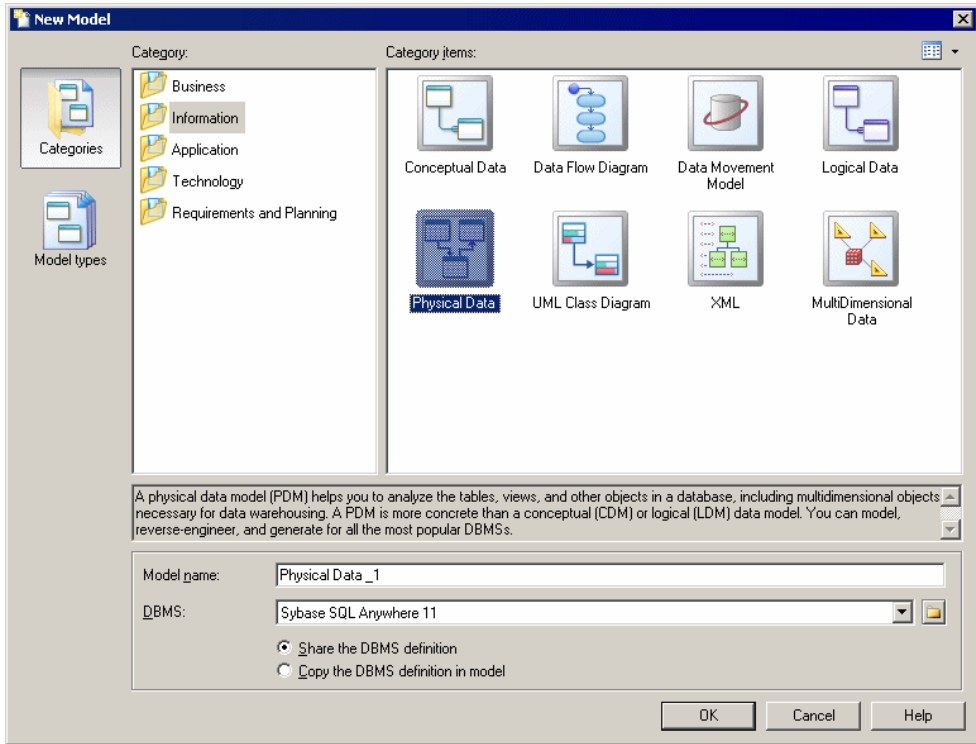
Note: You can also hide model types from certain types of users with object permission profiles (see *Controlling the Availability of Models, Objects, and Properties* on page 546).

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.

1. Select **Tools > General Options**, and click the **Model Creation** category in the left hand pane.
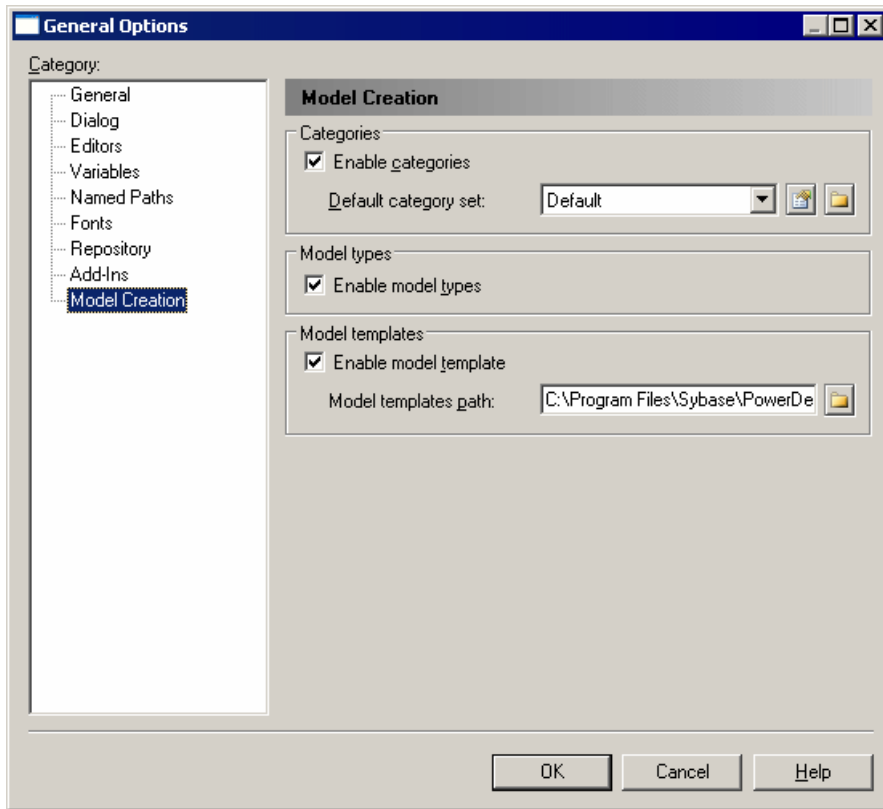
2. Verify that the **Enable categories** check box is selected, and select the category set you want to display from the Default category set list.

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.

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.

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.

   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* on page 543).
   - 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* on page 544)

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.
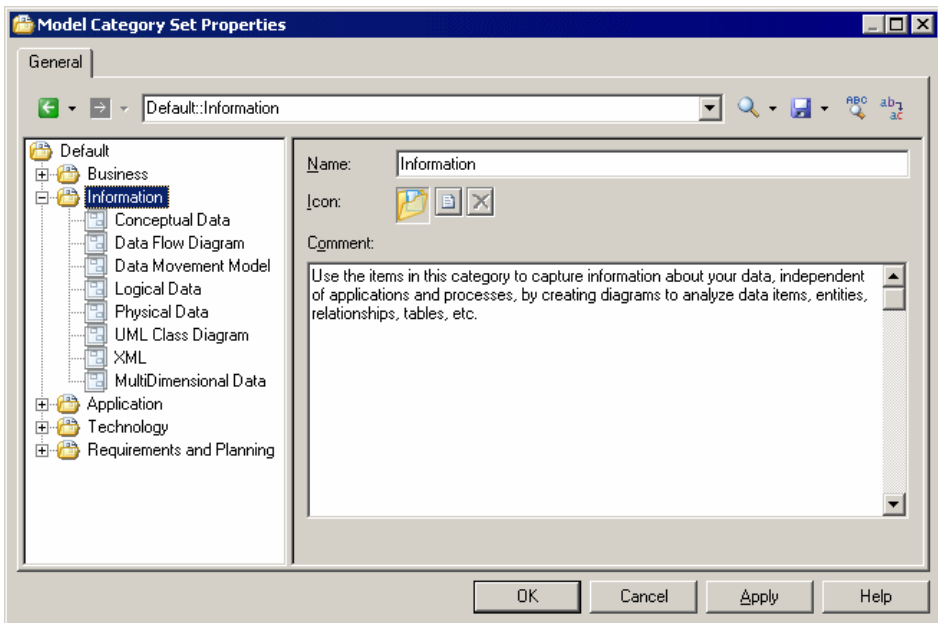
You can test your category set by selecting it in the General Options dialog (see *Selecting a Model Category Set to Display in the New Model Dialog* on page 540) and selecting **File > New Model**.

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.

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* on page 9).

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:
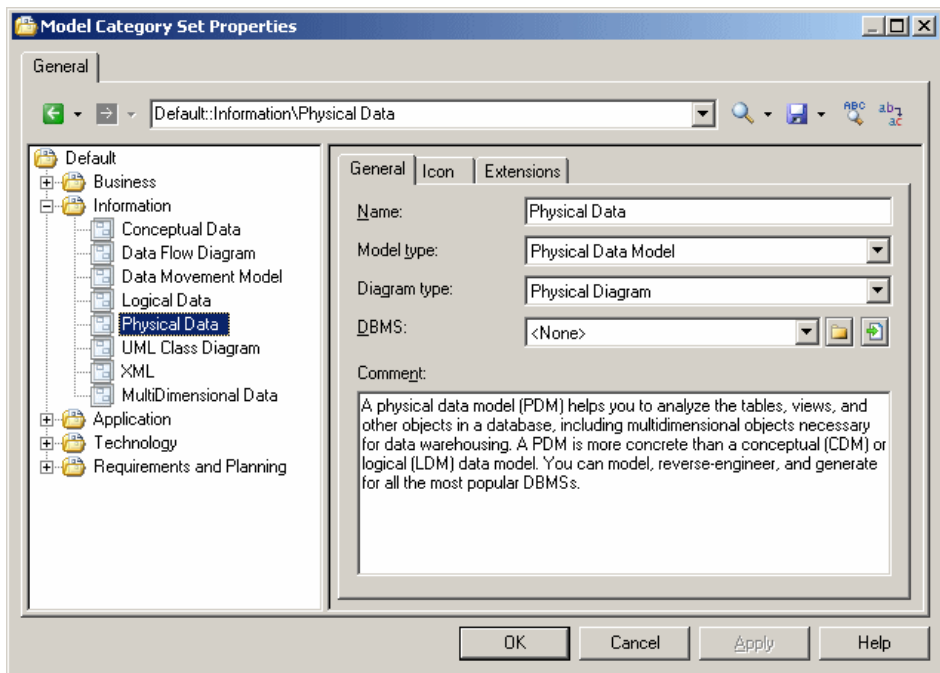
Adding a Model from Template to a Model Category Set

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.

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.

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:
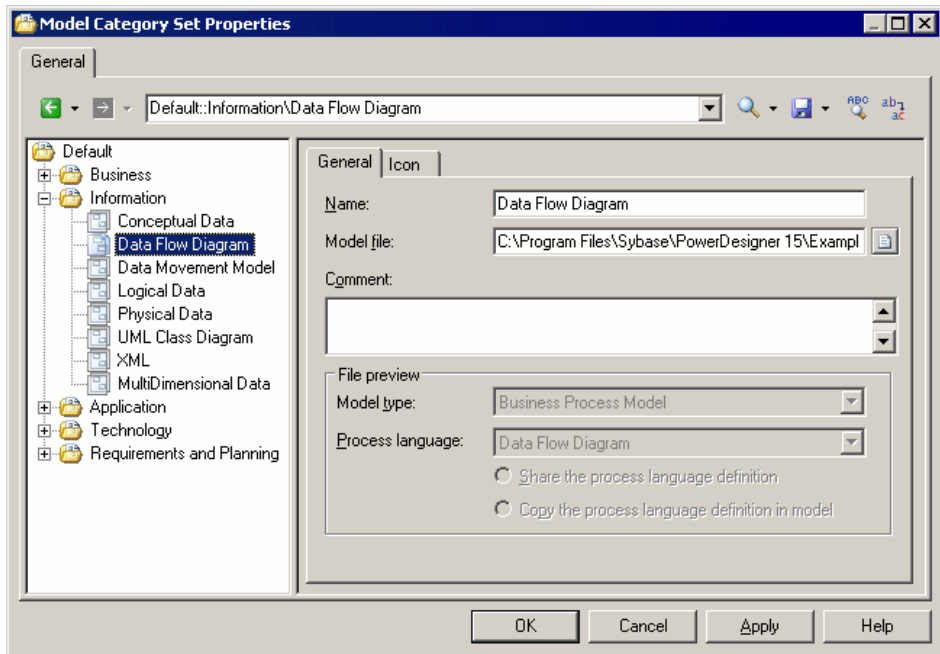
Pushing Model Category Sets to Users

You can deploy your model category sets to the library to make them available to other users.

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.

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 on page 563.

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.

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* on page 411).
- 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* on page 389).

Note: Permissions profiles are enforced (the user cannot access types of models or objects that are hidden from her), but UI and general preference profiles provide only defaults, which can be overridden by the user. The defaults can be reapplied at any time using the **Tools > Apply User Profile** command (see *Applying a User Profile* on page 390).

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.

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* on page 284. You can repeat this step as many times as necessary to import additional profiles.

   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* on page 550).

   ---

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:
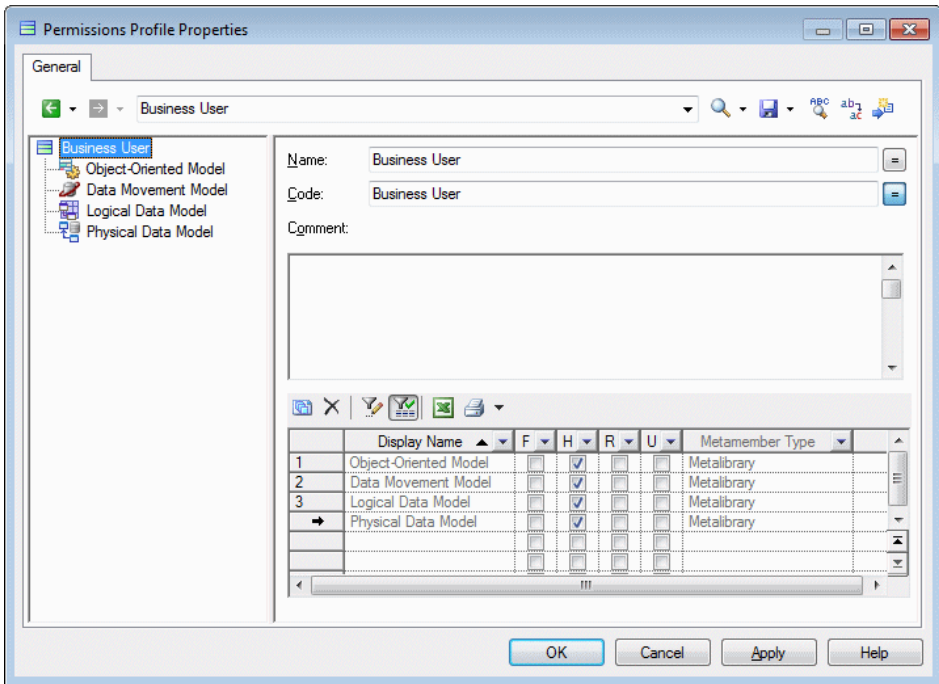
5. To modify the status of object types (metaclasses) for users associated with this profile, right-click the parent module, select **Add Metaclasses to Customize** (or click the equivalent tool in the toolbar), select the metaclasses and click **OK** to add them to the profile. Then select the appropriate 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.

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 Sybase 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:
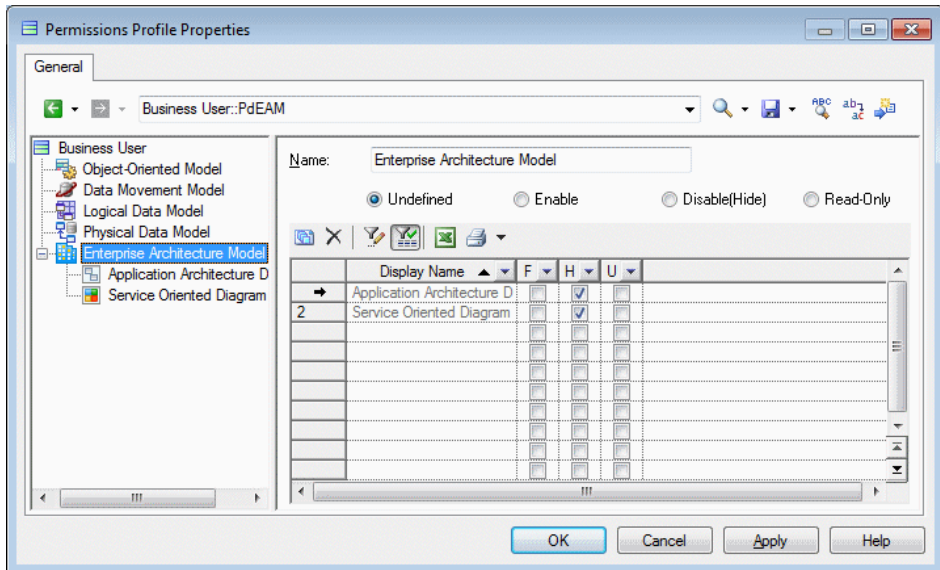
6. To modify the status of object properties (attributes and collections) for users associated with this profile, right-click the parent metaclass and select **Add Meta-attributes to Customize** or **Add Metacollections to Customize** (or click the equivalent tool in the toolbar), select the attributes 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.

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 Sybase 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:
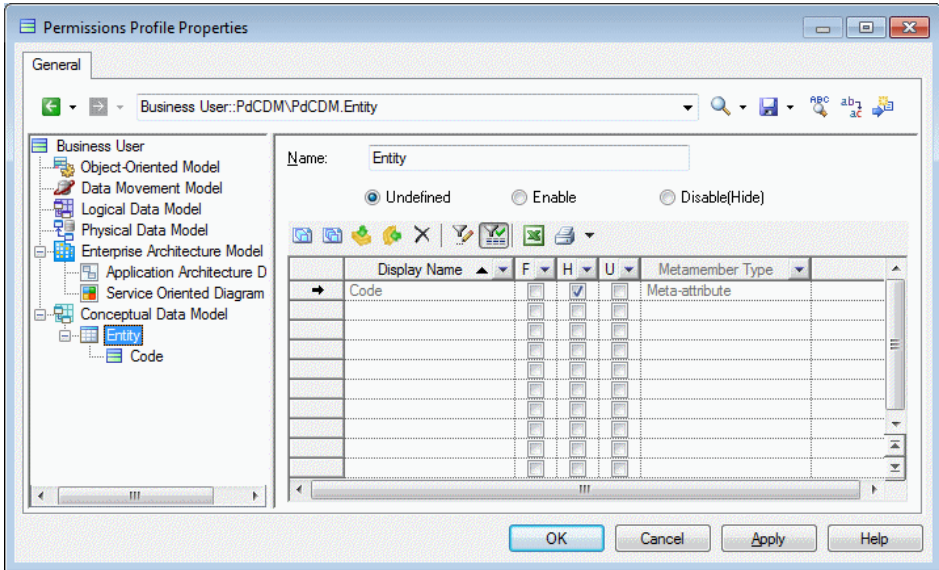
Note: You can propagate customizations of properties to other metaclasses, including to metaclasses in other types of models. Select a metaclass with customized properties and click the **Apply to All Metaclasses in This Module** or **Apply to All Metaclasses in All Modules** tool or right-click a meta-attribute 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.

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)

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.

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 User Interface 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* on page 411).
4. Click **OK** to save the profile in the repository.

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.

1. Connect to the repository and then select **Repository > Administration > General Preference Profiles**.

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 the preferences to copy from an open model (see *Copying Preferences from a Model* on page 393).

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, and then click **OK** to complete the import.

   For detailed information about working with this dialog, see *Merging Models* on page 284. 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* on page 394).

7. Click **OK** to save the profile in the repository.

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.

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* on page 527).

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:
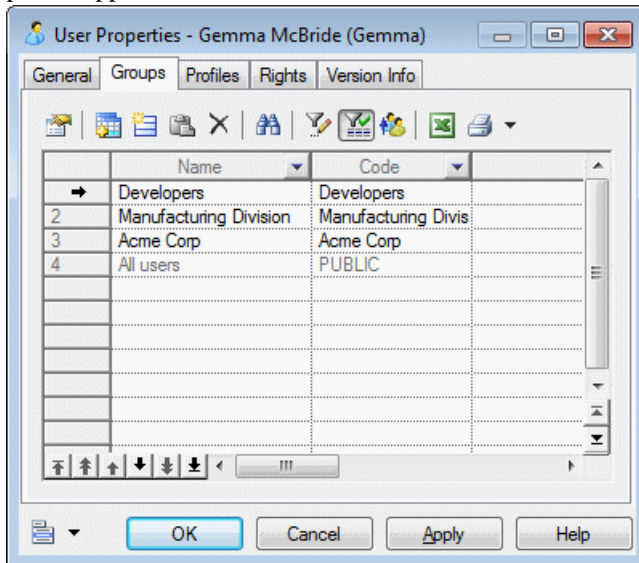
1. Create appropriate object permission, UI preference, and general preference profiles in the repository.
2. Select **Repository > Administration > Users** (or **RepositoryAdministrationGroups**) and select the user or group to whom you want to apply the profile.
3. Click the **Properties** tool to 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**.
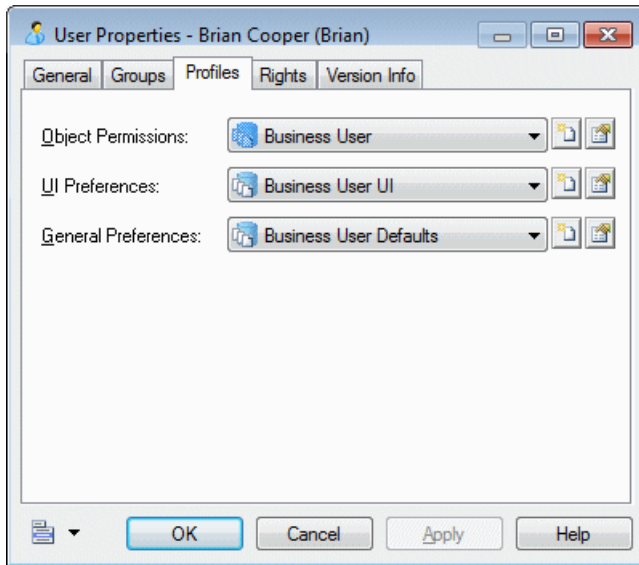
When the user next connects to the repository, the selected profiles will be pushed to their installation of PowerDesigner.

Note: The object permissions profile is automatically applied, but the UI and general preferences profiles are only applied if the user has not customized their interface in any way (or has not previously received customizing profiles). Users can manually apply the UI and general preference profiles at any time by selecting **Tools > Apply User Profile** (see *Applying a User Profile* on page 390).

CHAPTER 20   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.

Deploying an Enterprise Glossary

An administrator can create an enterprise-wide glossary to provide autocompletion for object names, and enable model checks to ensure that only approved terms are used in object names. Once deployed, the glossary is automatically pushed to all users when they connect to the repository, and appears in the **Glossary** tab in the Browser.

For information about using the glossary, see *The Glossary* on page 40.

Creating Terms, Synonyms, and Related Terms

An administrator can define new glossary terms in the **Glossary** tab of the Browser.

Note: You can quickly populate the glossary by performing a mass import of terms from a model (see *Importing Terms from a Model* on page 558) or a CSV or Excel file (*Importing Terms from an Excel or CSV File* on page 559), or from the results of the List of Missing Terms glossary compliance model check (see *Checking Glossary Compliance* on page 153). You can also add terms in a glossary terms list (see *Glossary Terms Lists* on page 560).

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.
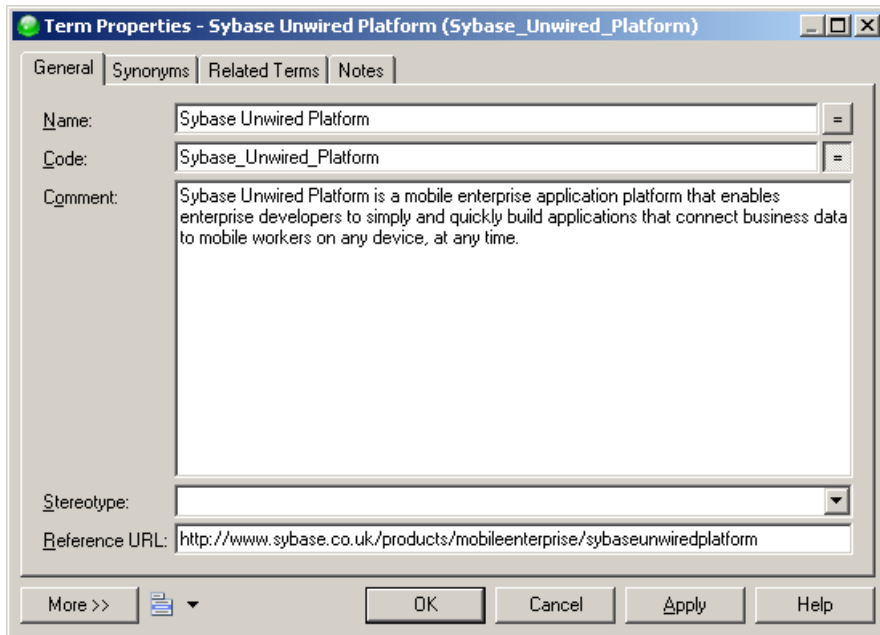
4. [optional] Click the **Synonyms** tab and enter one or more synonyms for the term.

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* on page 153).

5. [optional] Click the **Related Terms** tab and use the **Add Objects** and **Create an Object** tools to associate other terms with the term.

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.

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.

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.

    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 on page 284), which will allow you to review and modify the proposed changes before clicking **OK** to complete the import.

Importing Terms from an Excel or CSV File

You can import terms from an Excel or CSV file to quickly populate your glossary.

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	Syno-nyms	Related Terms	Catego-ry
Client	CLIENT	Specifies...	www.diction-ary.com	Customer, User	Account, Invoice	Sales
Supplier	SUPPLI-ER	Specifies...	www.diction-ary.com	Provider, Seller	Account, Invoice	Purchas-ing
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.

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.

Note: For detailed information about the PowerDesigner Excel Import Wizard, see *Importing Objects from Excel Files* on page 168.

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.

Warning! If you delete a category, you also delete any categories and terms that it contains.

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* on page 560).
- **Dependency Matrix** - to create a dependency matrix (see *Dependency Matrices* on page 186).
- **Business Rule** - to create a business rule (see *Business Rules* on page 142).
- **Report** - to create a report on the glossary (see *Chapter 6, Reports* on page 225).
- **List Report** - to create a list report on terms or other glossary objects (see *Creating a List Report* on page 236).
- **Extension** - to add an extension file (see *Extension Files* on page 415) 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* on page 155)

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.

Tools	Description
✂ 📋 📋	**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* on page 120).
▽✓	**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.
B *I* <u>U</u>	**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* on page 563).

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* on 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.

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.

1. Right-click the Glossary root, and select **Check In** to open the Check In Document dialog (see *Checking Documents into the Repository* on page 292).

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.

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 de-couple name-code synchronization, click to release the = button to the right of the **Code** field.
Filename	Specifies the location of the glossary model file.

Property	Description
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.

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.
External Short-cut 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.
	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* on page 560).

Deploying an Enterprise Library

An administrator or other user with Write permission on the .Library folder can check models, resource files, and external files such as Word documents or PDFs into the library and

have them pushed to repository users. 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.

Note: For information about managing permissions, see *Granting Access Permissions on Repository Items* on page 528.

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.

   Note: You cannot check projects into the library. For information about checking resource files into the library, see *Sharing Resource Files via the Library* on page 565)

2. Right-click the document in the **Local** tab of the Browser and select **Check In**.

3. Select the `Library` folder as the destination for your check in and enter an appropriate comment:
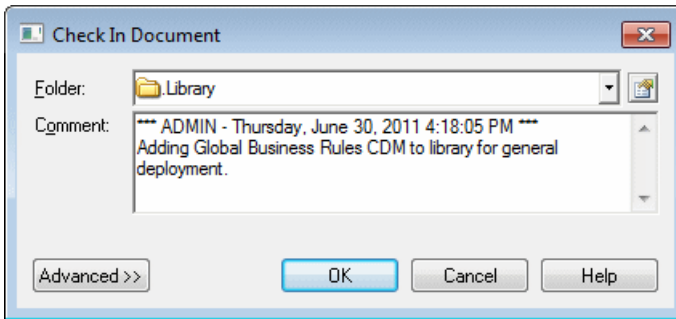
4. Click **OK** to check in the model.

   The document is checked into the library in the repository:
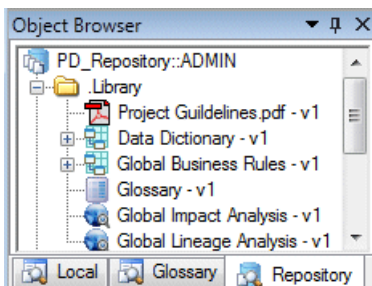
   The next time users connect to the repository (or select **Repository > Synchronize Library**, they will be invited to check the model out into the library in their local workspace (see *Connecting to a Repository* on page 17).

   Note: Branching (see *Branching Version Trees* on page 325) is not supported for the library.

Sharing Resource Files via the Library

You 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 repository documents, and appear automatically in the new model dialog, resource file lists, and anywhere in the interface where a resource file can be selected.

For more information about resource files, see *PowerDesigner Resource Files* on page 413.

1. [recommended] In the repository browser, create one or more folders inside the library folders to hold the resource files.

   You could create a single folder to hold all resource files, or one folder for each type of resource file that you want to share. PowerDesigner recursively scans all folders in the library to populate resource file lists with shared resources.

2. Select **Tools > Resources >** *type* to open the list of resource files containing the file that you want to share.

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, and enter an appropriate comment.

   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 available for selection in lists of resource files.

5. Click **OK** to check in the resource.

   The document is checked into the library in the repository.

   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.

   Note: Resource files are not displayed inside the library folder in the Local workspace browser.

Controlling Glossary and Library Write Permissions

You control access to the Library folder in the same way as any repository folder by granting 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.

Since the library models and other documents are intended to be a reference to guide users, many administrators will want to closely control who can edit them. It is common to create a Library Administrators group with Write permission on the folder. Sometimes a

separate `Glossary Administrators` group is created and `Write` permission on the glossary model is restricted to this subset of administrators.

For further information about permissions, see *Granting Access Permissions on Repository Items* on page 528.

Index

G