SYBASE®

General Features Guide

Sybase® PowerDesigner®

12.5

Windows

Part number: DC38093-01-1250-01

Last modified: April 2007

Contents

PART I

GUI GUIDE

Getting Started with PowerDesigner.

About This Manual

Subject

This book describes the PowerDesigner basic features.

The first part of this feature guide explains how to use and customize the PowerDesigner main window. You will learn to:

♦ Customize your work environment

♦ Use object property sheets

♦ Use lists of objects

♦ Use the model object Browser

The second part of this feature guide explains the basic concepts you need to understand how to use PowerDesigner. You will learn to:

♦ Manage a workspace

♦ Use a folder

♦ Create and manage models, and packages

♦ Create and use modeling objects

♦ Use object shortcuts

♦ Use object replications

♦ Use transformations

♦ Compare and merge models

♦ Use the Resource Editor

♦ Use the Mapping Editor

♦ Use the Generation Links Viewer

♦ Use the PowerDesigner graphical interface

Audience

This book assumes that you are an experienced Windows user.

Although it does not assume you have knowledge about any particular topic, having some familiarity with object oriented databases, client/server architecture, SQL, and design methodology is helpful.

Documentation primer

The PowerDesigner modeling environment supports several types of models:

♦ **Conceptual Data Model (CDM)** to model the overall logical structure of a data application, independent from any software or data storage structure considerations

♦ **Physical Data Model (PDM)** to model the overall physical structure of a database, taking into account DBMS software or data storage structure considerations

♦ **Object Oriented Model (OOM)** to model a software system using an object-oriented approach for Java or other object languages

♦ **Business Process Model (BPM)** to model the means by which one or more processes are accomplished in operating business practices

♦ **XML Model (XSM)** to model the structure of an XML file using a DTD or an XML schema

♦ **Requirements Model (RQM)** to list and document the customer needs that must be satisfied during a development process

♦ **Information Liquidity Model (ILM)** to model the replication of information from a source database to one or several remote databases using replication engines

♦ **Free Model (FEM)** to create any kind of chart diagram, in a context-free environment

This book only explains the general features of PowerDesigner. For information on other models or aspects of PowerDesigner, consult the following books:

Conceptual Data Model User's Guide   To work with the CDM.

Physical Data Model User's Guide   To work with the PDM.

Object Oriented Model User's Guide   To work with the OOM.

Business Process Model User's Guide   To work with the BPM.

XML Model User's Guide   To work with an XSM.

Requirements Model User's Guide   To work with the RQM.

Information Liquidity Model User's Guide   To work with the ILM.

Reports User's Guide   To create reports for any or all models.

Repository User's Guide   To work in a multi-user environment using a central repository.

Typographic conventions

PowerDesigner documentation uses specific typefaces to help you readily identify specific items:

♦ `monospace text (normal and bold)`

   Used for: Code samples, commands, compiled functions and files, references to variables.

   Example: `declare user_defined...`, the `BeforeInsertTrigger` template.

♦ **bold text**

   Any new term.

   Example: A **shortcut** has a target object.

♦ SMALL CAPS

   Any key name.

   Example: Press the ENTER key.

Introducing PowerDesigner

About this chapter
This chapter introduces the main capabilities and concepts of PowerDesigner.

Contents

What is PowerDesigner?

PowerDesigner is a graphical and easy-to-use enterprise modeling environment which provides:

♦ Integrated modeling through standard methodologies and notations:

Data (E/R, Merise)

Business (BPMN, BPEL, ebXML)

Application (UML)

♦ Automatic code generation through customizable templates:

SQL (with more than 50 supported DBMSs)

Java

.NET

♦ Powerful reverse engineering capabilities to document and update existing systems

♦ A scalable enterprise repository solution with strong security and versioning capabilities to aid multi-user development

♦ Automated, customizable reporting capabilities

♦ An extensible environment, permitting you to add new rules, commands, concepts and attributes to your modeling and coding methodologies

Modeling with PowerDesigner

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

Requirements Model (RQM)

Business Analysts define business requirements, which may be refined into technical requirements by Analysts and Designers. You will use a **Requirements Model (RQM),** which describes a project by listing and explaining precisely what features must be implemented during a development process, and who is responsible for them. These requirements can then be attached to any object in any of the other models in order to trace where, and how, they are met.

Business Process Model (BPM)

Business analysts can also define the high-level business process flows to describe existing and new systems, and can simulate business processes to reduce time and resource and increase revenue. You will use a **Business Process Model (BPM), which** represents your organization's processes in real business terms. You can use a BPM as a design tool to identify your business needs, organize them in a hierarchy, display your processes graphically, and then generate components in process languages such as BPEL4WS or Sybase Unwired Orchestrator.

Data analysts

Conceptual Data Model
(CDM)

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 is a platform-independent representation of a system, giving an abstract view of its static data structures. A PowerDesigner CDM 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

Physical Data Model
(PDM)

Once the data structure is well defined, **Database Administrators** can 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.

Logical Data Model
(LDM)

You may also use a **Logical Data Model** (**LDM**), which is a special RDBMS-independent version of a PowerDesigner PDM, and which can be used as a bridge between a CDM and a regular deployable 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.

Information Liquidity
Model (ILM)

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

Developers

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

Object-Oriented Model (OOM)

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.

XML Model (XSM)

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

Traceability

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.

Repository

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.

Automated Reports

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.

Free Model

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

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.

Enterprise Modeling

PowerDesigner's sophisticated inter-model and code and database generation capabilities allow you to keep all your models and applications in synch while providing you with control over each and every modification.

The following illustration shows how the various models provided in PowerDesigner's enterprise modeling solution can interact during the design, deployment, and maintenance of your system:
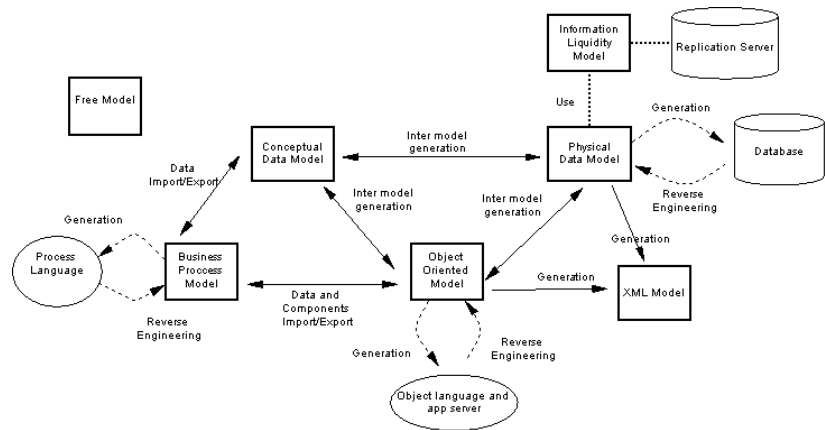
The diagram below shows another dimension of this inter-model interaction, by displaying how each of the different models benefits from and contributes to the global understanding and traceability of requirements:
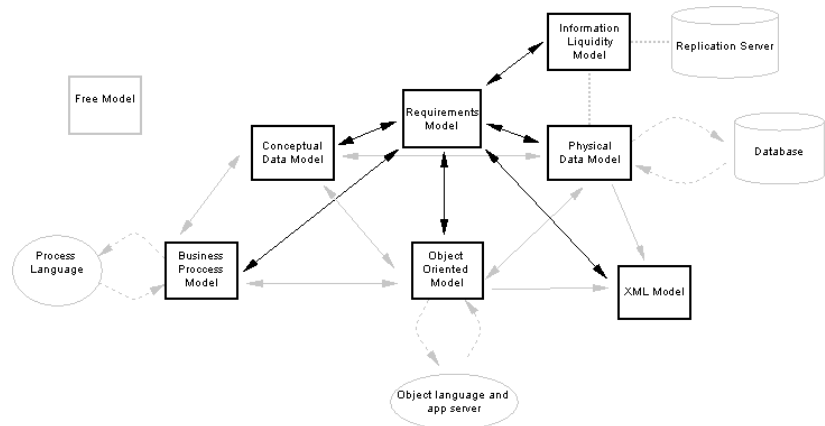
PowerDesigner Documentation

The following manuals are provided with your PowerDesigner installation:

The following manuals are provided with your PowerDesigner installation:

♦ **General Information**:

Installation Guide

General Features Guide

Repository User's Guide

Reports User's Guide

Advanced User Documentation

Integration with Other Products

Eclipse Plugin User's Guide

♦ **Model-Specific Guides**:

Requirements Model User's Guide

Business Process Model User's Guide

Conceptual Data Model User's Guide

Physical Data Model User's Guide

Information Liquidity Model User's Guide

XML Model User's Guide

Object-Oriented Model User's Guide

These manuals are provided in the following formats:

Format	Description
Online help	Available from within PowerDesigner by selecting the Help menu, by typing F1, or by clicking HELP on selected screens.
Online documentation	These manuals are available and searchable at: http://sybooks.sybase.com/pd.html
PDF Manuals	These printable manuals can be installed with PowerDesigner and are also available for download at: http://sybooks.sybase.com/pd.html

Videos   Demonstration videos are also available. They can be installed with PowerDesigner (see Help menu) or viewed from the installation CD.

API Documentation

Click Help ➤ Technical Documents ➤ PowerDesigner Scripting Object Help to access the PowerDesigner API documentation.

CHAPTER 2

Getting Started with PowerDesigner

About this chapter

This chapter explains the major PowerDesigner concepts, and introduces you to the tools you will use to build your models.

Contents

The PowerDesigner Interface

This section describes the main components of the PowerDesigner interface.

Main window components

The picture below shows a typical PowerDesigner window configuration.
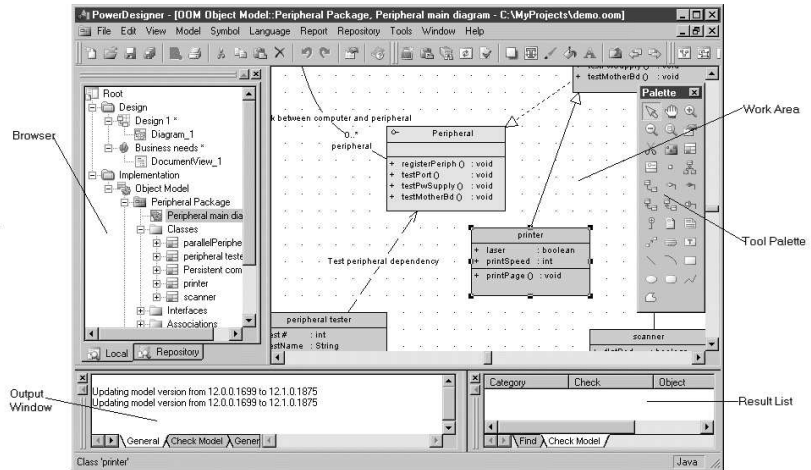
Browser	The **Browser** pane allows you to manage the objects you use to perform a modeling task. It displays your models and the objects belonging to them in a **tree view**, and allows you to rapidly navigate between them. The Browser also has a tab that gives you access to a PowerDesigner repository, where you can store all your models and associated files.
Work Area	The **work area** is the primary pane that displays your present model diagram or report outline.
Palette	The **Palette** provides graphical tools to help you quickly build model diagrams.
Output window	The **Output** window shows the progress of any PowerDesigner process, such as checking a model or generating or reverse engineering a database.
Result List	The **Result List** displays the results of a search or a model check.
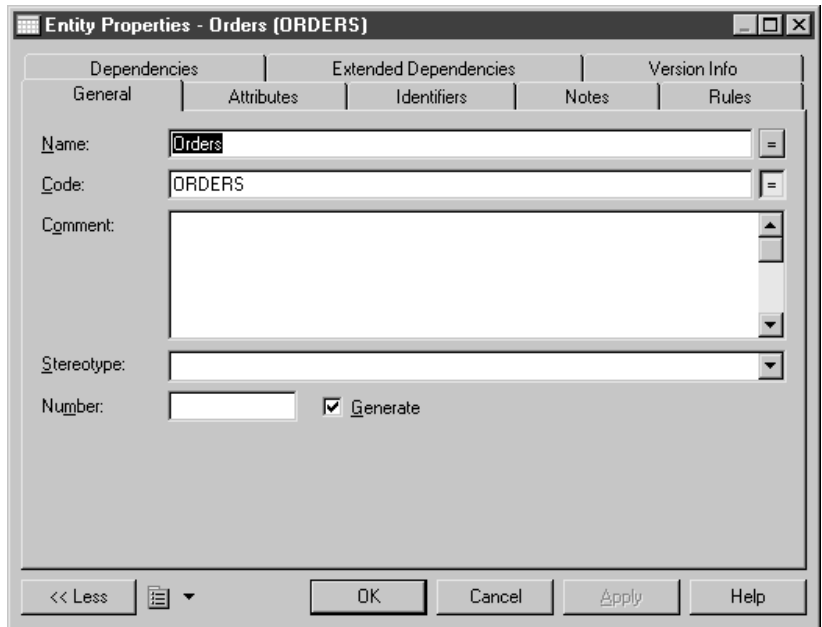
Property sheets

Each model object has a **property sheet**, which permits you to view and edit any of its properties. You can access an object's property sheet by double-clicking its symbol in your diagram or right-clicking it and selecting Properties from the contextual menu. For more information, see the section

""Working with property sheets" on page 22".

Object lists

You can view, add, create, modify, and delete multiple objects of a particular type by opening an **object list.** All your model's object lists are accessible via the Model menu. For more information, see the section ""Working with object lists" on page 25".

	Name ▼	Code	Data Type	Length	Precision	
→	Address =	ADDRESS	A80	80		
2	Commission	COMMISSION	MN8,2	8	2	
3	Company Name	COMPANY_NAME	A35	35		
4	CustomerID	CUSTOMERID	NO			
5	Description	DESCRIPTION	VA254	254		
6	Employee Description	EMP_DESC	VA254	254		
7	Employee Identifier	EMP_ID	NO			
8	Employee Name	EMP_NAME	A35	35		
9	Fax	FAX	A10	10		
10	Home Phone	HOME_PHONE	A10	10		
11	Hourly Rate	H_RT	MN8,2	8	2	
12	ItemsID	ITEMSID	NO			
13	Name	NAME	MN8,2	8	2	

List of Data Items

OK   Cancel   Apply   Help

Working with the Browser

The Browser provides a hierarchical view of all your model objects.
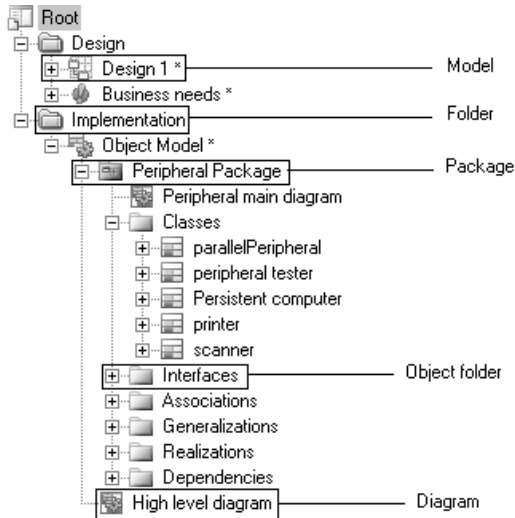
Hot keys for expanding and collapsing all nodes at once
To expand all nodes at once, press the numpad plus sign (+). To collapse all nodes at once, press the numpad minus sign (-).

Browser objects

A typical hierarchy of objects in a PowerDesigner Browser tree is as follows:

Workspace

The root of every Browser tree is a special type of folder called a **workspace**, a virtual environment that contains and organizes all the information and files you create to support your designs. A workspace allows you to save your local design environment so that it is available next time you start a session.

When you start PowerDesigner for the first time, a default workspace is created and opened. Whether you want to model data in a CDM or generate a multi-model report, you will always work within the context of a workspace, so that your files are grouped together and any changes in your model and folder hierarchy are recorded.

A workspace will generally contain links to model files, multi-model reports and external files, and may optionally also maintain a hierarchy of user-created folders. You can create several workspace files on your

machine, but you can only work in one workspace at a time. We recommend that you keep all the models relating to a particular project in the same workspace.

Folders
Workspaces can contain user-defined **folders**, which allow you to organize models or other files into groups. For example, if you are working on two separate projects, but want to be able to access both of them from a single workspace, you could organize the files using folders. For more information about folders, see the "Using the PowerDesigner Interface" chapter.

Models
A **model** is the basic design unit in PowerDesigner. Each model has one or more graphical views called diagrams and any number of model objects.

Packages
When you are working with large models, you may want to split them into smaller "sub-models" in order to avoid manipulating large sets of items. These sub-models are called **packages**, and can be used to assign different tasks or subject areas to different development teams. For more information, see the section ""Working with Packages" on page 29".

Diagrams
Diagrams show the interaction of various model objects. You can create several diagrams in a model or in a package.

Model Objects
Model objects is a general term used for all items belonging to a model. Some model objects, such as a class in an Object-Oriented Model, have graphical symbols while others, such as business rules, do not appear in diagrams and can only be accessed from the Browser or from an object list. For more information, see the section ""Working with Objects" on page 18".

Reports
You can document your PowerDesigner models by creating automatically-generated single or multi-model reports in the Report Editor.

Browser actions

Dragging and Dropping
You can drag and drop or copy objects in the Browser, or from the Browser to the diagram window.

Searching
You can find the Browser entry for any object you see in a diagram by right-clicking the symbol and selecting Edit ➤ Find in the Browser from the contextual menu. This feature is also available from the object's property sheet (from the dropdown menu in the bottom left corner) or from the contextual menu in a Result list.

Similarly, you can find the symbol (if one exists) for any object in the Browser by right-clicking the Browser entry and selecting Find in Diagram (or Find in Document View for an RQM).

Creating a Model

A model is the basic work unit in PowerDesigner. Every model is contained within a workspace, 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.

Creating a model

❖ **To create a model**

1. Select File ➤ New to open the New dialog box.

2. Select the appropriate model for your needs by clicking on it. Note that the tabs on the right of the dialog box change depending on the model currently selected.

3. Type a name in the Model name box. This is the name of the model. The code of the model, which may be used for script or code generation, is derived from this name according to the model naming conventions. You can modify the name and/or code at any time from the model property sheet by right-clicking the model entry in the Browser and selecting Properties from the contextual menu.

4. Choose any appropriate options in the right hand tabs (for example, if you are creating a PDM, you will specify a particular DBMS to model or, for an OOM, you will specify an object language).

5. If you are creating a BPM, OOM, or PDM, you can also specify the type of diagram you want to start with (you can add additional diagrams to your model later by right-clicking on the model in the Browser and selecting New ➤ *Diagram_Type*).

6. Click OK. The new model will be created in your workspace in the Browser, and its default diagram will be opened in the work area.

The following table lists the PowerDesigner models that you see in the Model type pane:

Icon	Model definition	File extension
	Business Process Model. A BPM allows you to identify processes and atomic tasks and organize them in a hierarchy that you can then implement	.bpm

Icon	Model definition	File exten- sion
	Conceptual Data Model. A CDM represents the overall logical structure of a database, which is independent of any software or data storage structure	.cdm
	Free Model. A FEM allows you to create any kind of chart or diagram, in a context-free environment	.fem
	Information Liquidity Model. An ILM is a design tool to make replication engine design and configuration easier to use	.ilm
	Multi-Model Report. Not a model, but an automated report that covers more than one model	.mmr
	Object Oriented Model. An OOM uses UML diagrams to let you design a static conceptual model of a software system using an object-oriented approach for Java or other object languages	.oom
	Physical Data Model. A PDM specifies the physical implementation of the database	.pdm
	Requirements Model. An RQM is a textual model, which can be used to list and explain all the customer needs that must be satisfied during a development process	.rqm
	XML Model. An XSM is a graphic representation of an XML file that provides a global and schematic view of all its elements	.xsm

Saving a model

Each model is saved in a separate file, with an extension specific to its model type. When you save a model, PowerDesigner also automatically creates a backup copy.

❖ **To save and close a model**

1. Select File ➤ Save.

   or

   Click the Save tool in the PowerDesigner toolbar.

   or

   Right-click the model in the Browser and select Save.

2. Right-click the model and select Close.

Working with Objects

Model objects are the building blocks of your models. They are listed as items in the Browser and often also appear as symbols in your diagrams.

There are two main types of objects:

♦ Ordinary Objects – such things as tables and entities, which are self-sufficient, and can be created in isolation from other objects

♦ Link Objects – such things as associations or dependencies, which link two ordinary objects, and cannot be created in isolation

The Palette

The Palette is a toolbar that lets you quickly create objects and the links between them in your diagram:
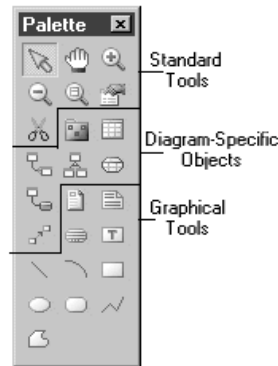
The standard palette tools are as follows:

Icon	Action
	Pointer [default] – Selects, moves, and resizes individual symbols. Double-click to select all the objects 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

Icon	Action
	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

The central tool area in the palette depends on the model you are working with, and will contain tools to create all the objects relevant to that type of model. For details of these model-specific tools, see the relevant User's Guide.

The standard graphical tools are as follows:

Icon	Action
	Insert an area where you can write free notes
	Draw one of the following: ♦ a graphical link between free symbols in the diagram ♦ a note link between a note and an object ♦ an extended dependency between objects (where possible)
	Insert a file symbol and object
	Insert a title box
	Insert text
	Draw a line

Icon	Action
	Draw an arc
	Draw a rectangle
	Draw an ellipse
	Draw a rounded rectangle
	Draw a jagged line
	Draw a polygon

> **Switching to the Pointer tool**
> To switch to the default Pointer tool, right-click anywhere in the diagram.
>
> To revert to the previously selected tool, hold down CTRL and perform a right double-click.

Creating objects

You can create objects:

♦ Directly in your diagram, using the Palette tool (if the object can be represented in a diagram by a symbol)

♦ From the Browser

♦ From an object list

Creating an object from the Palette

Ordinary objects, such as classes (in an OOM), or tables (in a PDM) can be created independently in any free space in the diagram:

❖ **To create an ordinary object from the Palette**

1. Click an object creation tool in the Palette and note how the pointer takes on the form of the selected object.

2. Click in a space in the diagram to create an object symbol there, and note that an entry for the object is displayed in the Browser.

3. Click elsewhere in the diagram to create another object symbol, or right-click anywhere to release the object creation tool and return to the pointer tool.

Link objects, such as messages (in an OOM), or references (in a PDM) are drawn between two ordinary objects in the diagram:

❖ **To create a link object from the Palette**

1. Click a link object creation tool in the Palette and note how the pointer takes on the form of the selected link object.

2. Click and hold inside an ordinary object, and then drag the pointer to inside a second object. The link object is created when you release the mouse button. Note that an entry for the link is displayed in the Browser.

3. Do the same again to link two other objects or right-click anywhere to release the object creation tool and return to the pointer tool.

❖ **To change the default name of a newly-created object**

1. Click the object to select it.

2. Click it again (or press F2) to enter edit in place mode. Press ENTER after you have completed typing the new name.

Creating an object from the Browser

❖ **To create an object in the Browser**

1. Right-click the model, package or object category where you want to create the object, and select New ➤ *object_type* to open a new default object property sheet (see the section ""Working with property sheets" on page 22").

2. Type an object name and code in the General tab of the property sheet, and then add any other relevant properties in the remaining fields of this or the other tabs.

3. Click OK to confirm the creation of the object.

The object is created in the appropriate object category in the Browser, under the current model or package, and its symbol is added to the current diagram.

Creating an object from an object list

You can also create objects from the lists of your model objects available by clicking Model ➤ *object_type*. For more information, see the section ""Working with object lists" on page 25".

Working with property sheets

All model objects created in PowerDesigner have property sheets.
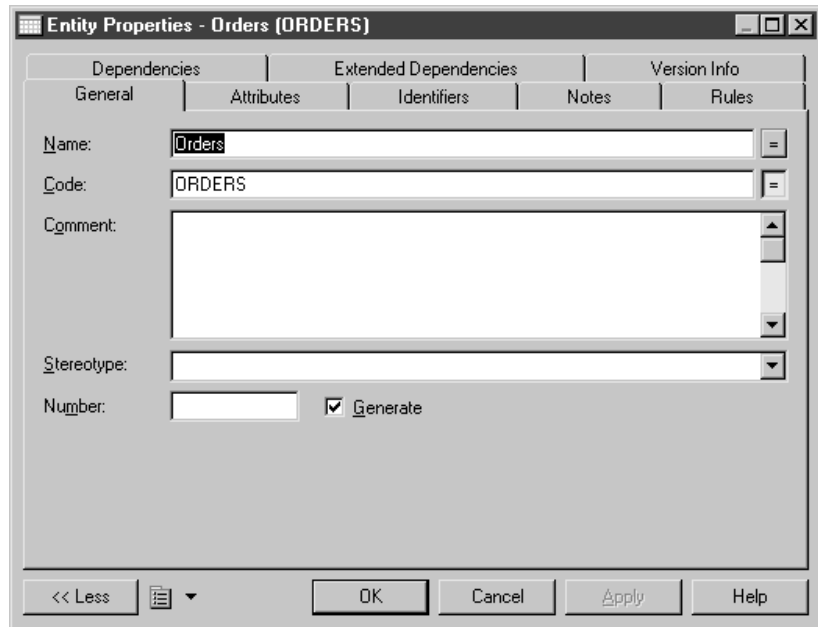
Property sheets allow you to assign extensive and sophisticated properties to objects. Most PowerDesigner objects have the following properties, but note that many other properties will be available, depending on the type of object:

♦ **Name**: the name of the object, which should be clear and meaningful, and should convey the object's purpose to non-technical users

♦ **Code**: the technical name of the object, which is used for generating code or scripts, which may be abbreviated, and should not generally include spaces

♦ **Comment**: an optional description of the object, which can be used to provide further information.

♦ **Notes** (on the Notes Tab) are made up of **Descriptions**, which provide detailed information about the object, and **Annotations**, which contain your own notes regarding the implementation of the object

♦ **Rules** (on the Rules tab) that your business follows. A business rule could be a government-imposed law, a customer requirement, or an internal guideline

♦ **Version Information** (on the Version Info tab) provides details about the object owner, creation, and modification date

♦ **Dependencies** (on the Dependencies tab) list all objects depending on the current object

♦ **Extended Dependencies** (on the Extended Dependencies tab) list all objects which collaborate with the current object

Opening a property sheet

❖ **To open a property sheet**

1. Double-click the object symbol in the diagram.

   or

   Right-click the object symbol in the diagram and select Properties.

   or

   Right-click the object name in the Browser and select Properties.

   or

   Double-click any label (name, code, role or cardinality) of a linking object in the diagram.

   The property sheet opens to the General tab. In the case of a link object, a graphical representation of the link with its source and destination objects points is displayed in the upper part of the dialog box.
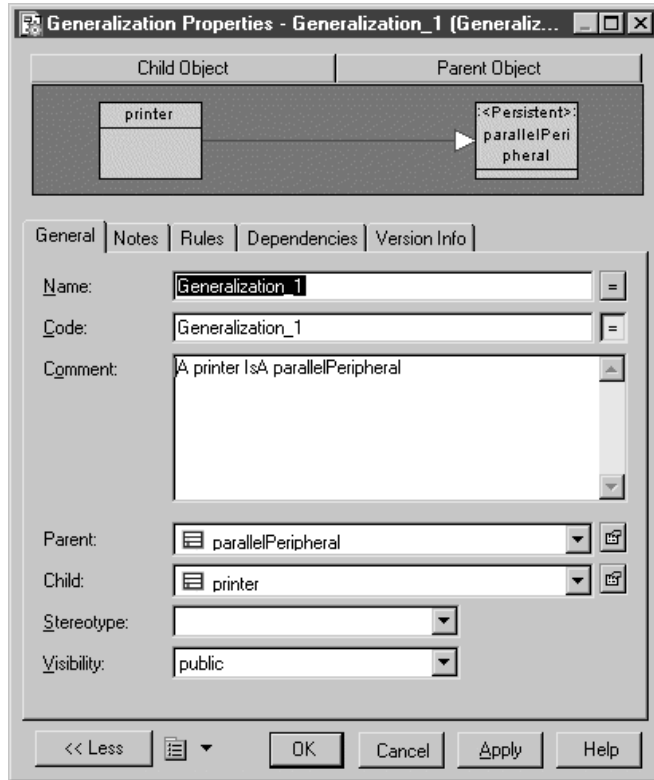
Specifying object properties

Property sheets behave like standard dialog boxes, and allow you to perform all the usual actions, such as typing and editing text in boxes, selecting check boxes, and selecting values in lists.

Once you have defined or modified any properties, you can click **Apply** to commit any changes and continue, or **OK** to commit any changes and close the property sheet. Click **Cancel** if you want to discard any changes made.

Customizing a property sheet

Since PowerDesigner property sheets can contain such a large, diverse amount of information, you may want to customize their display via the Customize Property Sheet dropdown menu in the bottom-left corner. This menu gives you the following options:

Command	Description
Display Standard Pages	Displays a minimum number of tabs or the tabs that you have defined as standard in the Customize Standard Pages dialog box
Display All Pages	Displays all available tabs
Customize	Allows you to define standard tabs

Any changes in the display options of property pages are immediately applied to the current property sheet and become the default settings for any property sheet of the same type that you subsequently open.

Working with object lists

Every object created in your model will appear in the object list of that type that is available by selecting Model ➤ *object_type*, or by right-clicking your model in the Browser and selecting List of ➤ *object_type* from the contextual menu.

An object list shows all objects of a certain type in the current package or model, including those that do not have symbols in the current diagram. An object list allows you to view, compare, and edit the properties of multiple objects. You can add, create, modify, and delete objects directly in the list. You can also filter the objects in the list and control which property columns are displayed.

Opening an object list

❖ **To open a global object list**

1. Select Model ➤ *Item* .

   or

   Right-click a model or package in the Browser and select List of ➤ *item* .

   The corresponding list opens:
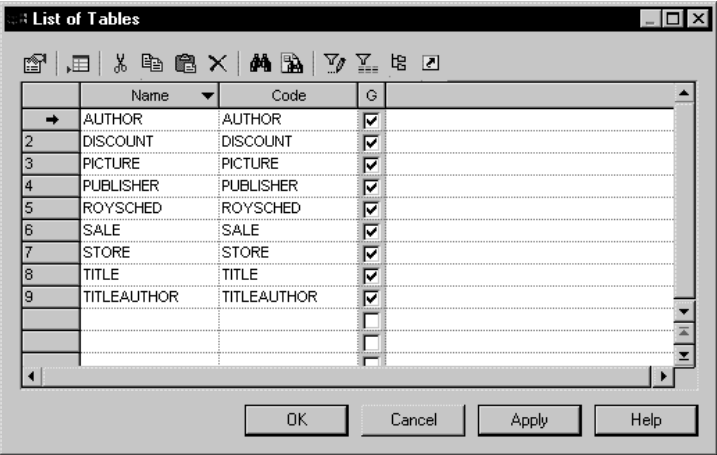
Object List toolbar   The following tools are available in the object list toolbar:

Tool	Action
	Show the property sheet of the selected object
	Create a new object
	Cut the selected object
	Copy the selected object
	Paste the selected object
	Delete the selected object
	Find a row

Tool	Action
🖳	Find the symbol of the selected object in the diagram
⟁	Customize the columns to be displayed and/or filter by their contents
⟁	Enable or disable a filter
⬚	Display objects belonging to sub-packages of the present model or package
⬚	Display composite objects (not available in all lists)
⬚	Display objects that exist only as shortcuts in the current model

Creating a new object in
a list

> ❖ **To create a new object in a list**
>
> 1. Select Model ➤ *object_type* (or select the appropriate tab in an object's property sheet) to open the list.
>
> 2. Click the Add a Row tool to add a new default object at the end of the list.
>
> 3. Type an appropriate name for the object (and, for link objects, select a source and a destination object), and click Apply. The new object will be placed in its appropriate alphabetical place in the list. Add other objects if necessary.
>
> 4. Click OK to close the list and return to the diagram which will be centered upon the symbol of the last object added.

Modifying object properties

You can modify object properties:

♦ In the object's property sheet

♦ From a list

Modifying properties in a
property sheet

❖ **To modify an object's properties in its property sheet**

1. Right-click the object in the Browser and select Properties.

   or

   Right-click the object symbol in the diagram and select Properties.

   or

   Double-click the object symbol in the diagram.

2. Modify the necessary properties in the different tabs.

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

   > **Opening property sheets at the last accessed tab**
   > Property sheets open to the General tab by default. However, you can choose to open property sheets at the last accessed tab by selecting Tools ➤ General Options ➤ Dialog, and selecting the Keep Last Tab option in the Property Sheets groupbox.

Modifying properties in a list

❖ **To modify object properties in a list**

1. Select Model ➤ *object_type* (or select the appropriate tab in an object's property sheet) to open the list.

2. Select the object that you want to modify and edit any of its properties directly in the list.

3. Click OK to save your changes and close the list.

Working with 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 useful to assign portions of a model, representing different tasks and subject areas, to different development teams.

Packages work like models, they can contain other packages, objects and diagrams. You can create as many packages as you need in a model, and you can also assign as many diagrams as you need to provide a specific view of each package.

Creating a package

You can only create a package inside a model.

♦ From a diagram: select the palette's Package tool and click in the diagram

♦ From the Browser: right-click a model and select New ➤ Package

♦ From the list of packages: select Model ➤ Packages to display the List of Packages and click the Add a row tool to insert a new package in the list

In the example below, a package called "Lending Records" (with a diagram of the same name) has been added to the model "Library OOM":
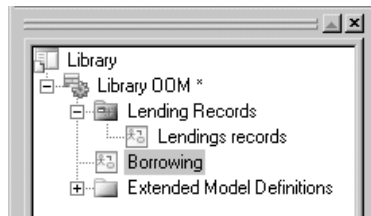
Building a package

You build a package in the same way as you build a model, with the difference that you must decide who will own each object, the model or the package. Objects created in and owned by a model will be available to all its packages as shortcuts, and vice versa. Thus only one instance of each object need be created in the model or in a package, and it can be referenced by way of the shortcut.

Saving a package

You cannot save a package individually, but only within its parent model.

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 (PDM) and information liquidity model (ILM) allow you to define data connections for target databases in order to generate your models and reverse engineer your data sources. The PowerDesigner repository requires a database to store your models and other design documents.

☞ See the *Physical Data Model* and *Information Liquidity Model* users guides, and *Working with the Repository* .

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
DBMS client	Native connection profile
JDBC driver	JDBC connection profile
ADO.NET driver and DBMS client	ADO.NET connection profile
OLE DB driver and DBMS client	OLE DB connection profile
DirectConnect driver	DirectConnect connection profile

☞ For information about connecting with an ODBC driver, see "Configuring ODBC machine and file data sources" on page 30.

☞ For information about connecting with a DBMS client, or JDBC, ADO.NET, OLE DB, or DirectConnect driver, see "Configuring connection profiles" on page 32.

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:

❖ **To define an ODBC data source**

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.
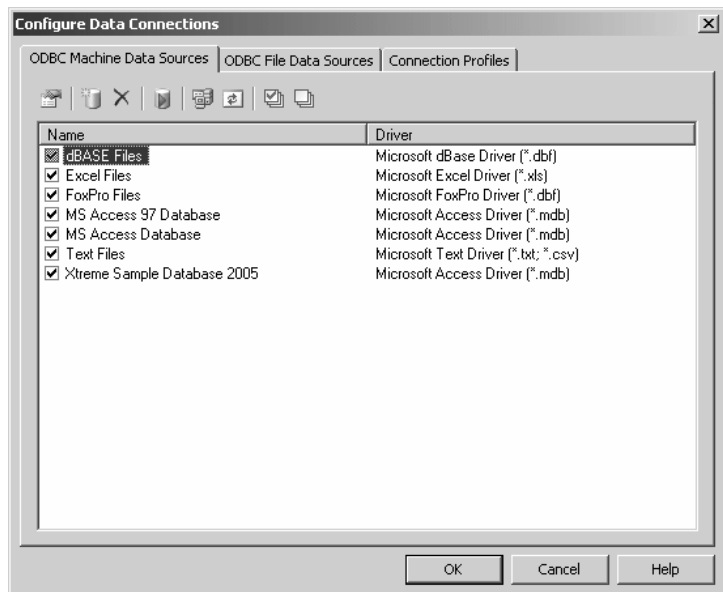
2. Click the Add Data Source tool (see "Data connection tools" on page 38)

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:
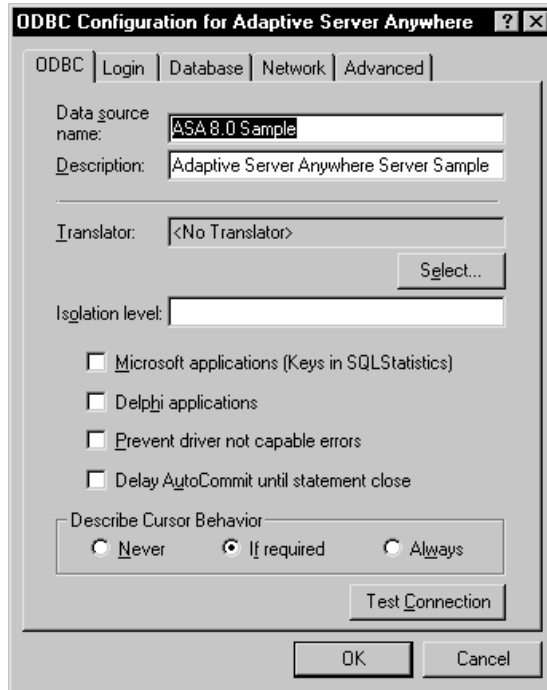
4. Enter the appropriate parameters and then click OK to return to the Configure Data Connections window.

5. Click OK to return to your model.

Configuring connection profiles
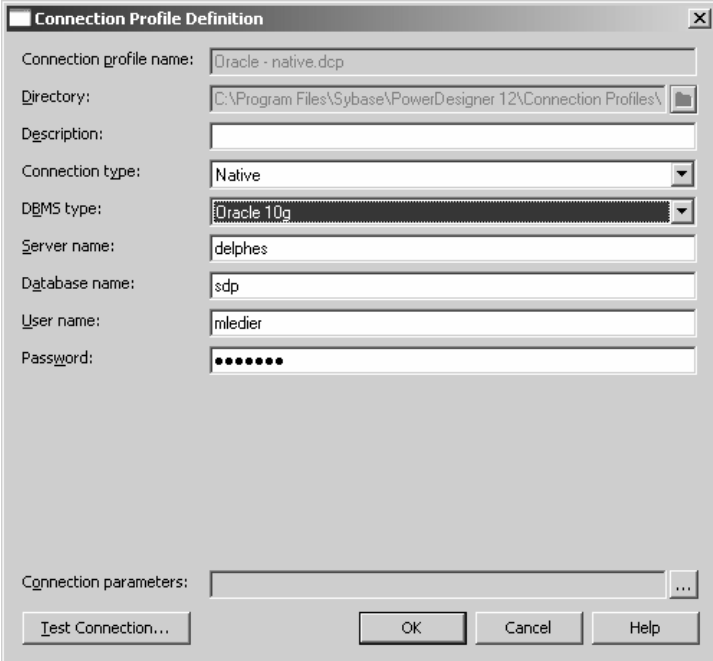
Before you can connect to a database through a DBMS client, or JDBC, ADO.NET, OLE DB, or DirectConnect driver, you must create an appropriate connection profile.

❖ **To create a 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 38) to open a Connection Profile Definition window:

3. Enter the properties (see "Connection profile properties" on page 34)
   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.

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 following properties are available in the Connection Profile Definition
window:

Property	Description
Connection pro-file 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 ♦ ADO.NET ♦ OLE DB ♦ JDBC ♦ DirectConnect 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 and DirectConnect only] Specifies the name of the database server to connect to.
Database name	[Native and DirectConnect only] Specifies the name of the database to connect to.
User name	Specifies the username to use when connecting.
Password	Specifies the password to use when connecting.
ADO.NET namespace	[ADO.NET only] Specifies the namespace to use for the connection.
OLE DB data provider	[OLE DB only] Specifies the data provider to use for the connection.
Data source	[ADO.NET and OLE DB only] Specifies the data source to use for the connection.
Provider-specific connection string	[ADO.NET and OLE DB only] Specifies a connection string to use for the connection.

Property	Description
Use integrated login	[ADO.NET] Specifies to use your windows login for connecting to the database.
ADO release	[ADO.NET] Specifies the release to use for the connection.
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 (see "Connection Parameters window" on page 36).

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.

❖ **To add a connection parameter**

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 my database, "Green", on my server "Rainbow".

Connection Profile Definition	☒

Connection profile name:	Oracle - native.dcp
Directory:	C:\Program Files\Sybase\PowerDesigner 12\Connection Profiles\
Description:	Connection via OraClient10g Client
Connection type:	Native
DBMS type:	Oracle 10g
Server name:	Rainbow
Database name:	Green
User name:	graham
Password:	•••••••

Connection parameters: [   ] [⋯]

[Test Connection...]   [OK]   [Cancel]   [Help]

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".

Data connection tools

The following tools are available in the Configure Data Connections window:

Tool	Description
	Properties – Opens the ODBC Setup or Connection Profile Definition dialog for the selected profile.
	Add Data Source – Creates a new connection.
	Browse Data Source File [ODBC file data sources and connection profiles] – Opens a file browser to select a .dcp profile file.
✕	Delete – Deletes the selected connection.
	Test Connection – Tests the selected connection.

Tool	Description
	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.

❖ **To connect to a data source**

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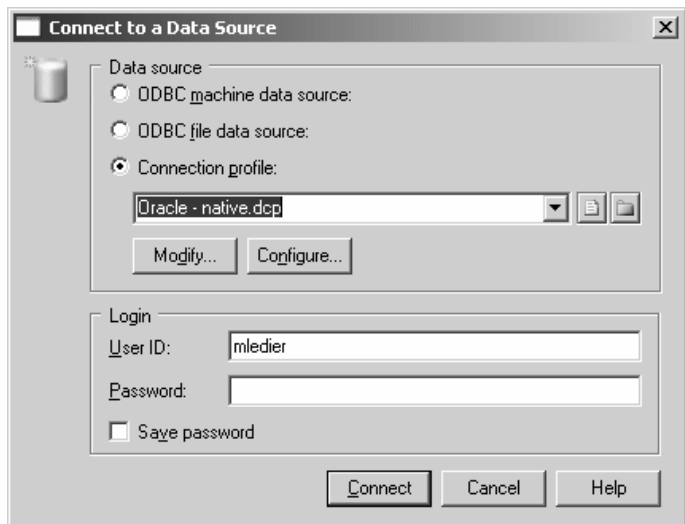
2. Select one of the following radio buttons, depending on your chosen method for connecting to your data source:

   ♦ ODBC machine data source (see "Configuring ODBC machine and file data sources" on page 30)

   ♦ ODBC file data source (see "Configuring ODBC machine and file data sources" on page 30) - 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 32) - 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 your 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.

Connection time
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. The following tools are available in the Edit/Run Script editor toolbar:

Tool	Description
⌷ ▾	Open Editor Contextual menu (SHIFT + F11)
⌷ ▾	Edit With (CTRL + E). Opens the previously defined default editor (see "Defining a text editor" in the Using the PowerDesigner Interface chapter). Click the down arrow to select another editor.
▷	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)

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 Model User's 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

Business Process Model (BPM)

Now turn to the *Business Process Model User's 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)

Now turn to the *Conceptual Data Model User's Guide* where you will learn how to:

♦ Create business rules, domains, and data items

♦ Create entities and relationships between them

♦ Generate a PDM from your CDM

Physical Data Model (PDM)

Now turn to the *Physical Data Model User's Guide* where you will learn how to:

♦ Create tables, columns, primary keys, indexes, and references and define referential integrity

♦ Create views, triggers, and abstract data types

♦ Reverse engineer from and generate to database scripts

♦ Work with multidimensional diagrams

Information Liquidity Model (ILM)

Now turn to the *Information Liquidity Model User's 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 and Mobilink replication engines

XML Model (XSM)

Now turn to the *XML Model User's 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

Object-Oriented Model (OOM)

Now turn to the *Object-Oriented Model User's 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

CHAPTER 3

Managing PowerDesigner Licenses

About this chapter   This chapter explains how to manage PowerDesigner licenses.

Contents

43

Understanding License Management in PowerDesigner

PowerDesigner supports a framework of license management common to other Sybase products. This system defines different types of licenses corresponding to different types of use of the product. You can manage the PowerDesigner licenses installed on your machine with a wizard.

This section explains the different types of licenses available in PowerDesigner.

Trial version and Regular licenses

The **trial** version of PowerDesigner is a free copy of the product used for evaluation purpose whereas a regular license needs to be purchased from Sybase. In both cases you get a full-featured copy of PowerDesigner, but the trial version is time-limited.

Trial

Definition

A trial version is free. It contains all the PowerDesigner features but you cannot use it for more than 15 days after the installation date.

If you need more than 15 days to evaluate PowerDesigner, you can obtain an extension from Sybase. This extension will be provided as a standalone local license (see ""Standalone seat - Local license" on page 45") established for a limited period of time.

Business case

You run a PowerDesigner trial version to evaluate the product and compare it with competitors. The trial version is a full-featured version of PowerDesigner that allows you to perform any design task.

Regular

Definition

A regular license needs to be purchased from Sybase or from a reseller. PowerDesigner is available in various packages that provide different levels of functionality.

Business case

You use a license to design models with PowerDesigner.

Local and served licenses

When you are using a PowerDesigner license, you have to choose between a **local** license dedicated to your computer and a **served** license provided by a license server.

Standalone seat - Local license

Definition

A **local license** is designed to be used without a license server. Local licenses are also called un-served licenses. A local license is dedicated to a given machine and cannot be used on other workstations.

When you select the Standalone seat - Local license installation option you have to declare the license file to authorize PowerDesigner on your machine. See the ""Requesting a License from Sybase Product Download Center" on page 50" section for more information on getting the license file.

Note that all PowerDesigner packages can be installed on your machine when you select this installation option. You can evaluate these packages in trial mode for 15 days after installation. However, only authorized packages will actually work beyond the trial period.

Business case

Local licenses are more convenient in smaller design teams, when you do not have many licenses and do not want to setup a license server. However this configuration requires that you activate the licenses for each machine separately and implies that you do not have centralized control over your licenses.

Standalone seat - Served

Definition

A **served standalone license** is designed to be provided by a server. A served standalone license is dedicated to a given machine and cannot be used by other designers.

This configuration requires that an administrator installs a license server in your organization (more information on how to setup a license server, see the "Setting up a License Server" section in the "Installing PowerDesigner" chapter in the *Installation Guide*). Then, when you select the Standalone seat - Served installation option, you will be prompted to define the license server and the PowerDesigner package you want to use in order to fetch the appropriate license from the server and authorize PowerDesigner on your machine.

A served standalone license requires:

♦ That you be connected to the license server when you start PowerDesigner for the first time to fetch the license.

♦ That you regularly connect to the license server. If you remain disconnected from the license server for more than one month, your license will be disabled on your machine. You will have to reconnect to the server to be able to use your license again.

Business case

Served standalone licenses are used in larger design teams, when you want

to prevent users from setting up their machine with an Internet connection for downloading license files. In this mode, the license is activated once by the administrator and not for each end user. The connection to a license server may also be considered to be more secure.

You have full control over license usage using the server mode.

Floating license - Served

Definition

A **served floating license** is requested from a shared pool of licenses each time you start PowerDesigner. The license server manages the pool of licenses.

This type of license is not dedicated to a given machine and needs to be requested and returned to the license server after use. This configuration also requires that an administrator installs a license server in your organization (more information on how to setup a license server, see the "Setting up a License Server" section in the "Installing PowerDesigner" chapter in the *Installation Guide*).

A floating license can also be used in mobile mode. When you need to work disconnected from the license server, you can borrow a floating license for 30 days and work offline during that period. See the ""Mobile mode" on page 46" section for more information.

Business case

Served floating licenses are also used in large design team when you have more users than licenses but not all users use PowerDesigner at the same time, they can share a common pool of licenses. The floating license solution can be better suited to this kind of organization.

Mobile mode

When you are using a floating license, you can use it in **mobile** mode. This mode allows you to use the license without being connected to the license server.

Borrowing a mobile license

To use a floating license in mobile mode you need to borrow the license, you can keep the mobile license for 30 days.

❖ **To borrow a mobile license**

1. Select Tools ➤ Take Mobile License.

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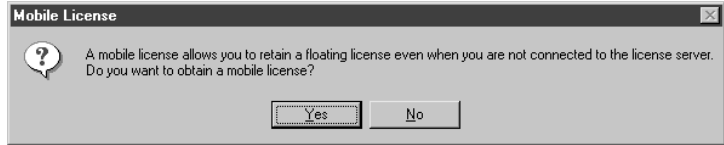
2. Click Yes to obtain a mobile license and click OK to confirm.

Returning a mobile
license

When you no longer need the mobile license you should return the license to the server.

If you do not return the mobile license before 30 days, it is automatically recovered by the server, and disabled on your machine even if you do not connect to the server. In this case, you do not have to return the mobile license the next time you connect to the license server.

❖ **To return a mobile license**

1. Select Tools ➤ Return Mobile License.

   A confirmation dialog box is displayed.
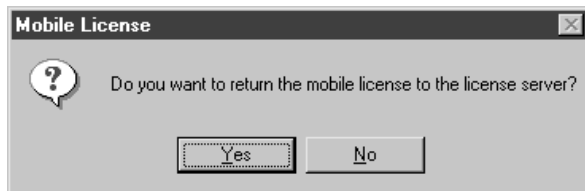
2. Click Yes to return the mobile license and click OK to confirm.

License management

This section explains how licenses are managed within PowerDesigner.

PowerDesigner contains a set of models used for designing different types of information. You cannot buy models individually, you must buy a package that is a set of PowerDesigner models.

If you are using a served license   When you start PowerDesigner the selected package authorization is requested from the license server. In case of failure PowerDesigner does not start.

If you are using a standalone served license   PowerDesigner verifies license on the server. If your workstation cannot connect to the license

47

server, you will be able to use your license for 30 days and then it will be disabled. You need to reconnect to the server to be able to use your license again.

If you are using a local license   There is no selected package and PowerDesigner is looking for all licensed packages in the local license file. If a model is not authorized, it will not be available in the PowerDesigner workspace.

Grace period

A grace is a 15-day period that allows you to use PowerDesigner without a license. Grace periods are not granted to floating license users.

You obtain a grace period in the following situations:

♦ When you do not have a license for a model you choose to install (this model is flagged "Trial" in the installation component list), you can test this model for 15 days after the installation date.

♦ When you have a license but you failed to obtain a license from the server (standalone served license) or to read the license file (standalone local license), you automatically obtain a grace period for using the product for 15 days after the last time you successfully took the license.

Check during execution of floating license

Floating license status is verified regularly in order to check that you are not using an expired license. If your license server is no longer responding for example, a modal dialog box is displayed to warn you that you have 24 hours to fix the problem. When you click OK, a modeless dialog box is displayed 24 hours later to prompt you to save your work, and a final confirmation dialog box is displayed before the application shuts down.

Trial expiration

When the trial version you have been evaluating expires you can purchase a license from Sybase to use PowerDesigner. Once you have the license, you must use the License Management Wizard to define the license configuration you want to use.

You can also obtain a trial extension from Sybase if you want to further evaluate PowerDesigner.

Troubleshooting

If PowerDesigner repeatedly fails to obtain authorization from the license file you can try to proceed as follows:

♦ Move any file with the .lic extension from the PowerDesigner installation folder to a safe place.

♦ Make a backup copy of the file sysam.properties, and open it in a text editor. Remove the lines starting by "Pd.LicenseMode" and "Pd.Package" at the end of the file.

☞ For more information about SySAM, you can look at the following Web address http://www.sybase.com/sysam.

This turns your current installation into a trial version.

If you are still in the 15-day grace period   You can use PowerDesigner normally.

If the grace period has expired   You are prompted to use the License Management Wizard to modify the type of license you are using. This should let you re-enter a valid license key.

Requesting a License from Sybase Product Download Center

The **Sybase Product Download Center** (SPDC) is an online kiosk where you can obtain license keys to enable a Sybase software.

A PowerDesigner license key is a file with the .lic extension containing the information for authorizing PowerDesigner on your machine.

If you purchased PowerDesigner from Sybase When your order is processed, you will receive an E-mail with an URL, your login, and your password for the Sybase Product Download Center. Note that this E-mail is sent by subscribenet.com (not Sybase), make sure the subscribenet.com domain is not listed in your spamfilter otherwise you will not receive this message.

If you purchased PowerDesigner from a Sybase reseller When your order is processed, you will receive a certificate with a Web key registration number and a download URL.

Host ID and computer name

Before accessing the download site, make sure you know exactly your license configuration, the host ID and the name of the computer that will receive the license key file. You can find the host ID and the computer name in the License Management Wizard. When you select the Standalone Seat - Local License option in the wizard, these details are automatically calculated and displayed in the second page of the wizard. See the ""Selecting a license file" on page 53" section.

If you download the license file for a license server, you also need the host ID and the name of the server. You can find these information on the third page of the license server installation program (see the "Setting up a License Server" section, in the "Installing PowerDesigner" chapter in the *Installation Guide*).

If the computer has several Ethernet cards or if they are supposed to change, you can use the volume serial number, which is another identification number that you obtain by typing dir c: in a DOS prompt window

Using the License Management Wizard

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

You can use the License Management Wizard to:

♦ Activate a PowerDesigner license, typically when you switch from a trial version to a regular license.

♦ Modify your license configuration if you switch from one type of license to another.

The following use cases explain different situations that may lead to using the License Management Wizard.

Use cases

The following use cases explain different situations that may require a change in license configuration.

From trial to license

You have been evaluating a PowerDesigner package with a trial version, and you have decided to adopt PowerDesigner as your design tool. You can switch from a trial version to a regular license using the License Management Wizard.

If you had installed all the package components on your machine   You do not need to reinstall PowerDesigner, the new license file will activate a regular PowerDesigner version that is not time-limited.

If you had installed only some package components on your machine
You will have to use the setup program to modify your current installation to have the missing components installed. After setup, you will need the License Management Wizard to modify your current license configuration.

Trial extension

Case 1

You have been using a trial version of a PowerDesigner package, and need more time to further evaluate the product you can obtain an extension from Sybase. The extension is a standalone local license limited for a certain period of time. You must change the license type to standalone local to be able to use the trial extension.

Case 2

You are working with a DataArchitect license for example and want to evaluate a trial version of Studio. To do so, you must:

♦ Call Sybase to obtain a trial extension (The extension is a standalone local license limited for a certain period of time).

♦ Run the install program in maintenance mode to install the missing Studio components.

♦ Use the wizard to switch to standalone local to be able to use the extension file.

From standalone local to standalone served

More users are using PowerDesigner in your organization and your administrator has decided to setup a license server to save deployment time and avoid numerous Internet connections to SPDC from the different local machines.

Before using the License Management Wizard to change license type, you must make sure that the license server is running with the appropriate served licenses.

From standalone to floating

In large design teams, when all designers are not using PowerDesigner at the same time, you can switch from standalone to floating licenses. This allows you to save money by buying less licenses and putting them in a common pool of licenses that can be borrowed and returned according to the designers' needs.

Accessing the wizard

You access the License Management Wizard through the Tools ➤ License Parameters command.

When you are in trial mode   A license agreement dialog box is displayed when you validate the License Management Wizard.

When you are using a floating license in mobile mode   A dialog box is displayed to prompt you to return mobile licenses before accessing the License Management Wizard.

When you are using a served license   An error message is displayed if the server is not available and you cannot access the wizard.

Selecting a type of license

In the first page of the License Management Wizard you can activate a license or modify the way your license is provided.
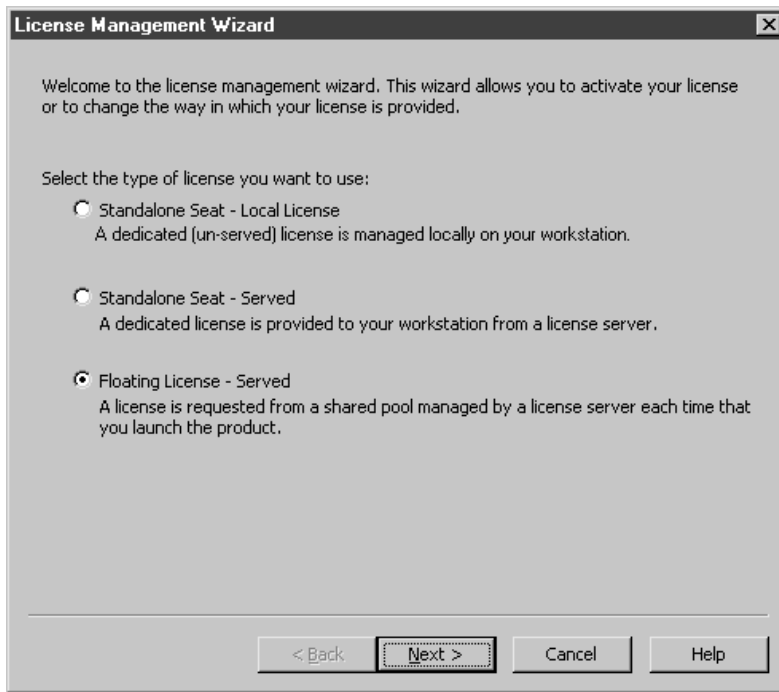
License type	Description
Standalone seat - Local license	A local license is a dedicated license designed to be used in a serverless configuration on a single machine.
Standalone seat - Served	A served standalone license is a dedicated license designed to be provided by a server and to be used on a single machine
Floating license - Served	A served floating license comes from a shared pool of licenses managed by a license server. This type of license is not dedicated to a given machine and needs to be borrowed and returned to the license server after use.
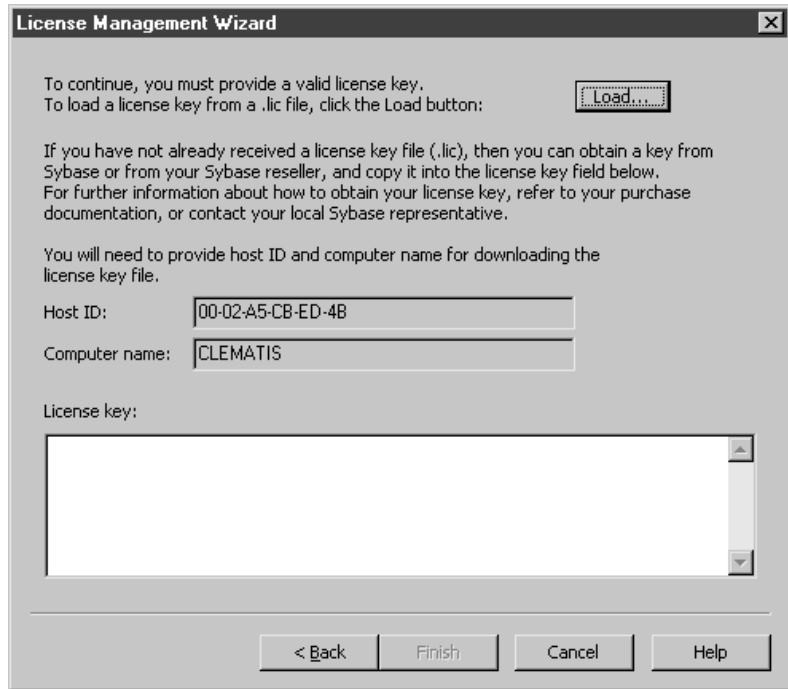
☞ For more information on the different types of licenses, see the ""Understanding License Management in PowerDesigner" on page 44" section.

Select a type of license and click Next.

Selecting a license file

When you select the Standalone Seat - Local License option, this page

allows you to specify the license file that will be used.

If you already have a license file You can click the Load button and browse to the folder where your license file is located. The content of the license file is automatically displayed in the License key box. You can click Next.
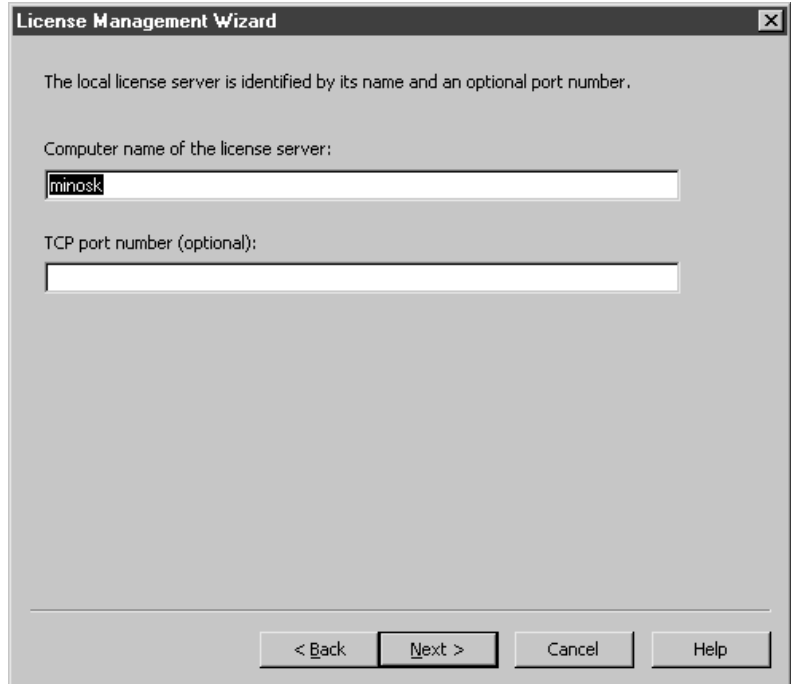
If you do not have a license file You need to download one from the Sybase Product Download Center. As explained in section ""Requesting a License from Sybase Product Download Center" on page 50" you will need to provide either a login and password or a Web key registration to connect to the SPDC site and obtain the license file.

In order to generate the license file, you must specify the **host ID** and **computer name** of the computer where you will be running the PowerDesigner license. By default, these boxes are initialized with the current computer identification, however if the downloaded license file is to be used on another machine you must change these default values otherwise the license will not work on this computer.

If the computer has several Ethernet cards or if they are supposed to change, you can use the volume serial number, which is another identification number that you obtain by typing dir c: in a DOS prompt window.

Selecting a license server

When you select the Standalone Seat - Served or the Floating License - Served option, this page allows you to identify the local server where the licenses will be stored.

You must type the **computer name** of the server.

You can also type the **TCP port number**. This is optional, you should only type a TCP port number if the administrator who installed the license server defined a specific TCP port number. In this case, you must specify this user-defined TCP port number.

Selecting a PowerDesigner package

When you select the Standalone Seat - Served or the Floating License - Served option, and after defining the license server, this page allows you to specify the PowerDesigner **package** you want to use.
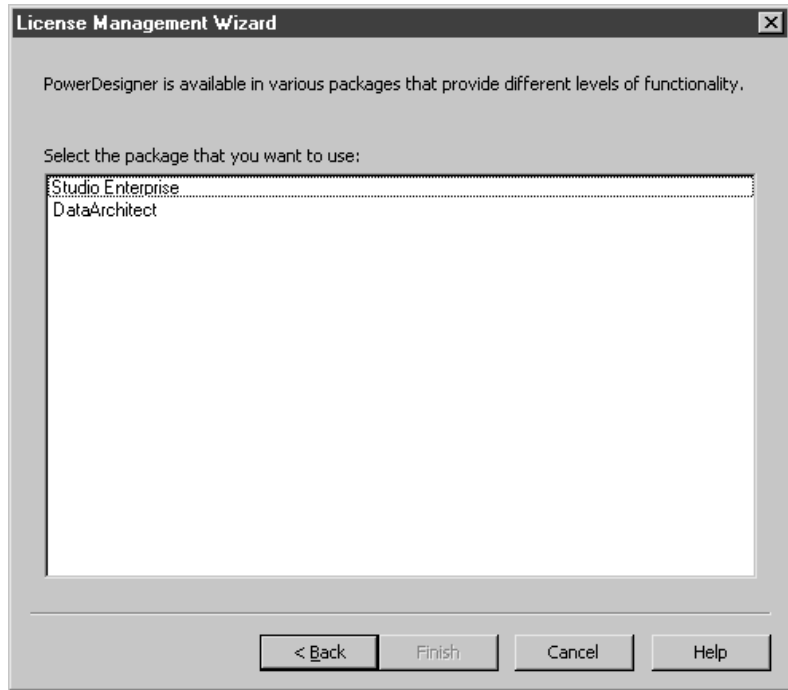
License Management Wizard

PowerDesigner is available in various packages that provide different levels of functionality.

Select the package that you want to use:

Studio Enterprise
DataArchitect

< Back   Finish   Cancel   Help

PowerDesigner is available in various packages that provide different levels of functionality.

A package is available on the server if a license file defines it, the list displays all packages available on the server but it does not reflect license availability.

When you select a package that has no available license on the server, an error message is displayed when you click Finish in the wizard. You can click Back and modify the selected configuration. You can also click Cancel to restore your previous configuration and wait for a license to be released or ask your license administrator to update the license file on the server.

Wizard validation

When you commit the last page of the wizard the following changes occur:

♦ 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.

PART II

MODELING GUIDE

This part presents the basic modeling concepts in PowerDesigner.

CHAPTER 4

Using the PowerDesigner Interface

About this chapter

This chapter explains how to use the different PowerDesigner interface components.

Contents

Managing Objects in the Browser

The Browser is a powerful feature for navigating and managing PowerDesigner objects.

Creating an object in the Browser

You can create objects in the Browser. If the symbol of the new object does not appear in the diagram you may have to select Symbol ➤ Show Symbols and select the object check box in the Show Symbols dialog box. Not all objects have symbols

❖ **To create an object in the Browser**

1. Right-click the workspace, a model, a package or an object folder and select New ➤ *object type* to create the object and open its property sheet.

2. Type a name and a code for the object.

3. Click OK.

   The object is displayed in the Browser.

Object has a symbol

You can see if an object has a symbol in the diagram by opening the object list to which it belongs and customize the display of the list so the S[ymbol] column is displayed. If the check box in the S column corresponding to the object is selected, then the object has a symbol in the diagram.

☞ For more information about object creation and management, see the "Managing Objects" chapter.

☞ For more information about customizing the display of a list, see the ""Lists" on page 91" section.

Identifying stereotyped objects in the Browser

You can easily locate in the Browser stereotyped objects whose stereotype is used as a metaclass. All objects having this stereotype are grouped together within the object type folder. This is even more visible when you attach an icon to this stereotype. You must collapse the object type folder and expand it again to refresh the display and re-order the stereotyped objects.

☞ For more information about stereotypes used as a metaclass and how to attach an icon to a stereotype, see " Defining a stereotype" in the "Managing Profiles" chapter in the *Advanced User Documentation* .

In the following example, Process_1 and Process_3 are grouped together as they have the same stereotype used as a metaclass; while Process_2 and Process_5, which have the same stereotype used as a metaclass, are similarly grouped:
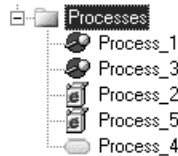
Displaying object properties

You can display the property sheet of an object selected in the Browser in any of the following ways:

♦ Double-click the object in the Browser

♦ Select the object in the Browser and press ALT+ENTER.

♦ Right-click the object in the Browser and select Properties.

Reports and Diagrams
When you double-click a report or a diagram, you open it in the report editor or diagram window. To open its property sheet, use one of the other methods.

Displaying an object contextual menu from the Browser

You can display the contextual menu of an object selected in the Browser.

❖ **To display the contextual menu of an object from the Browser**

1. Right-click an object in the Browser.

Editing an item in place

You can edit in place items in the Browser but you cannot edit the folders containing objects of the same type.

❖ **To edit an item in place**

1. Select an item in the Browser.

2. Click again on the selected item.

   or

   Press F2.

Dragging and dropping items

You can drag and drop items to and from the Browser.

☞ For more information about the drag and drop feature, see "Dragging and dropping objects" in the "Managing Objects" chapter.

Finding objects in the Browser

You can locate any model object in the Browser using the Find in Browser feature. For example, to find:

♦ A diagram in the Browser - Right-click the diagram background and select Edit ➤ Find in Browser.

♦ An object in the Browser from its symbol - Right-click the symbol in the diagram and select Edit ➤ Find in Browser.

♦ An object in the Browser from its property sheet - Open an object property sheet and select Find in Browser from the dropdown menu at the bottom-left corner.

♦ An object in the Browser from the Result List - Right-click the object in the list and select Find in Browser.

In each case,, the parents of the object in the Browser are automatically expanded and the found object is selected, to provide you with a hierarchical view of the contents of your workspace.

Note that you cannot find Repository objects in the Browser as they are not visible.

Workspaces

A workspace contains all the information you need to perform a modeling task with PowerDesigner. When you start PowerDesigner, you open a default workspace. You can work in this default workspace or open an existing workspace if you want to switch to another work environment to perform different tasks.

A workspace allows you to save your modeling environment in a file with a hierarchy of folders and models. During a work session, every change in the folder or models hierarchy is saved locally in the workspace.

> **Generated models and workspace**
> It is highly recommended to keep original models and their corresponding generated models in the same workspace.

The workspace environment contains:

♦ A hierarchy of folders

♦ Links to model files

♦ Links to multi-model report files

♦ Links to external files

You can create several workspace files on your machine, but you can only work in one workspace at a time.

The workspace is identified by its name, this property is saved locally in a file with the SWS extension.

The workspace does not have a property sheet.

Creating a workspace

By default, PowerDesigner opens with a workspace. Whether you want to model data in a CDM or generate a multi-model report, there will always be a workspace open in the PowerDesigner window.

To create a new workspace, you have to close the current workspace and a new workspace will be automatically created.

If the workspace-related items have been modified   A confirmation dialog box prompts you to save each item individually, save them all or cancel saving.

If the workspace-related items have not been modified   The workspace is closed.

❖ **To create a new workspace**

1. Select File ➤ Close Workspace.

   Confirmation boxes prompt you to save the models before closing the current workspace and opening a new work environment.

Opening a workspace

When you open an existing workspace, the hierarchy of folders and links to model and report files appear in the Browser. The models linked to the workspace are not open by default.

❖ **To open a workspace**

1. Select File ➤ Open Workspace.

   A confirmation message prompts you to save the current workspace.

2. Click Yes if you want to save the current workspace.

   or
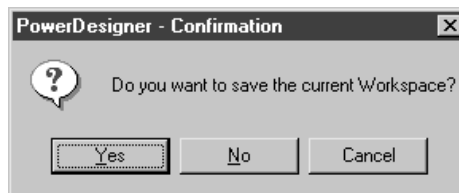
   Click No if you do not want to save the current workspace.

   A standard file selection dialog box is displayed.

3. Select a file with the SWS extension and click OK.

   The new workspace displays the hierarchy of folders, models, and packages defined in this workspace.

Renaming a workspace

The workspace is identified by its name. You can modify the name of a workspace from the Browser.

❖ **To rename a workspace**

1. Right-click the workspace in the tree view and select Rename.

   or

   Select the workspace and press F2 to edit the name in place.

Changing workspace

You can decide to change workspace during a work session. You do not have to exit PowerDesigner to change workspace.

❖ **To change workspace**

1. Select File ➤ Open Workspace.

   A confirmation message prompts you to save the current workspace.

2. Click Yes if you want to save the current workspace.

   or

   Click No if you do not want to save the current workspace.

   A standard file selection dialog box is displayed.

3. Select another workspace.

4. Click OK.

   The new workspace environment is displayed.

Building a workspace

You build a workspace by adding folders and models to your work environment.

You can create the following new items in a workspace by right-clicking the workspace in the Browser and select New ➤ *item.*:

♦ Folder

♦ Models: CDM, PDM, OOM, BPM, XSM, ILM, RQM or FEM

♦ Multi-model reports

You can add the following existing items to a workspace by right-clicking the workspace in the Browser, selecting Add, browsing to the item, and clicking OK:

♦ Models: CDM, PDM, OOM, BPM, XSM, ILM, RQM or FEM

♦ Multi-model reports

♦ Other workspaces you want the current workspace to reference

♦ External files you want the workspace to reference such as MS Word files, or Excel spread sheets. When you have external files referenced in a workspace, you can open them by double-clicking

Modifying the filename of a closed item

To modify the filename of closed models, multi-model reports, or any document type (workspaces or external files) you can right-click the item in the Browser and selecting Properties, the Document Properties dialog box allows you to directly modify the item filename.

Saving a workspace

When you save a workspace, you save the global architecture of the work environment, you do not save individual model files.

You should save the workspace whenever you modify the modeling architecture by performing the following actions:

♦ Creation, addition, modification or deletion of a folder

♦ Creation, or detachment of a model

♦ Creation, or detachment of a report

When you save a workspace you do not save the model or report files it contains. However, for any new model created in the workspace a Save As dialog box prompts you to indicate a name and file for the workspace to save the model location.

❖ **To save a workspace**

1. Select File ➤ Save Workspace.

☞ For more information about saving models, see the "Managing Models" chapter.

Saving a workspace under another name

You can save an existing workspace under another name.

❖ **To save the workspace under another name**

1. Select File ➤ Save Workspace As.

A standard Save As dialog box is displayed.

2. Type a new name for the workspace and click OK.

The new workspace is displayed by default.

Saving a workspace and its contents

You can save the workspace and the files it contains.

❖ **To save the workspace and its contents**

1. Select File ➤ Save All.

   The workspace and the model or report files it contains are saved. You are not prompted to save the individual files.

Deleting a workspace

Deleting a workspace does not affect the folders and models it contains. When you delete a workspace, you only delete the corresponding file existing on your machine, you do not delete any of the model or report files it contains.

If you want to delete a workspace file, you have to use the standard file deletion procedure from the Windows Explorer.

❖ **To delete a workspace**

1. Right-click the workspace file in the Windows Explorer and select Delete.

Folders

A folder is an optional container designed to help you organize the hierarchy within a workspace. You use a folder to structure the contents of the workspace in your local work environment. A folder cannot contain model objects, it only contains other container objects like folders, models, or multi-model reports. Folders are saved in the workspace file.

Creating a folder

Folders are not automatically created in PowerDesigner.

You can create folders from the workspace or from another folder.

❖ **To create a new folder**

1. Right-click the workspace or the folder where you want to create a new folder and select New ➤ Folder.

   The new folder is created below the selected item in the tree structure.

2. Type a name in the highlighted area. Folder names must be unique in the folder hierarchy.

Renaming a folder   You modify the folder property from the Browser.

❖ **To rename a folder**

1. Right-click the folder to rename and select Rename.

   or

   Select the folder and press F2 to edit the name in place.

Creating an item in, or adding an item to, a folder

A folder is a container for sub-folders, models and multi-model reports.

❖ **To create a new item in a folder**

1. Right-click the folder and select New ➤ *New item.*

   The new item is inserted in the folder.

You can add an item to a folder by copying or moving it using the drag and drop feature. However, you cannot paste or move an open model into a folder.

> **Multi-selection**
> You can add several items simultaneously by selecting them using the
> CTRL key and dragging them to the target folder. However, if you select an
> unauthorized item or an open model, the drag and drop will fail.

❖ **To add an item to a folder**

1. Select an item.

2. Press the CTRL key and drag the item to the folder if you want to copy the
   item.

   or

   Press the SHIFT key and drag the item to the folder if you want to move
   the item.

   The new item is displayed in the folder.

Deleting a folder

When you delete a folder, you detach the models and reports it contains from
the workspace, but you do not delete the models and reports themselves.

> *Caution*
> *Deleting a folder cannot be undone.*

❖ **To delete a folder**

1. Select the folder in the tree view.

2. Right-click and select Edit ➤ Delete.

   or

   Click the Delete tool from the PowerDesigner toolbar.

   or

   Select Edit ➤ Delete from the PowerDesigner menu bar.

Windows

The PowerDesigner main window is divided into different windows with independent behavior.
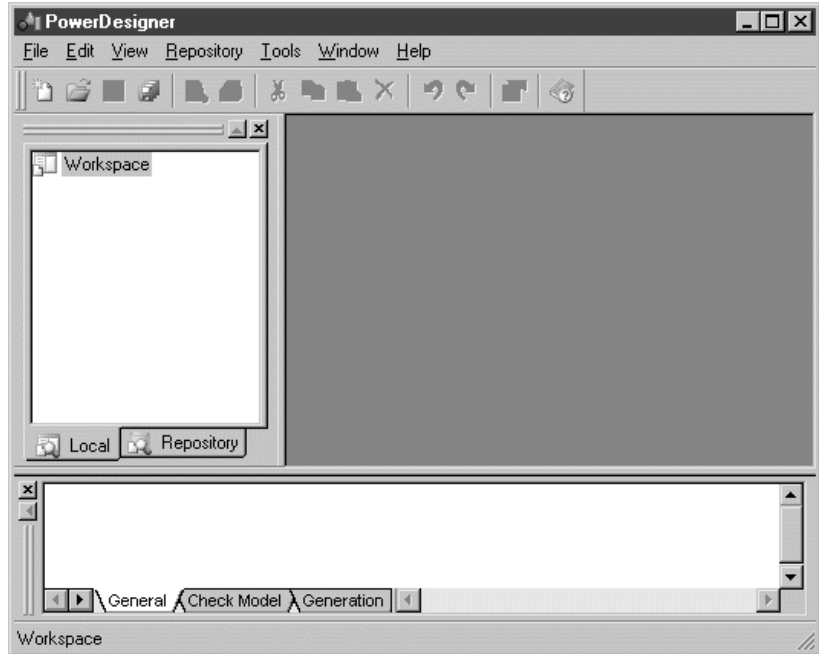
In the upper part of the window, you can see the PowerDesigner general title bar and the menu bar. Below the general menus, the standard toolbar displays tools for carrying out standard operations such as Open, Save or Undo.

Two dockable windows are displayed by default when you open PowerDesigner: the Browser on the left and the Output pane at the bottom.

On the right hand side, the work area is empty upon opening. This area is used to display the different MDI child windows.

Docking windows

You can customize the PowerDesigner environment by docking windows in the interface. Dockable windows are windows that align themselves with the edge of another interface element, typically a window or a pane.

There are three dockable windows in the PowerDesigner interface:

Window	Purpose
Browser	Provides a hierarchical view of the contents of the workspace
Output pane	Displays PowerDesigner messages during check model, generation, reverse operations, or script execution
Result List	Displays result lists for a Find or a Check Model operation

❖ **To dock a window**

1. Click the window title bar and keep the mouse button pressed.

2. Drag the selected window to another location in the main window.

If you drag the window close to another interface element   The dockable window automatically aligns itself with the edge of the closest interface element.

If you drag the window anywhere in the work environment   The window is resized and displays a title.

❖ **To disable the docking feature**

1. Right-click the dockable window background and deselect the Docking View command.

The window becomes an MDI window.

> **Disabling the docking feature temporarily**
> If you want to move a window in the PowerDesigner environment without using the docking feature you have to press CTRL while you drag the window.

❖ **To hide a docking window**

1. Click the X button in the window title bar.

❖ **To restore a window**

1. Select View ➤ Browser.

    or

    Select View ➤ Output.

    or

    Select View ➤ Result List.

Toolbars

Toolbars are designed to provide quick access to specific menu commands and to ease creation of objects related to a target language.

By default, the Standard toolbar is displayed when you start PowerDesigner for the first time. You can display more toolbars in the interface, this is saved in the registry.

Creating a toolbar

You can create new toolbars in the PowerDesigner environment.

❖ **To create a new toolbar**

1. Select Tools ➤ Customize Toolbars to display the Toolbars dialog box.

2. Click the New button to display the New Toolbar dialog box.

3. Type a name in the Toolbar Name box.
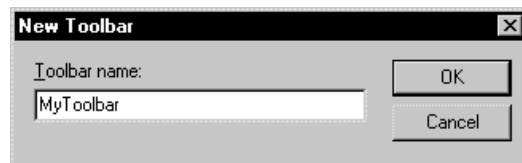
4. Click OK.

   The new toolbar is displayed in the list of toolbars, it is empty.

5. Select the new toolbar in the Toolbars list and click Customize to display the Customize Toolbars dialog box.

6. Select a menu category in the Categories pane.

The tools corresponding to the commands of the menu appear in the right pane of the window with a description in the lower part of the window.

7. Select the tools you want to add, drag them to the new toolbar, and release the mouse button.

The tools are inserted into the new toolbar.

Deleting a tool in a toolbar
To delete a tool in a toolbar, you must be in Customize mode, then right-click the tool to be deleted and select Delete from the contextual menu.

8. Click Close in each of the dialog boxes.
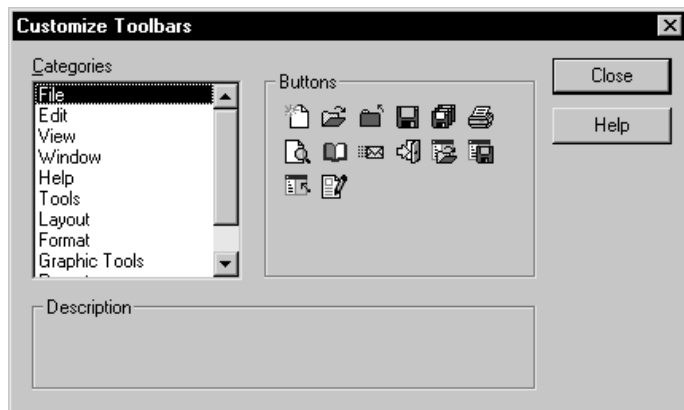
Customizing a toolbar

You can customize the PowerDesigner toolbars according to your needs.

The customization options are the following:

Option	Description
Show Tooltip	Activates or deactivates the display of small pop-up windows that provide descriptive labels for a control or graphic object
Flat button	Toggles the display of buttons between flat and 3D
Customize	Opens the toolbar customization dialog box in which you can select icons corresponding to commands of the PowerDesigner menus

Grouping related tools on a toolbar
You can group related tools on a toolbar by adding a separator bar before the first and after the last one in the group to distinguish the group from other tools on the toolbar.

To do so, right-click the tool before which you want to add a separator bar and select Begin Group.

Custom Graphic tools category
The Custom Graphic Tools category lets you define custom tools for stereotyped objects, as defined in extended model definitions.

☞ For more information, see the "Extended Model Definitions Reference Guide" chapter in the *Advanced User Documentation*.

❖ **To customize a toolbar**

1. Select Tools ➤ Customize Toolbars to display the Toolbars dialog box.

2. Select a toolbar in the list of toolbars.

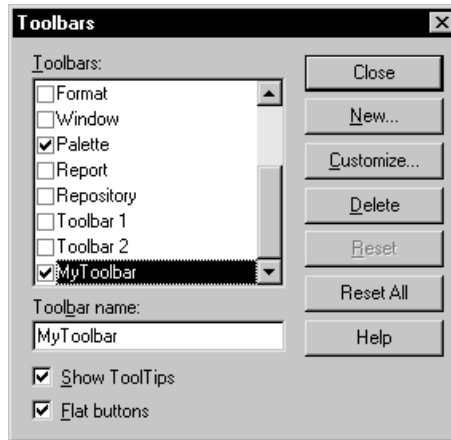3. Select or clear the customization check boxes in the lower part of the dialog box.

4. Click the Customize button.

   The Customize Toolbars window is displayed.

5. Select a menu category in the Categories pane.

   The tools corresponding to the menu commands appear in the right pane of the window, with a description in the lower part of the window.

6. Select the tool you want to add, drag it to the desired toolbar, and release the mouse button.

   The tool is inserted into the target toolbar.

7. Click Close in each of the dialog boxes.

Restoring the default settings of a toolbar

You can restore the default settings of a customized toolbar using the Reset button in the Toolbars dialog box.

If you want to reset the default settings of all toolbars you have to click the Reset All button.

Hiding and showing a toolbar

By default, the Standard toolbar is displayed in the PowerDesigner main window and the Palette in the diagram window. You can choose to hide or show a toolbar from the PowerDesigner interface.

❖ **To hide a toolbar**

1. Select Tools ➤ Customize Toolbars to display the Toolbars dialog box.

2. Clear the check box corresponding to the toolbar you want to remove from the interface.

   or

   Select the check box corresponding to the toolbar you want to add to the interface

   The toolbar instantly is displayed or disappears from the interface.

3. Click Close.

Deleting a toolbar

You can delete any user-defined toolbar but you cannot remove a default PowerDesigner toolbar.

❖ **To delete a user-defined toolbar**

1. Select Tools ➤ Customize Toolbars to display the Toolbars dialog box.

2. Select a user-defined toolbar in the list of toolbars and click the Delete button.

3. Click Close.

Property Sheets

The properties of model objects can be viewed via their property sheets.

The title bar of the property sheet identifies the object, and provides the name and code of the object.

Property sheets group properties in different tabs and you can customize their display. Most PowerDesigner objects share the following standard tabs, which are displayed by default the first time you open the product:

Standard tabs	Description
General	Contains information that identifies the object (Name, Code).
Rules	Contains business rules applied to the object.
Notes	Contains notes (Description and Annotation) for the object.
Version Info	Contains information about the origins of the object.
Dependencies	Shows items that depend on the current object and influence other objects.
Extended Dependencies	Contains user-defined dependencies of the object.

Displaying a property sheet

❖ **To display a property sheet**

1. Double-click an object symbol in the diagram.

   or

   Select an object symbol and press ALT + ENTER.

   or

   Right-click the object symbol and select Properties.

   or

   Right-click an object in the Browser and select Properties.

Opening property sheets at last accessed tab
You can choose to open property sheets at the last accessed tab by selecting Tools ➤ General Options ➤ Dialog, and selecting the option Keep Last Tab in the Property Sheets groupbox.

Link objects property
sheet

You can also double-click any label (name, code, role or cardinality) of the
link object to open its property sheet.

Property sheets of link objects display a graphic representation of the
relations between model objects in the upper part of the property sheet
dialog box.
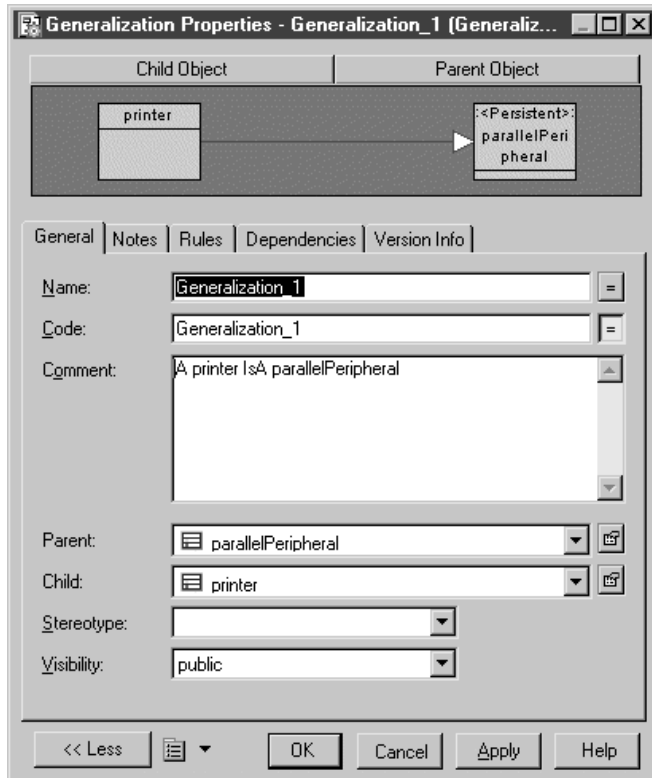
The graphic always shows the link together with its extremities. Some text
such as the cardinality value and the role is also displayed. The symbols and
background color together with the text format correspond to the default
display preferences saved in the registry.

Each time you want a modification of the display preferences to be taken
into account when you display the link object property sheet, you must click
the Set As Default button in the Display Preferences dialog box 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.
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:
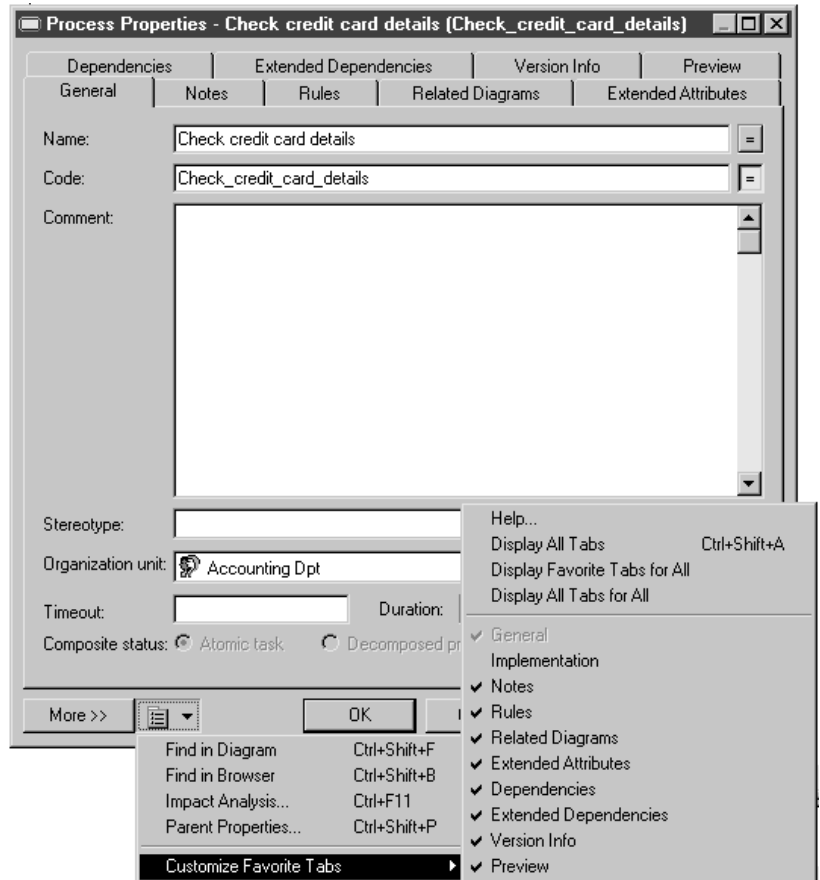
Any changes in the display of tabs are immediately applied to the current property sheet but not to other property sheets that are currently open. The changes become the default setting for any property sheet of the same type.

Displaying tabs on one or several rows

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.

Inserting properties in the General tab

The General tab of a property sheet contains information that identifies the object: the name should be clear and understandable and the code should be shorter for generation needs. You can use up to 254 characters in the Name

and Code boxes but your current language or DBMS may be more restrictive. The comment is used to provide additional information about the object, there is no limitation in the Comment box, and you can type as many characters as needed to specify the properties of the selected object.

❖ **To insert properties in the General tab**

1. Type a name in the Name box, the Code box is automatically filled if the Name to code mirroring general option is selected.

   or

   Type a code in the Code box if the Name to code mirroring general option is not selected.

   You can also use the Equals button at the end of the Name or Code box.

2. Type a comment in the Comment box.

3. Click Apply to validate the new properties.

Applying a business rule to an object

A business rule is a rule that your business follows. A business rule could be a government-imposed law, a customer requirement, or an internal guideline.

Business rules complement model graphics with information that is not easily represented graphically.

You can add business rules to the definition of the current object from its property sheet.

U Column in the List of business rules

When you apply a business rule to an object, the U (Used) column beside this business rule is automatically checked in the List of business rules to indicate that the business rule is used by at least one object in the model. The U column allows you to visualize unused business rules. You can then delete them if necessary.

❖ **To apply a business rule to the current object**

1. Open the property sheet of an object and click the Rules tab.

2. Click the Add Rules tool.

3. Select one or more business rules in the selection dialog box.

4. Click OK in the selection dialog box, the rules appear in the list of rules.
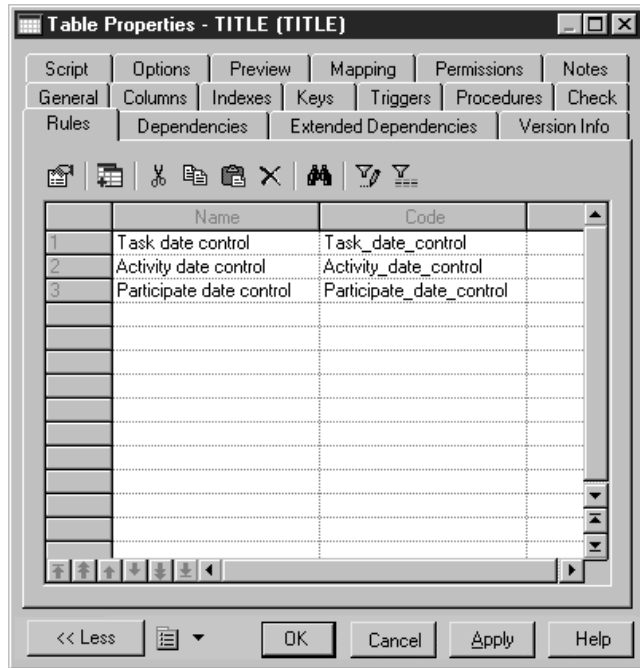
5. Click OK to return to the model diagram.

Attaching a requirement to an object

You can attach one or more requirements to an object in a model using the Requirements tab in the object property sheet. This tab is displayed if you select the **Enable links to requirements** model option, and can be used if at least one Requirements model is open in the workspace.

☞ For more information about this model option, see the user's guide of each PowerDesigner model.

When you double-click a row in the Requirements tab of an object property sheet, you open the external shortcut property sheet for the selected requirement. If you want to open the property sheet of the requirement target, click the Target Object Properties button beside the Name box. You can also select Tools ➤ General Options ➤ Dialog from the Menu bar and select the Target Object option beside External Shortcut to define a default open mode for shortcut property sheets.

❖ **To attach a requirement to an object**

1. Open an object property sheet and click the Requirements tab.

2. Click the Add Objects tool in the toolbar to open a selection list of Requirements models.

3. Select one or more requirements and click OK.

   The selected requirements appear in the Requirements tab.

4. Click OK to close the property sheet.

Adding notes to an object

From the Notes tab, you can document objects using an RTF editor in the Description and Annotation tabs.

☞ For more information, see "Attaching notes to an object" in the "Managing objects" chapter.

❖ **To add notes to an object**

1. Double-click an object in the diagram to display the object property sheet.

2. Click the Notes tab, the Description sub-tab is displayed by default, you can type a description.

3. Click the Annotation tab in the lower part of the object property sheet to change tab and type annotations.

4. Click OK.

Displaying object dependencies

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

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

♦ Internal, when the links are within a model. These dependencies are saved in the model and appear in the Dependencies tab of an object property sheet

♦ External, when the links exist between models. These dependencies are created during intermodel generation or external shortcut creation, they appear in the Dependencies tab if the related model is opened in the workspace. If the related model is not available, you can use the repository to retrieve external dependencies

☞ For more information about how to retrieve external links using the repository, see "Auditing repository activities" in the "Managing Repository Documents" chapter in the Repository User's Guide.
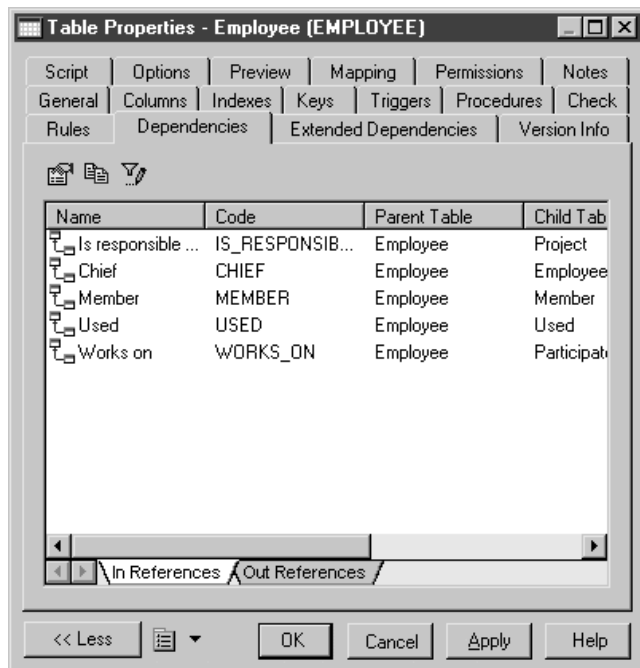
PowerDesigner lists these links in the Dependency tab of the object's property sheet.

Model dependencies
The Dependencies tab of a model also lets you check the model origin (Generated From tab) and destination (Generated As tab).

❖ **To display object dependencies**

1. Open the property sheet of an object and click the Dependencies tab.

2. Click the different sub-tabs to see the depending objects or the uses of the object throughout the open models of the workspace.

3. Click OK.

Using extended dependencies

You can use extended dependencies to specify additional forms of collaborations between PowerDesigner objects. These links are allowed

among objects of any type of model, but are not interpreted and checked by PowerDesigner, as they are meant for documentation purposes only.

You can complement these links by creating **stereotypes** to specify their meaning. Stereotypes are used to define extended dependencies in all PowerDesigner modules, and appear in the Extended Dependencies tab of each non-link object property sheet. You can type them directly in the Stereotype column or select a value from the list if you have previously defined stereotypes in an extended model definition (.XEM) applied to or embedded in the model.

☞ For more information about stereotypes in an extended model definition, see "Working with Extended Model Definitions" in the "Managing Objects" chapter.

Example

The following example illustrates how extended dependencies can help you document relations between package or model objects.

This example about generation from a CDM to two PDMs generated with different DBMS shows that the file object SourceCDM.cdm has extended dependencies with the file objects GeneratedPDM.pdm (for Sybase IQ) and GeneratedPDM2.pdm (for ASA7). These extended dependencies are defined by the <<generation>> stereotype to explain that the two PDM come from the same CDM.
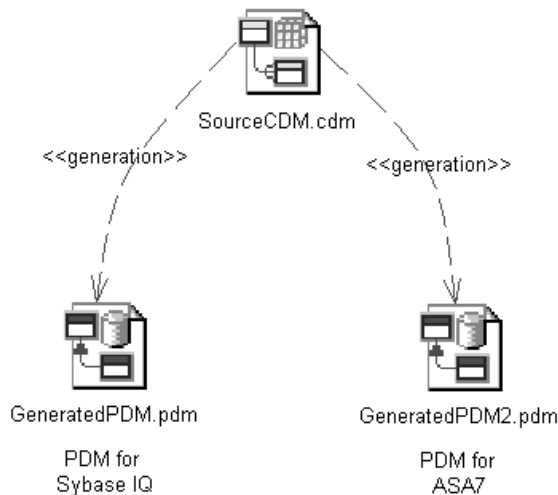
☞ For more information, see the "Using file objects" section in the "Managing Objects" chapter.

Creating an extended dependency

The object from which you create the extended dependency is the **dependent object** and the object towards which it goes is the **influent object**.

You can create an extended dependency in either of the following ways:

❖ **To create an extended dependency using the Link/Extended Dependency tool**

1. Click the Link/Extended Dependency tool in the Palette.

2. Click inside the dependent object and while continuing to hold down the mouse button, drag the cursor to the influent object where you release the mouse button.

Moving extended dependencies

Extended dependencies behave like oriented links. When you move the dependent object, you also move the extended dependency attached to it. And when you move the influent object, it is replaced with a shortcut to preserve the extended dependency definition.

Note that the Link/Extended Dependency tool cannot create an extended dependency between objects belonging to different types of models as these cannot have symbols in the same diagram.

❖ **To create an extended dependency from the dependent object property sheet**

1. Open the property sheet of a non-link object, and click the Extended Dependencies 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, a package from the list, and an object from one of the sub-tabs.

4. Click OK.

   The extended dependency you have created is displayed simultaneously in the list of extended dependencies but also in the diagram if objects at both extremities are displayed.

5. Click the Stereotype column for the extended dependency you have just created and type a stereotype or select a stereotype from the list if available.

6. Click OK.

Displaying object version information

PowerDesigner automatically manages version information about objects in the model. All objects with property sheets have a Version Info tab, which displays the following information:

Version Info	Description
Creation User	Name of the User who created the object.
Creation Date	Date of creation of the object
Last Modification User	Name of the User who made the last modification to the object.
Last Modification Date	Date of the last modification of the object
Generated From Origin Object	Name of the origin object. Displayed when the object has been generated from another object. Click the Origin Object Properties button to open the origin object property sheet

This information is read-only, you cannot modify any value in this tab.

Finding text using regular expressions

A regular expression is a pattern of text that describes one or more strings to match when searching for text. You can use regular expressions to find script text displayed in the Preview or Script tab of a table property sheet, 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 Find dialog box (or Replace in Items dialog box in the Resource Editor) and type a regular expression in the Find What box.

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

Metacharacter	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"

Metacharacter	Description
*	Matches the preceding character zero or more times Examples: "zo*" matches "z" and "zoo"
+	Matches the preceding character one or more times Examples: "zo+" matches "zo" and "zoo", but not "z"
?	Matches the preceding character zero or one time Examples: "to?" matches either "t" or "to"
.	Matches any single character except the newline \n Examples: ".ork" matches "Work", "Fork" etc
[ ]	Matches any one of the enclosed character Examples: "[abc]" matches "a", "b", or "c" A range of character can be indicated with a dash "[a-z]"

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

Validating changes in a property sheet

To validate changes in a property sheet you can use the Apply button to commit property changes and the OK button to save the property sheet contents and close it.

Auto commit mode
You can modify the property sheet validation mode in the Options dialog box. If you select the Auto commit mode, any change is committed as soon as you change field in the property sheet. You have to use the Undo feature to cancel a change.

☞ For more information, see "Dialog box general options" on page 105.

❖ **To validate changes in a property sheet**

1. Click Apply if you are not in Auto Commit mode.

   or

   Change field in the property sheet if you are in Auto Commit mode.

Closing a property sheet

❖ **To close a property sheet**

1. Click OK or Cancel if you are not in Auto Commit mode.

   or

   Click the Close button if you are in Auto Commit mode.

Lists

PowerDesigner lists show objects of the same type belonging to the current package or model. You can add or create objects in lists

You can multi-select items in a list in order to perform mass changes on object properties. You can select all items in the list by clicking the top-left corner box in the list.

You can right-click an object in the list to access a contextual menu offering the following options:

♦ Find in Diagram

♦ Find in Browser

♦ Impact Analysis

♦ Edit

♦ Properties

♦ Extended menus (created using the Resource Editor)

The properties of the listed objects are organized in columns. Depending on the properties of the listed objects, the list may have a natural order. For example, a list of columns has an order that must be preserved.

Global lists   Global lists are accessible from the Model menu or from the contextual menu in the Browser, and show all the objects of the type in the model. The title bar of the list identifies the object type, for example, List of Columns.

91
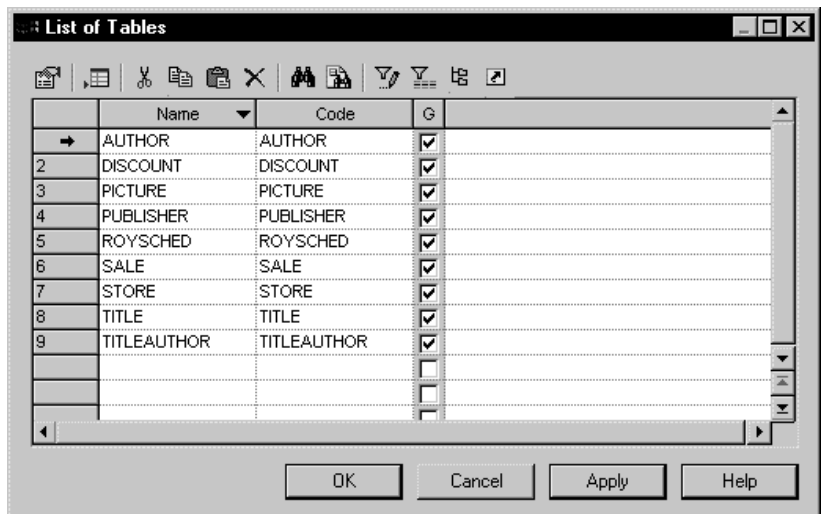
You can open a global list in either of the following ways:

♦ Select Model ➤ *Item* .

♦ Right-click a model in the Browser and select List of ➤ *Item* from the contextual menu.

Displaying shortcuts and objects from sub-packages in a list
You can enable the display of shortcuts in a list by clicking the Include shortcuts tool. Shortcuts are grayed as they cannot be modified. When you include shortcuts whose target model is closed, some information may be unavailable. You can enable the display of Include Sub-packages tools are enabled in the selected list, all the object shortcuts of the current package and those of the sub-packages appear.

Child lists   Child lists are available on certain tabs of object property sheets, and list all of the child objects of that type belonging to the object

For example, when you select the Columns tab in a table property sheet you display the list of columns in the selected table:
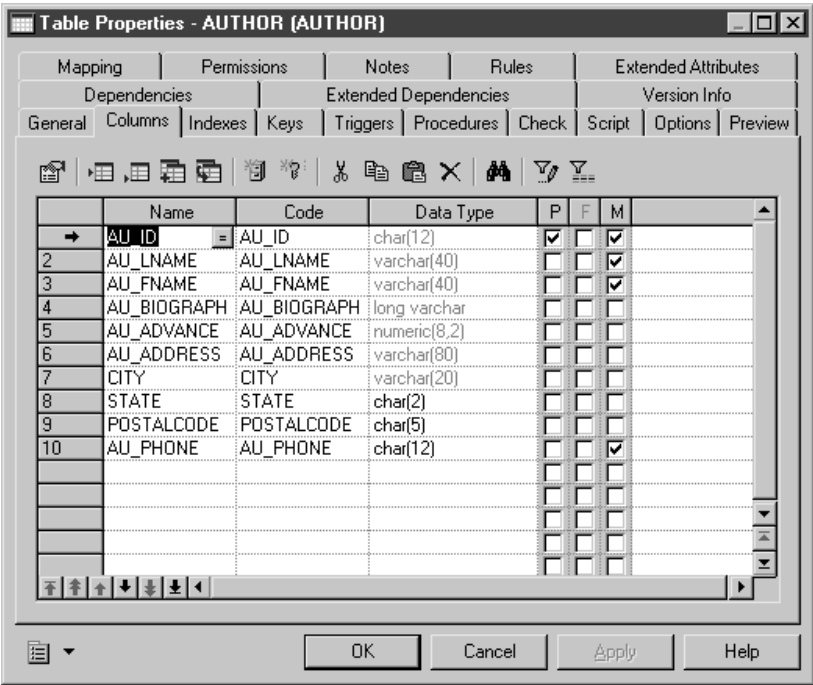
List tools

The following tools are available in PowerDesigner lists:

Tool	Description
	Opens the property sheet of the selected item
	[ordered lists only] Inserts a row before the selected row in the list
	Adds a row at the end of the list
	Opens an object selection dialog box to select objects and add them to the list. When you select an item you copy it to the list.
	Creates a new object and opens the corresponding property sheet
	[CDM only] Opens an object selection dialog box to reuse objects. When you select an item you create a link to the original item, you do not copy it.
	Deletes the row and stores it in the Clipboard
	Copies the selected row to the Clipboard
	Pastes the contents of the Clipboard
	Deletes the row
	Opens a Find dialog box to search an item in the list
	Finds the symbol in the diagram
	Opens a Filter dialog box to define a filter on the list
	Enables the filter on the list
	Includes objects in sub-packages in the list

Tool	Description
🖳	Includes composite objects in the list (like sub-process, sub-activity, sub state, etc.)
🔲	Includes object shortcuts in the list
🖫	[Related Diagrams tab] Opens a target model or a diagram**
✏	Opens an object selection list to change the target object of a shortcut

Modifying properties in a list

You can modify the properties of items in a list.

By default, you have to use the Apply button to commit changes and the OK button to save the list contents.

You can modify the validation mode of lists in the Options dialog box. If you select the Auto commit mode, any change is committed as soon as you change a field in the list. You have to use the Undo feature to cancel the change.

☞ For more information, see "Dialog box general options" on page 105.

❖ **To modify properties on a list**

1. Select Model ➤ *item* to display a list dialog box.

2. Select an item in the list and type changes in the columns corresponding to object properties.

3. Click OK.

Adding items to 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. 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.

❖ **To add items at the end of a list**

1. Click the Add a Row tool or click an empty row at the end of the list

❖ **To insert an item at a particular point of an ordered list**

1. Select the item before which you want to insert a new item in the ordered list, and click the Insert a Row tool.

   The new item is inserted before the selected item in the list. It is displayed with a default name you can modify.

| 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: |

Button	Moves cursor to
⬆	Move to the top of the list.
⬆	Move up one page
⬆	Move up one line
⬇	Move down one line
⬇	Move down one page
⬇	Move 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 or its packages and objects contained in other models or other packages of these 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 extended dependencies.

Object selection   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.

Tool	Description
-	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
	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 check boxes in the current object type tab. To selects all check boxes in all object type tabs you can click the arrow and select All Lists or you can press the CTRL key and click the Select All tool.
	Deselect All - Deselects all check boxes in the current object type tab. To clear all check boxes in all object type tabs you can click the arrow and select All Lists or you can press the CTRL key and click the Deselect All tool.
	Use Graphical Selection - Uses the graphical selection.
	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 - Allows the definition of filter expression on the columns of the selection list.
	For more information about the Customize Columns and Filter dialog box, see "Filtering a list" on page 101.

Tool	Description
⧗	Enable/Disable Filter - Enables or disables the filter. The currently available filter is displayed in the Filter box underneath the object list.
⧗	Use Filter for Selection - Selects objects based on the filter parameters.
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. Note that in case of a database reverse engineering by ODBC, object selections are saved into separated files, as you can reverse a database without having any model open in the Workspace.

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. For a database reverse engineering by ODBC, you have to select a folder before being able to save the object selection into a separated file.

☞ For more information about database reverse engineering by ODBC, see the "Reverse Engineering a database into a PDM" chapter in the *PDM User's Guide* .

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 ODBC Reverse Engineering dialog box

In this case, some objects that have already been selected will no longer be displayed, and a dialog box opens offering you the following options:

♦ 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.

When you confirm your selection by clicking OK, the confirmation dialog box will not be displayed again, while you continue to edit your selection, even if you again modify your parameters.

Selecting an object from a selection tree

Browse tab   You can expand the tree in the Browse tab of the Select *object* dialog box to select an item.

This dialog box opens when for example you:

♦ Select an implementation process in another diagram in the BPM

♦ Insert an RQM in an existing Word document.

♦ Select any object attached to an object state in an OOM

♦ Select a classifier as data type for a class attribute in a class diagram in an OOM

If you double-click an object name in the tree view, you commit the selection and close the dialog box.
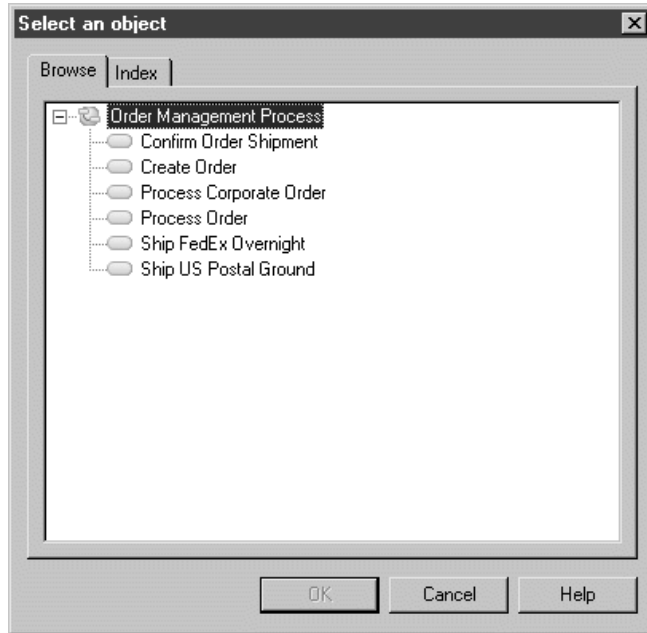
Index tab   From the Index tab, you can search for an object whose name you know
                   fully or only partially but do not know where the object is defined. The
                   search uses the Display Name/Code option you have defined in the model
                   options. Object names are sorted alphabetically in the list. When you type a
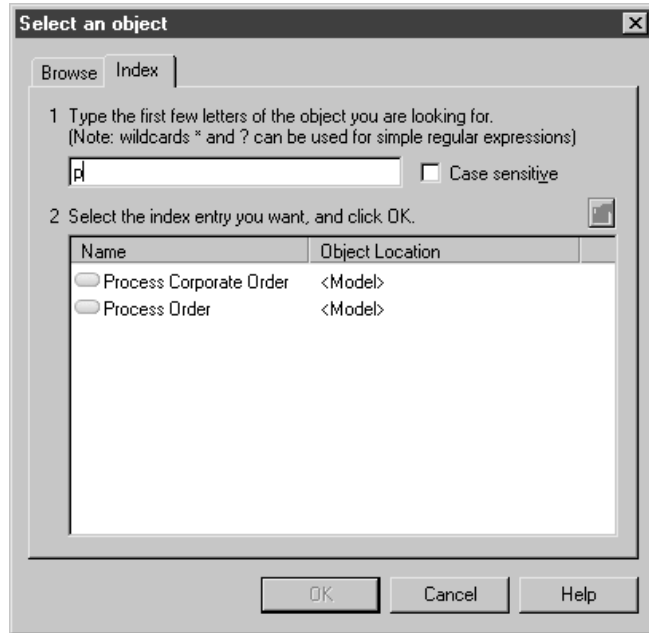   name or even just a letter in the expression field, the list automatically
                   returns the corresponding results.

                   The Properties button above the list allows you to open in read-only mode
                   the property sheet of the selected object. If you select an object name in the
                   list of the Index tab, then click the Browse tab, the object name is
                   automatically selected in that tab. In the same way, when you select an
                   object name in the Browse tab tree view, then click the Index tab, if empty,
                   the expression field is initialized with the selected name, otherwise the
                   expression content is preserved.

                   If you double-click an object name in the list, you commit the selection and
                   close the dialog box.

                   Note that some selection dialog boxes, which display objects in a tree view
                   may not have Browse and/or Index tabs.

Select an object　×

Browse | Index |

1　Type the first few letters of the object you are looking for.
(Note: wildcards * and ? can be used for simple regular expressions)

`p|`　　　　　　　　　　□ Case sensi*t*ive

2　Select the index entry you want, and click OK.

| Name	Object Location
◯ Process Corporate Order	<Model>
◯ Process Order	<Model>

OK　　Cancel　　Help

Filtering a list

A filter restricts the list to those items that meet certain criteria.

PowerDesigner allows you to hide columns, modify column order, or define
a filter expression to modify the content of the rows.

❖ **To customize the display of columns and/or define a filter expression**

1. Click the Customize Columns and Filter tool to open the Customize
   Columns and Filter dialog box.

2. Perform any of the following functions to filter the list:

   ♦ Select columns to display by checking the D column

   ♦ Enter any expressions to filter by in the Expression column. The Used
     check box in the U column is automatically selected when you enter a
     filter expression. If you want to keep the expression in memory but
     disable it, you can clear this check box.

   ♦ Use the arrows at the bottom of the dialog to change the order of
     columns in the list. You can press SHIFT and ENTER or press SHIFT and
     click OK to move all the selected rows on top of the list.
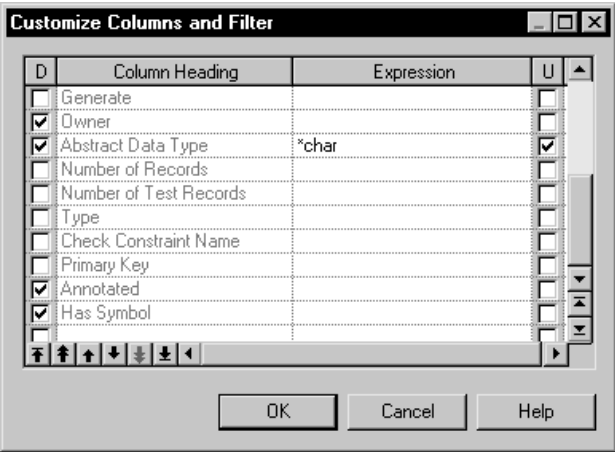
3. Click OK to return to the list. The filter is applied by default, and the Enable/Disable filter tool is depressed. Click this tool to toggle between enabling and disabling the filter

You only customize the display of the current list. When you have defined customization parameters, you have to apply them to the list.

Defining a filter expression

The Expression field in the Customize Columns and Filter dialog box is not case-sensitive and accepts the following wildcards:

Wildcard	Description
*	Any string (from none to any number of characters)
?	Any character
*	* is a normal character
\?	? is a normal character

The following are examples of filter expressions

Expression	String values allowed
W*	Work
	Washington
*CODE	AREA CODE
	COUNTRY CODE
	CITY CODE

Expression	String values allowed
????ER	Writer
	Seller
*_emp_???	div_emp_fun (code for the function of the employee in the division)
	_emp_idn (code for the identification of the employee in the division)
*\?	Is this book ready for production?

Closing a list

❖ **To close a list**

1. Click Close if you are in Auto Commit mode.

   or

   Click OK or Cancel if you are not in Auto Commit mode.

Setting PowerDesigner General Options

You can control the look and behavior of the PowerDesigner interface through the general options. These options 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
Delete: Confirm object deletion	Display a confirmation dialog box when you delete an object.
Browser : 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.
Browser drag and drop: Default action	Allows you to 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 palette
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.
Name to Code mirroring	Automatically updates the name or code of an object to reflect changes in the other.

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 Short-cut	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 Short-cut	Controls whether double-clicking an external shortcut opens the property sheet for the shortcut or for the target object.

☞ For more information, see the "Managing Shortcuts" chapter.

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.

❖ To define a text editor

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

The following variables are created when you install PowerDesigner:

Variable	Description	Default
CMD	Windows command interpreter	command.com or cmd.exe
HOME	Variable defining the default home directory	—
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

You can edit these variables and add your own.

❖ To define an environment variable

1. Select Tools ➤ General Options, and click the Variables category in the left-hand pane.

2. Click in the row of an existing variable in order to edit its values, or click the Add a Row tool to create a new variable.

3. Click OK to close the dialog box.

Variables defined here are used in commands in the Generation\Commands sub-category of the JAVA object language, and can be used in the Generation Template Language.

The syntax for using these variables in GTL requires that you add $ before the variable name within the % signs, for example %$CMD%.

☞ For more information about the GTL, see the "Generation Reference Guide: GTL" chapter in the *Advanced User documentation* .

Defining named paths

When you add a document or a model to the workspace, create an external shortcut, or perform various other file operations, PowerDesigner saves the paths to these external files in the workspace or model files.

However, in a team environment, when people exchange model files, the links may be broken when one user opens the file of another .

To solve this potential problem, you can define **named paths** in PowerDesigner. A team leader can define a list of names corresponding to shared resource files and specify a folder structure. Then each team member recreates the named paths on his/her workstation.

PowerDesigner provides a number of predefined named paths, each of which is preceded by an underscore:

Name	Path
_DBMS	Folder where DBMS definition files are stored
_EXAMPLES	Folder where the demo examples are stored
_HOME	PowerDesigner installation folder
_LIBRARY	Folder where object libraries are stored
_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
_XEM	Folder where extended model definition files are stored
_XMLLANG	Folder where XML language definition files are stored

_HOME predefined named path

_HOME is a very useful named path that 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

By default, the path of a predefined named path corresponds to the installation path you have selected. If you use files proceeding from another directory, the path of the predefined named paths is not automatically modified. You have to carry out this change manually.

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 named path is created with the name and path of the modified predefined named path.

If you modify the path of a predefined named path and want to cancel the change, you have to select the predefined named path in the list, delete it and click OK in the General Options dialog box. The next time you open the Named Path page in the General Options dialog box, the deleted predefined named path is displayed in the list with the default installation path.

How named paths are used

Named paths are used when you save a file and open it.

Saving a file

When you save a PowerDesigner model or workspace, the name of a named path is substituted for the actual path . For example, if the HOME named path is defined as:

HOME = c:\ Program Files\ Sybase

And you have the following path in your model:

c:\ Program Files\ Sybase\ PowerDesigner\ tempo_samples

When you save this file, the path will be saved as:

%HOME%\ PowerDesigner\ tempo_samples

Opening a file

When you open a file, named paths are replaced by the value of that path in the local environment. For example:

%HOME%\ PowerDesigner\ tempo_samples

becomes:

c:\ Program Files\ Sybase\ PowerDesigner\ tempo_samples

If you open a file containing a named path not defined on the current machine, the Unresolved Named Path dialog box 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

Creating a named path

You can create additional named paths. Your team should agree on the names of any named paths to be created.

❖ **To create a named path**

1. Select Tools ➤ General Options, and then click the Named Paths category.

2. Click the Add a Row tool and enter a name and a path for the named path.
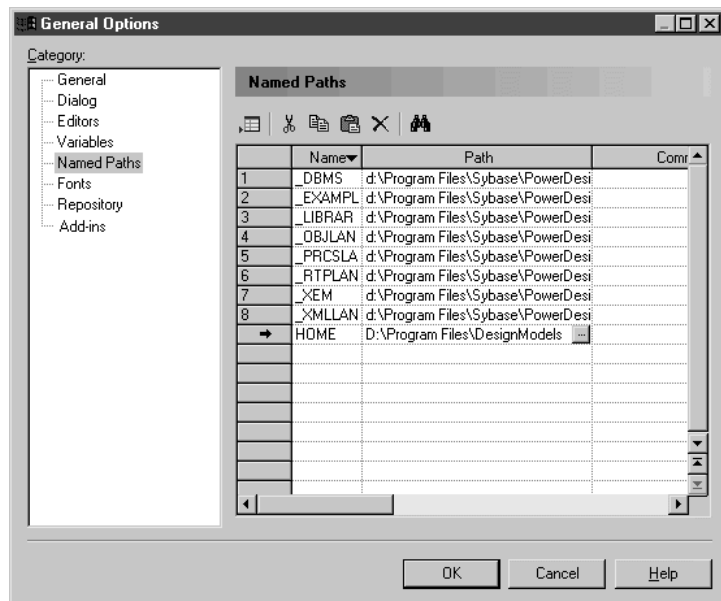
3. Click OK.

Defining default interface fonts

You can modify the default font proposed for:

♦ User interface (lists)

♦ Code editor (SQL preview)

♦ RTF editor (description, annotation)

❖ **To define a default font**

1. Select Tools ➤ General Options, and then click the Fonts category.

2. Specify the appropriate font format options.
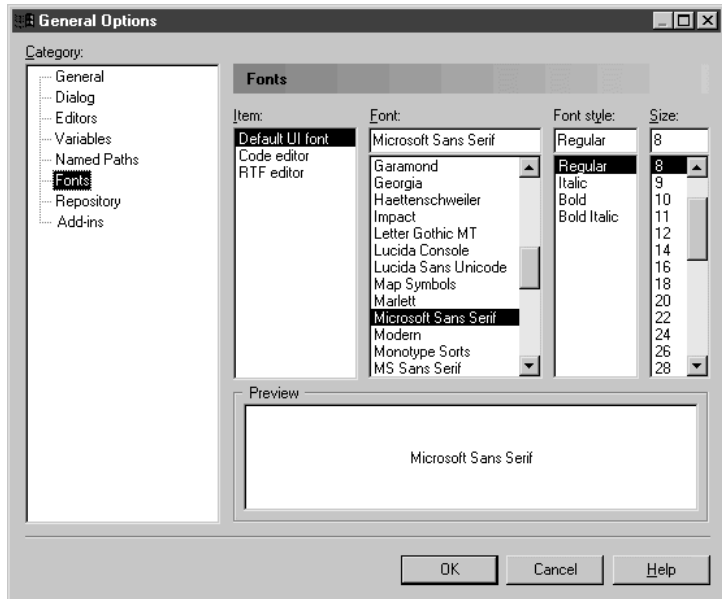
3. Click OK.

Managing add-ins

An add-in is a module that adds a specific feature or service to

PowerDesigner. PowerDesigner is installed with several add-ins.

Add-in	Description
Meta-integration Import Export	Uses the Meta Integration Model Bridge to import and export 3rd party file formats. Commands are available from the File ➤ Import and File ➤ Export menu. You need to purchase bridges from Meta Integration.
	www.metaintegration.net
Microsoft Analysis Services	Imports multidimensional data from MS SQL Server into a PDM. Microsoft Analysis Services must be installed on the current machine.
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.
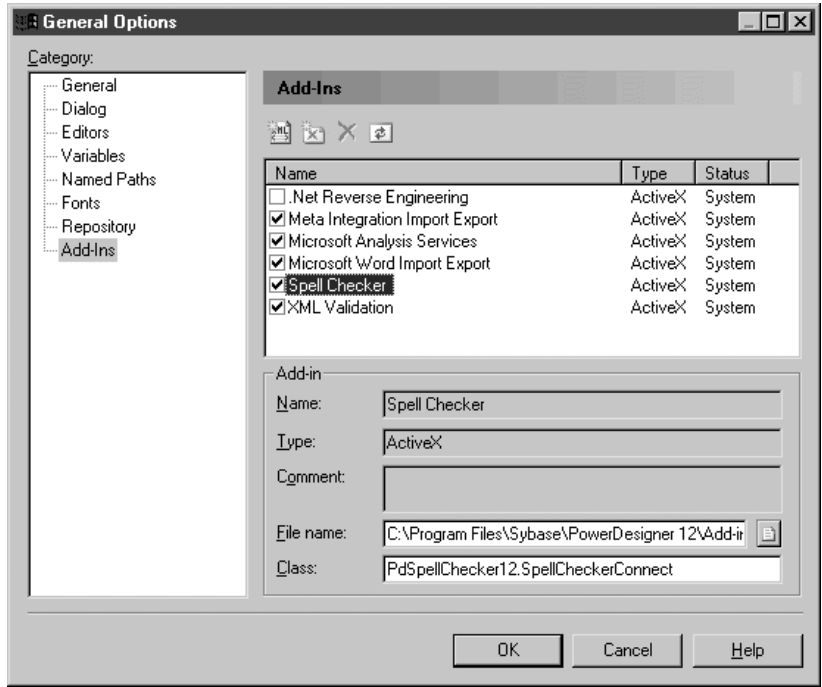
.Net reverse engineering In order to use the .Net binary reverse, you need to register the application reverseapp.exe.

.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.

System and user add-ins | 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.

An add-in has the following properties:

Property	Description
Name	Name of the add-in, this name must be unique in the PowerDesigner environment. System add-in names cannot be modified
Type	Add-in type, it can be ActiveX or XML.
	An ActiveX add-in implements a specific interface that defines methods. These methods 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
Comment	Additional information about the add-in. System add-in comments cannot be modified
File	XML file containing the menu and command specifications for an XML add-in
	DLL or EXE files for an ActiveX add-in
Class	[ActiveX only] Name of the class that implements the ActiveX add-in interface

Declaring a user-defined add-in

You can create your own XML or ActiveX add-ins. For more information about the add-in creation procedure, see the *Scripting User's Guide* .

When you create an add-in you must declare it and enable it in the Add-ins page of the General Options dialog box.

❖ **To declare a user-defined add-in**

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.

3. Enter the name of your add-in in the Name box.

4. Select the XML file corresponding to the XML add-in or select a DLL or an EXE file corresponding to the ActiveX file. You can also use the browse tool to select a file.

5. (ActiveX only) Enter the name of the ActiveX implementation class in the Class box.
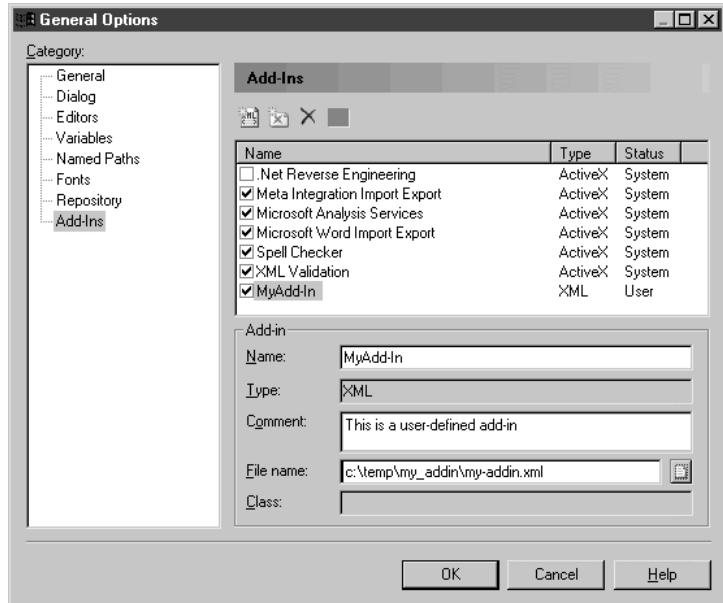
6. Click OK.

Using the PowerDesigner Eclipse Plugin

Eclipse is an open source platform for sharing web and software development tools on an integrated Java environment. Eclipse has its own features and integrated plugin.

Each PowerDesigner module has its own plugin for the Eclipse Platform. The PowerDesigner Eclipse plugins allow you to work with any of the PowerDesigner modules on the Eclipse Platform.

☞ For more information about the PowerDesigner Eclipse plugin, see *PowerDesigner Eclipse Plugin User's Guide* .

Using the PowerDesigner Visual Studio 2005 Plugin

Visual Studio 2005 is an integrated development environment with a particular focus on the .NET technologies.

The PowerDesigner Visual Studio 2005 plugin allows you to work with all the PowerDesigner modules from within Visual Studio 2005.

☞ For more information about the PowerDesigner Visual Studio 2005 plugin, see *PowerDesigner Visual Studio 2005 User's Guide* .

Managing Models

This chapter describes model management functions.

Contents

Working with Model Files

PowerDesigner supports the following types of models:

Model	Extension	Backup Extension
Requirements Model (RQM)	.rqm	.rqb
Business Process Model (BPM)	.bpm	.bpb
Conceptual Data Model (CDM)	.cdm	.cdb
Physical Data Model (PDM)	.pdm	.pdb
Information Liquidity Model (ILM)	.ilm	.ilb
Object-Oriented Model (OOM)	.oom	.oob
XML Model (XSM)	.xsm	.xsb
Free Model (FEM)	.fem	.feb
Multi-Model report	.mmr	.bmr

In PowerDesigner the model unit is the basic work environment. You can split it for organization purpose in **packages** and display it with several **diagrams or document views**, the entire set remains the fundamental basis for your modeling work.

Creating a model

You can create a new model:

♦ In the current workspace

♦ In a folder

New multi-model report   You can also use the New dialog box to create a multi-model report.

☞ For more information on multi-model reports, see chapter Managing a multi-model report in the *Reports User's Guide* .

Creating a new model

You can create a new model from the New dialog box. This dialog lets you choose between two creation modes:

New Model   You create a new model by default. In this mode, all options (like display

preferences or model options) are initialized from the PowerDesigner environment (either from the registry or from hard coded default values).

New model from template

You can create a new model using a model template. In this mode, the model options and preferences are initialized with a model template. A model template is a set of model options, display preferences, extension, or objects saved in a model located in the template folder. You use model templates when you need to reuse preferences and options in several models.

To create a model template you should create a new model, define the options, preferences, and environment of the template, and save this model into the template folder. You can create your own templates and save them in a folder. You define this folder for user-defined templates using the Change User-Defined Model Templates Folder tool. The list of templates displays the content of both system (if any) and user-defined template folders.

If you wish to turn an existing model into a template, you can use an accelerator for creating templates: click the **Copy Model to User-Defined Model Templates Folder** tool, browse for a model, select it and click Open. The model is copied to the folder of user-defined templates and becomes a template available in the list.

To modify a model template you have to modify the model saved as template, to do so: open the model saved in the template folder, modify options and preferences, and save it again in the template folder.

Creating a model from existing elements

There are several ways to create a new model from existing elements. You can:

♦ Import one or more existing PowerDesigner models

♦ Import one or more Rose or ERwin models

♦ Generate from another type of model or resource

♦ Reverse engineer from another type of model or resource

♦ Open a model from a previous version

☞ For more information on the different methods of model creation, see section on model creation in the different user's guides.

Model default name *
When you create a new model, the following symbol * (star) is displayed beside the model default name in the Browser, for example BusinessProcessModel_1*. This symbol designates the state of a model to be saved. The first time you save your model, the symbol is no longer displayed.

When you select File ➤ New, you open the model creation dialog box, which allows you to select the following:

When you select. . .	. . . you open
Business Process Model	A new model of the selected type
Conceptual Data Model	
Free Model	
Information Liquidity Model	
Object-Oriented Model	
Physical Data Model	
Requirements Model	
XML Model	
Multi-Model Report	The Report Editor in order to create a report on several models

Opening, saving, closing, and deleting models

You open, save, close, and delete models using the File menu or directly in the Browser.

Opening a model

❖ **To open a model not in the present workspace**

1. Select File ➤ Open from the PowerDesigner menu bar to open a file selection dialog.

2. Browse to and select a model, and then click OK.

   The model is added to the workspace and opened in the Browser.

By default, when you open a workspace, the models linked to workspace are closed, in order to save working memory.

❖ **To open a model which is closed in the workspace**

1. Right-click the model node and select Open from the model contextual menu.

You can open a model as read-only when you do not need to modify its content.

❖ **To open a model as read-only**

1. Select File ➤ Open and select the Open as read-only check box in the Open dialog box.

   or

   Right-click a model closed in the workspace and select Open As Read-only in the contextual menu.

   The model opens with (Read-only) specified in the title bar.

Saving a model | 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 automatically assigns to the file a unique identifying number called GUID (Global Unique ID), and creates a backup copy of your file with the same identifying number. The GUID is used:

♦ In the Repository, to allow documents to be identified and updated

♦ During model generation

The following formats are available when you save a model:

Format	Description
XML	[default] The saving and loading time is longer (1.5x) than binary, and the file size is bigger (2.5x) than binary.
	You can open and modify the contents of an XML model file.
	XML is recommended for small models.
Binary	Saving and loading time is shorter than XML, and the file size is smaller than XML
	Recommended for big models.

❖ **To save a model**

1. Select File ➤ Save.

   or

   Click the Save tool in the PowerDesigner toolbar.

   or

   Right-click the model entry in the Browser and select Save from the contextual menu.

   If you did not define a file name when you created the new model, the Save As dialog box asks you to indicate a name and a path for the file of the new model.

You can create a backup version of a model with the same GUID as the original.

❖ To save a model as a new file with the same GUID

1. Select File ➤ Save As.

    or

    Right-click the model node in the Browser and select Save As from the contextual menu.

You can save the model as a new model with a new GUID. This allows you to develop two separate models in parallel, starting from the same set of model objects.

Note that when you consolidate the new model in the Repository, the Update mode will not be available. External shortcuts located in the new model may also not work properly since the identity of the model has changed.

❖ To save a model as a new file with a new GUID

1. Select File ➤ Save As New Model.

    or

    Right-click the model node in the Browser and select Save As New Model from the contextual menu.

You can save the workspace and all the models it contains with a single click.

❖ To save all models

1. Select File ➤ Save All.

    or

    Click the Save All tool in the PowerDesigner toolbar.

Renaming a model

You can rename a model in the Browser.

❖ To rename a model in the Browser

1. Select the model node in the Browser.

2. Click in the selection to make the name editable.

    or

    Press F2.

    or

    Right-click the node and select Rename from the contextual menu.

3. Type a new name for the model and press ENTER.

Closing a model
When you close a model you free up working memory, but the model remains available in the workspace to be re-opened:

❖ **To close a model**

1. Right-click the model.

2. Select Close in the contextual menu.

   If the model needs to be saved, a Save dialog box is displayed.

   The node in the Browser is tagged with a red dot.
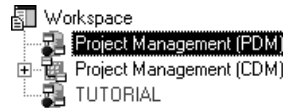
Deleting a model
When you delete a model, you detach it from the current workspace, but do not delete the corresponding file from your disk.

❖ **To delete a model**

1. Right-click the model node in the Browser and select Detach From Workspace from the contextual menu.

   or

   Select the model node in the Browser and press DEL (or click the Delete tool in the toolbar).

   The model node and all its related items are removed from the current workspace.

Adding legacy models to the workspace

Your current models can coexist with non-migrated V6 models like Process Analyst Models (PAM) or Warehouse Architect Models (WAM) and also external models like ERwin or Rational Rose.

This feature allows you to keep on working with the non-migrated or external model application together with PowerDesigner within the same workspace.

❖ **To add a legacy model to your workspace**

1. Right-click the workspace entry in the Browser and select Add from the contextual menu.

   or

   Drag the model from Windows Explorer, and drop it into the Browser.

   or

   Extract the model from the Repository (with the Add to workspace option selected and the Open document option deselected).

Each time you add a model to the workspace, PowerDesigner lets you choose for each one of them whether you want to:

♦ Add non-migrated V6 models and external models to the workspace and keep their original format

♦ Open non-migrated V6 models and external models as PowerDesigner models

When you choose to add a model to the workspace and keep its original format, the model opens in its associated application every time you want to open it.

☞ For more information on opening a PAM into a CDM, see chapter Working with Conceptual Data Models in the *Conceptual Data Model User's Guide* .

☞ For more information on opening a PAM into a BPM, see chapter Working with Business Process Models in the *Business Process Model User's Guide* .

☞ For more information on importing a WAM into a PDM see chapter Working with Physical Data Model in the *Physical Data Model User's Guide* .

Model properties

You can modify the properties of a model from its property sheet.

❖ **To open the model property sheet**

1. Double-click the model node in the Browser.

   or

   Right-click the model node, or the model diagram background, and select Properties from the contextual menu.

All model types share the following common properties:

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users. By default, any new model is called Model *n*
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	A comment is an optional label that describes a model and provides more information than the name
Filename	Location of the model file. This box is empty if the model has never been saved
Author	Author of the model. You can insert a name, a space, or nothing. If you insert a space, the Author field in the title box remains empty. If you intentionally leave the box empty, the Author field in the title box displays the user name from the Version Info page of the model property sheet
Version	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 model display preferences
Default diagram/ view	You can select from the list the diagram or view to display by default when you open the model

Generating a model

When you generate a PowerDesigner model, you can define generation targets, options, and tasks.

Defining a generation target

The Targets page displays a list of target extended model definitions that can be selected for generation.

If you use an extended model definition to extend a target language, you must use an extended model definition designed for generation. To do this, you must verify that the Complement Language Generation check box is selected in the Extended Model Definition Editor.

☞ For more information on extended model definitions used with target

languages, see chapter Generating for a language in the appropriate module user's guide.

If you use an extended model definition for extended generation, you must use an extended model definition designed for generation that does not complement the main generation. To do this, you must verify that the Complement Language Generation check box is deselected in the Extended Model Definition Editor.

☞ For more information on the generation targets, see section Generation category, in chapter Extended Model Definition Reference Guide in the *Advanced User Documentation* .

The Targets page contains check boxes that allow you to select a category that may contain multiple extended model definitions, whereas the radio buttons allow you to select extended model definitions. The radio buttons are mutually exclusive, as it is only possible to select one extended model definition at a time.

☞ For more information on how to create or import an extended model definition, see section Working with extended model definitions in chapter Extended Model Definition Reference Guide in the *Advanced User Documentation* .

Defining generation options

You can check your model before generation using the Check Model check box. This check box is always available. The generation stops if an error is found. You must correct this error before starting generation again.

☞ For more information on the Check Model feature, see sections on checking a model in the different user's guides.

You can add any options you may need during generation in the Generation\Options sub-category of the target language or extended model definition file. For example, you may want to sort the order in which the operations are displayed according to their visibility in the OOM.

☞ For more information on how to add options for generation, see section Generation category in chapters Object Languages Reference Guide, Process Language Reference Guide, XML Language Reference Guide and Extended Model Definition Reference Guide in the *Advanced User Documentation* .

The Options page of the Generation dialog box may be empty, this means that no option has been defined in the corresponding target file (target language or extended model definition). If you use Java for example, you can use standard generation options available by default in this object language.

Defining generation artifacts

The Artifacts tab contains the list of files that will be generated.

☞ For more information, see "Customizing the generation of files for an object" in the "Managing Objects" chapter.

Defining generation tasks

The Tasks page contains standard commands defined in the target language and/or the extended model definition(s) attached to the model. You can select any tasks to execute at generation time.

You can add any tasks you may need during generation in the Generation\Tasks sub-category of the target language or extended model definition file. For example, you may add tasks in your current language in order to automatically compile files, or run a Java application during generation.

☞ For more information on using these tasks in target languages, see section Tasks category in chapters Process Languages Reference Guide, XML Languages Reference Guide and Object Languages Reference Guide in the *Advanced User Documentation.*

☞ For more information on using these tasks in extended model definitions, see section Commands and tasks in chapter Extended Model Definition Reference Guide in the *Advanced User Documentation.*

You may use a language or an extended model definition that displays the Tasks page of the Generation dialog box empty by default. This means that no task has been defined in the corresponding target language or extended model definition.

Sending a model via a messaging application

PowerDesigner uses the messaging application programming interface (MAPI) to send model files by electronic mail.

This interface lets you use your internal messaging system to send model files directly to other team members. Make sure your current email software supports MAPI or verify your email software settings in Control Panel ➤ Internet Options.

❖ **To send a model using a messaging system**

1. Select File ➤ Send to open a profile selection dialog box.

2. Select a profile and click OK.

   The file transfers to your internal messaging system, which may ask you for additional information, such as a destination.

Packages

When you are working with large models, you can split any model into smaller subdivisions, or **packages**, in order to avoid manipulating the entire set of data of the model. Packages can be used to represent different tasks or subject areas, or to distinguish between 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

Creating a package

You can create a package in any of the following ways:

♦ Use the package tool in the diagram palette

♦ 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 "Creating an object" in the Managing Objects chapter.

Package properties

Packages have properties displayed on property sheets. All packages share the following common properties:

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	A comment is an optional label that describes a package and provides more information than the name
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Default diagram	You can select from the list the diagram to display by default when you open the package
Use parent namespace	Option that defines the package as being the area in which the name of an object must be unique in order to be used.

Saving a package

You cannot save a package individually. When you save the model you also save all the packages it contains.

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, ILM, 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.

> **XSM namespace**
> The XML model does not support packages. For more information on namespaces in the XSM, see the *XML Model User's Guide* .

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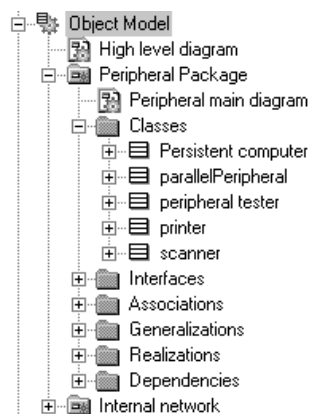
If you were to select the Use Parent Namespace check box in the property sheet of Peripheral Package, the internal names of objects in this package would no longer be prefixed by the name of the package. The parent of the folder, in this case the model itself, becomes the namespace, and you could not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

❖ **To define the namespace of a package**

1. Open the property sheet of the package.

2. Select or clear the Use Parent Namespace check box.

Package Properties - Sales (sales)

| Dependencies | Extended Dependencies | Version Info | Preview |
| General | Notes | Rules | Extended Attributes |

Name: Sales

Code: sales

Comment:

Stereotype:

☑ Use parent namespace

Default diagram: ClassDiagram_1

<< Less   OK   Cancel   Apply   Help
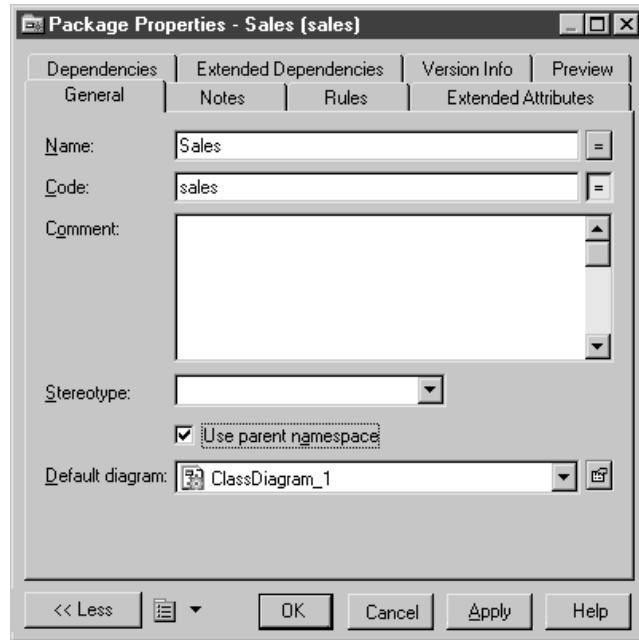
3. Click OK to return to the model diagram.

Diagrams

A diagram is a graphical view of a model or a package. All models and packages have at least one diagram.

Requirements Model views

In the Requirements Model, there are no diagrams, but rather views. For more information, see the *Requirements Model User's Guide* .

You can add additional diagrams to a model or package to split the display and focus on certain portions of the system. You can also use multiple diagrams to focus on different subject areas. They allow you to see the symbols of the same objects, displayed with different kinds of information.

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.

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 ➤ *Diagram* .

♦ Right-click the background of your diagram and select Diagram ➤ New Diagram ➤ *Diagram* from the contextual menu

♦ Right-click the model node in the Browser and select New ➤ *Diagram* from the contextual menu

In each case you will be invited to specify a name, code, and optional comment for 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
Stereo-type	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 the "Managing Profiles" chapter in *Advanced User Documentation* .
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

Navigating between and opening existing diagrams

You can open a diagram in any of the following ways:

♦ 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.
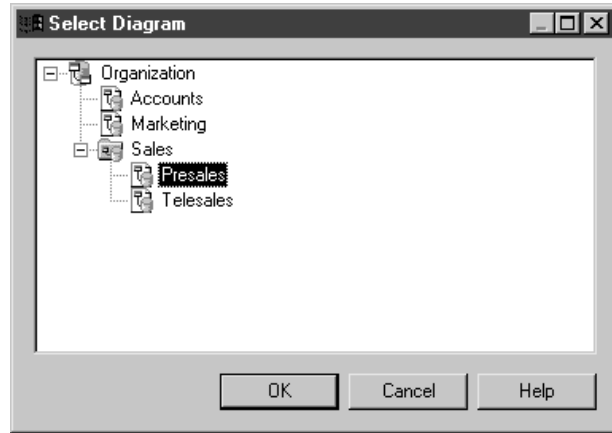
♦ In the case of package diagrams, you can additionally:

  • Select the Open Package Diagram tool from the Palette and click on a package symbol.

  • Press CTRL and double-click a package symbol.

  • Right-click a package symbol and select Diagram ➤ Open Diagram from the package contextual menu.

Showing and hiding diagram symbols

Not all PowerDesigner model objects have symbols. For those that do, the symbol is, by default, displayed in the diagram when you create the object. You can subsequently choose to show or hide the symbol.

Hiding a symbol

Hiding certain symbols can improve the readability of your diagram. When you hide a symbol you do not delete it or the object.

❖ **To hide a symbol from the Symbol menu**

1. Select one or several symbols in the diagram.

2. Select Symbol ➤ Hide Symbols from the contextual menu.

   The symbols are hidden in the diagram.

Hidden link objects

When you hide an object with link objects connected to it, the link objects are also hidden. For example, if you hide a table symbol that has a reference to another table, the selected table will be hidden with the reference.

❖ **To hide a symbol from the Show Symbols dialog box**

1. Select Symbol ➤ Show Symbols (or right-click the diagram background and select Diagram ➤ Show Symbols from the diagram contextual menu) to open the Show Symbols dialog box.

2. Deselect the check box beside the visible symbol.

3. Click OK to return to the model diagram.

Displaying a hidden symbol

You can redisplay a hidden symbol.

❖ **To display a hidden symbol**

1. Select Symbol ➤ Show Symbols (or right-click the diagram background and select Diagram ➤ Show Symbols from the diagram contextual menu) to open the Show Symbols dialog box.

2. Select the appropriate object type sub-tab and select the check box beside the appropriate symbol.
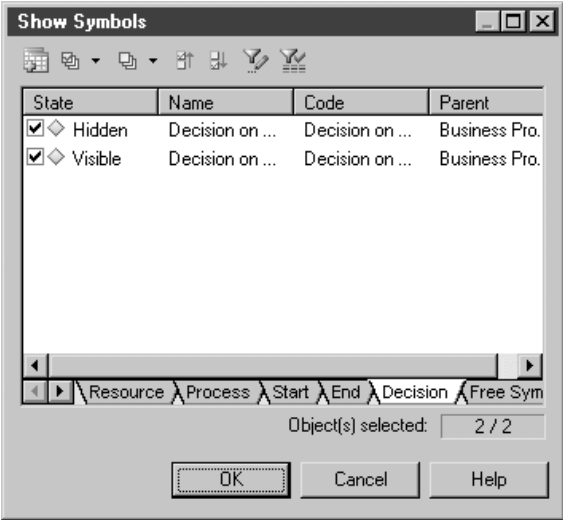
3. Click OK to return to the model diagram.

Working with the Show Symbols dialog box

The Show Symbols dialog box displays all objects belonging to the current package. You can select objects from other packages for display in the current diagram using the Add Objects tool.

Object symbols are organized by object type in sub-tabs. Only relevant tabs are displayed. For example, if your package contains only tables and references, only the Table tab will be available, as references are contained in table definitions. If shortcuts, extended dependencies and/or free symbols are present in the package, they appear on their own sub-tabs.

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 check boxes in the current object type page or selects all check boxes in all object type pages when you click the Down Arrowhead and select All Pages
	Deselects all check boxes in the current object type page or deselects all check boxes in all object type pages when you click the Down Arrowhead and select All Pages
	Moves all selected object to the top of the list
	Moves all selected object to the bottom of the list
CTRL + selection tool	Selects all items in all object type pages

Symbol deletion
You cannot delete a symbol from the Show Symbols dialog box. When a symbol is hidden it is not deleted.

Show symbol directly in the diagram

If an object has no symbol, you can show it directly in the diagram without using the Show Symbols dialog box, by using the:

♦ Drag and Drop feature

♦ Paste as Shortcut command

Action	Result
Standard Drag and Drop from Browser to diagram	Object symbol created in the destination diagram
Drag and Drop from diagram to diagram + Create shortcut option selected (General Options) *or* Copy symbol + Paste as Shortcut command from diagram to diagram	Object symbol created in the destination diagram with symbol format parameters and relative position preserved

☞ For more information on the Drag and Drop feature and the Paste as Shortcut command, see respectively sections Dragging and dropping objects and Pasting an object as a shortcut in chapter Managing Objects.

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
Parent entity	Deselected	Inheritances for which the hidden entity is the parent are automatically hidden

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

♦ The Result List

♦ The object property sheet dropdown menu

♦ An objects list

❖ **To find an object symbol from the Browser**

1. Right-click an object in the Browser and select Find in Diagram (or Find in Document View) from the contextual menu.

❖ **To find an object symbol in the diagram from the Result List**

1. Right-click an object in the Result List and select Find in Diagram (or Find in Document View) from the contextual menu.

❖ **To find an object symbol in the diagram from its property sheet**

1. Open an object property sheet and select Find in Diagram (or Find in Document View) from the dropdown menu at the bottom-left corner.

❖ **To find an object symbol in the diagram from an objects list**

1. From an objects list, select an object in the list and click the Find Symbol in Diagram tool in the list toolbar.

Defining related diagrams

You can define **Related Diagrams** to attach an object to one or several diagrams others than the ones it was created in. This feature can be used to further define the behavior and implementation of objects . It allows you to view objects from different angles and describes a semantic relationships.

You can associate any type of diagram open in the workspace with an object, including diagrams from other packages or models.

Adding related diagrams

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

❖ **To add a related diagram**

1. Open the property sheet of the object and click the Related Diagrams tab.

2. Click the Add Objects tool to open a selection window.

3. Select a model in the list, and then select the diagram to attach.

4. Click OK to return to the object property sheet.

   The diagram is displayed in the list of related diagrams of the object.

Opening related diagrams

You can open a related diagram from the property sheet of an object.

❖ **To open a related diagram from an object property sheet**

1. Open the property sheet of the object and click the Related Diagrams tab.

2. Select a diagram from the list, and then click the Open Diagram tool.

   The selected diagram is opened in the diagram window, behind the object property sheet.

3. Click OK to access the related diagram.

Saving a diagram

You cannot save a diagram individually, as it only represents a view of a model or package. When you save a model, you save the diagram in its present state of magnification.

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.

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.

☞ For more information on shortcut creation, see the Managing Shortcuts chapter.

❖ **To convert a diagram to a package**

1. Select View ➤ Diagram ➤ Convert to Package.

   or

   Right-click the diagram background window and select Diagram ➤
   Convert to Package from the contextual menu.

   or

   Right-click the diagram node in the Browser and select Convert to
   Package from the contextual menu.

   In each case, the Convert Diagram to Package wizard opens. By default
   the sub-package takes the name of the diagram.

2. [optional] Enter a different name and code for the new package.

3. Click Next to open the Selecting Objects to Move page.

   This page 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.
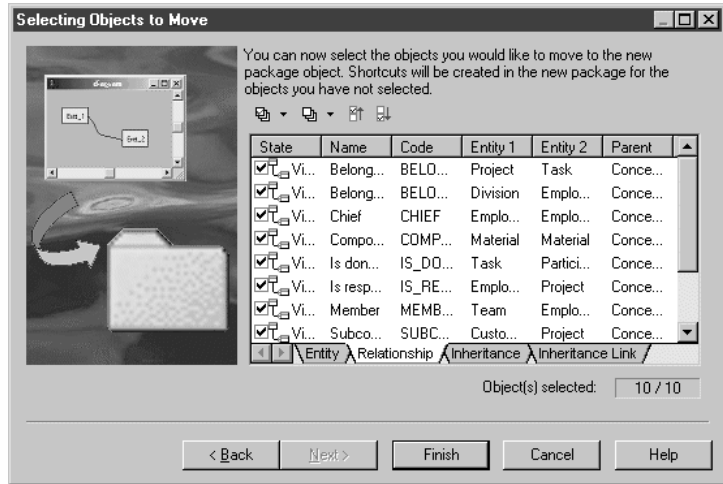
4. [optional] Deselect any objects you do not want to move to the new package. Objects deselected here will remain in the original package and be represented in the new package via shortcuts.

5. 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 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.

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

☞ For more information on shortcut creation, see chapter Managing shortcuts.

You can simply drag and drop the diagram from one package to another in the Browser. All objects in the diagram are automatically moved to the target package.

You can use the wizard to control which objects in the diagram are moved to the new package.

❖ **To move a diagram to a package using the wizard**

1. Select View ➤ Diagram ➤ Move to Package.

   or

   Right-click the diagram background window and select Diagram ➤ Move to Package from the contextual menu.

   or

   Right-click the diagram node in the Browser and select Move to Package from the contextual menu.

   In each case, the Move Diagram to Package wizard opens:
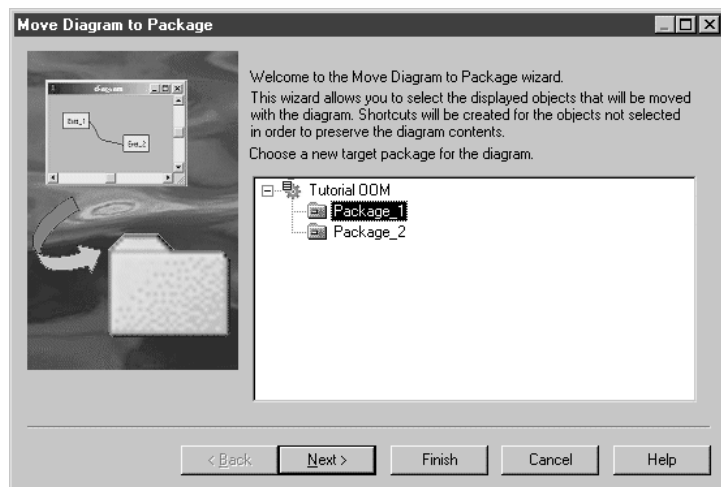
2. Select a target package in which you want to move the current diagram and click Next to open the Selecting Objects to Move page.

   This page 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.

> **Last diagram in 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, the following rules apply:

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

Checking a Model

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.

❖ **To check your model**

1. Press F4, select Tools ➤ Check Model, or right-click the diagram background and select Check Model from the contextual menu to open the Check Model Parameters dialog box.

   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:
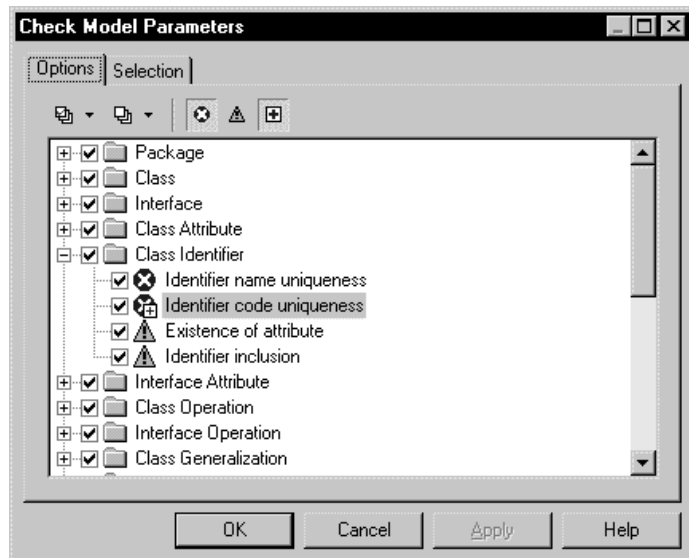
2. [optional] Select or deselect types of objects to check, and expand object nodes to enable, disable, vary the severity of, and enable or disable automatic correction of individual checks (see "Check model parameter tools" on page 150).

   Getting documentation for object checks
   Right-click a check in the Check Model Parameters dialog box and select Help from the contextual menu 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:
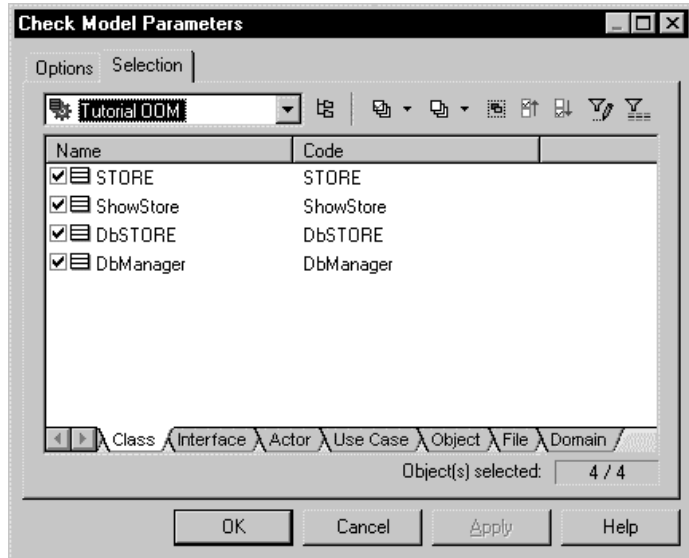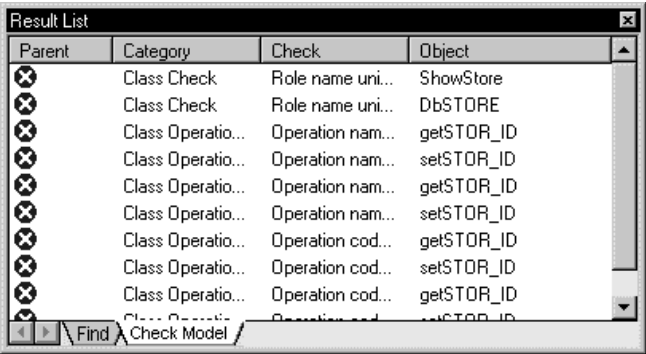
Selecting objects in the diagram

If you have selected object symbols in your diagram before starting the model check, you can select them for checking by clicking the Use Graphical Selection tool in the Selection tab tool bar.

For more information about selecting objects in Selection tabs, see the "Adding an item from a selection list" section in the "Using the PowerDesigner Interface" chapter.

4. [optional] Click Apply to save your selections so that they will be available for future model checks.

5. Click OK to launch 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 151.

Check model parameter tools

The following tools are available in the Check Model Parameters dialog box:

Tool	Description
	Select All – Click the arrow to the right of this tool to choose to select all checks, all error checks, or all warning checks.
	Deselect All - Click the arrow to the right of this tool to choose to deselect all checks, all error checks, or all warning checks.
	Error – Sets the selected check to error level. The check icon will change accordingly. When errors are encountered, any model generation is stopped.
	Warning - Sets the selected check to error level. The check icon will change accordingly. Warnings do not prevent generation from your model.
	Automatic correction – [enabled if automatic correction is available for the selected check] Enables automatic correction for the selected check. The check icon will change to bear a small red cross in its bottom-right corner.
	You should be aware of how automatic corrections can affect your models. For example, in a PDM, if a column code length is longer than the length specified in the `MaxColumnLen` field in the DBMS, then PowerDesigner can automatically truncate the code to the specified length. However, such truncation can make create duplicate names and PowerDesigner will automatically rename the duplicated column code. If you do not want this to occur, you should deselect automatic correction for column code uniqueness.
	All problems that can not be corrected automatically must be corrected manually. For example, in a PDM, PowerDesigner will not create a column for an existing index, and you will need to create the appropriate column.

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.

Right click menu
These and other options are also available by right-clicking an item in the Check Model Result List.

Using the Free Model

The Free Model (FEM) allows you to create any kind of chart, diagram, in a context-free environment. For example, you could create a model to represent:

♦ how your models and documents relate to each other

♦ the organizations in your enterprise and how they interrelate

♦ a flowchart

♦ a hierarchy diagram

♦ the graphics you use in PowerPoint presentations and link them to the presentations they are used in

♦ the users in your Enterprise Repository and link them to symbols representing the groups they belong to

♦ an ORM model using ORM specific extended model definitions

You can enrich the diagrams in the Free Model by selecting different bitmaps for each symbol and add extended attributes to collect specific metadata. VBScript and the Generic Generator make it possible to program your own semantics into this model and create specific generated code or other output particular to your own interpretation.

Naming Conventions

You can define **naming conventions** for your model objects in the Model Options dialog box. You can:

♦ Set rules to enforce predefined name and code formats for all the objects within a model. When you type a name or a code that does not respect the conventions, PowerDesigner adapts them to the required format. See "Name/Code Format" on page 154.

♦ Set conversion scripts in order to generate objects name from their code or objects code from their name using specific macros and conversion tables. This feature is only available when you select the Enable name/code conversion option. See "Name and code conversion scripts" on page 159.

> **Name into code conversion during inter-model generation**
> The name into code conversion is particularly useful for inter-model generation, as it allows you to define how you want the codes of your model objects to be generated. For more information on the name into code conversion during the inter-model generation, see section on inter-model generation options in the different modules user's guides.

Name/Code Format

Name and code format encompasses constraints on length, character case and type for the name and the code of objects.

You can set name and code format for:

♦ Models and packages

♦ Model objects

♦ Reusable templates

Displaying the name or the code of objects

You can define whether the names or codes of objects are displayed.

❖ **To define the display of names or codes**

1. Select Tools ➤ Model Options to open the Model Options dialog.

2. Select the Naming Convention node in the Category tree view.
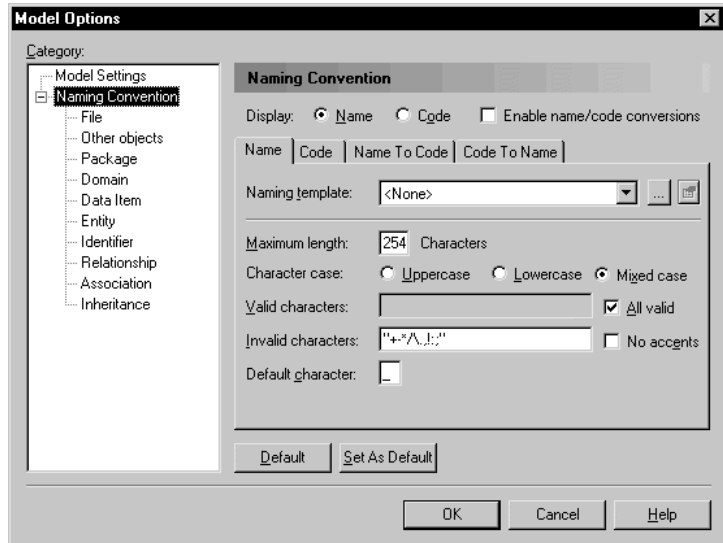
3. Select the Name or Code radio button in the upper part of the dialog box.

4. Click OK to return to the model diagram.

Defining name and code format for a type of object

You can specify name and code formats for each type of object in a model.

❖ **To define name and code format for a type of object**

1. Select Tools ➤ Model Options to open the Model Options dialog box.

2. Select an object type beneath the Naming Convention node in the Category tree view.

> **Other Objects category**
>
> Select the Other Objects category to define name and code format for models and all objects that are not visible in the tree view.
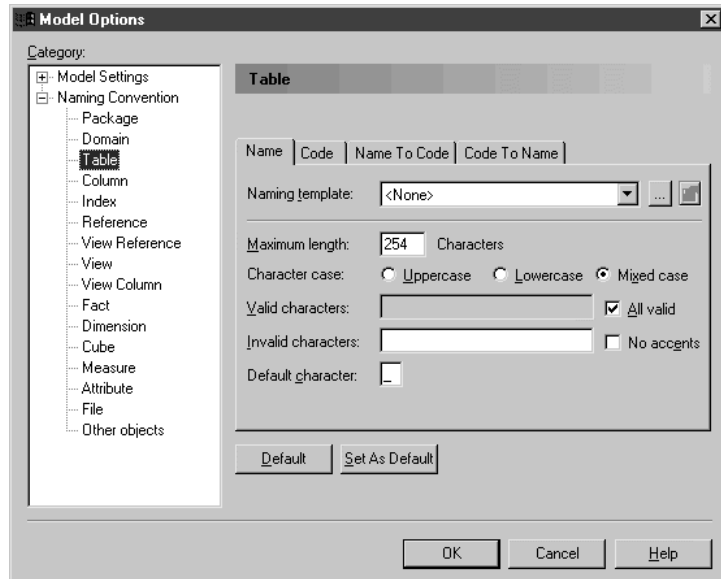
3. Complete the appropriate properties on the Name and Code tabs (see "Name and Code format properties" on page 158).

4. Click OK to return to the model diagram.

Creating a template for name and code format

You can define a template for name and code formats and reuse it for several types of objects.

❖ **To create a template for name and code format**

1. Select Tools ➤ Model Options to open the Model Options dialog box.

2. Select the Naming Convention node in the Category tree view.

3. Click the Ellipsis button beside the Naming Template box to open the List of Naming Templates:
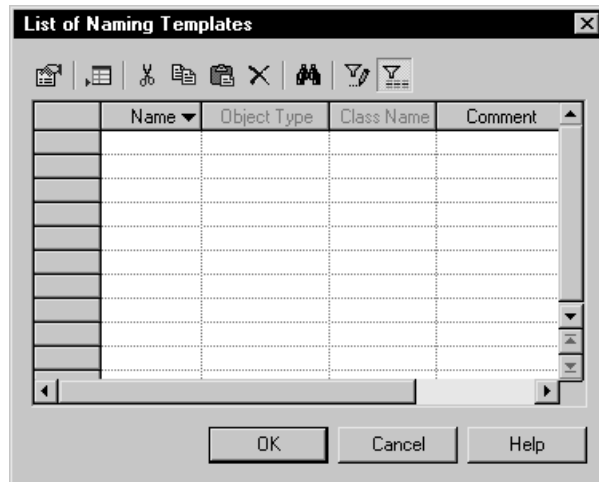
4. Click the Add a Row button and type a naming template name.

5. Click the Properties tool to open the new naming template property sheet to the Detail page:
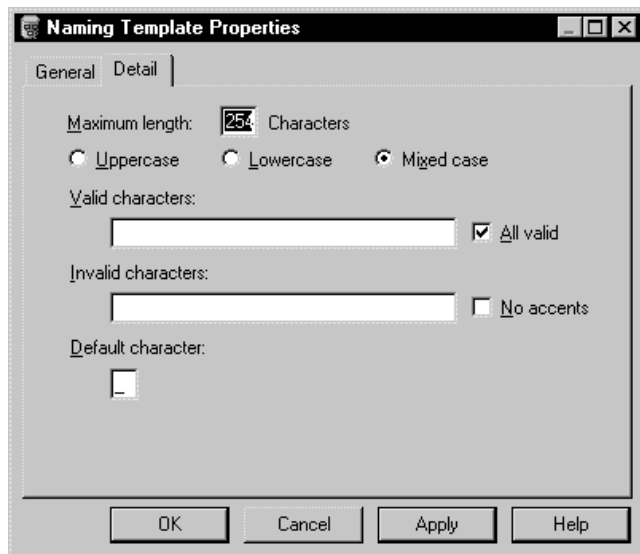
6. Complete the appropriate properties (see "Name and Code format properties" on page 158).

7. Click OK to return to the model diagram.

> **Naming template modification**
> You can modify name and code format for a template by selecting it from the Naming template list and clicking the Properties tool beside the Naming template box.

Applying a naming template to a type of object

You can apply a naming template to any number of object types in your model.

❖ **To apply a naming template to an object type**

1. Select Tools ➤ Model Options to open the Model Options dialog box.

2. Select an object type beneath the Naming Convention node in the Category tree view.

3. Select the Name or Code tab and then select a template in the Naming Template list.

   The values of the selected template appear in the name format boxes.

4. Click OK to return to the model diagram.

Name and Code format properties

The following options are available for formatting names and codes:

Option	Description
Display Name	Displays the names of objects in the interface
Display Code	Displays the codes of objects in the interface
Enable name/code conversions	Enables the use of conversion scripts and conversion tables to generate a code from a name or a name from a code. See section "Name and code conversion scripts" on page 159
Naming Template	Specifies a template for name and code format

Option	Description
Maximum Length	Maximum number of characters. In a PDM, this maximum number of characters can come from the Registry (if you defined a Maximum length and clicked the Set As Default button) or from the DBMS definition file. If a maximum length is declared in both the Registry and the DBMS, PowerDesigner uses the length with higher constraint. For example, in the case where the Registry = 128 and the DBMS = 30, PowerDesigner uses 30 characters for maximum length
Character Case	Indicates authorized character cases: Uppercase, Lowercase, and Mixed Case
Valid Characters	List of authorized characters.
	You can specify a single character or character string in quotation marks. For example:
	`"a", "xyz"`
	You can specify an interval of characters between single quotation marks separated by a dash. For example:
	`'a'-'z'`
	`'a'-'z', 'A'-'Z', '0'-'9'`
	By default, PowerDesigner allows the following valid characters for codes: 'a'-'z','A'-'Z','0'-'9','_'
All Valid	Accepts all characters.
Invalid Characters	List of unauthorized characters. By default, PowerDesigner defines the following invalid characters for names:
	''+-*/\.,!:;''
No Accents	Removes accents from accented characters.
Default Character	Character used to replace any invalid characters entered.

Name and code conversion scripts

You can set conversion scripts in order to generate the name of an object from its code or vice versa using conversion scripts and tables. Name/code conversions can be applied to all named objects in a model or for each type of object individually.

You must select the Enable name/code conversions check box in the upper part of the Naming Convention page of the Model Options dialog box for

this feature to be available.

Enable name/code conversion option	Name/code behavior
Selected	Use of conversion scripts and conversion tables to generate a code from a name or a name from a code together with the application of naming conventions
Unselected	Names are copied from codes or codes are copied from names together with the application of naming conventions

Name to Code mirroring mode

When you create an object in PowerDesigner, you may have to type a name and click the Equal button after the Code box to set the code equal to the name because the Name to Code mirroring mode may not be selected in the General Options dialog box.

You can select this operating mode to automatically set the code equal to the name and thereby apply the conversion scripts you defined to both names and codes.

☞ For more information on the Name to Code mirroring option, see section Defining dialog box operating mode in chapter using the PowerDesigner interface

Conversion scripts allow you to alternatively generate a code from a name and a name from a code. You define conversion scripts from the corresponding tabs in the Naming Convention page of the Model Options dialog box. Usually name into code conversion is the most commonly used:

Tab	Description
Name To Code	You use it when you want to generate a code from a name
Code To Name	You use it when you want to generate a name from a code

Each tab allows you to:

♦ Define a conversion script in the Conversion Script edit box to generate the name from the code or the code from the name using macros

♦ Select a conversion table from the Conversion table list. This allows you to convert an expression to another, when using the .convert_name or .convert_code macros. For more information on macros, see section "Using a conversion script" on page 161. The use of conversion tables is optional
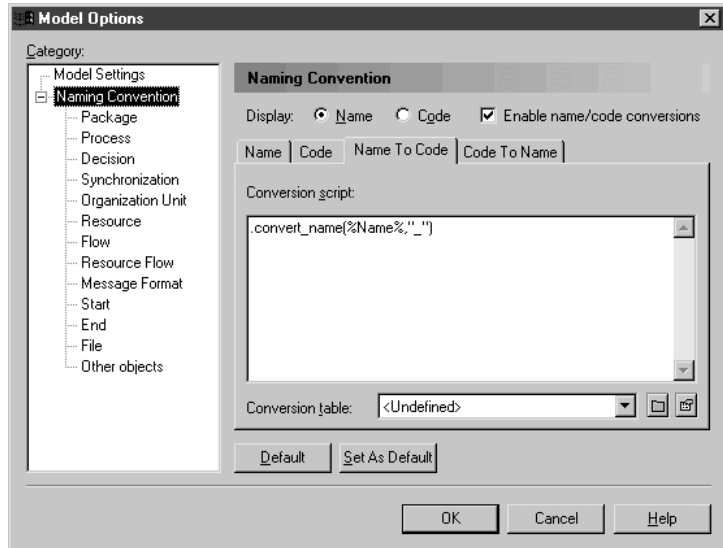
Using a conversion script

Default conversion scripts allow you to generate names from codes and codes from names using macros. These are macros used in the Generation Template Language (GTL) and recommended for the name/code conversions.

Conversion scripts are used when you click the Equal button after a Name or a Code box in the object property sheet or when you have selected the Name to code mirroring option in the Dialog page of the General Options.

You can customize the default conversion scripts using any of the following macros:

♦ .foreach_part

♦ .convert_name & .convert_code

♦ .lowercase

♦ .uppercase

♦ .replace

♦ .delete

☞ For more information on the syntax of macro keywords used in the Generation Template Language (GTL), see section Using macros in chapter Generation Reference Guide in the *Advanced User Documentation* .

.foreach_part macro

The **.foreach_part** macro allows an iteration on each part of an expression. The part separators will be specified into a pattern expression.

This macro is very useful when dealing with inter-model generation, as models can have very different naming conventions. For example a Java class attribute code may be "customerName" whereas a PDM table column code may be "CUSTOMER_NAME".

Syntax

```
. foreach_part (<Expression> [, <Separator Pattern>[,<Head> [,
    <Tail>]]])
[<Block>]
.next[(<Separator>)]
```

Parameters

Parameter	Description
<Expression>	Designates the expression scanned by the part iterator. This part iterator will stop on each character specified into the <PartSeparatorPattern>
<Separator Pattern> (optional)	Defined into a double quoted string. Any character specified into the pattern will be used as a part separator. See below
<Block>	Encompasses the following variables: %CurrentPart%: Current part value, %IsFirst%: Determines if the current part is the first part of the expression, %IsLast%: Determines if the current part is the last part of the expression
<Separator> (optional)	A <separator> can be concatenated between each part. If you defined a character separator in the <PartSeparatorPattern>, the value of the <separator> will replace the character separator. If you defined a range separator in the <PartSeparatorPattern>, the table conversion you selected will return the corresponding value to replace the range separator
<Head> or <Tail> (optional)	Expressions that can be added respectively at the beginning or at the end of the generated expression

Separator	There are two types of separator:

♦ A **character** separator that must be defined into an additional simple quote and writes as follows: (%Name,"'<char>'")

♦ A **range** separator that must be used with a conversion table and writes as follows: (%Name,"[<c1>-<c2>]")

Both can also be combined: (%Name,"'<char>',[<c1>-<c2>]").
'<char>' can be any character specified into the pattern for example: a, b , c, 0, 9.

[<c1> - <c2>] specifies a character taken into the range defined between the two characters <c1> and <c2>. For example, [A-Z], [a-z] or [0-9]" can be used as part separators.

By default, the <PartSeparatorPattern> is initialized by the pattern " -_,\t". If the specified pattern is empty the pattern is initialized using the default value.

Examples

♦ Script 1: Convert a name into a class code (JAVA naming convention)

```
.foreach_part(%Name%, "' _-'")
%.FU:CurrentPart%
.next
```

The conversion script will output:

```
Name = Employee shareholder => Code = EmployeeShareholder
```

♦ Script 2: Convert a name into a class attribute code (JAVA naming convention)

```
.set_value(_First, true, new)
.foreach_part(%Name%, "' _-'")
.if (%_First%)
%.L:CurrentPart%
.set_value(_First, false, update)
.else
%.FU:CurrentPart%
.endif
.next
```

The conversion script will output:

```
Name = Employee shareholder => Code = employeeShareholder
```

.convert_name & .convert_code macros

The **.convert_name** & **.convert_code** macros use a conversion table to get the corresponding code from a name or the corresponding name from a code.

The .convert_name macro uses the conversion table of a name into a code and the .convert_code macro uses the conversion table of a code into a name. When no occurrence is found in the table, the name or the code is returned.

You can use these macros together with a user-defined conversion table that you select in the Conversion Table list.

Conversion tables are not case sensitive. You can indifferently use lower-or-uppercases in tables.

☞ For more information on conversion tables, see section "Using a conversion table" on page 165.

Syntax

```
.convert_name (<Expression>[,<Separator Character>[,<Separator
        Pattern>],<code naming convention>])
.convert_code (<Expression>[,<Separator Character>[,<Separator
        Pattern>]])
```

Parameters

Parameter	Description
<Expression>	Name or code to be converted in the corresponding conversion table
<Separator Character> (optional)	Character generated each time a separator declared in <Separator Pattern> is found in <Expression>. For example this character can be "_"
<Separator Pattern> (optional)	Declaration of the different separators likely to exist in the <Expression>. These separators will be replaced by the <Separator Character>. You can declare several separators, for example "/-_"
<Code naming convention>	<**firstLowerWord**>: First word in lowercase, then other first letters of other 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

Code naming conventions

If you use the code naming convention parameter with the convert_name macro, the conversion script will automatically convert names according to the selected parameter. This can be very useful when you need to adapt codes to a particular target, for example PowerBuilder naming conventions

require object codes to be always in lowercase and separated by an underscore.

When you use a code naming convention parameter in the conversion script, the conversion table is no longer used and the name is systematically converted according to the pattern defined in the code naming convention parameter.

You can also define a code naming convention for all the instances of a selected metaclass using the resource editor, for more information on this feature, see section Adding a metaclass to a profile in chapter Managing Profiles in the *Advanced User Documentation* .

Example　　　　You type the following script in the Name to Code page:

```
.convert_name(%Name%,"_","'-/'")
```

Note that both the <Separator Character> and the <Separator Pattern> must be defined into a double quoted string, furthermore you should add simple quotes around <Separator Pattern> to define the pattern syntax.

In this script, the separator patterns (- and /) are used to distinguish the different words in the name. The separator character (_) will be used to separate the different characters of the name. So whenever a "-" or a "/" separator pattern is encountered, it will be replaced by the "_" separator character.

The script has to convert the name CLIENT-France/Address and uses the following conversion table:

Name	Code
CLIENT	CLI
France	FR
Address	ADDR

The resulting code is:

```
CLI_FR_ADDR
```

Using a conversion table

Conversion tables provide a way to define a correspondence between the name and the code of an object or the code and the name of an object.

Conversion tables are stored into separate CSV (Comma-Separated Values) files and are shared by all models.

PowerDesigner provides CSV file samples stored in the Resource Files\Conversion tables folder of the PowerDesigner installation directory. You can also store the conversion tables you create in that directory.

Conversion tables are not case sensitive. You can indifferently use lower-or-uppercases in tables.

A conversion table encompasses a **Name** column and a **Code** column. Depending on the conversion type you use, a name or a code is returned during the generation. However, if several different names have the same code for example, only the first found name is returned.

The following example illustrates a Name To Code conversion where two different names (Customer and Client) have the same code (CUST):
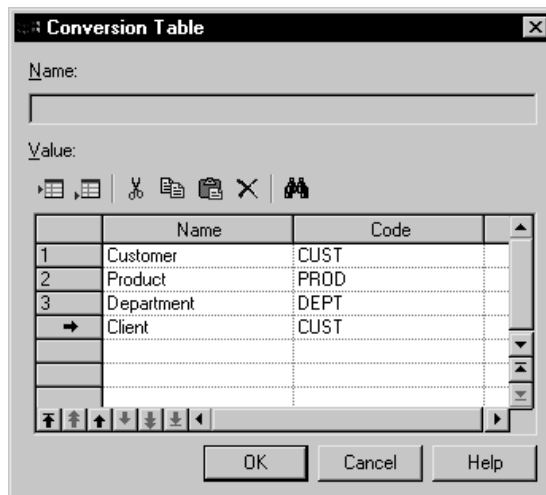
♦ When converting the name into code, CUST will appear in the Code box of both Client and Customer property sheet objects

♦ When converting the code into name, only the first occurrence (Customer = CUST) will be taken into account to appear in the Name box of the Customer object property sheet

A selected conversion table will be operational only if you simultaneously invoke it using the Convert_name or Convert_code macros in the Conversion script edit box.

☞ For more information on the Convert_name or Convert_code macros, see section "Using a conversion script" on page 161.

❖ **To use a conversion table**

1. Select a conversion table from the Conversion table list located in the lower part of the Name To Code or Code to Name pages.

Creating a conversion table

You create a conversion table from the list of conversion tables.

❖ **To create a conversion table**

1. Select Tools ➤ Resources ➤ Conversion Tables to display the list of available conversion tables.
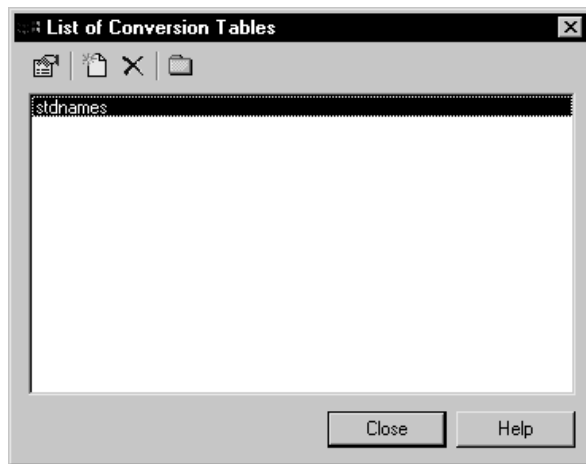
2. Click the New tool, enter a name and click Save to open the Conversion Table dialog.

3. Click the Add a Row tool.

4. Type a name in the Name column, and click the code into which you want it to be converted in the Code column:
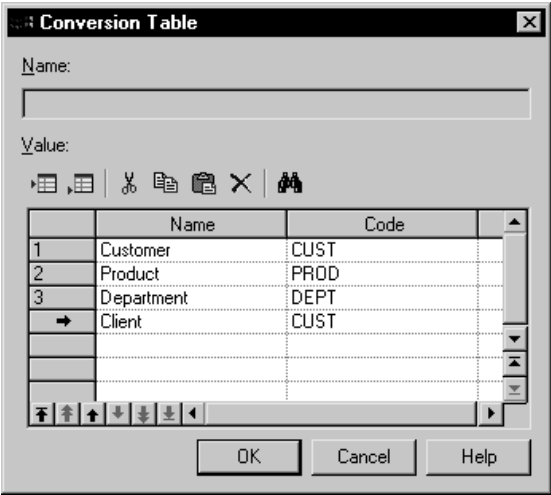
You can create as many conversion pairs as required.

5. Click OK to return to the list of conversion tables, where the new table is now available.

Modifying a conversion table

You modify a conversion table from the List of Conversion Tables.

❖ To modify a conversion table

1. Select Tools ➤ Resources ➤ Conversion Tables to display the list of available conversion tables.

2. Select a conversion table in the list and click the Properties tool.
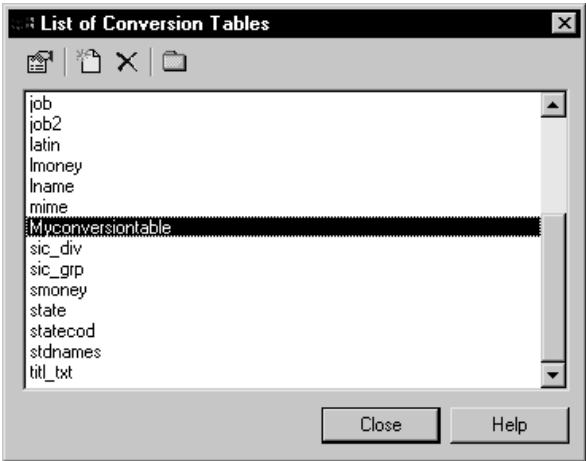
Note that you can also modify a conversion table directly from the
Naming Convention page of the Model Options dialog box by selecting a
conversion table from the Conversion Table list and clicking the Edit the
selected conversion table tool.

> **Modifying a conversion table outside PowerDesigner**
> As a .CSV file, a conversion table can be modified in any external
> text editor, like Notepad for example. However, if you modify your
> conversion table outside PowerDesigner while the conversion tables
> editor is running in the current session, your modifications will be taken
> into account only if you close and re-open the conversion tables editor.

Deleting a conversion table

You delete a conversion table from the list of conversion tables.

❖ **To delete a conversion table**

1. Select Tools ➤ Resources ➤ Conversion Tables to display the list of
   available conversion tables.

2. Select the conversion table that you want to delete from the list and click
   the Delete tool.

Defining a code naming convention in a resource file

When you define code naming conventions in a model, these are not
automatically applied to the current model. To apply these code naming
conventions, you have to do one of the following actions:

♦ Use the model to model generation feature, in this case you re-generate
   the model and apply the new naming conventions

♦ Create a new model

♦ Use the Change Target Language to apply the same resource file to the
   model but have the new naming conventions applied

❖ **To define a code naming convention in a resource file**

1. Open the DBMS, object language, process language, or XML language
   editor.

2. Select a metaclass in the Profile category.

3. Select a Code Naming Convention from the list.
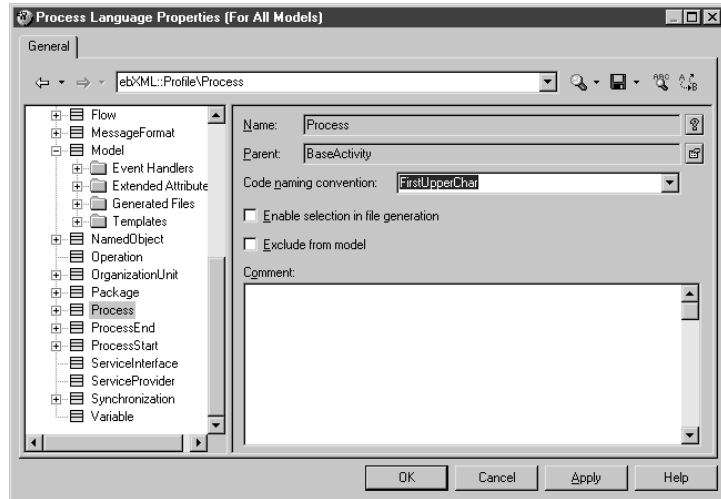
4. Click OK.

Spell Checking

Spell checking in PowerDesigner is only available 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.

Enabling the spell checker

The spell checker is a PowerDesigner add-in that needs to be enabled to work properly.

❖ **To enable and configure the spell checker**

1. Select Tools ➤ General Options to open the General Options dialog box.

2. Click the Add-Ins node in the Category tree view and select the Spell Checker checkbox.

3. Click OK to return to the model diagram.

4. Select Tools ➤ Spell Checking Options to open the Spell Checking Options dialog box.

5. Select the properties of model objects 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.

6. Select the language of your model in the Dictionary Language list, and then click OK to return to the model diagram.

Using the spell checker

❖ **To use the spell checker**

1. Right-click an object or a model node and select Spell Check in the contextual menu.

   Spell checking starts. If an error is found the Spell Checking dialog box 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.

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 the "Managing Profiles" chapter in the *Advanced User Documentation* .

Applying transformations during generation

Transformation profiles can be used during model generation:

♦ pre-generation transformations are applied to the source model

♦ post-generation transformations are applied to the target model

Inter model generation

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.
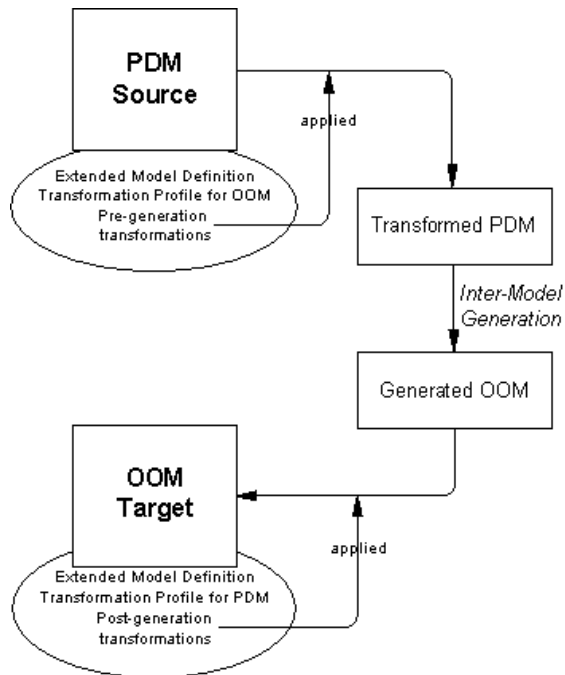
Model to model generation

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 extended model definition in the source and in the target model.

PDM to PDM generation
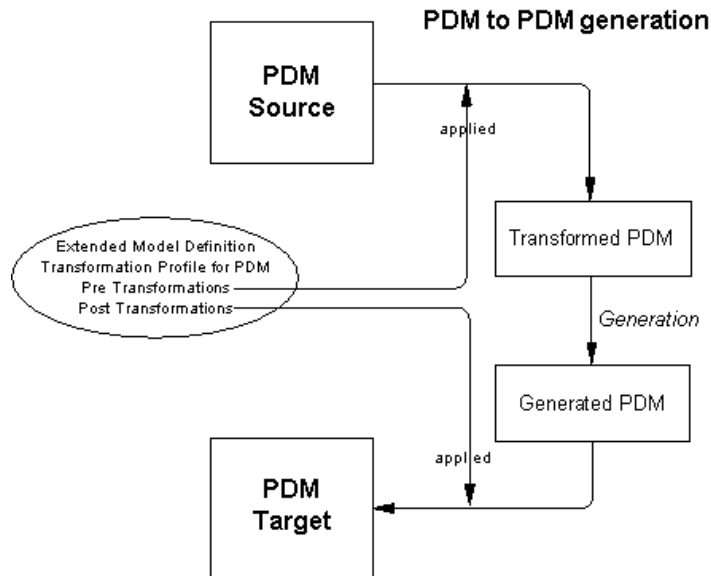
❖ **To select transformations for generation**

1. Select Tools ➤ Generate Model to open the Model Generation Options dialog box.

2. Click the Details tab, and click the Enable Transformations to display the Extended Model Definitions, Pre-generation, and Post-generation tabs.

3. Click the Extended Model Definitions tab to select the extended model definitions 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
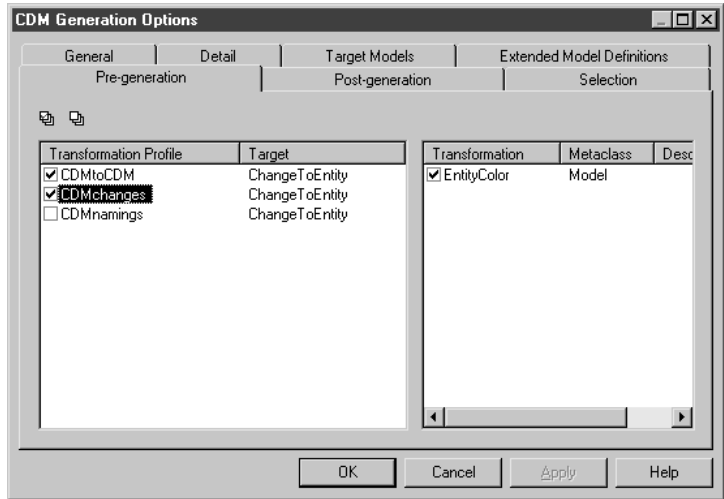
5. Click the Post-generation tab and select profiles and transformations to be applied after generation.

6. Click OK in the Model Generation Options dialog box.

Applying transformations on demand

Transformations can also 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.

> **Post-Generation Transformations only**
> You can only invoke post-generation transformations on demand.

There are two methods for applying transformations on demand in a model:

♦ Add a transformation as a command in a main or contextual menu (see "Defining menus in a profile" in the "Managing Profiles" chapter in the *Advanced User Documentation*).

♦ Use the Apply Transformations window available from the Tools menu.

❖ **To use the Apply Transformations window**

1. Add one or more extended model definitions 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.
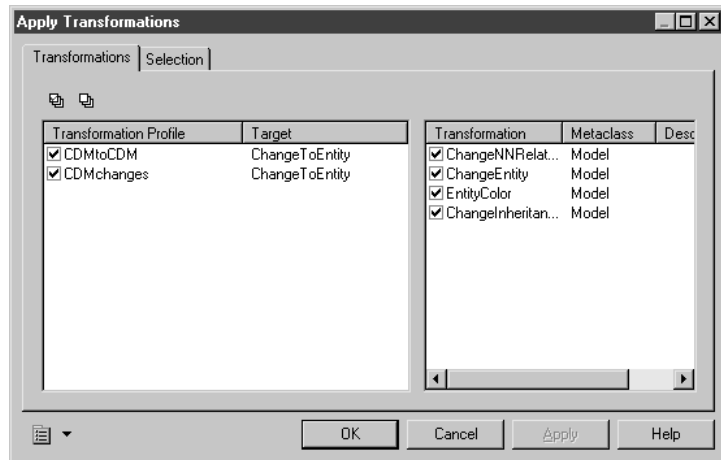
4. [optional] Click the Selection tab and deselect any objects that you want to exclude from the transformation.

5. Click OK to apply the transformations.

Resource Files

The PowerDesigner modeling environment is powered by 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 these XML-format resource files in order to customize and extend the behavior of the environment. The following types of resource files are provided:

♦ **Target languages**: define the standard objects available in a model. Types of target language files include:

- **Process languages** (.xpl) – define a specific business process language in the BPM (see the "Process Language Reference Guide" chapter in the *Advanced User Documentation*).

- **Object languages** (.xol) - define a specific object-oriented language in the OOM (see the "Object Language Reference Guide" chapter in the *Advanced User Documentation*).

- **DBMSs** (.xdb) - define a specific DBMS in the PDM (see the "DBMS Reference Guide" chapter in the *Advanced User Documentation*).

- **XML languages** (.xsl) - define a specific XML language definition in the XSM (see the "XML Language Reference Guide" chapter in the *Advanced User Documentation*).

♦ **Extended model definitions** (.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 the "Extended Model Definitions Reference Guide" chapter in the *Advanced User Documentation*).

♦ **Report templates** (.rtp) - specify the structure of a report. Editable within the Report Template Editor (see the *Reports User's Guide*).

♦ **Report language files** (.xrl) – translate the headings and other standard text in a report (see the *Reports User's Guide*).

♦ **Conversion tables** (.csv) - define conversions between the name and code of an object (see "Using a conversion table" in the "Managing Models" chapter).

Introduction to the Resource Editor

The resource editor allows you to view and edit the resources files provided with PowerDesigner. The left-hand pane shows a tree view of the entries

contained within the resource file, and the right-hand pane displays the
properties of the currently-selected element:
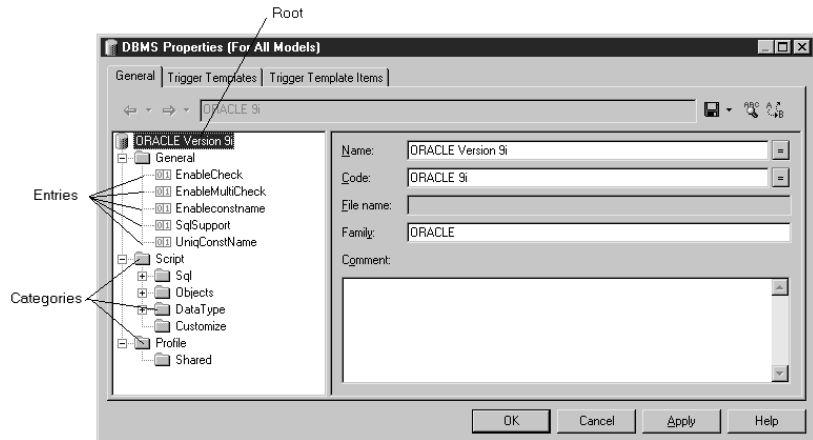
Each entry is a part of the definition of a resource file. For example, you can
define entries for a database command, a characteristic of an object
language, a report item, and so on.

Entries are organized into logical categories. For example, the Script
category in a DBMS language file collects together all the entries relating to
database generation and reverse engineering.

> **Caution**
> You should never modify the resource files shipped with PowerDesigner.
> To create your own resource file For each original resource file you want to
> modify you should create a corresponding new resource file. To do so you
> have to create a new resource file from the List of Resource Files, define
> a name and select the original resource file in the Copy From list. This
> allows you to create a new resource file that is identical to the original file
> apart from the name.

For detailed information about resource files, see the "Resource Files and the
Public Metamodel"chapter in the *Advanced User Documentation* .

Opening the current target language file in the Resource Editor

When working with a BPM, PDM, OOM, or XSM, you can open the target
language file that defines your modeling environment in the Resource Editor
for viewing and editing.

❖ **To open the current target language file**

1. In a BPM, select Language ➤ Edit Current Process Language.

2. In a PDM, select Database ➤ Edit Current DBMS.

3. In an OOM, select Language ➤ Edit Current Object Language.

4. In an XSM, select Language ➤ Edit Current Language.

   The target language file opens in the Resource Editor.

Opening any available resource file in the Resource Editor

You can open, inspect, and edit any resource file from the lists of resource files.

❖ **To open a resource file**

1. Select Tools ➤ Resources ➤ *Type* to open the relevant resource file list.

2. Select a file in the list, and then click the Properties tool.

   The selected resource file opens in the Resource Editor.

Interchanging Models using the XMI Format

PowerDesigner allows you to interchange Unified Modeling Language (UML) models between different UML tool vendors using the XML Metadata Interchange (XMI) standard file format.

XMI combines the benefits of web-based XML standard for defining, validating and sharing document formats on the Web with the benefits of the object-oriented UML. However, the current version of XMI does not support diagrams interchange.

The current version of XMI is 1.1 and PowerDesigner uses the UML 1.3 DTD to generate XMI files.

Importing XMI files

When you import an XMI file, you have to choose an object language and the diagram to open (class or use case). You can then select the .XML file to import.

The corresponding OOM is displayed in the diagram window. By default all symbols are visible in the diagram window.

❖ **To import an XMI file as an OOM**

1. Select File ➤ Import ➤ XMI File to open the New Object-Oriented Model dialog box.

2. Select an object language from the Object Language list.

3. Select whether you want to Share or Copy the object language definition.

4. Select the type of the first diagram to open in the First Diagram list.
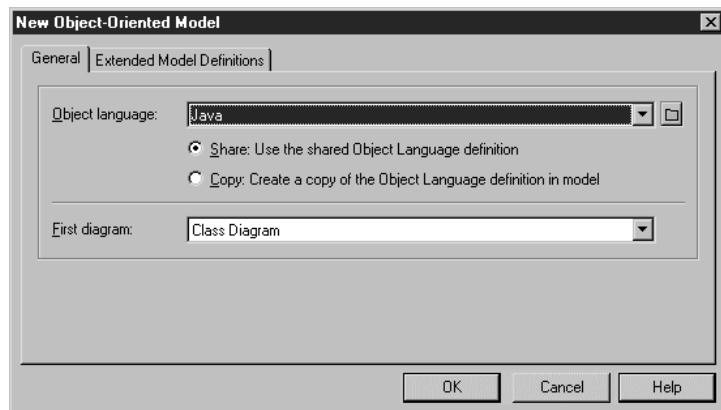
5. Click OK, browse to your XMI file, and click Open.

A progress box shows the progress rate of the import process.

The General page in the Output window, located in the lower part of the PowerDesigner main window, shows the objects import order. At the end of the import process, the OOM corresponding to the imported XMI file is opened in the diagram window.

Exporting XMI files

You can export your PowerDesigner OOM as an XMI file to allow it to be easily opened in other UML modeling tools or code generators like Java, CORBA or C++. Note that only objects from the class diagram and from the use case diagram are exported, and any symbols will be lost as only objects are exported.

❖ **To export XMI files**

1. Select File ➤ Export ➤ XMI File.

2. Enter a filename for your XMI file and click Save.

A progress box shows the progress rate of the export process.

The General page in the Output window, located in the lower part of the PowerDesigner main window, shows the objects export order. Your OOM is exported as an XMI file.

Upgrading from Previous Versions of PowerDesigner

The resource files that customize PowerDesigner for particular target languages or DBMSs are constantly evolving. When you open a model created with a previous version of PowerDesigner, the resource file associated with the model will be automatically upgraded if possible. Note that it may be possible to upgrade a model created with v6.x or earlier but only upgrading from v7.x or later is supported.

Replacing a DBMS from previous versions

All DBMSs stored in the installation directory are updated when you upgrade PowerDesigner.

For DBMSs copied to the model file in previous PowerDesigner versions, you will be invited to upgrade the DBMS when you open the old model.

When you open a PDM the following situations can occur:

DBMS type	Matching DBMS	No match
Share	DBMS upgraded and message in the Output window	Keep DBMS from previous version
Copy	A confirmation dialog box lets you decide if you want to replace the DBMS, if you replace the DBMS, a message appears in the Output window	Keep DBMS from previous version

How to transfer DBMS changes?

If you had customized your DBMS, you can use the following methods to keep the changes in your definition file while switching to the current version:

Shared DBMS   Merge the current version DBMS with DBMS from a previous version, using the merge feature available from the list of DBMS (Tools ➤ Resources ➤ DBMS).

Copied DBMS   Create a new model using a current version DBMS fairly similar to your modified DBMS, and merge it with a model using the DBMS from a previous version.

In the merge dialog box, you should have the current version DBMS in the right pane and the modified DBMS in the left pane. You should be careful when you modify the current version DBMS: apply only the changes you remember, in case of changes in both DBMS versions, try to understand the

differences and avoid modifying the current version DBMS.

Replacing an object language from previous versions

All object languages stored in the installation directory are updated when you upgrade PowerDesigner.

For object languages copied to the model file in previous PowerDesigner versions, you will be invited to upgrade the object language when you open the old model.

When you open an OOM the following situations can occur:

Shared object language   If the object language file is the original file delivered with PowerDesigner, the object language is automatically replaced as mentioned in the Output window. If the object language file is a renamed copy of an object language shipped with PowerDesigner, a message box appears to advise you to change object language.

Copied object language   A message box appears to advise you to change object language. If you click Yes, the object language is automatically replaced in your model. You lose all the changes performed on the definition file. If you click No, the object language is not replaced and remains in the previous version format, you cannot generate for the target language. If you click Cancel the model is not opened.

Converting generic object language

If you had customized an object language from PowerDesigner v8.0 using the generic generation mechanism, the names of the templates may conflict with the public names defined in the current version. You can use the script _PublicNames.pl located in the \Tools directory to avoid conflicts with public names.

CHAPTER 6

Linking and Synchronizing Models in PowerDesigner

About this chapter   This chapter describes the different methods you can use for linking models of heterogeneous type and keeping them synchronized in PowerDesigner.

Contents

Introduction

The PowerDesigner Enterprise Modeling environment permits you to create and exploit complex interdependencies between your models.

When you, for example, generate one model from another, or create a shortcut or define mappings from one model to another, you can track and analyze these dependencies via property sheets. You can also use **impact analysis** , to evaluate how changes made in one model may affect associated models.

The following types of interdependencies can be created between models:

♦ **Generation links** - created during model generation. Each generated object is linked with its origin object.

♦ **Shortcuts** - references that you create explicitly to objects in other models (target objects). When models are open in the workspace, you can see the links between shortcuts and other objects in the Dependencies tab of target objects.

♦ **Manual links**. automatically created in several other cases: when you link objects using extended or calculated collections, extended dependencies, or related diagrams. When you create manual links, shortcuts are silently created in the model.

Linking Models Using Generation

You can link models using the model generation and the synchronize model features. The links in the following figure show the different generations allowed among PowerDesigner models:
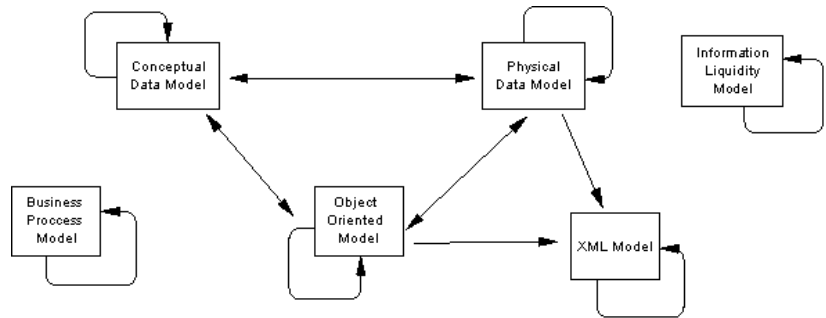
Model generation

You can generate a model from another model of the same type or from another type.

Model generation is used to link models together provided the **Save Generation Dependencies** option is selected in the Detail tab of the generation dialog box. This option creates the generation link itself and makes it possible to keep track of the identity of the origin of each generated object.

For example, you generate a PDM from a CDM when you want to start implementing the physical aspect of a conceptual model. The generation link allows you to iterate CDM to PDM generations preserving generation choices over the process.

Model generation allows you to:

♦ Generate a new model that will contain objects translated from the source mode.

♦ Update a model to impact changes.

☞ For more information on merging two models, see the "Comparing and Merging Models" chapter.

You can select options in the Detail tab of the generation dialog box.

The **Save Generation Dependencies** option is used to create the generation link and to keep track of the identity of the origin of each generated object. It is useful when objects have been renamed in the source or generated

model. It can also be used when you want to merge two models generated from the same model: objects can be compared and recognized as the same object, even if the object has been modified in the merged model.

The **Check model** option is always available; it is used to verify the source model before generation. If an error is found the generation process stops.

Other generation options are specific to the source and target model; you should refer to the corresponding chapter in the different model user's guides.

Preserving shortcuts links during generation

Shortcuts allow you to share objects between different models. A shortcut references a target object in a target model. You can preserve the link between a generated shortcut and its target object during generation.

By default, shortcuts are generated as shortcuts during generation: the Generate check box is selected and the Generated as property is set to Shortcut in the shortcut property sheet. If you want to modify these properties to a large number of shortcuts, you can do it from the List of Shortcuts. This feature allows you to preserve the link between a shortcut and its target object through generation.

To generate shortcuts you have to select the generated models containing the target objects of the generated shortcuts in the Target Models tab of the generation dialog box. Shortcuts can then be correctly linked to target objects in the correct target model.

The Target Models tab displays the following columns:

Column	Description
Target Models	Original target model of the shortcut (not editable)
Generated Models	Lets you select the model that will be used as the target for the generated shortcut

The model generation process allows you to define the target object of a shortcut in a generated model.
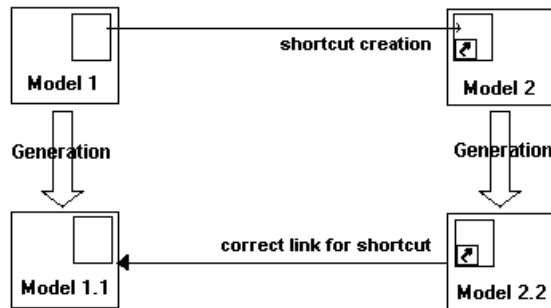
For example, here is the proper sequence of events for shortcuts generation:

1. Model 2 contains a shortcut of an object from Model 1.

2. Model 1 is generated into Model 1.1

3. Model 2 is prepared for generation to Model 2.2 by associating appropriate parameters in the Target Models tab

   If Model 1 is open in the workspace   The Target Model column displays the original target model (Model 1), and its path. The Generated Models column displays the name of the last generated model from Model 1; the next time you generate Model 1, the Generated Models column displays the last model selected. You can click the arrow in the Generated Models column to modify the model selection and choose another generated model.

   If Model 1 is closed in the workspace   The Target Model column displays the original target model (Model 1), and its path. The Generated Models column displays <none>. When you click into the Generated Models column, the original target model Model 1 opens in the workspace in order to find the models generated from Model 1. You can use the arrow to select Model 1.1, the new target that will allow the creation of a correctly linked shortcut in Model 2.2.

4. The shortcut in Model 2.2 is correctly generated with a link to its target object in Model 1.1.

Selecting objects to generate

You can select objects for model generation from the Selection tab. You can generate a model from a global model or from a package within the model. Limiting model generation to a single package is useful when different designers own packages of the same model. Designers can generate their

packages independently from others. Generating a package results in an independent model.

You can display in the list, objects in the current model, or objects in individual packages contained in the model. If you select the Include Sub-packages tool, you can display either all objects in the current model, or all objects in a package.

You have the following selection options:

Parent object	Include Sub-packages tool selected	Include Sub-packages tool not selected
Model	All objects in the model including all objects contained in packages and sub-packages	All objects in the model except objects contained in packages and sub-packages
Package	All objects contained in the package including all objects contained in sub-packages	All objects in the package except objects contained in sub-packages

Generating a new model

You can generate a new model from another model or from a package. PowerDesigner creates a new model containing all the objects that you selected to generate from the source model. The new model is displayed in the Browser and the corresponding diagram opens in the work area.

❖ **To generate a new model**

1. Select Tools ➤ Generate *Model* type to display the generation dialog box.

2. Select the Generate New *Model* type radio button.

3. Select a resource file (DBMS, object language, and so on) if required.

4. Type a new name and code.

5. Click the Configure Model Options button to define the options of the generated model.

6. Click the Detail tab to define options and generation parameters.

7. Click the Target Models tab to select the target models of shortcuts in the current model.

8. Click the Selection tab to select the objects you want to generate.

   You can select the entire model or a package name in the Select Location list. If you click the Include Sub-packages tool you display all the objects belonging to the current package and its sub-packages.

9. Click OK.

   The Output window shows the progress of the generation process. The diagram of the new model is displayed in the Work Area.

Updating an existing model

During an update the source model is converted into a model of the same type as the target and then both models are merged.

If you select the Preserve Modifications option   You can manually compare and merge the existing source model (model to be merged in the right pane) with the newly generated model in the left pane.

If you do not select the Preserve Modifications option   The existing target model is automatically replaced by the newly generated model. The Merge dialog box is not displayed.

In the Update Existing *Model* groupbox, the Select Model list contains all the models already generated from the current model. You can use the tooltip in the list to verify the workspace location and physical path of the generated models.

❖ **To update an existing model**

1. Select Tools ➤ Generate *Model* type to display the generation dialog box.

2. Select the Update Existing *Model* type radio button.

3. Select a model from the Select Model list.

4. Select or clear the Preserve Modifications check box.

5. Click the Detail tab to define options and generation parameters.

6. Click the Target Models tab to select the target models of shortcuts in the current model.

7. Click the Selection tab to select the objects you want to generate.

   You can select the entire model or a package name in the Select Location list. If you click the Include Sub-packages tool you display all the objects belonging to the current package and its sub-packages.

8. Click OK.

   If you cleared the Preserve Modifications check box the updated PDM diagram is displayed in the work area.

   If you selected the Preserve Modifications check box, the Merge dialog box is displayed and displays the new model in the left pane and the existing model in the right pane. You can select or clear check boxes in the right pane for objects that you want to include or delete in the model to be merged. When you clear the check box of a merge action in the right pane, this action will remain deselected the next time in the Merge dialog box.

   ☞ For more information on merging models, see the "Comparing and Merging Models" chapter.

Model synchronization

If you want to link models that do not have a generation link or if this link is broken, you can use the synchronize models feature.

Synchronizing models allows you to associate two models of heterogeneous types in order to create a pair of synchronized models. You can then iterate generations, as you would do with models linked by generation.

Make sure you select the Preserve Modifications check box to display the Merge dialog box that you will use to select which object you want to create or remove in the target model. Selected merge actions are saved into the target model and will no longer be suggested the next time you generate the source into the target model.

❖ **To synchronize models**

1. Open the origin model and make sure the model you want to synchronize is loaded in the workspace, it can be open or closed.

2. Select Tools ➤ Generate *Model Type* from the origin model.

   The generation dialog box is displayed. By default, the Generate New *Model Type* button is selected since the model is not linked to any other model through a generation link.

3. Select the Update existing *Model Type* button.

   The Select Model list is empty.

4. Click the Ellipsis button beside the Select Model list to display a Select Model dialog box. This dialog box lets you select a model of the correct type to update among models available in the workspace (either open or

closed). You can use the Workspace Location and Physical Path columns in the Select Model dialog to select the model to update.

5. Select the Preserve Modifications check box and click OK.

In the Merge dialog box you can define the content of the target synchronized model. The additions and removals are kept in memory and will not be suggested anymore the next time you generate the model.

The next time you will generate the origin model, the corresponding synchronized model will appear in the Select Model box in the Update existing *Model Type* group box.

Linking Models Using Shortcuts

PowerDesigner lets you reference an object defined in another model. When you modify the definition of the referenced object, all its occurrences called shortcuts are updated in the different models.

☞ For a complete description of shortcut properties and behavior, see the "Managing Shortcuts" chapter.

Linking Models Manually

PowerDesigner automatically creates links when you use the following
features:

♦ Requirement to design object link.

♦ Extended dependencies between objects.

♦ Extended collections.

♦ Calculated collections.

♦ Related diagrams.

♦ Data import/export.

♦ Objet mappings.

In each case, shortcuts are silently created in the model.

Linking a requirement to a design object

The following figure shows the links between the Requirements Model and
other PowerDesigner models.
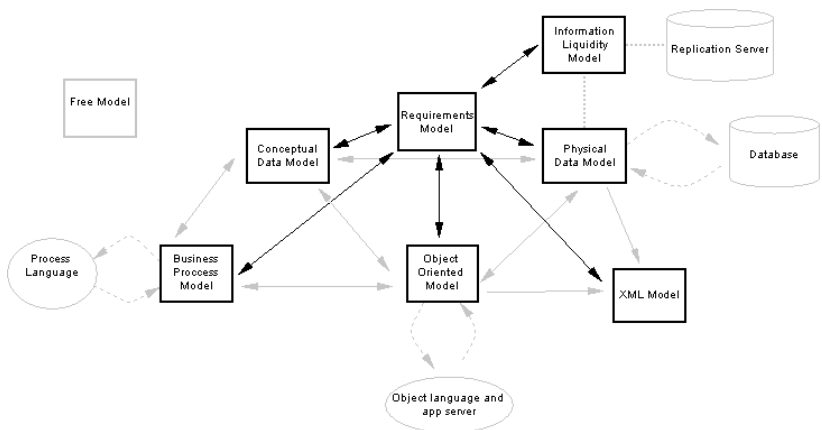
☞ For more information on the RQM, see the *Requirements Model User's
Guide* .

You can link requirements to design objects to make sure that each action
described in the RQM is implemented by an object during the development
process. For example, the Internet Access requirement of the Cyberfridge
example can be linked to a Web Service component in an OOM.

You can link a requirement to a design object in any of the following ways:

♦ Using the Add Links to Design Objects in the Traceability Links tab of the requirement property sheet.

♦ Using the Import Design Objects as Requirements command in the Requirements menu.

♦ Using the Export Requirements as Design Objects command in the Requirements menu.

When you link a requirement to a design object, a shortcut of the design object is automatically created in the RQM.

The great benefits of linking requirement to design objects are:

♦ Make sure each the requirement is implemented by at least one design object, you can use the graphical interface of the Traceability Matrix view to do so.

♦ Verify how a change in a requirement impacts design objects using impact analysis.

☞ For more information on impact analysis used with requirements and design objects, see the ""Monitoring Links Using Impact Analysis" on page 202" section.

Linking objects with extended dependencies

You can create extended dependencies toward objects in models of heterogeneous types.

When you define an extended dependency, a shortcut of the linked object is automatically created in the current model. Inversely, a dependency is added to the target object. For example, you create an extended dependency in table Sale to entity Sale in a CDM: a shortcut of entity Sale is added into the PDM.

If you open the CDM that contains the Sale entity, which is actually the target object of the shortcut in the PDM, the link is displayed in the Shortcut sub-tab in the Dependencies tab of the entity:
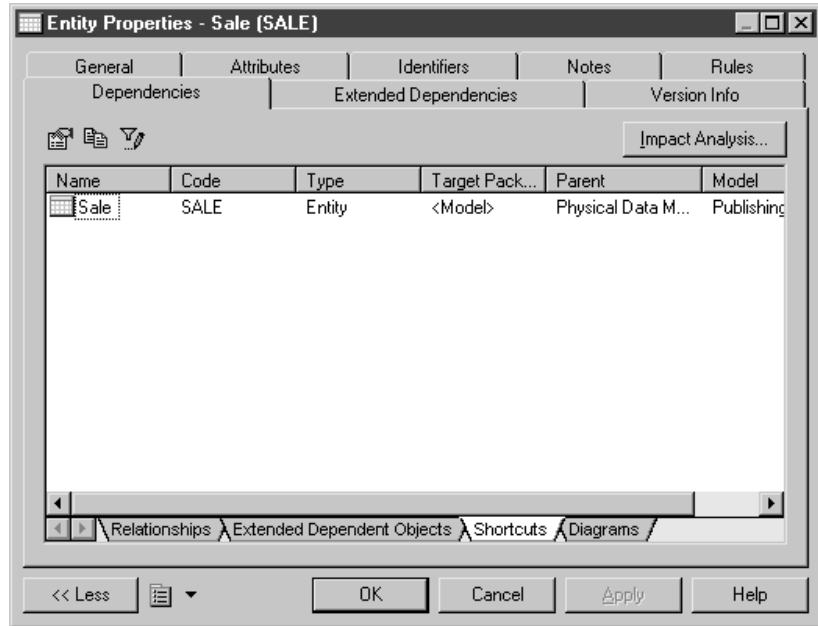
In the Dependencies tab, you can use the impact analysis feature to verify how a change in a target object can affect linked objects.

☞ For more information on impact analysis used with requirements and design objects, see the ""Monitoring Links Using Impact Analysis" on page 202" section.

Linking objects with extended collections

The extended collection feature supports collections of objects from any type of model.

☞ For more information on extended collections, see "Defining an extended collection in a profile" in the "Managing Profiles" chapter in the *Advanced User Documentation* .

For example, you can define an extended collection in the entity metaclass of a CDM that shows processes associated with entities. This adds a new tab to the entity property sheet where you can select processes from the process models open in the workspace. When you add processes to an entity, shortcuts of the processes are automatically created in the current CDM.
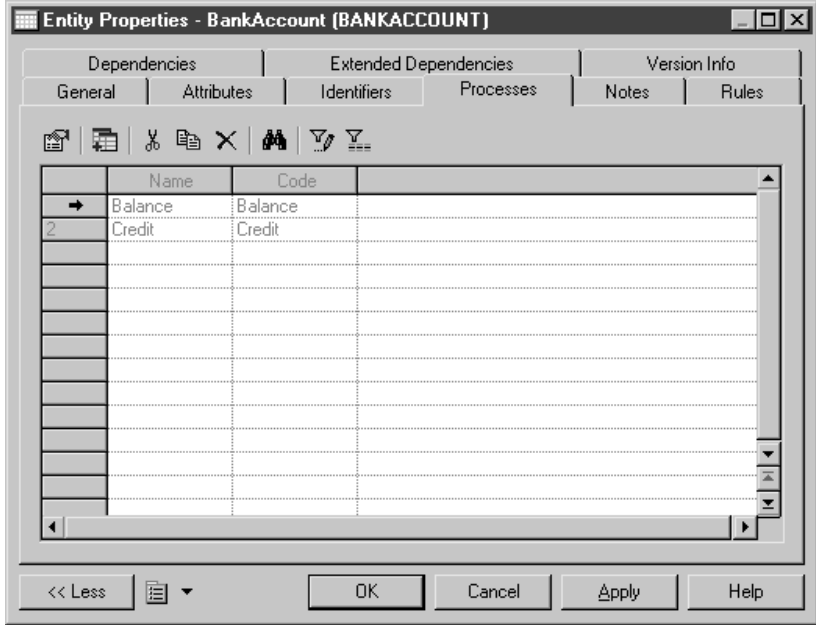
You can use the impact analysis feature from the Dependencies tab of the
shortcuted processes to assess which object would be affected by a change
of the current process.

☞ For more information on impact analysis used with requirements and
design objects, see the ""Monitoring Links Using Impact Analysis" on
page 202" section.

Linking objects with calculated collections

Calculated collections allow you to create lists of associated objects with a
user-defined semantic from any type of model.

☞ For more information on calculated collections, see "Defining a
calculated collection in a profile" in the "Managing Profiles" chapter in the
Advanced User Documentation .

When you add objects to a calculated collection, you can view them in
distinct sub-tabs in the Dependencies tab of the selected metaclass. You can
click the Impact Analysis button in the Dependencies tab to analyze the
impact of a change on the current object on the objects linked to it.

☞ For more information on impact analysis used with requirements and
design objects, see the ""Monitoring Links Using Impact Analysis" on
page 202" section.

Linking objects with related diagrams

The related diagram feature is used to detail the behavior and define the implementation of objects into other diagrams. You can link an object to a diagram of another model type when you need to highlight a specific aspect of implementation.

For example, you can create the list of physical diagrams related to CDM entities. This allows you to follow the implementation process of entities.

☞ For more information on related diagrams, see the "Defining related diagrams" section in the "Managing Models" chapter in the *General Features Guide* .

You can use the impact analysis feature from the Dependencies tab of the diagram to assess which object would be affected by a change of the current diagram.

☞ For more information on impact analysis used with requirements and design objects, see the ""Monitoring Links Using Impact Analysis" on page 202" section.

Linking objects with the data import/export feature

In a BPM, data are conceptual pieces of information that can be specified with more details in a Conceptual Data Model (CDM), in a Physical Data Model (PDM) or in the class diagram of an Object Oriented Model (OOM).

You can export BPM data to create the corresponding objects in the CDM, PDM or OOM. Exported data are automatically linked to the created objects in the BPM via a shortcut in the BPM. For example, if you export data as CDM data items, a shortcut for each new data item is created in the BPM.

You can also import data defined in a CDM, PDM or OOM into a BPM. Imported data are automatically linked to the selected objects in the source model via a shortcut in the BPM.

☞ For more information, see "Exchanging data" in the "Building an Analysis Business Process Model" chapter in the *BPM User's Guide* .

Linking objects with the objet mapping feature

You can map objects when you want to establish a correspondence between objects belonging to heterogeneous models and diagrams. Object mapping creates a structure for data movement and transformation: data comes from a data source and is loaded in another database or model.

You can use the mapping editor to easily create mappings between models objects using drag-and-drop or specific tools. If you drag a source object and drop it on a target model (without defining the target object of the mapping) a new object is automatically created in the target model, it is actually a shortcut of the source object.

☞ For more information on the mapping editor, see "The Mapping Editor" chapter.

Monitoring Links Using Impact Analysis

You can use the impact analysis feature to monitor the links existing between your models. Impact analysis is a powerful tool for evaluating the consequences of changes on objects before performing these changes or before consolidating a list of changes in the repository.

Using impact analysis with generated models

You need to consolidate your models into the PowerDesigner Repository to benefit from the impact analysis feature.

To have the Generated As sub-tab displayed in the Dependencies tab of the source objects you need to consolidate and extract your models with the **Extract Dependencies** option selected.
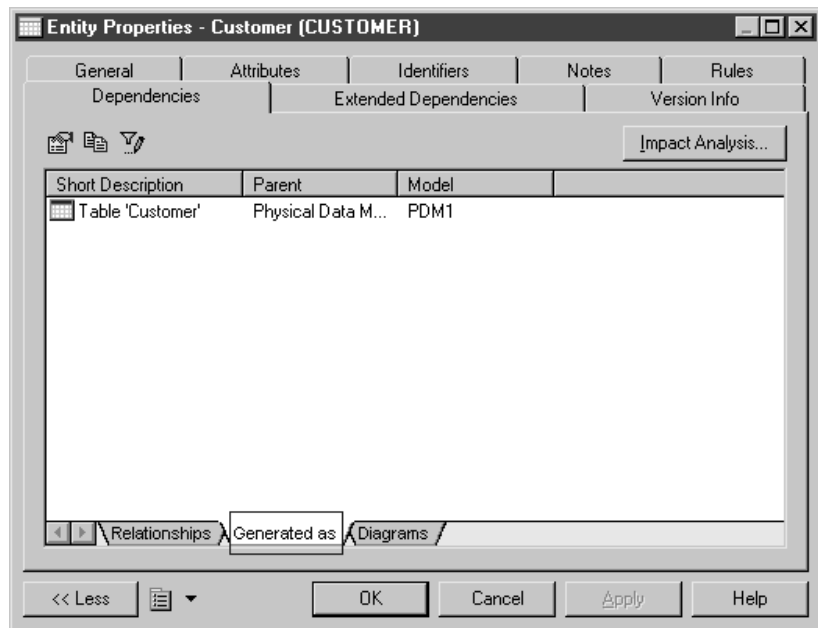
In the Impact Analysis dialog box, this link does not appear by default, you have to click the **Extract Dependencies** tool to display the generation links:
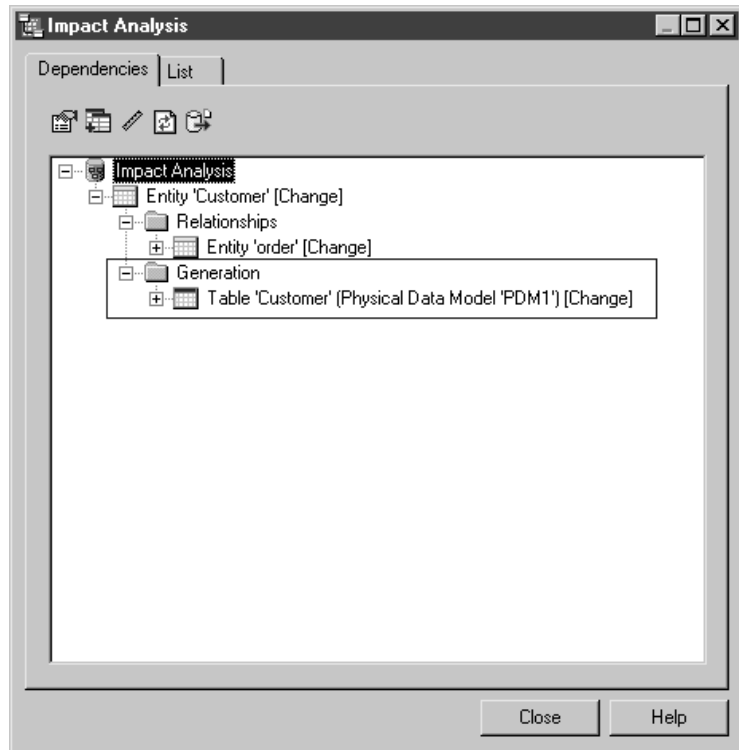
Using impact analysis with shortcuts

You can use impact analysis to verify if an object is the target of a shortcut and evaluate the impact of changes on this target object.

A shortcut sub-tab automatically is displayed in the Dependencies tab of a target object provided the referencing model is open in the workspace. This is to inform you that the current object has shortcuts in other models.

If you click the Impact Analysis button in the Dependencies tab you can see a graph of dependencies of the current object, and the impact of any type of change on the target object.

If the referencing model is closed but consolidated in the repository, you can click the **Extract Dependencies** tool in the impact analysis dialog box to display shortcut links.

Using impact analysis with requirements linked to design objects

When you create a traceability link between a requirement and a design object this link is displayed in the Design Object sub-tab of the

Dependencies tab of the requirement, you can click the Impact Analysis button to display the graph of dependencies of the current requirement:
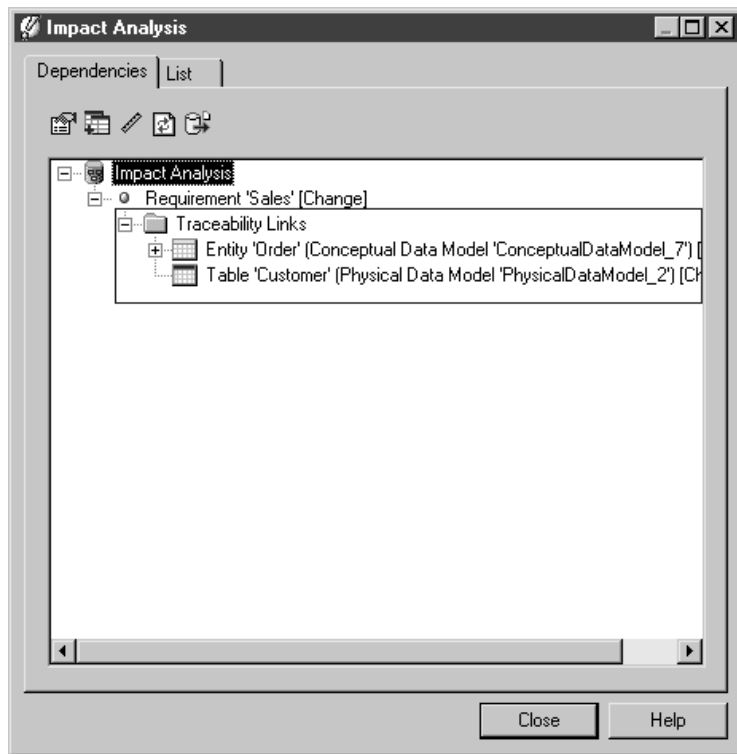
Using impact analysis with extended dependencies

When you create an extended dependency, PowerDesigner automatically creates a shortcut of the linked objet in the current model and an Extended Dependent Objects sub-tab is automatically created in the Dependencies tab of the linked object.

When you want to modify the linked object, you can check if it is linked to other objects through extended links using the impact analysis graph:
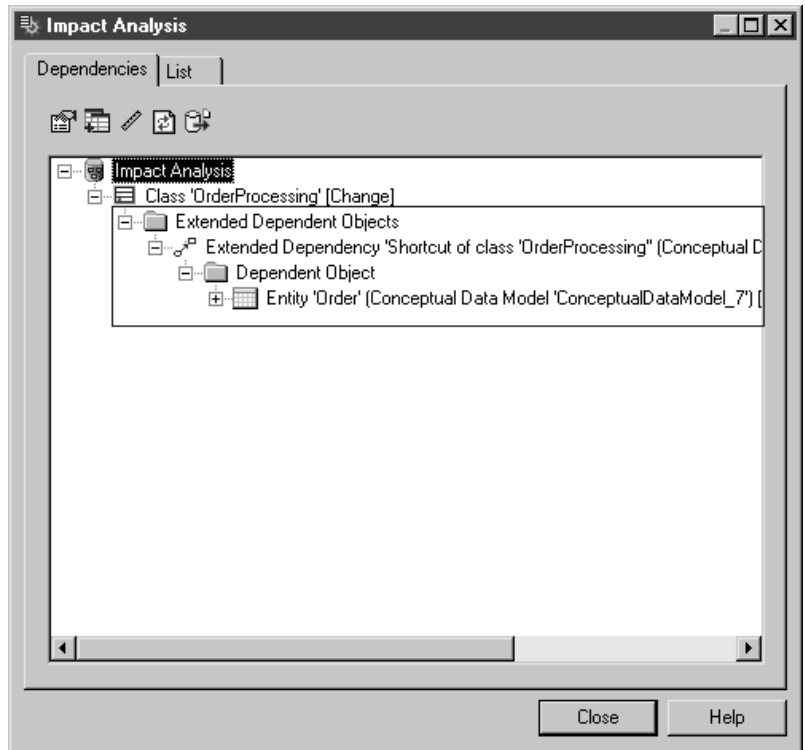
Using impact analysis with extended collections

You can create an extended collection for a selected metaclass and link
objects across models. The objects you link to the current metaclass display
an Extended Inverse Collection sub-tab in their Dependencies tab to indicate
that they are used in an extended collection.

The same information is displayed in the impact analysis graph of an object
added to an extended collection:
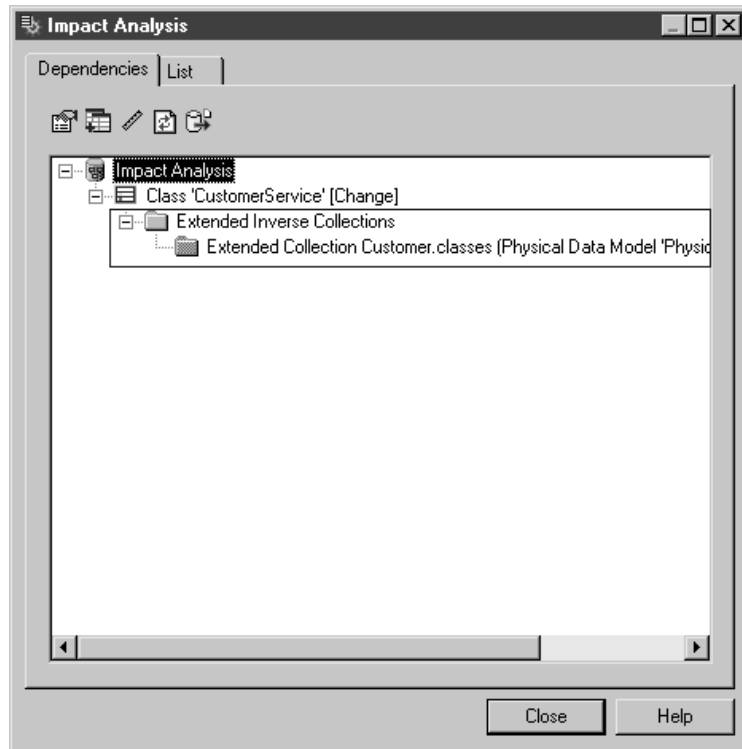
Using impact analysis with calculated collections

When you add objects to a calculated collection, you can view them in distinct sub-tabs in the Dependencies tab of the selected metaclass.

Using impact analysis with related diagrams

When you declare a diagram as related to an object it implies that the diagram participates or helps understand the implementation of the current object. A Related Objects sub-tab is displayed in the Dependencies tab of the related diagram, it displays the list of objects that use the current diagram for implementation purpose. You can click the Impact Analysis button and use the dependencies graph to further analyze the impact of changes in the current diagram:
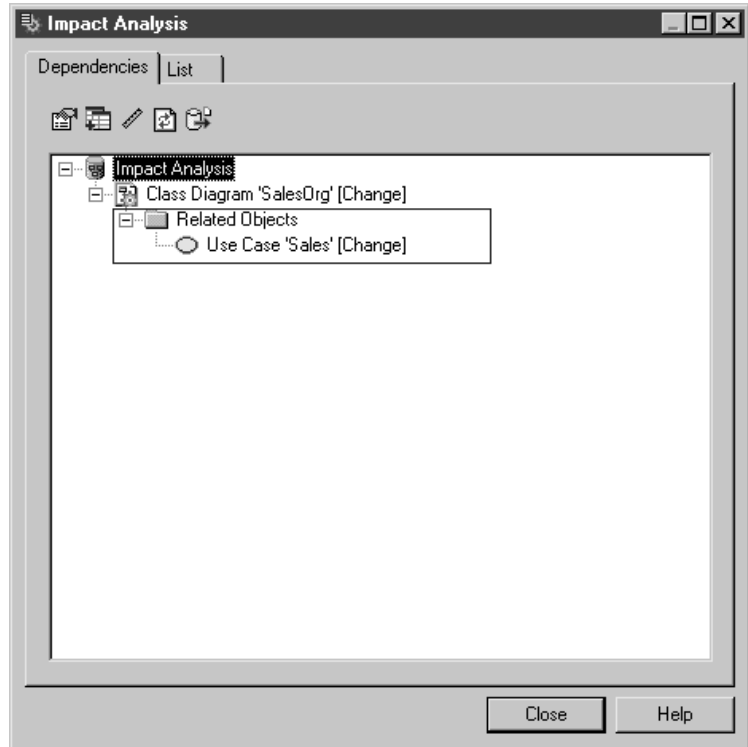
Using impact analysis with data import/export

When you export data to a CDM, PDM or OOM, you can open the property
sheet of the new object, click the Dependencies tab and click the Impact
Analysis button to display the link between the new object and the data it
comes from.
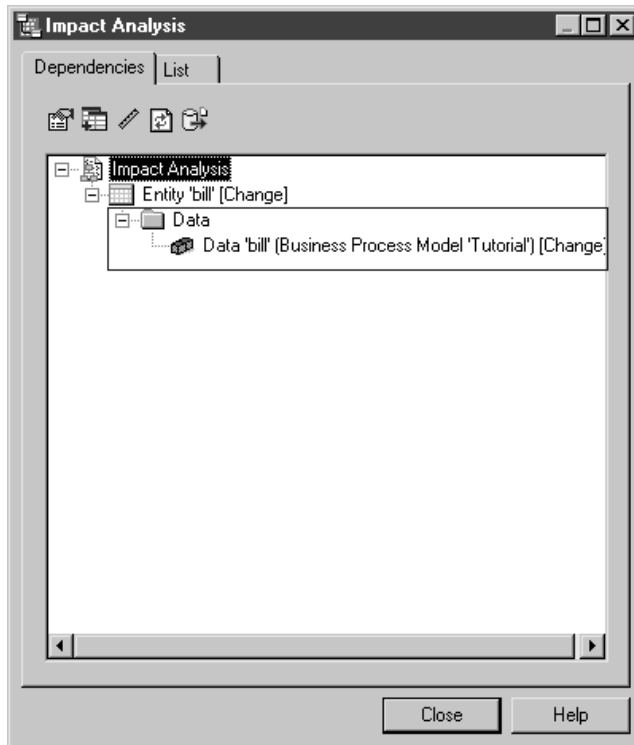
You can also use impact analysis to view the data linked to (imported as) a particular object in a CDM, PDM, or OOM.

Using impact analysis with object mappings

You can use impact analysis to verify the mappings of a source object: mappings are displayed in the Dependencies tab of source objects, and in the Impact Analysis dialog box.

Impact Analysis

Dependencies | List

Impact Analysis
  Table 'Table_1' [Change]
    Mappings
      Table Mapping 'tt.customer.Mapping_1' (Physical Dat

Close   Help
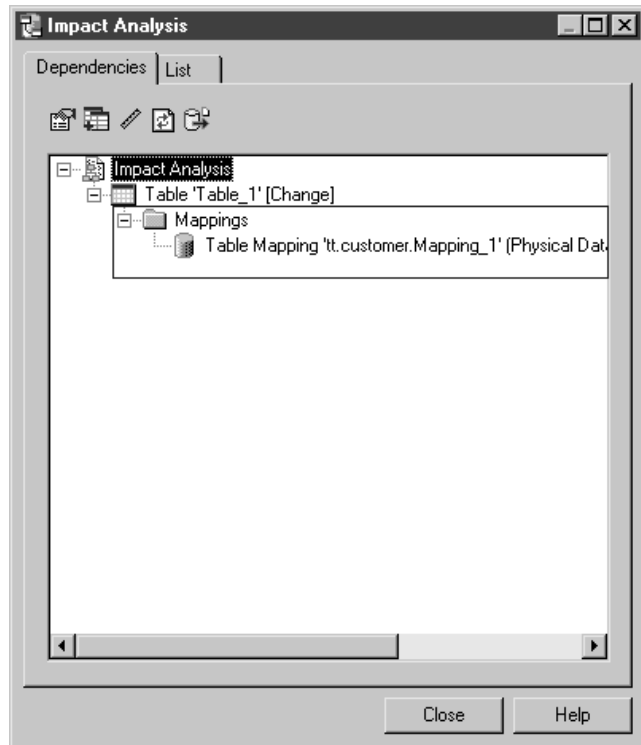
CHAPTER 7

Managing Objects

About this chapter   This chapter explains how to manage objects in PowerDesigner.

Contents

Defining Objects

Objects the things that you model with. For example, tables, data items, generalizations, and packages are PowerDesigner objects.

There is detailed information about each object in the relevant model User Guide. The following sections describe generic features available for all objects.

All objects appear as nodes in the Browser, and many are available via object lists (in the Model menu) and have a **symbol** that can be displayed in the diagram window.

Objects and Symbols
Objects appear in the Browser. What you can see in the diagram windows are instances of objects called **symbols**.

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.

☞ For more information about the management of namespace, see section

Managing the namespace in models in chapter Managing Models.

Creating an object

You can create objects in a model or in a package in three different ways:

♦ Graphically, in the diagram window

♦ In the Browser

♦ From an object list

Creating an object graphically

When objects support symbols, you can create them from the diagram window using the tools available in the Palette.

❖ **To create an object graphically**

1. Click an object creation tool in the Palette.

2. Click an empty space in the diagram.

   The symbol of the newly created object is displayed in the diagram window. The corresponding node also is displayed in the Browser.

3. Right-click to release the tool.

4. Double-click the symbol of the newly created object in the diagram window to display the object property sheet.

5. Type an object name and an object code.

6. Click OK to return to the model diagram.

Creating a link object graphically

You can create link objects from the diagram window using the tools available in the Palette.

❖ **To create a link object graphically**

1. Click a link object creation tool in the Palette.

2. Click inside the source object and, while continuing to hold down the mouse button, drag the cursor to the destination object. Then, release the mouse button.

   A link object symbol is created and displayed between the two source and destination objects.

3. Double-click the new link object symbol in the diagram to display the link object property sheet.

4. Type a link object name and code.

5. Click OK to return to the model diagram.

Complete links   The Complete Links command available from the Tools menu allows you to display the symbols of links existing in the model. If you select objects in the diagram, the Complete Links command only applies to these objects.

Creating an object from the Browser

You can create objects from the Browser. Link objects can be created from the Browser only if you have previously defined the required source and destination objects.

❖ **To create an object from the Browser**

1. Right-click a model, package or an object category in the Browser, and select New ➤ *object type* from the contextual menu.

The object is created and its property sheet opens.

2. Type an object name and an object code.

3. [for link objects] Select a source and a destination object.

4. Click OK.

The object node is displayed under its corresponding category in the Browser.

> **Associated symbol in diagram**
> By default, when you create an object from the Browser, it does not have a symbol. You can create a symbol for the newly created object by pressing the SHIFT key and dragging and dropping the node into the diagram window.

Creating an object in an object list

You can create an object in an object list. Link objects can be created in a list of objects only if you have previously defined the required source and destination objects.

❖ **To create an object in a list of objects**

1. Select Model ➤ *object list* to open the appropriate object list.

2. Click the Add a Row tool.

   or

   Click an empty line in the list.

   or

   [if the list is ordered] Click the Insert a row tool, .

   A new item is added at the end of the list or before the selected row in the list.

3. Type an object name and an object code.

4. [for link objects] Select a source and a destination object.

5. Click OK.

Associated symbol in diagram

By default, when you create an object in a list of objects, the symbol of the new object is displayed in the current diagram.

☞ For more details about object lists, see "Lists" in the Using the PowerDesigner Interface chapter.

Object properties

You define an object with properties that appear in its property sheet. Property sheets organize object properties on tabs.

❖ **To display an object's property sheet**

1. Double-click the object symbol or its entry in the Browser.

   or

   Right-click the object symbol or its entry in the Browser, and select Properties from the contextual menu.

Different types of objects have different tabs, but almost all PowerDesigner object property sheets have the following tabs:

♦ **General** – provides basic information about the object. Almost all objects have the following properties:

   • **Name** - clearly identifies the object. By default names can have up to 254 characters, and can include upper, lower, and mixed case strings.

- **Code** – is a reference for the object. It is used in scripts that are generated from the model. By default, codes can have up to 254 characters, and can include upper, lower, and mixed case strings.

- **Comment** – [optional] provides a more detailed description of the object. You can display (and edit) object comments on their many diagram symbols using the "Comment" Display Preference. This feature can be useful when importing an ERwin model into a CDM or PDM.

♦ **Notes** – lists additional information about the object. For more information, see "Attaching notes to an object" on page 216.

♦ **Rules** – lists the business rules that the system must follow. A business rule could be a government-imposed law, a customer requirement, or an internal guideline. For more information, see "Business Rules" on page 228.

♦ **Version Info** - provides details about the object owner, creation, and modification date.

♦ **Dependencies** - lists all the objects that depend on the object

♦ **Extended dependencies** - lists all the objects on which the object depends

---

Navigating between tabs

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.

Opening property sheets at last accessed tab

Property sheets open to the General tab by default. However, you can choose to open property sheets at the last accessed tab by selecting Tools ➤ General Options ➤ Dialog, and selecting the Keep Last Tab option in the Property Sheets groupbox.

Attaching notes to an object

♦ The Notes tab in an object property sheet contains two sub-tabs:

- **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, which includes the following tools:

Tool	Description
	[SHIFT + F11] Open Editor Menu.
	For more information, see "Working with free text" in the Model Graphics chapter.
	[CTRL + E] Edit With. Opens your default RTF editor you previously defined or allows you to select another editor if you click the down arrow beside this tool.
	☞ For information about defining a default editor, see "Specifying external text editors" in the Using the PowerDesigner Interface chapter.

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.

❖ **To insert a .txt or .rtf file in the RTF editor**

1. Double-click an object in the diagram to display the object property sheet.

2. Click the Notes tab.

3. Click the tab (Description or Annotation) that contains the text you want to edit.

   The corresponding tab is displayed.

4. Click the Open Menu tool and select Insert from the dropdown menu.

   A standard Open dialog box is displayed.

5. Browse to the directory that contains the file to open.

6. Click OK.

   The content of the file is displayed in the RTF editor.

> **Object is annotated**
> You can see if an object in the diagram has notes by opening the object list to which it belongs. If the check box corresponding to the object is selected in the N column, the object in the diagram is annotated. For more information about customizing the display of a list, see "Filtering a list" in the Using the PowerDesigner Interface chapter.

Previewing the code to be generated from an object

Many objects have a Preview tab in their property sheets, that allow you to preview the code to be generated from the object. This provides an easy way to apply modifications to your code and update the model.

The following tools are available on this tab (though some may be unavailable for certain objects):

Tool	Description	Keyboard shortcut
	Open Editor Contextual Menu	SHIFT + F11
	Edit With contextual menu. This allows you to select an editor for the script	CTRL + E
	Refresh	F5
	Select generation targets. This tool is available when at least one extended model definition flagged for generation is linked to the model and when it contains GeneratedFiles entries for the current object. When available, it displays the list of targets for the current object. If you add a generation target, the corresponding tab is added to the Preview tab. If you deselect a generation target, the corresponding tab disappears from the Preview tab	CTRL + F6
	Show generation options. If you select the Show Generation Options tool when available, the Generation Options dialog box is displayed. You can change generation options from this dialog box and see the impact on the code	CTRL + W
	Ignore generation options. If you click the Ignore Generation Options tool when available, the preview ignores generation options selected by using the Show generation options tool but uses a predefined set of options	CTRL + D

Script bookmarks

In the Preview tab, you can add and remove bookmarks at specific points in the code and then navigate forwards or backwards from bookmark to bookmark:

Keyboard shortcut	Description
CTRL + F2	Adds a new bookmark. A blue bookmark box is displayed. If you repeat this action from the same position, the bookmark is deleted and the blue marker disappears. Note that bookmarks are not printable and are lost if you use the Refresh, or Show Generation tools.
F2	Jumps to bookmark
SHIFT + F2	Jumps to previous bookmark

☞ For more information about how the Preview tab is used in the different modules, see the appropriate user's guide.

❖ **To preview the code of an object**

1. Double-click an object in the diagram to display its property sheet.

2. Click the Preview tab to preview the code.
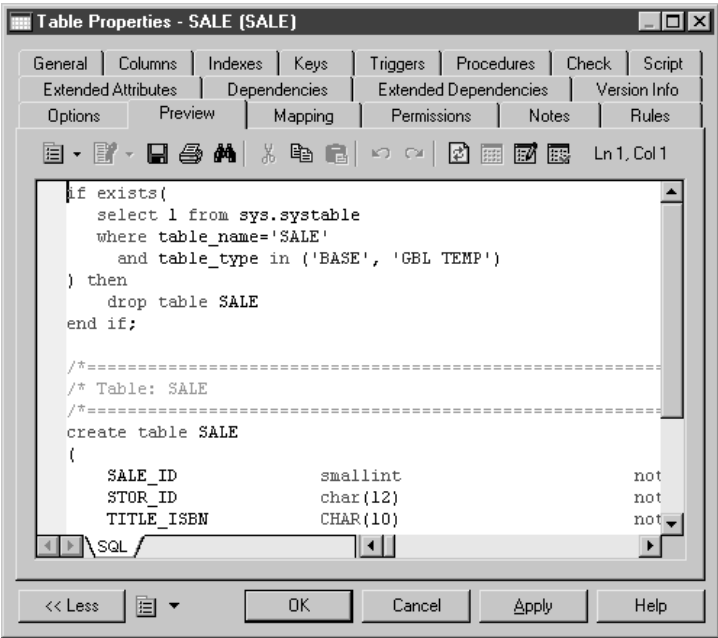
3. Click OK.

Modifying object properties

You can modify the object properties using one of the following methods:

♦ From the object property sheet

♦ From the object type list

♦ From the object symbol in the diagram

❖ **To modify an object's properties from its property sheet**

1. Double-click the object symbol or its entry in the Browser.

   or

   Right-click the object symbol or its entry in the Browser, and select Properties from the contextual menu.

2. Enter changes to object properties on the various tabs.

3. Click OK to close the property sheet and return to the model diagram.

Object lists display all the objects of a given type in the current model or package. If you want to edit the properties of multiple objects, you can often do this more quickly by editing in the object list.

❖ **To modify one or more object's properties from an object list**

1. Select Model ➤ *Objects* to display the corresponding List of *Objects* .

2. Select the object that you want to modify by clicking the numbered bar in the left-hand column.

   or

   Click and hold as you select multiple lines.

   The line(s) of the selected object(s) are highlighted.
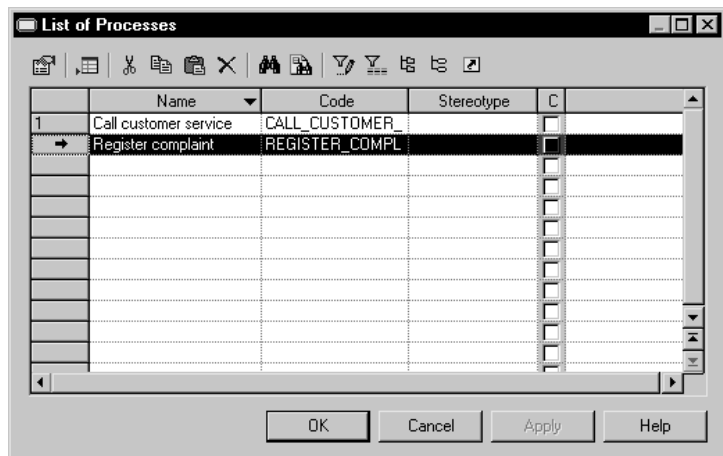
3. Modify any of the properties of the object(s) directly in the list.

> **Display the column you need**
> If you do not see the column you need, display it with the Customize
> Columns and Filter tool. For details, see section Applying the cus-
> tomization parameters to a list in chapter Using the PowerDesigner
> Interface.

or

Double-click the arrow at the beginning of the line to display the object
property sheet. If you have selected multiple objects, the property sheet
of the first will be displayed. Modify the properties as required.

4. Click OK.

You can modify an object property (like name, code, stereotype, or
cardinality) directly from its symbol in the diagram without having to open
its property sheet. Note that you cannot modify read-only properties or
properties that display in a list (such as table columns), in this way.

> **Renaming an object from its symbol in the diagram**
> You can rename an object from its symbol in the diagram by right-clicking
> the symbol and select Edit ➤ Rename from the contextual menu.

You can .

❖ **To modify properties of an object from its symbol in the diagram**

1. Click an object symbol and Press F2.

   or

   Select an object symbol and then click it again.

   or

   Press CTRL while right-clicking a property name in the object symbol.

   The object name, or the first object property if the name is not displayed,
   becomes editable.

2. Modify the property of the object in the edit zone. You can navigate to
   the next or previous editable property using TAB and SHIFT+TAB

3. Click outside the edit zone to commit the change

   The normal checks are performed and in case of error, the edit is lost.

> **Edit in Place general option**
> You can also select the Edit in place after creation option in the General
> Option dialog box to automatically be able to modify the name of objects
> from their symbol in the diagram, whenever you create them from the
> palette. For more information about how to select the Edit-in place
> option, see section Defining the graphic tool behavior in chapter Using the
> PowerDesigner Interface.

Renaming an object from the Browser

You can rename an object in the Browser, without having to open the object
property sheet.

❖ **To rename an object from the Browser**

1. Select the object node and press F2.

   or

   Select the object node and then click it again.

   or

   Right-click the object and select Rename from the contextual menu.

2. Rename the object.

Analyzing object dependencies

Objects collaborate with other objects in a variety of ways. In an object's
property sheet, the Dependencies tab shows the different links that exist
between the object and other objects in the model and any other models open
in the workspace.

☞ For more information about displaying object dependencies, see
"Displaying object dependencies" in the Using the PowerDesigner Interface
chapter.

☞ For more information about how to display the **uses** of an object in
closed models, see "Auditing repository activities" in the Managing
Repository Documents chapter in the *Repository User's Guide* .

Using extended attributes

Extended attributes are used to complement the definition of a metaclass in
order to:

♦ **Control generation** for a given generation target. In this case, extended
attributes are defined in the target language or DBMS of the model. For

example, in the Java object language, several metaclasses have extended attributes used for generating Javadoc comments.

♦ **Further define model objects** in extended model definitions. For example, in the extended model definition for Sybase ASA Proxy tables, the extended attribute called GenerateAsProxyServer in the DataSource metaclass is used to define the data source as a proxy server.

Some extended attributes are defined by default in the resource files that ship with PowerDesigner. But you can also create additional extended attributes according to your needs. To do so, you have to use the profile feature available in the resource editor.

☞ For more information about profiles in a resource files, see the "Managing Profiles" chapter in the *Advanced User Documentation* .

Extended attribute properties

Each extended attribute has the following properties:

Property	Description
Name	Name of extended attribute.
Data type	Extended attribute data type including boolean, color, date, file, float, font, etc or customized data types.
Value	Value of the extended attribute. This field displays the default value defined for the extended attribute data type.
R	Redefined value. This check box is selected if you modify the default value in the Value column, using either the down arrow or the ellipsis button.

How to define the value of an extended attribute?

You can define the value of an extended attribute in the Value column that is displayed in the Extended Attributes tab of an object property sheet. If extended attributes are defined in a user-defined tab, it is even easier to define their values, see "Displaying extended attributes in specific tabs" on page 226.
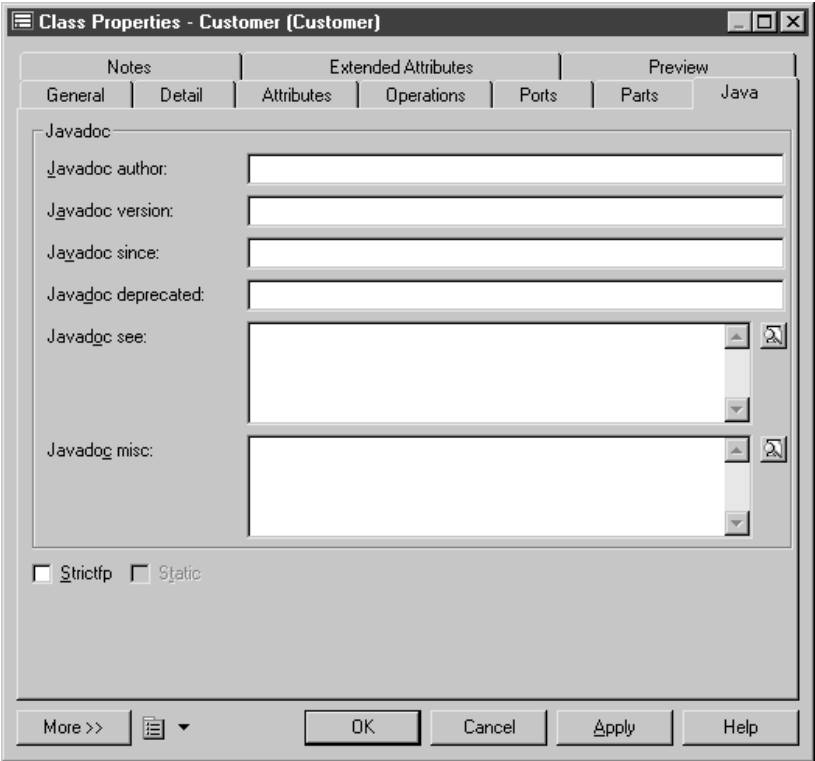
You can also display and modify extended attributes in object lists, but you need to modify the default display of the list using the Customize Columns and Filter tool in the list toolbar. In the following example, some javadoc extended attributes are displayed in the list of interfaces.
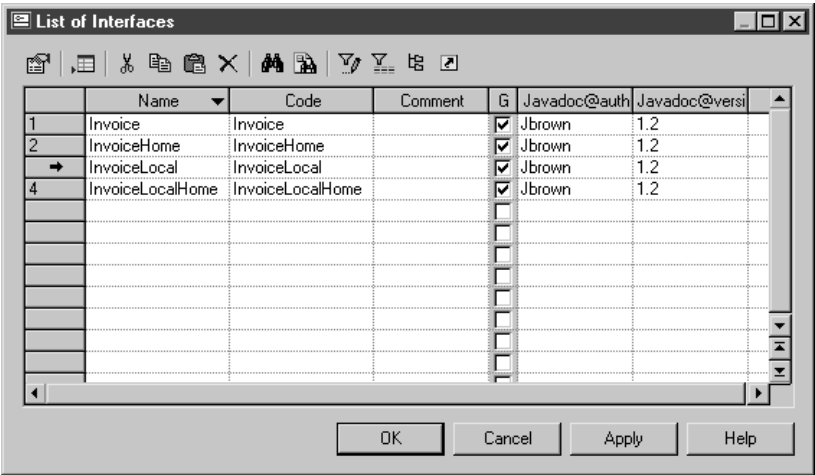
❖ **To define the values of the extended attributes of an object**

1. Open the property sheet of an object.

   or

   Select Model ➤ Object to display a list of objects.

2. Click the Extended Attributes tab to display the corresponding tab.

   or

   Click a user-defined tab to display a user-define tab for extended attributes.

   or

   Click the Customize Columns and Filter tool, select extended attributes in the list of columns, and click OK.

3. Click the Value column of an extended attribute if you want to modify its value and select a value from the list.

   or

   Type values directly in the boxes of the user-defined extended attribute tab.

   or

   Type or select a value in the value cell in the list.

4. Click OK.

Displaying extended attributes in specific tabs

You can create additional tabs in a property sheet when you want to improve extended attributes presentation. By default, extended attributes are listed in the Extended Attributes tab of an object's property sheet. This list gathers heterogeneous attributes in alphabetical order. You can create additional tabs to better present extended attributes and make it easier to enter their values.

Additional tabs allow you to address the following problems:

♦ You have a very large list of extended attributes; you can display them in different property tabs with a meaningful title in order to retrieve them easily.

♦ Extended attributes are of very heterogeneous types and need to be organized logically instead of alphabetically.

♦ Extended attributes are related and need to be grouped together in order to clearly indicate their dependency link.

♦ You need to define the extended attribute value using the appropriate editor.

To create an additional property tab you use the form extended feature available from the Resource Editor. You can create a "Property tab" form for a selected metaclass, stereotype or criterion and select the extended attributes to display in this tab.

☞ For more information see the "Managing Forms in a Profile" section, in the "Managing Profiles" chapter in the *Advanced User Documentation* .

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 in any of the following ways:

♦ Select Model ➤ Business Rules to access the List of Classes, 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

For general information about creating objects, see the "Getting Started with PowerDesigner" chapter.

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	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Descriptive label for the rule

Property	Description
Type	Specifies the nature of the business rule. You can choose between:
	♦ Constraint – a check constraint on a value. In a PDM, constraint business rules can be generated in the database. For example, "The start date should be inferior to the end date of a project."
	♦ Definition – a property of the element in the system. For example; "A customer is a person identified by a name and an address".
	♦ Fact – a certainty in the system. For example, "A client may place one or more orders".
	♦ Formula – a calculation. For example, "The total order is the sum of all the order line costs".
	♦ OCL constraint [OOM only] – An Object Constraint Language expression. See "Business rule property sheet OCL Constraint tab" on page 230.
	♦ 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".

Business rule property sheet Expression tab

A business rule typically starts out as a description. As you develop your model and analyze your business problem, you can complete a rule by adding a technical expression. Expressions are used primarily in CDMs and PDMs.

Each business rule can include two types of expression, which you define on the appropriate sub-tab:

♦ Server

♦ Client

Business rule property sheet 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.

❖ **To apply a business rule to an object**

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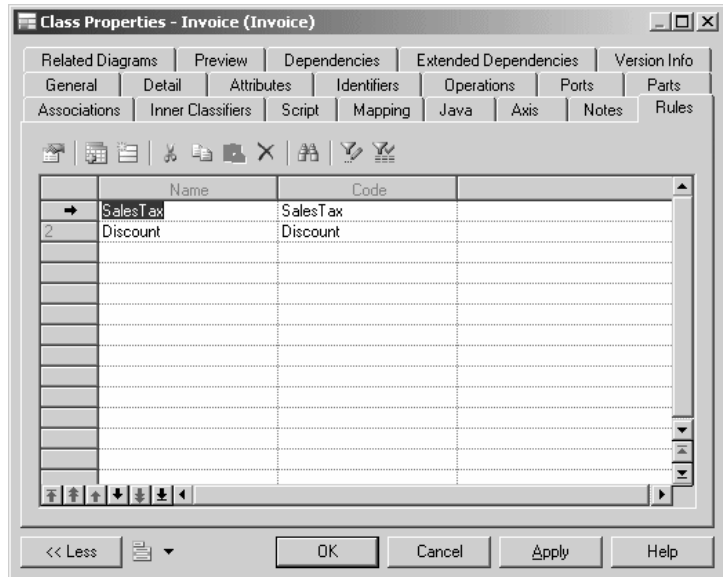
2. Click the Add Rules tool to open a selection window listing all the business rules available in the model:
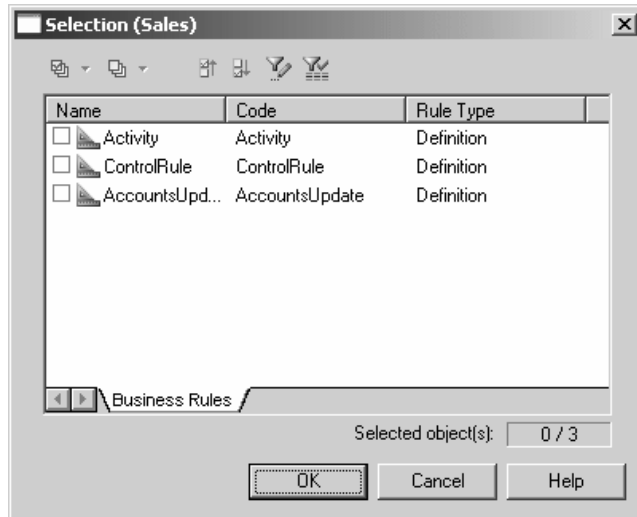
3. Select the business rules you want to add to the object, and then click OK to return to the object's property sheet.

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

List of Business Rules U Column
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.

Defining Extended Objects and Links

Extended objects and links are additional objects that can be added in any kind of model to let you design specific business needs or concepts that are not supported by PowerDesigner standard objects.

Extended objects and links exist by default in the free model; they can also be added to any other model to provide greater design flexibility.

☞ For more information about the free model, see "Using the Free Model" in the Managing Models chapter.

For example, you can use extended objects in a PDM to design database specific objects that do not exist by default in the model. These objects are not generated or reverse engineered by default, but you can add generated files in order to define extended generation. You could also use extended objects to design UDDI related data in an OOM.

How to use extended objects and links

Extended objects and extended links are available in all PowerDesigner models. However, only the free model displays these objects by default. If you want to use extended objects and/or links in another type of model, you have to add the corresponding metaclasses and their tools in the current model.

Adding the extended object and link metaclasses to the current model

By default, extended objects and links do not appear in models other than the free model. You have to add them in the Profile category of the resource file attached to your model. The resource file can be an extended model definition for those models that do not support languages like the CDM.

Once added, you can customize the extended objects and links using stereotypes. You can also define extended collections for these metaclasses in order to better integrate extended objects in the semantics or your model.

☞ For more information about how to customize profiles, see chapter Managing Profiles in the *Advanced User documentation* .

☞ For more information about extended collections, see section Defining an extended collection in a profile in chapter Managing Profiles in the *Advanced User documentation* .

❖ **To add the extended object and link metaclasses to the current model**

1. Open a resource file in the resource editor.

   You may have to create and attach an extended model definition if the current model is a CDM.

2. Right-click the Profile category and select Add Metaclasses in the contextual menu.

   The Metaclass Selection dialog box is displayed.

3. Click the PdCommon tab at the bottom of the dialog box to display the list of objects common to all models.

4. Select the ExtendedLink and ExtendedObject check boxes and click OK.

   The metaclasses are added to the profile. You can further define them using stereotypes, custom tools and extended collections.

5. Click Apply in the resource editor.

Adding the extended object and link tool to the palette of the current model

The tools for creating extended objects and extended links do not appear by default in the palette of models other than the free model. However, you can add them to the model palette.

❖ **To add the extended object and link tools to the palette of the current model**

1. Select Tools ➤ Customize Toolbars to display the Toolbars dialog box..

2. Select the Palette check box in the toolbars list and click the Customize button to display the Customize Toolbars window.

3. Select the Graphical Tools category and drag the tools corresponding to the extended object or the extended link to the palette of your model, and release the mouse button.
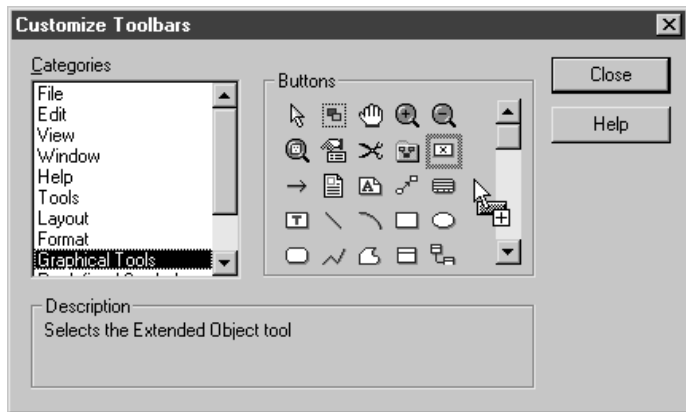
The tools appear in the palette of your model.

4. Click Close in each of the dialog boxes.

Extended object properties

You can double-click any extended object symbol in the diagram to open its property sheet:

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Additional information about the extended object
Stereotype	Sub-classification used to extend the semantics of the extended object. You can create stereotypes in the Profile category of the resource file attached to the current model

Modifying the extended object display preferences

You can modify the following display preference for extended objects using the Tools ➤ Display Preferences command:

Preference	Description
Stereotype	Displays the stereotype of the extended object
Comment	Displays the comment of the extended object

Extended link properties

An extended link is a full-featured object that can be created between any type of object in a model.

When you create an extended link, you have to define the following properties.

Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Additional information about the extended object
Source	Name of the extended object origin of the extended link
Destina- tion	Name of the extended object destination of the extended link
Stereotype	Sub-classification used to extend the semantics of the extended link. You can create stereotypes in the Profile category of the resource file attached to the current model

CanLinkKind event handler

You can use the **CanLinkKind** event handler to restrict the kind and stereotype of the objects you want to link together. The CanLinkKind event handler has two input parameters: the source and the destination extremity of the link, note that these cannot be shortcuts.

The CanLink event handler is called when you create a link using the corresponding tool in the Palette or when you try to modify the ends of a link from its property sheet.

☞ For more information about the CanLinkKind event handler, see

"Defining an Event Handler in a Profile" in the "Managing Profiles" chapter in the *Advanced User Documentation* .

Modifying the extended link display preferences

You can modify the following display preferences for an extended link using the Tools ➤ Display Preferences command:

Preference	Description
Stereotype	Displays the stereotype of the extended link
Name	Displays the name of the extended link

Dragging, Dropping, Copying, Pasting, Moving, 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

Right-click + drag and drop
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.

You can modify the Browser drag and drop default behavior from the General Options dialog box.

❖ **To modify the default drag and drop behavior**

1. Select Tools ➤ General Options to open the General Options window.

2. Select a Browser Drag & Drop Default action radio button .
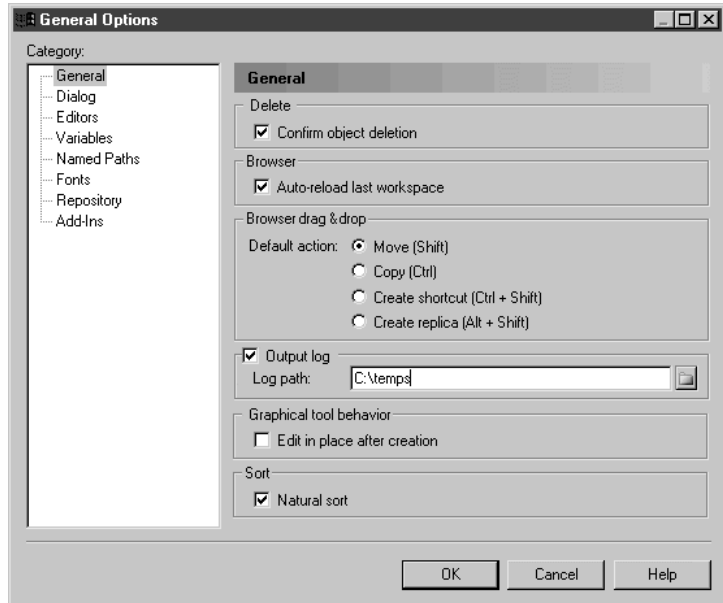
3. Click OK.

See also "Setting PowerDesigner General Options" in the "Using the PowerDesigner Interface" chapter.

Copying objects

You can copy objects from the following areas of the PowerDesigner interface:

♦ The Browser

♦ A diagram

♦ A list of objects

♦ The Result List

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

You can copy an object to the Clipboard in any of the following ways.

❖ **To copy a selected object to the Clipboard**

1. Select Edit ➤ Copy from the PowerDesigner menu bar.

   or

   Press CTRL + C on the keyboard.

   or

   Right-click and select Edit ➤ Copy from the contextual menu of the object.

   or

   Open a list of objects, select one or several lines in the list and Press CTRL + C on the keyboard.

Pasting objects

When you paste an object in PowerDesigner, 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.

You can paste objects into the following locations:

♦ A container node (folder, model or package) in the Browser

♦ A diagram

♦ A list of objects

♦ An external application

❖ **To paste an object from the Clipboard inside PowerDesigner**

1. Select Edit ➤ Paste from the PowerDesigner menu bar.

   or

   Press CTRL + V on the keyboard.

   or

   Right-click and select Edit ➤ Paste from the contextual menu of an object.

   or

   Open a list of objects, select a line in the list and Press CTRL + V on the keyboard.

   > **Drag and drop +**
   > CTRL When you hold down the CTRL key during a drag and drop operation, you obtain the same result as a copy/paste operation.

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 a shortcut

You can copy a shortcut and paste it in another model or package, the result is a shortcut with the same properties as the original shortcut.

You can use the drag and drop feature for shortcut creation by:

♦ Pressing the SHIFT and CTRL keys

♦ Selecting the Create shortcut option as the default drag and drop behavior

☞ For more information about selecting, shortcut creation as the default drag and drop behavior see "Dragging and dropping objects" on page 238.

You can paste a shortcut into the following PowerDesigner targets:

♦ The Browser

♦ A diagram

☞ For more information about shortcuts, see the Managing Shortcuts chapter.

Pasting in a diagram   When you paste a shortcut into a diagram the result is the following:

Shortcut source	Paste result
Same model or package Same diagram	New graphical synonym of shortcut No object creation
Same model or package Different diagram	New shortcut symbol or graphical synonym No object creation
Different model or package	If the shortcut does not exist, a shortcut with its symbol are created
	If the shortcut already exists without a symbol, symbol is created
	If a shortcut and a symbol already exist, a graphical synonym is created

☞ For more information about synonyms, see section "Creating a Graphical Synonym for an Object" on page 250.

❖ **To paste a shortcut**

1. Select Edit ➤ Paste.

   Disabled paste
   You cannot paste a shortcut in a model or a package where the same shortcut already exists.

Pasting an object as a shortcut

When you copy an object you can also paste it as a shortcut in a referencing model or package.

☞ For more information about shortcuts, see the Managing Shortcuts chapter.

You can paste an object as a shortcut into:

♦ The Browser

♦ A diagram

The paste as shortcut feature works according to the following rules:

Paste destination	If a shortcut already exists	If a shortcut does not exist
Browser List of objects	Nothing happens	Shortcut created
Same model or package but different diagram	Nothing happens	Nothing happens
Same model or package and same diagram	Nothing happens	Nothing happens
Different model or package	Symbol created if there was no symbol Graphical synonym created if symbol already existed	Shortcut + symbol created

❖ **To paste an object as a shortcut**

1. Select Edit ➤ Paste As Shortcut.

   or

   Right-click the Browser target or the diagram window and select Edit ➤ Paste As Shortcut from the contextual menu.

   > **Drag and drop +**
   > CTRL+SHIFT When you hold down the CTRL AND SHIFT keys during a drag and drop operation, you obtain the same result as a paste as shortcut operation.

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 the identifying property is just the code, for other objects, identifying properties are the name and the code.

☞ For more information about the identifying properties of an object, see "Defining Objects" on page 212.

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.

Renaming an entity   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	New entity with new name and code
Allow Reuse	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 the Browser, a diagram, or an object list. You can choose to

♦ Delete the object (including any sub-objects) from the model. For example, when you delete a table from a PDM, you delete its columns, keys, triggers and indexes.

♦ Delete the symbol from the diagram, but keep the object in the model

If you delete an object that is connected to another object via a link object, you will also delete the link object.

> **Deleting domains and data items in a CDM or PDM**
>
> 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 the Conceptual Data Model and Physical Data Model User's Guides.

❖ **To delete an object from the Browser**

1. Select the object node in the Browser and press the DEL key.

   or

   Right-click the object node in the Browser and select Edit ➤ Delete from the contextual menu.

   The Confirmation dialog box is displayed.
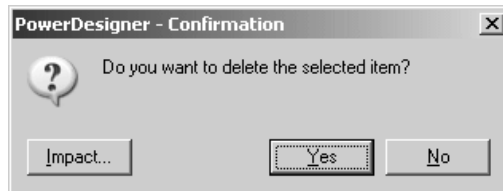
2. [optional] Click the Impact button to evaluate the impact of the deletion (see the Impact Analysis chapter).

3. Click OK.

   The object, its symbol, and any sub-objects are deleted from the model.

❖ **To delete an object from the diagram window**

1. Select the object symbol in the diagram and press the DEL key.

   or

   Right-click the object symbol in the diagram and select Edit ➤ Delete from the contextual menu.

   The Confirm Deletion dialog box is displayed. Note that if you delete a free symbol, such as free text, a line or a shape, no confirmation is required.
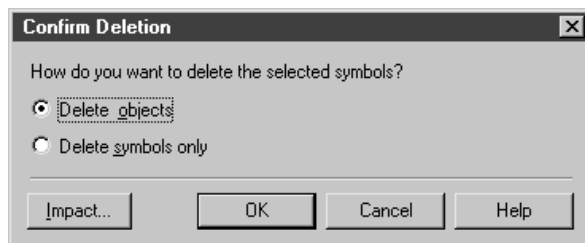
2. [optional] Click the Impact button to evaluate the impact of the deletion (see the Impact Analysis chapter).

3. Choose one of the following delete options:

   ♦ Delete objects – deletes the object, with all its properties and sub-objects from the model

   ♦ Delete symbols only – deletes the object symbol from the diagram, but leaves the object available in the model for future use

4. Click OK.

   The object (or only its symbol) is deleted from the model.

❖ **To delete an object from a list**

1. Select Model ➤ Object to display a list of objects.

2. Select one or more items in the list and then click the Delete button.

   No confirmation dialog box opens. The object is deleted directed from the model.

3. Click OK to close the list and return to the model diagram.

Undo deletion

You can select Edit ➤ Undo to undo the deletion. You can restore the symbol in the diagram, by selecting Symbol ➤ Show Symbols and selecting the object's check box in the Show Symbols dialog box.

Suppressing the delete confirmation dialog box

By default, you are required to confirm any deletion of an object. You can suppress these confirmation messages.

❖ **To suppress the confirmation of object deletions**

1. Select Tools ➤ General Options to open the General Options dialog box .

2. Clear the Confirm Object Deletion check box.
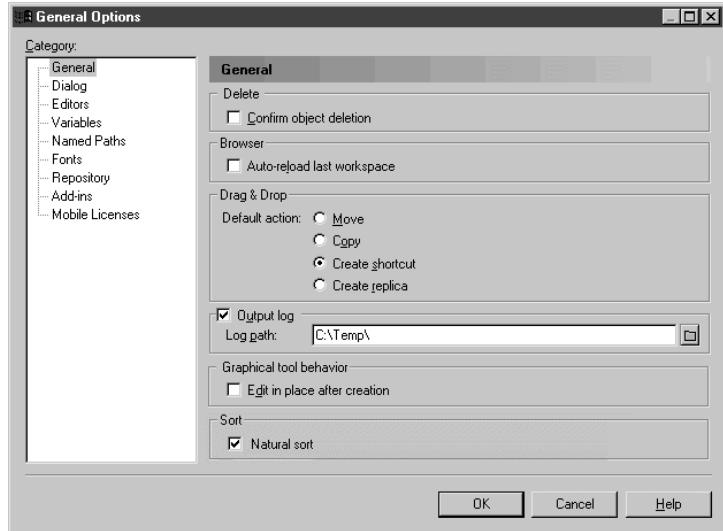
3. Click OK to return to the model diagram.

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 section Creating a shortcut in chapter Managing shortcuts.

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 chapter Managing shortcuts.

Moving 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:
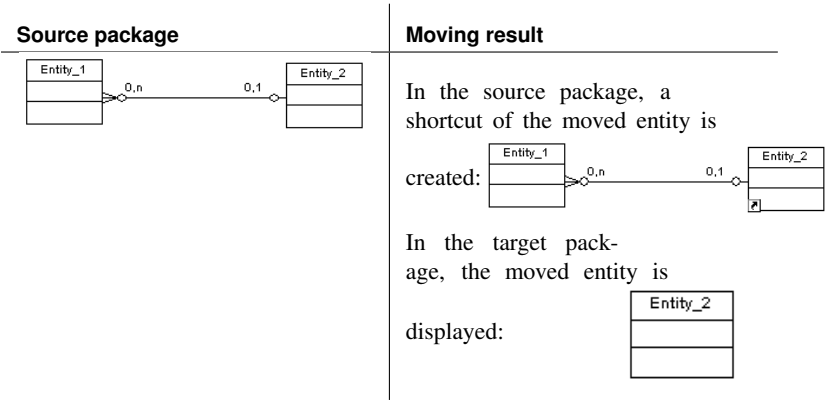
Moving an object with an
oriented link

Source package	Moving result
TABLE_2 TABLE_1	

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):

❖ **To move an object from a package to another**

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 a Graphical Synonym for an Object

A graphical synonym is an additional symbol for an object. It has no specific definition in itself but takes the one of the object it represents.

By reproducing the same object at different places in the diagram, graphical synonyms can improve the readability of the diagram as links become shorter.

You can create as many graphical synonyms as you want within the same diagram.

You can create graphical synonyms of graphical synonyms as they are graphical representations of the same object. You can also create graphical synonyms of object shortcuts.

In the diagram, the graphical synonym displays the name of the object followed by a colon and the number of the occurrence.

This class is a graphical synonym for the class Printer.

Moving graphical synonyms	A graphical synonym exists in a diagram as long as the object it represents also exists.

If you move a graphical synonym from a package to another, it is moved to the target package and shortcuts are created for the remaining graphical synonyms in the source package.

Link object graphical synonym

When you create a graphical synonym for a link object, you automatically create one for its extremities, provided they also support graphical synonyms. If not, the creation of the graphical synonym for the link object will fail.

The following object symbols do not support graphical synonyms:

Model	Object symbol
CDM	Inheritance
OOM	Swimlane, synchronization, decision
BPM	Swimlane, synchronization, decision

You cannot graphically distinguish a link object graphical synonym from a simple link object, unless you display the name attribute in the diagram using the Display Preferences dialog box. However, when you select a link

object in the diagram, the text (Graphical Synonym) is displayed in the tooltip for the object link.

In the following example, the creation of the graphical synonym for this reference has forced the creation of a graphical synonym for its extremities (tables). The tooltip shows that the reference is a graphical synonym.
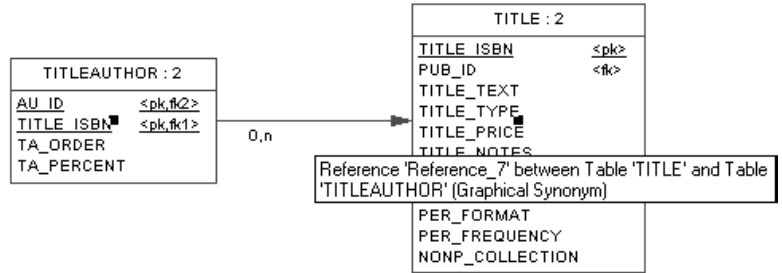
If you delete a link object graphical synonym, only the symbol is deleted without asking any confirmation.

The following link objects do not support graphical synonyms:

Model	Link object
CDM	Inheritance link
OOM	Transition (Activity and Statechart diagrams) Instance Link (Collaboration and Object diagrams) Messages (Sequence and collaboration diagrams) Association class link (class diagram) Interaction Frame, Fragment, and Reference (Sequence)
BPM	Flow
All	Free symbol Line

❖ **To create a graphical synonym for an object**

1. Right-click an object in the diagram.

2. Select Edit ➤ Create Graphical Synonym from the contextual menu.

   The graphical synonym is displayed in the diagram window.

Navigating between graphical synonyms in a diagram
You can find a particular graphical synonym in a diagram by right-clicking a graphical synonym in the diagram window and select Edit ➤ Find Graphical Synonym from the contextual menu, then select a graphical synonym from the list. The graphical synonym is centered and selected in the diagram window.

Finding Objects

PowerDesigner lets you search for objects within all the models currently open in your workspace in order for you to:

♦ Locate objects in the different models in the workspace and modify their properties

♦ Find all the shortcuts related to a given object

♦ Reuse objects from one model to another

❖ To find an object
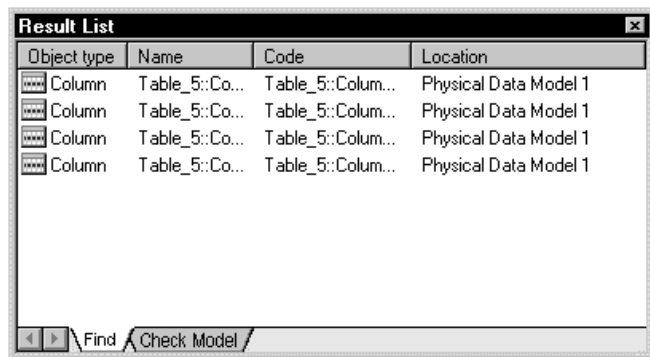
1. Select Edit ➤ Find Objects to open the Find Objects dialog box.

2. Specify the appropriate parameters on the Name and Location, User and Date, and Advanced tabs. For more information, see "Find object parameters" on page 253.

3. Click the Find Now button.

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

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

   The Find Object dialog box remains open until the end of the process, then it is closed and the Result List displays the result.

Find object parameters

You can define very precise parameters in the Find Objects dialog box:

♦ The name and location of PowerDesigner objects.

♦ Advanced find criteria on the properties of PowerDesigner objects.

♦ The creation or modification date of PowerDesigner objects.

Name and Location
parameters

The following parameters are available on this tab

Parameter	Description
Look in	Specifies the scope of the search. You can select the entire workspace, a folder, a model, or a package within a model. You may need to navigate up or down the dynamic list in order to make all possibilities available.
Model type	Specifies the type of PowerDesigner model to be searched. 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	Specifies that object shortcuts that correspond to your criteria will be searched.

Parameter	Description
Name	Specifies the object name to search for. You can use the following special characters: ♦ * - none to any number of characters ♦ ? – exactly one character ♦ * - * is a normal character ♦ \? - ? is a normal character ♦ \\- \is a normal character ♦ true/false - Boolean value (True is when the check box is selected in the interface) For example: W* finds "Work" and "Washington" *IST finds "List" and "Specialist" *96 finds "01/11/96" and "26/08/96" ????ER finds "Writer" and "Seller" but not "inner" COLN? finds "COLN1" and "COLN2" *_emp_??? finds "Div_emp_fun" and "Div_emp_idn"
Code	Code of the object. You can type the exact code of the object or use a string expression
Case sensitive	If you select this check box, you will find objects with case matching with the Name and Code expressions

User and Date parameters

The following parameters are available on this tab

Parameter	Description
Created check box	Select this check box if you want to define creation parameters
Created by	Name of the user who created the object
Date	Creation date criteria, you can search: ♦ Before a given date ♦ After a given date ♦ Between two dates ♦ During a period of time
Modified check box	Select this check box if you want to define modification parameters
Modified by	Name of the user who made the last modification to the object
Date	Modification date criteria, you can search: ♦ Before a given date ♦ After a given date ♦ Between two dates ♦ During a period of time

Advanced parameters

When you select an object type, the Advanced tab allows you to specify advanced find parameters on each property of the selected object type. This advanced search feature also supports the use of wildcards and string expressions.
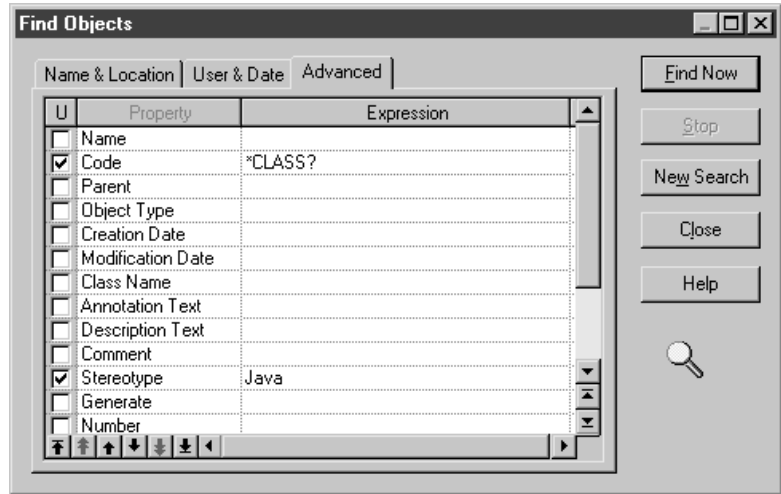
The following parameters are available on this tab: Note that if you do not select an object type, you define advanced Find parameters only on the name and code properties. These properties being common to all types of objects.

Find criteria	Description
U column (U for Used)	When selected, implies that the property will be taken into account during the Find process
Expression	Allows you to define a string expression that will become a find criteria on the selected property

Selected property with no expression defined
If you select the Used check box for a property without any expression, it is equivalent to a null value, the find process will consequently look for objects which selected property is null.

Using the Result List

From the Result List, you can:

♦ Copy an object and paste it as a new object or a shortcut

♦ Find an object in the Browser

♦ Find an object's symbol in one or more diagrams

♦ Modify the properties of an object

❖ **To copy an object from the Result List**

1. Right-click the object in the result list and select Copy from the contextual menu.

2. Select the appropriate destination in the Browser or diagram window and select Edit ➤ Paste or Edit ➤ Paste as Shortcut from the contextual menu

❖ **To find an object from the Result List in the Browser**

1. Right-click the object in the result list and select Find in Browser from the contextual menu.

   The object is highlighted in the Browser

Objects identification
The Location column also allows you to know where objects have been defined.

❖ **To find the symbol(s) of object from the Result List**

1. Right-click the object in the result list and select Find in Diagram from the contextual menu.

   The appropriate diagram is opened and the symbol centered in the window

   or

   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.

Object without symbols
If you select an object that does not have an associated symbol in the diagram, a warning message is displayed.

❖ **To modify object properties from the Result List**

1. Right-click the object in the result list and select Properties from the contextual menu.

   The object's property sheet opens.

2. Make any appropriate changes and then click OK

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 "Defining Templates and Generated Files in a Profile" in the "Managing Profiles" chapter in the *Advanced User Guide* .

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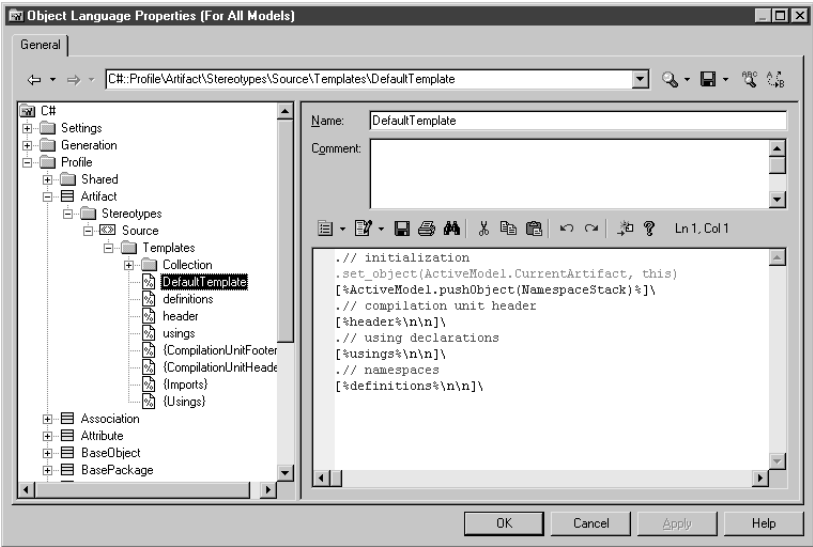
Example   You define the following artifact criterion in an extended model definition:
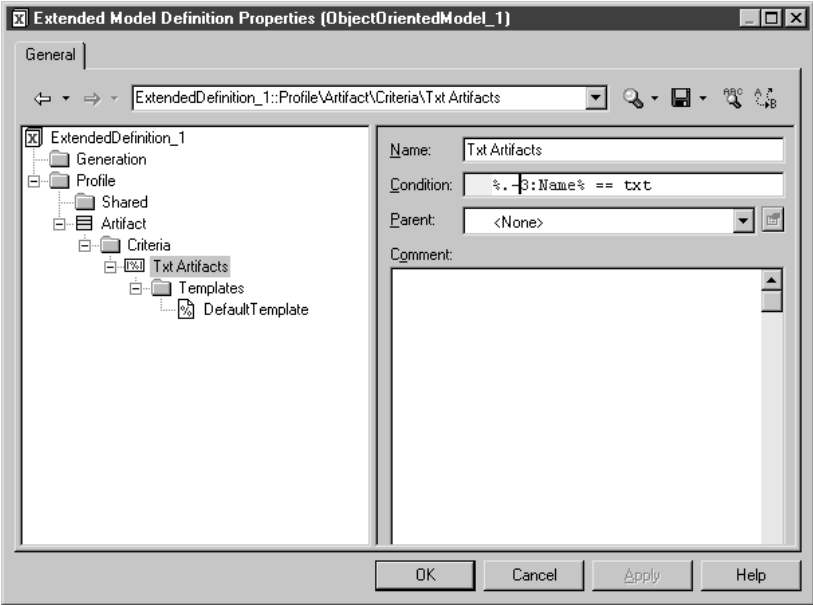
This criterion verifies that the last 4 characters of the artifact name are .TXT. You also need to create the default template in order to define the content of the generated file. In this example, DefaultTemplate is defined as follows:

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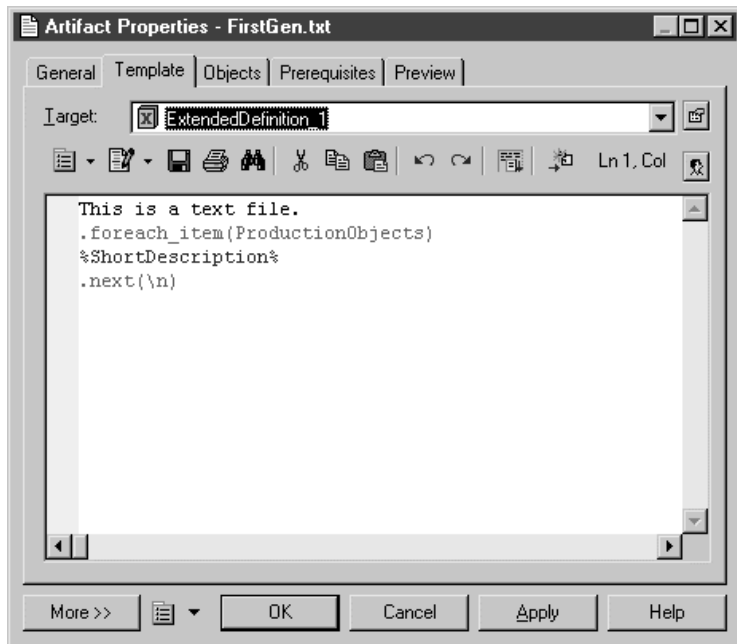
You can now define instances of objects you want to include in the file generated from the current artifact:
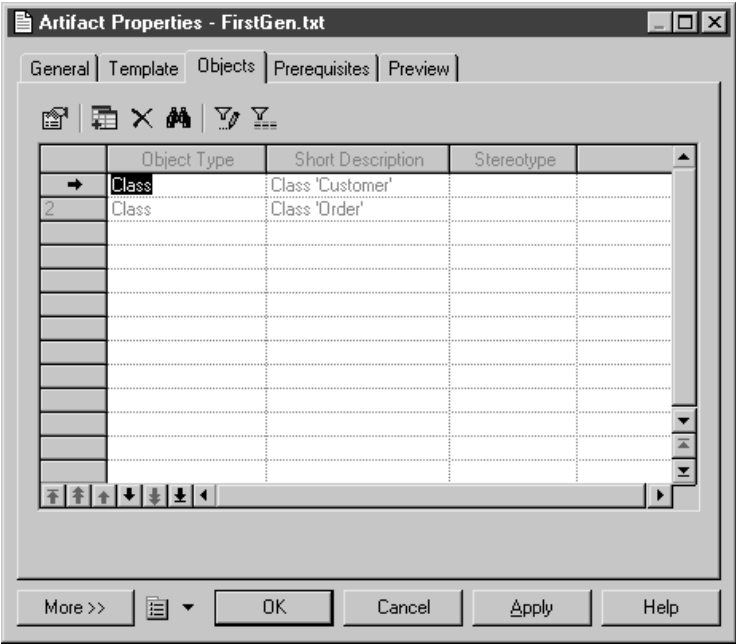
Defining an artifact

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.
Stereo-type	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.

❖ **To define an 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.
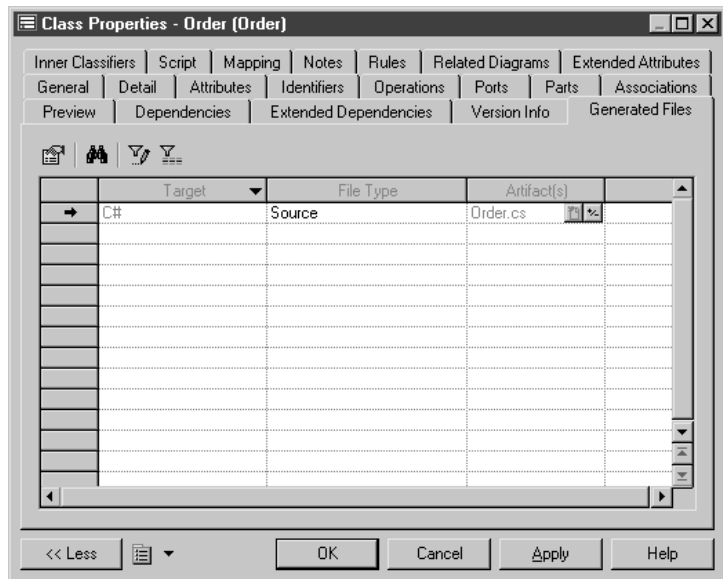
4. Click Apply and click the Properties tool to define artifact properties.

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:

♦ Change the artifact name using the Edit in place feature and avoid opening the artifact property sheet

♦ Drag and drop authorized objects from the diagram or the Browser to an artifact in order to add the code of this instance to the artifact

Artifact folder   An Artifact folder is used to create an artifact hierarchy for generation.

You can create as many folders as you need using the New ➤ Artifact Folder command in the Artifact category or model contextual menu. The artifact folder property sheet is displayed to let you define a name and comment for the folder. You can then create artifacts in the folder structure.
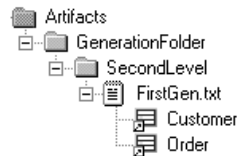
Generated files tab

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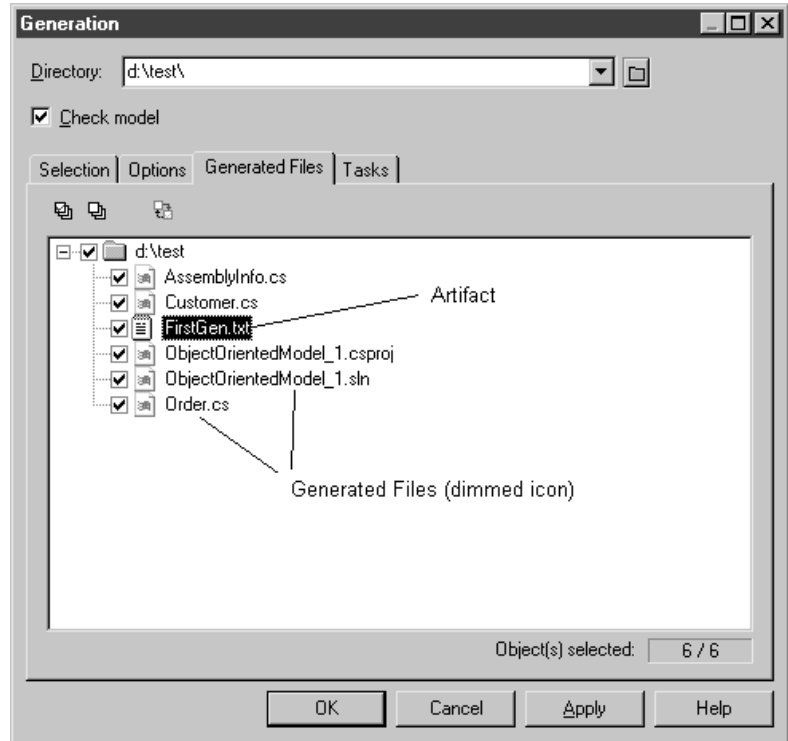
You can modify generation options from this dialog box, and you can also check artifact generation completeness: if an exclamation mark is displayed on the artifact icon it means that one or several prerequisite artifacts are missing.

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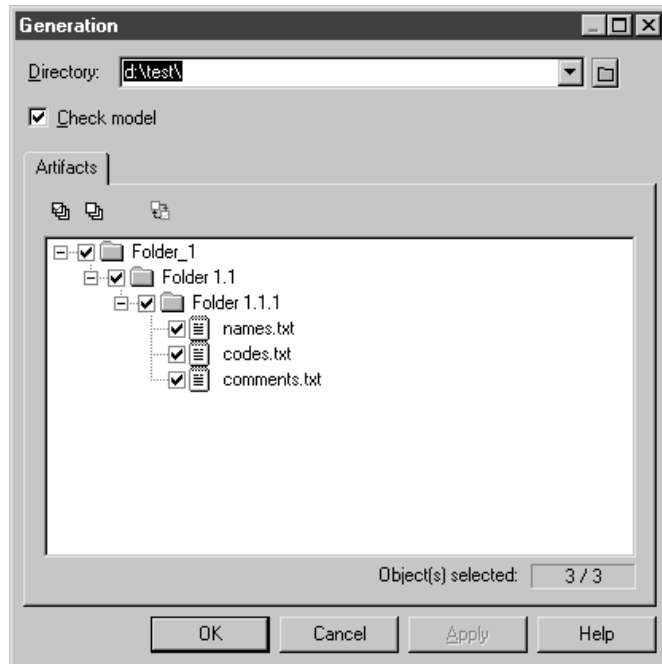
You can enforce dependencies in the Generation dialog box to make sure all prerequisites are selected for generation.

Using File Objects

A file object is a graphical representation of a Windows supported file (for example, a Java file, script SQL, or MS Word file). The file object can be external to the model or embedded in it.

You can use a file object for different purposes in PowerDesigner. For example you can:

♦ Attach a file object to a PowerDesigner object to enrich its description

♦ Attach a generated OOM class to a target Java file

Creating a file object

You can create a file object in any of the following ways:

♦ Use the File tool in the diagram Palette.

♦ 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 from the contextual menu.

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

For general information about creating objects, see "Creating an object" on page 213.

File object properties

You can modify an object's properties from its property sheet. To open a file object property sheet, double-click its diagram symbol or its Browser entry in the Files folder. The General tab contains the following properties:

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

Property	Description
Stereo-type	Sub-classification used to extend the semantics of the file object. You can create stereotypes in the Profile category of the resource file attached to the current model.
Location type	Specifies the nature of the file object. You can choose from the following options: ♦ Embedded – the file is stored within the model and is saved when you save the model. ♦ External – the file is stored in the Windows file system, and you must enter its path in the Location field. ♦ URL – the file is on the web and you must enter its URL in the Location field If you subsequently change the type: ♦ From external to embedded – you will be prompted to import the contents of the file into the model ♦ From embedded to external – you will be warned that the existing contents will be lost.
Location	[External and URL types only] Path or URL to the file.
Extension	Extension of the file object name that indicates the application to open when the file object is embedded
Generate	File object is automatically included among the objects generated from the model when you launch the inter-model generation process
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 box and is linked to the <internal> editor (see "Specifying text editors" in the Using the PowerDesigner Interface chapter), a Contents tab is displayed in the artifact property sheet. The Contents tab allows you to edit the artifact file directly in the internal text editor of PowerDesigner. For more information about the use of artifact files, see "Files" in the Building Physical Diagrams chapter of the *OOM User's Guide* .

Opening and editing file objects

When you work with an embedded object file, the content of the file is stored into a temporary file while the associated application opens. Each time you save your file in the associated application, its content is automatically copied from the temporary file into your PowerDesigner model. If you rename your file in the associated application, PowerDesigner will not be able to recognize your file and its content will not be copied into your model.

❖ **To open a file object**

1. Double-click the file object node in the Browser, its symbol in the diagram, or its entry in the List of Files.

   The application associated with the file object extension opens. If the file object extension has no association in Windows, a standard Open With dialog box opens that lets you select the application you want to open the file object.

A newly created file object has, by default, a .txt extension and opens in Notepad. You can change the extension in the file object property sheet and associate other editors with the extension.

❖ **To select an application program for a file object**

1. Right-click a file object and select ➤ Open With ➤ Choose Program from the contextual menu.

2. Browse to the directory that contains the program file you want, select it, and click Open.

   The file object automatically opens in the associated application program.

☞ For more information about text editors, see "Specifying a text editor" in the Using the PowerDesigner Interface chapter.

Attaching a file object to a PowerDesigner object

You can attach a file object to another object in any of the following ways:

♦ Right-click an object symbol in the diagram and select File ➤ Add File from the contextual menu.

♦ Use the Link/Extended Dependency tool in the diagram palette to draw a link from the object symbol to the file object symbol.

♦ Open the property sheet of the object, click the Extended Dependencies tab, and select the file object using the Add Objects tool..

The connection between the object and the file object takes the form of an extended dependency, and is visible:

♦ Graphically in the diagram

♦ On the Dependencies tab of the file object property sheet and on the Extended Dependencies tab of the dependent object property sheet

♦ Under the File menu item in the contextual menu of the dependent object symbol.

☞ For more information about extended dependencies, see "Displaying object dependencies" in the Using the PowerDesigner Interface chapter.

Modifying the file object display preferences

You can modify the following display preference for a file object using the Tools ➤ Display Preferences command:

Preference	Description
Location	Displays the location of the file object

Working with Extended Model Definitions

Extended model definitions provide means for customizing and extending PowerDesigner metaclasses, parameters and generation. You use extended model definitions to:

♦ Extend the PowerDesigner metamodel and develop the definition of metaclasses using profiles

♦ Complement the generation targets and commands of an object language

♦ Generate for an extended model definition

☞ For more information about extended model definitions, see chapter Extended Model Definitions Reference Guide in the *Advanced User documentation* .

What is an extended model definition?

An extended model definition is made up of a number of categories. A category can contain other categories, entries, and values. These entries are parameters recognizable by PowerDesigner.

The values for extended model definition categories and entries vary for each extended model definition. Some entries may not exist in the extended model definition file if they are not applicable to the particular extended model definition.

How to use extended model definitions?

You can create generic and specific extended model definitions.

♦ A **generic** extended model definition is a library of metamodel extensions and generation parameters saved in a file with the .XEM extension. This file is stored in a central area and can be referenced by models to guarantee data consistency and save time to the user

♦ A **specific** extended model definition is embedded into a model and develops object definitions and generation parameters in this particular model

Creating an extended model definition

When you create a new extended model definition you can choose to:

♦ Create a generic extended model definition file to reuse between models

♦ Create a specific extended model definition for the needs of a given model

The creation procedure differs according to the type of extended model definition you want to create.

Creating a generic extended model definition

You can create generic extended model definitions to share information between models of the same type.

❖ **To create a generic extended model definition**

1. Open a model.

2. Select Tools ➤ Resources ➤ Extended Model Definitions ➤ Model type.

   The List of Extended Model Definitions is displayed.

3. Click the New tool to display the New Extended Model Definition window.

New Extended Model Definition	✕

   Name: `New Extended Model Definition`

   Copy from: `<Default template>`

   OK   Cancel
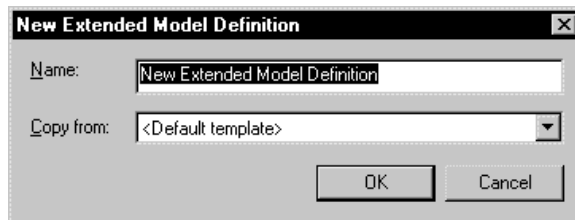
4. Type a name for the new extended model definition in the Name box.

5. <optional> Select a model template from the Copy From box. The list is empty if you create the first extended model definition.

6. Click OK.

   A standard Save As dialog box is displayed.

7. Type a name.

   > **Changing path**
   > If you change the default path, extended model definitions do not appear in the list of extended model definitions. If you want to save your extended model definitions in a specific folder, you have to define a specific named path in the General Options dialog box. For more information about named path, see section Defining named paths in chapter Using the PowerDesigner Interface.

8. Click Save.

   The Extended Model Definition Properties dialog box is displayed. The General tab displays all objects of the extended model definition together with an editor of extended model definitions on the right side of the tree view.
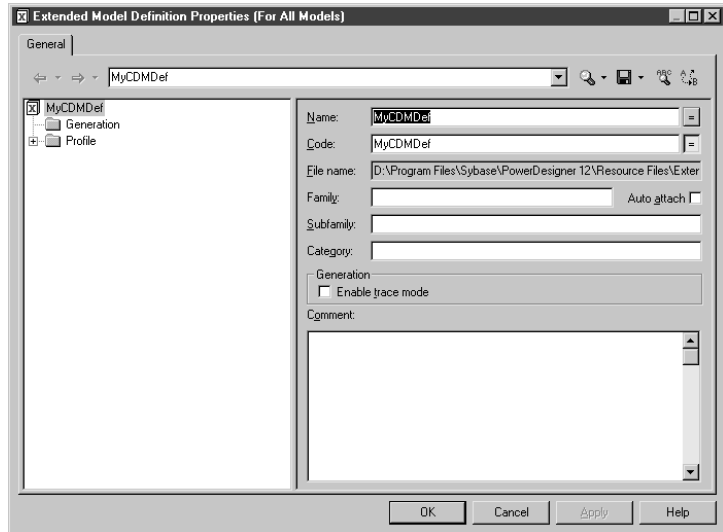
9. Define the extended model definition.

   ☞ For more information, see chapter Managing Profiles in the
   Advanced User documentation .

10. Click OK.

    The List of Extended Model Definitions displays the new extended model
    definition, whose name is followed by a star symbol to designate the state
    of an extended model definition to be saved. The first time you save your
    extended model definition, the symbol is no longer displayed.
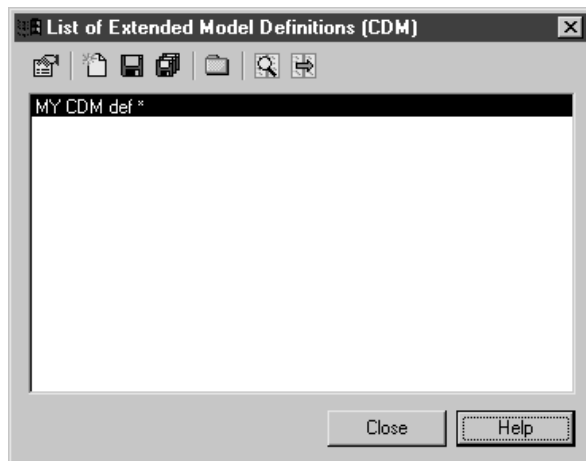
11. Click Close.

A confirmation box prompts you to save the changes made to the extended model definition, click Yes. The extended model definition is saved in a file with the .XEM extension.

> **Importing an extended model definition file**
> If you want to integrate a generic extended model definition into a model, you can import an extended model definition file.

☞ For more information about importing extended model definitions, see section "Selecting extended model definitions" on page 276.

Creating a specific extended model definition for a model

You can create an extended model definition for a specific model, in this case, it has the same type as the current model.

❖ **To create a specific extended model definition for a model**

1. Open a model.

2. Select Model ➤ Extended Model Definitions to display the List of Extended Model Definitions.

3. Click the Add a Row tool to create a new extended model definition.

4. Type a name for the extended model definition.

5. Click the Properties tool to display the property sheet of the extended model definition.

   You are asked to commit the creation of the extended model definition.

6. Click Yes.

   The Extended Model Definition Properties dialog box is displayed.
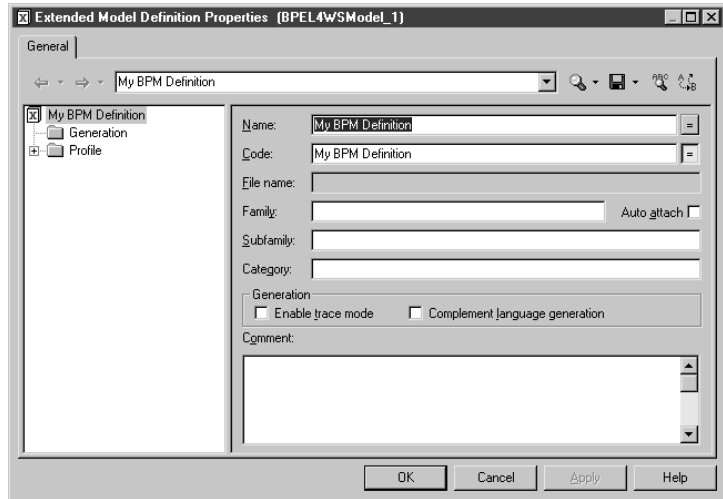
7. Define the extended model definition.

☞ For more information, see chapter Managing Profiles in the *Advanced User documentation* .

8. Click OK.

You return to the List of Extended Model Definitions.

9. Click OK.

Exporting an extended model definition
If you want to share an extended model definition created for a model with other models, you can export this extended model definition in order to reuse it with other models.

☞ For more information about exporting extended model definitions, see section "Exporting an extended model definition" on page 277.

Missing extended model definition

When you import a shared extended model definition in a model, you create a link between the model and the extended model definition file. If you rename, move or delete the extended model definition file associated with the model, you break this link.

The following dialog box is displayed when you open a model linked to a shared extended model definition file that is missing from the directory where it was created:
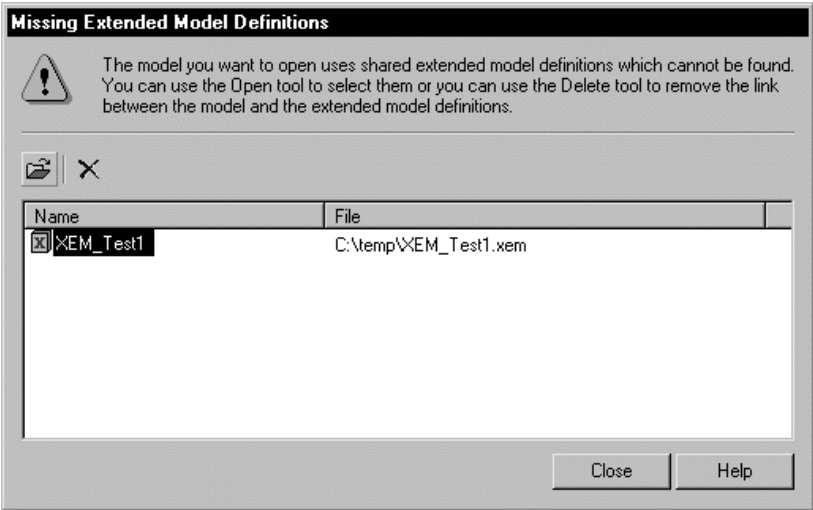
The Missing Extended Model Definitions dialog box displays the name and path of the missing(s) extended model definition(s).

You can use the following tools:

Tool	Description
🖼	Opens a standard Open dialog box in which you can browse and select the missing extended model definition. When you select the correct path and file, the extended model definition disappears from the list
✕	Deletes the link between the model and the shared extended model definition. The missing extended model definition disappears from the list

Selecting extended model definitions

Extended model definitions (.XEM files) provide means for customizing and extending PowerDesigner metaclasses, parameters and generation. Extended model definitions are typed like models in PowerDesigner. You create an extended model definition for a specific type of model and you cannot share these files between heterogeneous models.

When you create a new model, or when you reverse engineer into a new model, you can select one or several extended model definitions and attach them to the model from the New *Model* dialog box.

You can also import a generic extended model definition file into your current model. When you import an existing extended model definition, you

reuse the extended attributes and stereotypes defined in a library available on your machine.

Naming conventions

When you import an extended model definition and copy it into a model, the name and code of the extended model definition may be modified in order to respect the naming conventions of the Other Objects category in the Model Options dialog box.

You can choose one of the following options:

Option	Definition
Share	Current extended model definition constantly refers to the extended model definition stored in the Resource Files\Extended Model Definitions directory. Any changes made to the extended model definition are shared by all linked XEM
Copy	Current extended model definition is a unique copy of the extended model definition stored in the Resource Files\Extended Model Definitions directory. The current extended model definition is independent of the original one, so modifications made to the extended model definition in the Resource Files\Extended Model Definitions directory are not available to the copied XEM. This one is saved with the model and cannot be used without it

☞ For more information about extended model definitions, see chapter Extended Model Definitions Reference Guide, in the *Advanced User Documentation*.

Exporting an extended model definition

You can export an extended model definition created in a model if you wish to share this definition with other models. Export allows you to create a XEM file that will be stored into your extended model definition library directory. When you export an extended model definition, the specific extended model definition remains embedded in the model.

An extended model definition created in a model does not appear in the List of Extended Model Definitions. Whereas an exported extended model definition is displayed in the List of Extended Model Definitions.

❖ **To export an extended model definition into a model**

1. Select Model ➤ Extended model definitions to display the List of Extended Model Definitions.

2. Select an extended model definition in the list.

3. Click the Export an Extended Model Definition tool.

   A standard Save As dialog box is displayed.

4. Type a name and select a directory for the extended model definition.

5. Click Save.

   The extended model definition is saved in a library directory where it can be shared with other models.

CHAPTER 8

Managing Shortcuts

About this chapter   This chapter describes the use of shortcuts in PowerDesigner.

Contents

Creating Shortcuts

A **shortcut** is an object that represents and references a **target object.** There are two types of shortcuts:

♦ An **Internal** shortcut – is a shortcut to an object in a different package in the same model.

♦ An **External** shortcut – is a shortcut to an object in a different model. The external model may take the form of a library of reusable objects.

With shortcuts you can benefit from:

♦ **Reusability**: You can create libraries of reusable objects whose properties are inherited by multiple objects in different models. For example, the entity person 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.

♦ **Automatic updates**: When the target object changes, the updates cascades automatically to all the shortcuts. Note that both the referencing and target models must be open in the workspace for the changes to be propagated.

In the example below, the Person entity is created in one CDM, and a shortcut to it is created in a second CDM. The Person symbol in the second CDM carries a small arrow in its bottom left corner to indicate that it is a shortcut:
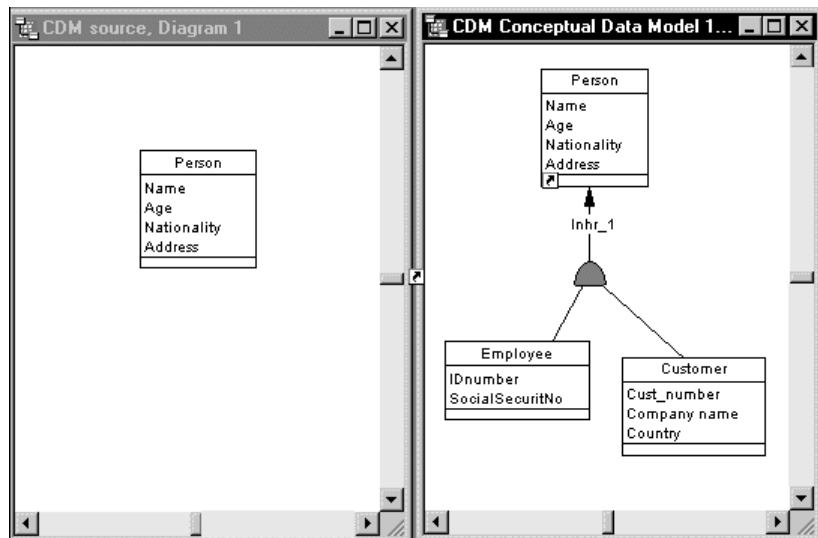
Objects that do not support shortcuts

You can create internal and external shortcuts on most of the object types that appear under a model diagram or package in the Browser. You can also create shortcuts of shortcuts.

The following objects do not support shortcuts at all or support only internal shortcuts:

Model	Internal and External Shortcuts not supported	External shortcuts not supported
CDM	—	—
OOM	Start, end, decision, synchronization, junction point, transition, state, interaction fragment, interaction reference	Message
BPM	Start, end, decision, synchronization, resource flow, service provider, service interface, operation	Flow, correlation, variable
XSM	[No internal shortcuts as there are no packages in an XSM]	Import, include, redefine, annotation
ILM	—	—
RQM	Traceability link, user allocation	—

However, it is still possible to create shortcuts in a model of a different type for traceability purposes.

You cannot create internal shortcuts to **global objects**, such as organization units, or business rules, because they always belong to the model and cannot be displaced into a sub-package. However, you can create external shortcuts to these objects.

Data item shortcuts

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 a shortcut

You can create a shortcut to an object through copy and paste or drag and drop.

❖ **To paste an object as a shortcut**

1. In the Browser or diagram window, select a target object in the target model or package and press CTRL+C or select Edit ➤ Copy.

2. Select the model or package where you want to create the shortcut in the Browser, or double-click a target diagram.

3. Select Edit ➤ Paste as Shortcut.

    The shortcut symbol is displayed in the model or package diagram.

Title			
<u>Title ISBN</u>	<u><pi></u>	<u>A10</u>	<M>
Title Text		LONG_TEXT	<M>
Title Type		SHORT_TEXT	<M>
Title Price		AMOUNT	
Title Notes		LONG_NOTES	
Title Publication Date		DATE	<M>
<u>Identifier_6</u> <pi>			

Paste as Shortcut from contextual menu
You can also right-click a model or package node in the Browser and select Edit ➤ Paste as Shortcut from the contextual menu.

❖ **To create a shortcut by drag and drop**

1. Select a target object in the Browser or diagram window.

2. Press CTRL + SHIFT while you drag the object to the desired model or package and release the mouse button.

    The shortcut is displayed in the active model or package and in the Browser under the appropriate node.

    ☞ For more information on dragging and dropping objects, including how to change the default behavior, see "Dragging and dropping objects" in the Managing Objects chapter.

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 data item outside the current namespace

♦ You cannot create links between two shortcuts, if the link implies a parent/child hierarchy (for example: reference link between two table shortcuts in the PDM)

♦ You cannot create a link between an entity and the shortcut of an inheritance. For more information, see "Linking shortcuts" on page 287.

When you create a shortcut, the following display rules apply to the shortcut symbol:

If referencing model or package...	PowerDesigner creates...
Does not contain the shortcut	Shortcut and shortcut symbol
Already contains shortcut without symbol	Shortcut symbol
Already contains shortcut with symbol	Shortcut synonym

Creating shortcuts from the List of Shortcuts

You can create shortcuts to target objects from another package in the current model, or from another model open in the workspace, in the List of Shortcuts.

❖ **To create a shortcut in the List of Shortcuts**

1. Select Model ➤ Shortcuts to open the List of Shortcuts.

2. Click the Add Shortcuts tool to open the Add Shortcuts selection box.
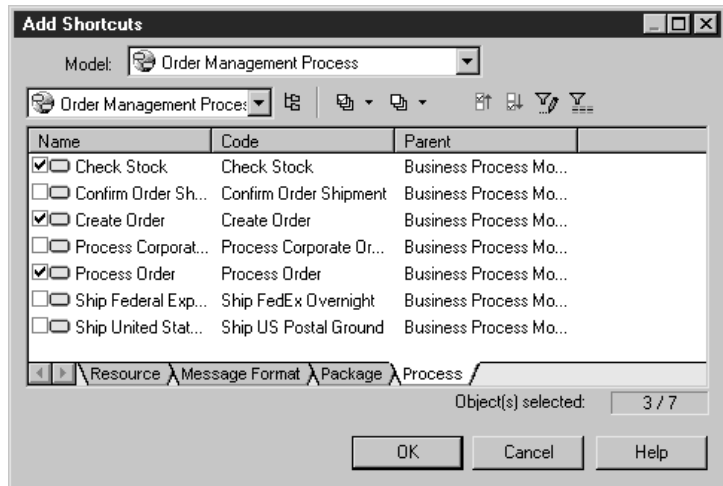
3. Select the appropriate model and package in the upper part of the dialog box, and select objects to add as shortcuts using the sub-tabs in the lower part of the dialog.

4. Click OK to return to the List of Shortcuts.

   The newly-selected objects are displayed in the List of Shortcuts.

☞ For more information, see "Adding an item from a selection list" in the "Using the PowerDesigner Interface" chapter.

Shortcut properties

To open the property sheet of a shortcut, double-click its Browser entry or diagram symbol. The General tab contains the following properties:

Property	Description
Target type	Type of model and object type
Name	Name of the target object. The Properties button lets you open the target object property sheet
Code	Code of the target object.
Target model	Name of the model to which the target object belongs. The Properties button lets you open the target model property sheet
Target package	Name and path of the package to which the target object belongs. The Properties button lets you open the target package or model property sheet.
Shortcut type	Type of the shortcut, external or internal
Status	Displays the status of the target model or object. The following states are possible: ♦ Closed – the target model is closed and the status of the target object cannot be determined ♦ Opened – the target model is open and the target object is present. ♦ Not Found – the target model or target object cannot be found. For more information, see "Changing the target object" on page 286.
Object1	[link shortcut only] Name of the source object from which the link is dragged
Object2	[link shortcut only] Name of the target object towards which the link is dragged
Generate	[external shortcut only] Shortcut is automatically included among the objects generated from the model when you launch the generation process

Property	Description
Generated as	[external shortcut only] Specifies how the shortcut will be treated during a model-to-model generation. You can choose between: ♦ Shortcut - the shortcut is generated as a shortcut and retains its link to the target model ♦ Object – the shortcut is generated as an independent object and loses its link to the target model
Change target object	Click this button to change the target object of the shortcut

A shortcut property sheet also includes the following tabs:

Property	Description
Dependencies	Objects with which the shortcut collaborates. See "Displaying the objects that are dependent on a shortcut" on page 290.
Version Info	Shortcut owner, modification and creation details

Modifying the target object properties

You can access and change the properties of the target object from the shortcut property sheet.

❖ **To change the properties of the target object**

1. Open the property sheet of the shortcut.
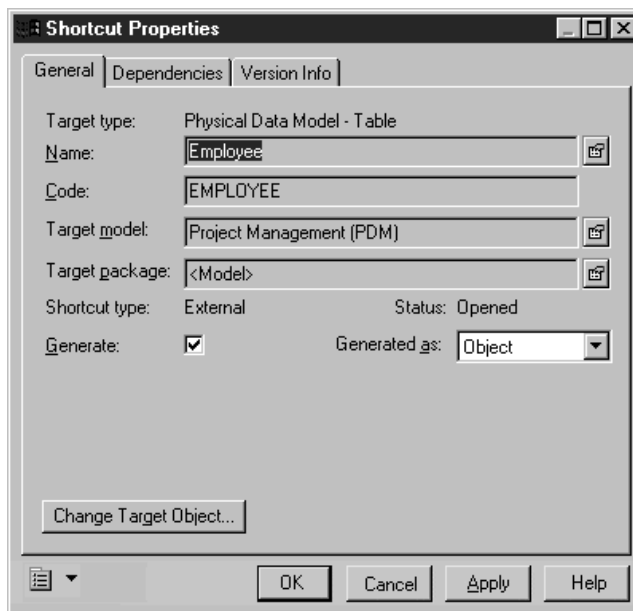
2. Click the Properties button to the right of the Name box to open the target object property sheet.

    If the target model is closed, you will be prompted to open it.

3. Modify the target object properties.

4. Click OK to return to the shortcut property sheet and OK again to return to the model diagram.

Modifying the target model and package properties
You can also access and modify the properties of the target model and package by clicking the properties tools to the right of these fields.

Changing the target object

You can change the target object that is referenced by the shortcut.

You may need to do this if the original target object or model has been deleted, and the shortcut status is, consequently, Not Found.

In this case, you can either delete the shortcut or select a new target object. The new target must be of the same object type, and cannot already be referenced by another shortcut in the same package.

❖ **To change the target object**

1. Open the shortcut property sheet.

2. Click the Change Target Object button in the bottom left corner of the shortcut property sheet to open an object selection dialog box.

3. Browse to the appropriate package and select a new target object.

4. Click OK to return to the shortcut property sheet.

   The new target object is displayed in the Name box and the shortcut status is changed to Opened.

5. Click OK to return to the model diagram.

Changing the target object from the List of Shortcuts
You can also change the target object of a shortcut by opening the List of Shortcuts, selecting a shortcut and clicking the Change Target Object tool.

Synchronizing shortcuts

When you create an external shortcut to 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.

When you make changes to a target object, the shortcut is synchronized automatically:

♦ If the referencing model is opened - synchronization occurs instantly.

♦ If the referencing model is closed - synchronization occurs when it is next opened.

Both the referencing and target models must be open for synchronization to occur.

Linking shortcuts

You can create a link between two shortcuts in the referencing model if the link does not imply a parent/child hierarchy. For example, you cannot create a reference link between two table shortcuts in the PDM, as the reference is an oriented link.

You can also keep the link existing between two target objects in the referencing model. Unlike shortcut symbols, the shortcut of a link does not show a particular icon that identifies it.

Creating the shortcut of a link

In the target model, when two target objects are linked, you can keep this link and create a shortcut for the link.

❖ **To create the shortcut of a link**

1. Select both target objects and their link using the SHIFT key for multi-selection.

2. Press CTRL + SHIFT while you drag the symbols and their link to the desired model or package and release the mouse button.

   The object shortcuts appear with their link.

Updating the display of a link between two shortcuts

When you create a link between two target objects in the target model, you can update the referencing diagram in order for it to display the new link.

❖ **To update the display of a link between two shortcuts**

1. Display the referencing model diagram window.

2. Select Tools ➤ Complete Links.

   The link is displayed between the shortcuts in the referencing model.

Displaying Shortcuts

Shortcuts are visible in both the Browser and the diagram window.

In the Browser, a shortcut is treated as a regular object, except that its symbol carries a small arrow in its bottom left corner. In the example below, the shortcut to Class C appears under the Class folder, sorted alphabetically among the other classes that are native to the model:

In a diagram, the symbol of a shortcut is identical to the symbol of a regular object except that it carries a small arrow in its bottom left corner.

If the target model is closed, only the name of the target model will be displayed. No other properties or sub-objects will be available:

Target model	Description	Default display Symbol
Opened	Shortcut symbols displays target object name, and target object attributes	
Closed	Simplified shortcut symbol with name, identical size but no attributes	

Modifying shortcut display preferences

You can modify the following display preferences for a shortcut using the Tools ➤ Display Preferences command:

Preference	Description
Icon	Displays the shortcut icon on shortcut symbols

Preference	Description
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 one of the following options: ♦ Full path – displays the full path to the package ♦ Last package only –displays the last package name only

Displaying the objects that are dependent on a shortcut

Shortcuts can collaborate with other objects in a variety of ways. The Dependencies tab in the shortcut property sheet lists the objects that are dependent on the shortcut.

This can be very useful to avoid deleting a shortcut whose deletion could seriously modify the design of your model.

❖ **To display the objects dependent on a shortcut**

1. Open the shortcut property sheet, and click the Dependencies tab.

   The Dependencies tab lists all the objects dependent on the shortcut in all the models open in the workspace.

   You can double-click an entry in the list to display the object's property sheet, or click the Impact Analysis button to determine the impact of deleting the shortcut.
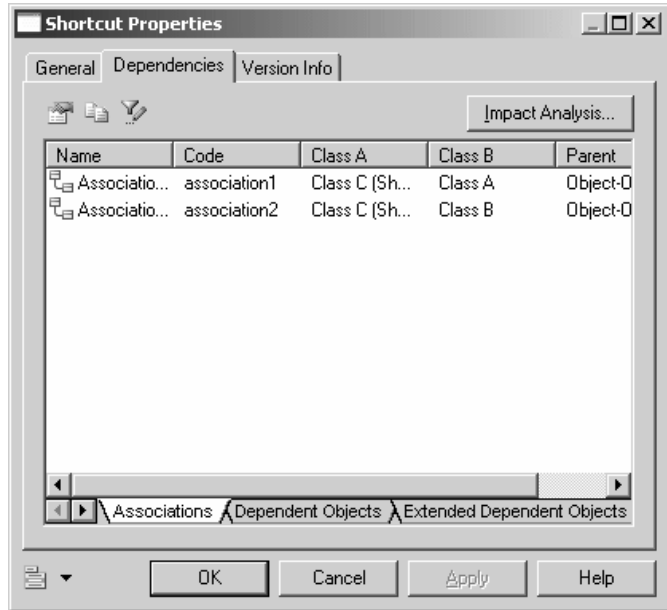
Displaying the shortcuts that reference a target object

The Dependencies tab in an object's property sheet lists all the shortcuts that reference the object

❖ **To display the shortcuts that reference a target object**

1. Open the target object property sheet and click the Dependencies tab.

2. Click the Shortcuts sub-tab.

   The Shortcuts sub-tab lists all the shortcuts that reference the target object in all the models open in the workspace.
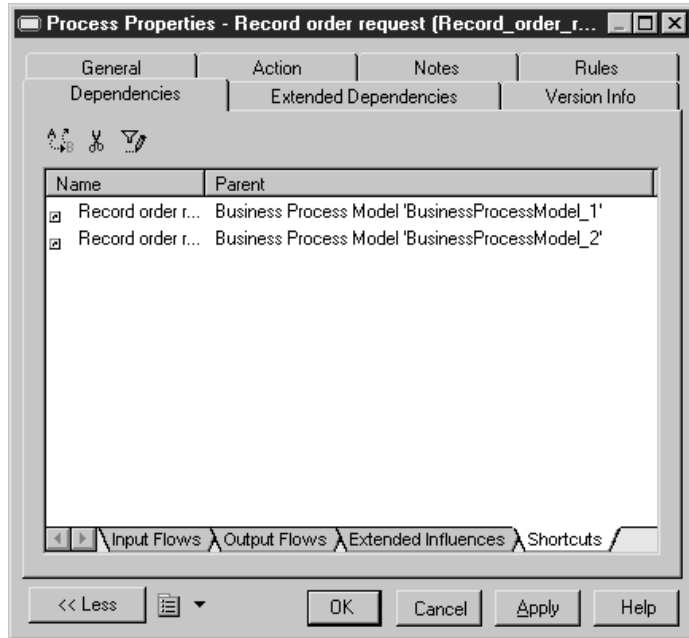
☞ For more information on how to visualize external shortcuts of a target object in closed model, see "Auditing repository activities" in the "Managing Repository Documents" chapter of the *Repository User's Guide* .

Displaying all the shortcuts for a particular type of object

You can display all the shortcuts for a particular type of object in the object lists.

❖ **To display all the shortcuts of an object type**

1. Select Model ➤ *Object* to open the List of *Objects* .

2. Click the Include Shortcuts tool to display the shortcuts of this object type within the list.
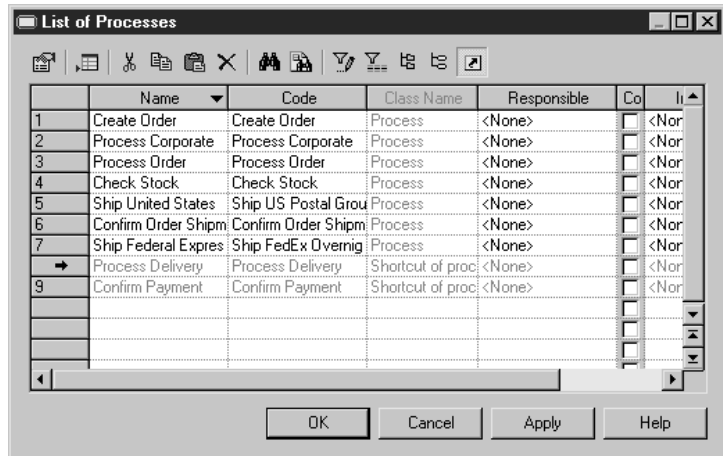
The shortcuts of this object type are displayed grayed out in the list. You cannot modify a shortcut directly in this list, but you can select one and click the Properties tool to display its property sheet.

Displaying shortcuts of all packages in an object type list
If both the Include Shortcuts and Include Sub-packages tools are enabled in the selected list, all the objects shortcuts of the current package and those of the sub-packages appear.

Displaying all the shortcuts in the model

You can display all the shortcuts in the model in the List of Shortcuts.

❖ **To display the List of Shortcuts**

1. Select Model ➤ Shortcuts to open the List of Shortcuts.

   All the shortcuts in the model are displayed grayed out in the list. You cannot modify a shortcut directly in this list, but you can select one and click the Properties tool to display its property sheet.
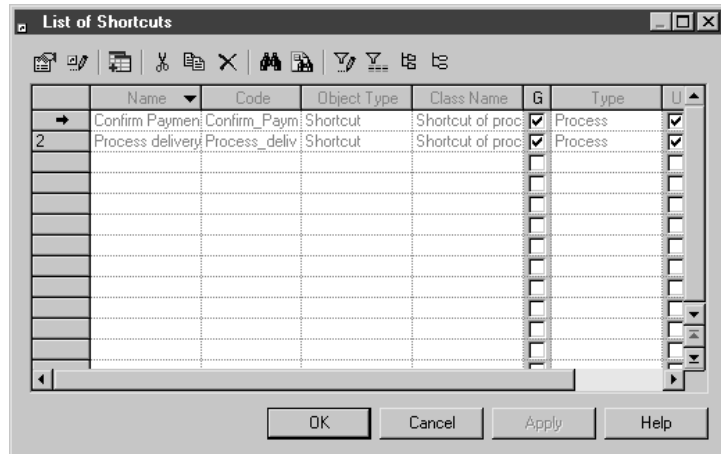
> **Displaying shortcuts of all packages in the List of Shortcuts**
> You can display the shortcuts of all packages by selecting the Include Sub-Packages tool.

U column   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.

Displaying all the target models referenced by the model

In addition to external shortcuts that you create yourself, PowerDesigner also creates shortcuts automatically when you perform certain tasks, such as linking requirements to design objects, mapping objects, or selecting a library during OOM reverse engineering.

❖ **To open the list of target models**

1. Select Model ➤ Target Models:
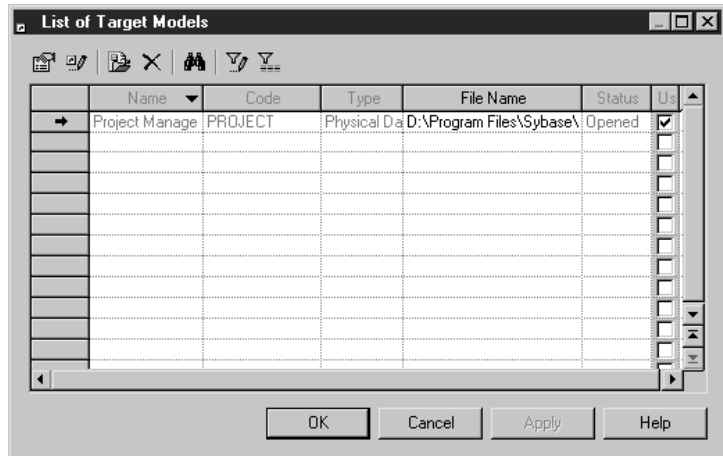
Opening a model from the list of target models
If a target model is presently closed, you can open it in the List of Target Models by selecting it and clicking the Open Model tool. You can also open a closed model and review its property sheet by clicking the Properties tool.

Changing a target model from the List of Target Models

You can change a target model that supplies shortcuts to your model from the List of Target Models.

You cannot undo a change of target model
You cannot use the Undo feature to cancel a Change Target Model action.

For example, you create a model, Project.pdm, which contains shortcuts to objects in a target model, Library.pdm. A copy of the target model, Library2.pdm, is made, in which different versions of the target objects are developed. You can switch between Library.pdm and Library2.pdm in the List of Target Models.

❖ **To change a target model**

1. Select Model ➤ Target Models to open the List of Target Models.

2. Select a target model in the list, and then click the Change Target Model tool to open a file chooser dialog.

3. Browse to the new target model, select it, and click OK.

A confirmation dialog box informs you that this change cannot be undone. If you click OK to proceed:

♦ And the new target model is open in the workspace - the shortcuts are updated to the new target objects. Otherwise they will be opened the next time you open the target model.

♦ And a shortcut cannot find its target object - a message box warns you. You can either delete the shortcut, or choose a new target object.

4. Click OK to close the list of target models.

Deleting a target model

If you delete a target model, you delete all its shortcuts.

❖ **To delete the shortcuts of a target model**

1. Select Model ➤ Target Models to open the List of Target Models.

2. Select a target model in the list and click the Delete tool.

   A message warns you that all the shortcuts related to the target model will be deleted during this operation.
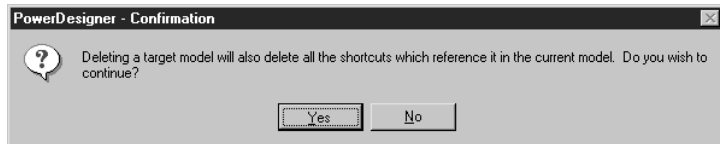
3. Click Yes to confirm the deletion of the target and of all its shortcuts.

Generating Shortcuts

You can generate shortcuts through the Target Models page located in the model type Generation Options dialog box. This page also allows you to generate replications (see "Generating replications" in the "Managing Object Replications" chapter).

When you generate a model from another models, shortcuts are, by default, generated in the new model as independent objects. You can choose to instead generate them as shortcuts and, thus, preserve their links to the target model.

The following example shows the generation of a PDM from an OOM:
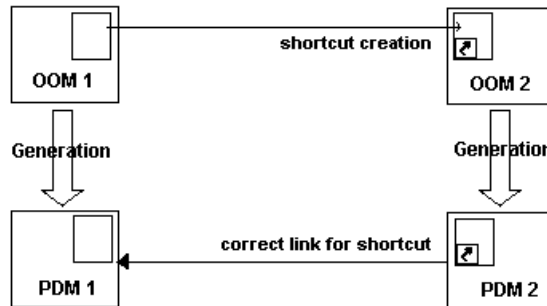
Here is the proper sequence of events for external shortcuts generation:

♦ OOM 1 is the target model of a shortcut in OOM 2

♦ OOM 1 is generated to PDM 1

♦ OOM 2 is prepared for generation to PDM 2 by associating appropriate properties and parameters in both the shortcut property sheet and the Target Models page:

If OOM 1 is still opened in the workspace   The Target Model column displays the original target model (OOM 1), and its path. The Generated Models column displays the last generated PDM the first time you generate the OOM into a PDM; the next time you generate the OOM into a PDM, the Generated Models column displays the last PDM selected. You can click the arrow in the Generated Models column to modify the PDM selection in order to allow the creation of a correctly linked shortcut.

If OOM 1 is closed in the workspace   The Target Model column displays the original target model (OOM 1), and its path. The Generated Models column displays <none>. When you click into the Generated Models

297

column, the original target model OOM 1 is automatically opened in the workspace in order to find the models generated from OOM 1. You can use the arrow to select PDM 1, the new target that will allow the creation of a correctly linked shortcut in PDM 2.

♦ The external shortcut in PDM 2 is correctly generated with a link to its target object in PDM 1

❖ **To generate shortcuts as shortcuts**

1. Before beginning the generation, verify that all the appropriate shortcuts have the following properties set:

   ♦ Generate is selected to include the shortcut in the generation process

   ♦ Generated As is set to Shortcut

Setting properties in the List of Shortcuts list
You can select and modify the properties of multiple shortcuts in the List of Shortcuts.

2. Select Tools ➤ Generate *model type* and click the Target Models tab, which contains a list of :

   ♦ Target Models - of the current model that contain at least one shortcut generated as shortcut

   ♦ Generated Models – from which you can select the model that will be used as a target for the generated shortcut

3. Verify the list of generated models for each shortcut you want to generate as a shortcut or make the appropriate changes.

4. Click OK to start generation.

Managing Object Replications

About this chapter   This chapter describes the use of object replications in PowerDesigner.

Contents

Creating Object Replications

While shortcuts allow you to reference objects in other models, they have some limitations:

♦ the complete definition of the target object can only be accessed if the target model is open.

♦ you cannot redefine locally any of the properties of the target object.

For example, in one model you may need a Client table with Name and detailed Address columns, while in another model you may need only the Name column.

In this case, instead of using a shortcut, you should **replicate** the object.

When you replicate an object, PowerDesigner creates a complete copy (or **replica**) of the object. The replica retains the name, code, type and Id of the **original object** and is automatically updated when the original is modified.

A replica looks exactly like other objects in the Browser and diagram, but its property sheet is, by default, uneditable, because all its properties are synchronized with the original:
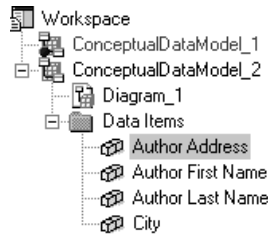
You can desynchronize any property that you want to change, while retaining synchronization of the others.

The link to and synchronization with the original object is maintained by a **replication**, which is not visible in the Browser or diagram, but whose properties are accessible via the List of Replications.

Replication Properties

General | Attributes | Collections | Sub-Replications

☐ Annotation   ☑ Precision
☑ Cannot modify   ☑ Server validation rule
☑ Client validation rule   ☑ Unit
☐ Code   ☑ Uppercase
☑ Comment
☑ Data Type
☑ Default value
☑ Description
☑ Extended Attributes
☑ Format
☑ High value   Desynchronized attributes
☑ Length
☑ List of values
☑ Low value
☑ Lowercase
☐ Name

OK   Cancel   Apply   Help
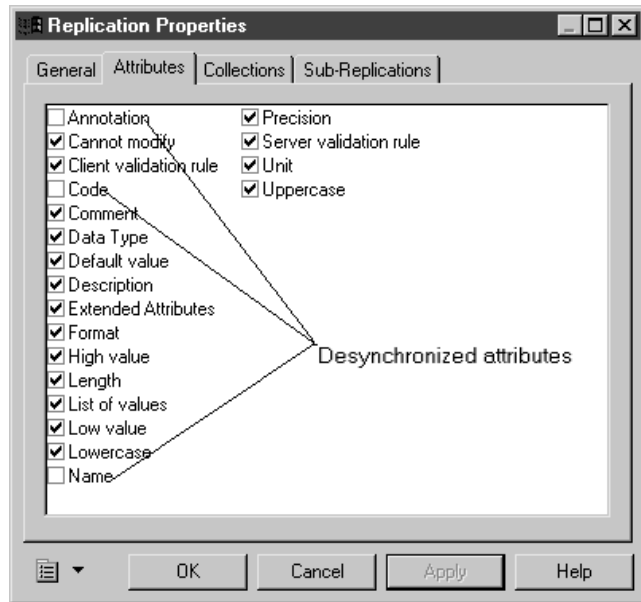
You use shortcuts when You want to reference an object in the same model or in different models or packages in order to share this object representation between models or packages. The shortcut is not a local copy of the target object and cannot be modified independently of its target object.

You use object replications when You want to have a local copy of an object that can also diverge from its original object.

Objects that support object replications?

You can create as many replicas as you want for most of the object types that appear under a model diagram or package in the Browser.

You can also create replicas of replicas.

You cannot replicate links, but you can create a link between two replicas in the referencing model.

The following table lists objects per module that support replications:

Module	Object
CDM Conceptual Diagram	Entity, Data Item
PDM Physical Diagram	Table, View, User, Role, Group, Abstract Data Type, Test Data Profile, Storage, Tablespace, Procedure, Trigger Template, Trigger Template Item, Join Index, Sequence, Database Package, Synonym

Module	Object
PDM Multidimensional Diagram	Cube, Dimension, Fact, Data Source
OOM Class Diagram	Class, Interface
OOM Use Case Diagram	Use Case, Actor
OOM Sequence Diagram	Object, Actor
OOM Activity Diagram	Object State, Organization Unit, Object, Activity
OOM Component Diagram	Component
OOM Object Diagram	Object
OOM Deployment Diagram	Component Instance, Node
OOM Collaboration Diagram	Object, Actor
OOM Statechart Diagram	Event, State
BPM Business Process Diagram	Organization Unit, Resource, Message Format, Data, Service Provider, Event, Data Transformation, Variable, Correlation, Process, Correlation Key
XSM XSD Diagram	Attribute Group, Attribute, Element, Simple Type, Complex Type, Group, Notation, Import, Include, Redefine
XSM DTD Diagram	Entity, Attribute Group, Attribute, Element, Group, Notation
RQM	Term, User, Group, Replication
ILM Information Liquidity Diagram	—
All modules	File, Business Rule, Domain, Data Source

Creating a replica

You can creating replicas using:

♦ the Replicate Objects dialog box

♦ drag and drop while holding ALT + SHIFT

Creating a replica from the Replicate Objects dialog box

The Replicate Objects dialog box allows you to select one or more objects to replicate from any model or package open in the workspace.

❖ **To create a replica from the Replicate Objects dialog box**

1. Select Edit ➤ Replicate Objects (or right-click the diagram background and select Edit ➤ Replicate Objects from the contextual menu) to open the Replicate Objects dialog box.

2. Select a model and, optionally, a package from which to select objects to replicate. All the available objects of the selected model and package are displayed.

   Include Sub-Packages
   If you want to display all objects in the model and all objects in packages and sub-packages, click the Include Sub-Packages tool in the list toolbar.

3. Select the objects you want to replicate from the various sub-tabs and click OK.

   Replicating sub-objects
   If you select an object that owns sub-objects (a table that contains columns, for example, or a class that contains attributes or operations), its sub-objects are also replicated. If you want to directly replicate a sub-object, you must use drag and drop.
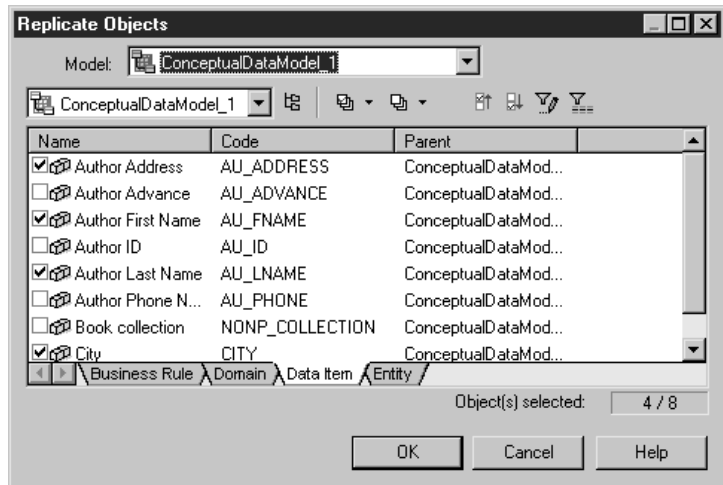
The replicas appear in the active model and in the Browser under the appropriate node.
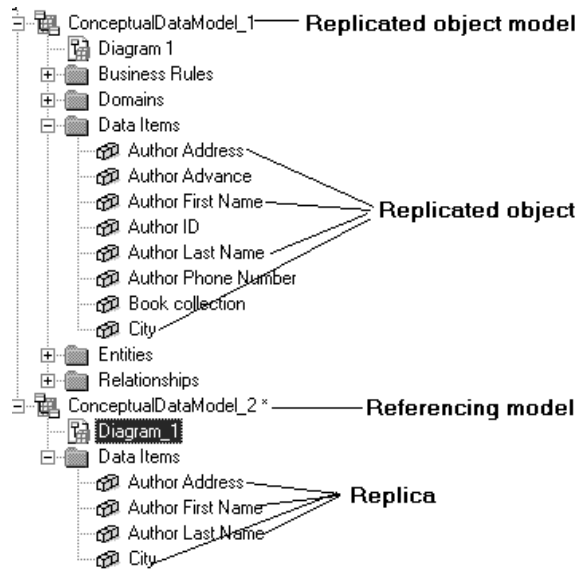
Creating a replica using drag and drop

You can use drag and drop while holding ALT+SHIFT to create replicas in the Browser or diagram, or from one to the other. A rounded arrow within a circle is displayed under the cursor when you are about to create the replica.

You can replicate sub-objects by drag and drop in the Browser while pressing the ALT + SHIFT keys combination. It can be useful to replicate a sub-object independently of its parent object (for example a column without the table that owns it) to create a reference library, in which you can store columns, attributes or operations that you regularly use.

The following example illustrates a column (Column_1) in Table_1 that is replicated in Table_2:
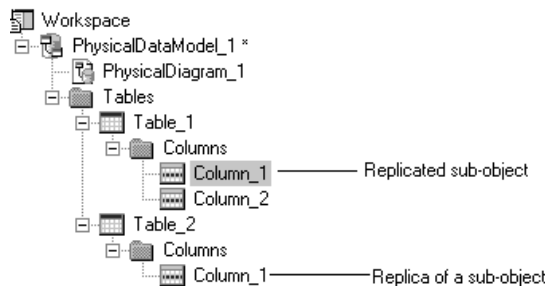
You can define the default behavior of drag and drop to directly create replicas (see "Dragging and dropping objects" in the "Using the

PowerDesigner Interface" chapter).

❖ **To create a replica by drag and drop**

1. Select a target object in the Browser.

2. Press ALT+SHIFT while you drag the object to the desired model or package and release the mouse button.

   The replica is displayed in the active model or package and in the Browser under the appropriate node.

Replicate Here contextual menu
You can also create a replica by right-clicking and dragging the target object and select the Replicate Here menu item from the contextual menu.

Replicas and namespace   Name uniqueness is automatically checked when you replicate an object, so that replicas are renamed in the referencing model when objects with the same name already exist in the model.

☞ For more information on the definition of the namespace, see "Managing the namespace in models" in the "Managing Models" chapter.

The following example illustrates the replication of Entity_1 in a model that already contains an entity named Entity_1. The Entity_1 replica is automatically renamed to Entity_2 in the referencing model:
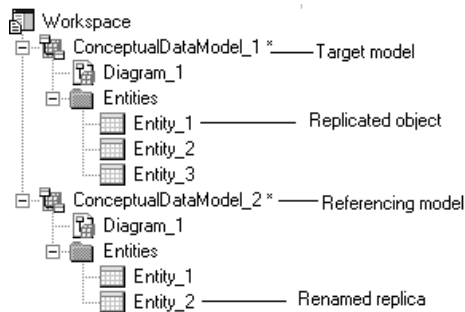
Replica and replication properties

Replicas have the property sheets appropriate to their object type. For example, a replica of a class has a standard class property sheet. However, by default, the property sheet of a replica is read-only because all its properties are synchronized with the original object.

In order to make editable any of the properties of a replica, you must open the properties of the replication responsible for its link with the original object.

❖ **To open the property sheet of a replication**

1. Open the property sheet of a replica and click the Version Info tab.

2. Click the Replication Properties button in the Replicated From groupbox.

The definition of a replication object includes the following general properties:

Property	Description
Original Object Model	Model of the original object. The Properties button lets you open the property sheet of the model containing the original object.
	The List of Target Models shows models containing shortcuts or replicas in the current session. It also allows you to get a model full name to distinguish from another. For more information on how to use the List of the Target Model, see "Using the list of target models" in the "Managing shortcuts" chapter.
Original Object Full Name	Full path describing the location of the original object. The Properties button lets you open the property sheet of the original object
Original Object Type	Type of the original object
Original Object Status	Displays the status of the original object. The following states are possible: ♦ Closed – the target model is closed and the status of the original object cannot be determined ♦ Opened – the target model is open and the original object is present. ♦ Not Found – the target model or original object cannot be found. For more information, see "Deleting replicas, replications, and original objects" on page 313.
Replica Object Full Name	Full path describing the location of the replica object. The Properties button lets you open the property sheet of the replica object

Property	Description
Generate	Replication is automatically included among the objects generated from the model when you launch the inter-model generation process. For more information, see the ""Generating Replications" on page 318" section.

A replication definition also includes the following properties:

Property	Description
Attributes	List of replicated attributes
Collections	List of replicated collections
Sub-Replications	List of replicated sub-objects (for example a column of a replicated table)

> **Opening the property sheet of a replication from the List of Replications**
> Select Model ➤ Replications to open the List of Replications, select a replication in the list, and then click the Properties tool

Modifying the original object properties

You can access and change the properties of the original object from the replication property sheet.

❖ **To change the properties of the original object**

1. Open the property sheet of the replica, and then click the Version Info tab.

2. Click the Replication Properties button in the Replicated From groupbox to open the replication property sheet:
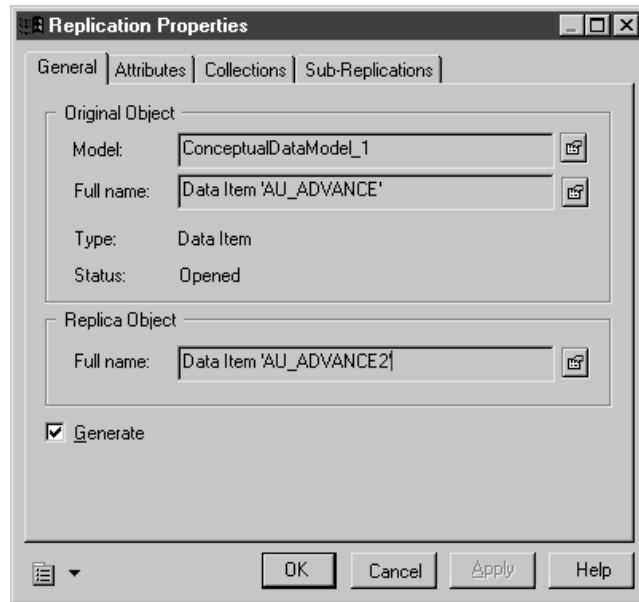
3. Click the Properties button to the right of the Full name field to open the original object property sheet.

4. Modify the original object properties.

5. Click OK to return to the replication property sheet and OK again to return to the model diagram.

Synchronizing replicas

When you create a replica to 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.

When you make changes to an original object, the replica is synchronized automatically:

♦ If the referencing model is opened - synchronization occurs instantly.

♦ If the referencing model is closed - synchronization occurs when it is next opened.

Both the referencing and target models must be open for synchronization to occur.

Desynchronizing replica properties

By default, when you replicate an object, all of its properties are also replicated, and synchronized to the original, and are uneditable in the replica property sheet.

You can desynchronize any of these properties in the replication property sheet, and then change their values in the replica.

❖ **To desynchronize replica properties**

1. Open the property sheet of a replica, and then click the Version Info tab.

2. Click the Replication Properties button in the Replicated From groupbox to open the replication property sheet:
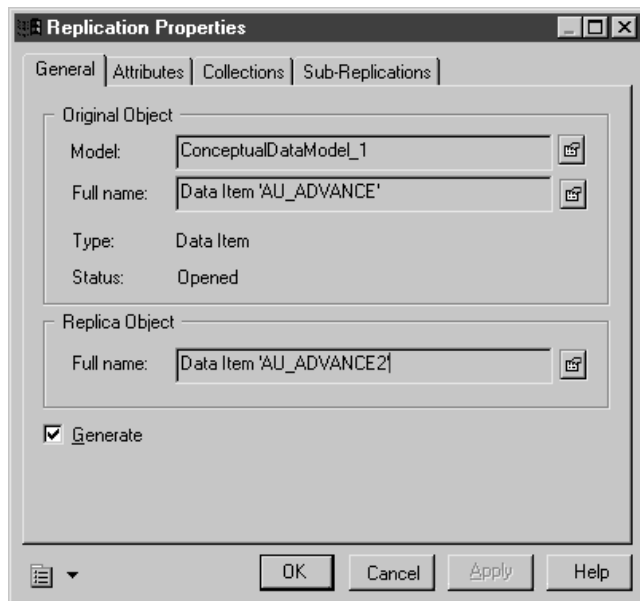
3. Click one of the following tabs to display the type of property you want to desynchronize:

♦ Attributes – properties that contain a single value

♦ Collections – properties that can contain multiple values

♦ Sub-replications – sub-objects, that have their own property sheets

4. Clear the check boxes next to the items that you want to desynchronize:
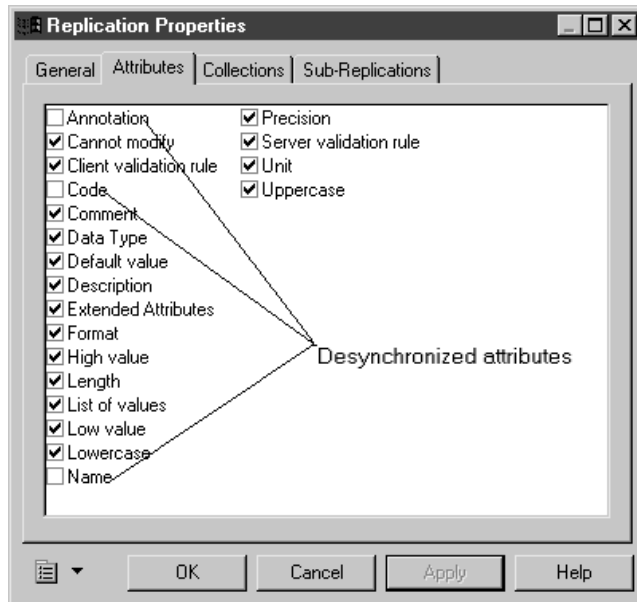
5. [for sub-replications] Select the sub-object in the list on the Sub-replications tab and click the properties tool to open its sub-replication property sheet. Then clear the relevant check boxes to desynchronize its properties.

6. Click OK to return to the replica property sheet.

   The desynchronized properties are now editable in the property sheet, and will no longer be synchronized with the original.

Moving replicas, replications, and original objects

If you move a replica to a different package or model, the replication that links it to the original object is also moved.

If you move an original object, any replications that link replicas to it will be automatically updated.

You cannot move a replication separately from the replica that it serves.

☞ For more information on moving objects, see "Moving objects from package to package" in the "Managing Objects" chapter.

Deleting replicas, replications, and original objects

If you delete a replica, the replication that linked it to the original object is also deleted.

If you delete an original object, any replicas linked to it will be deleted too, unless they have one or more property desynchronized, in which case they will be retained and become normal objects.

If you delete a replication, the link between the replica and its original is broken and the replica becomes a normal, object. No further synchronization is possible.

❖ **To delete a replication**

1. Open the property sheet of a replica, click the Version Info tab, and then click the Delete Replication button in the Replicated From groupbox.

2. Select Model ➤ Replications to open the List of Replications, select a replication in the list and click the Delete tool

Displaying Replicas and Replications

Replicas look identical to normal objects, but all or part of their property sheets will be grayed out and uneditable.

Replications, which maintain the link between the replica and the original object have no browser entry or diagram symbol, but their property sheets can be accessed from the property sheet of the replica or from the List of Replications.

Displaying the replications linked to an original object

The Dependencies tab in an object's property sheet lists all the replications that reference the object

❖ **To display the replications that reference an original object**

1. Open the original object property sheet and click the Dependencies tab.

2. Click the Replications sub-tab.

   The Replications sub-tab lists all the replications that reference the original object in all the models open in the workspace.
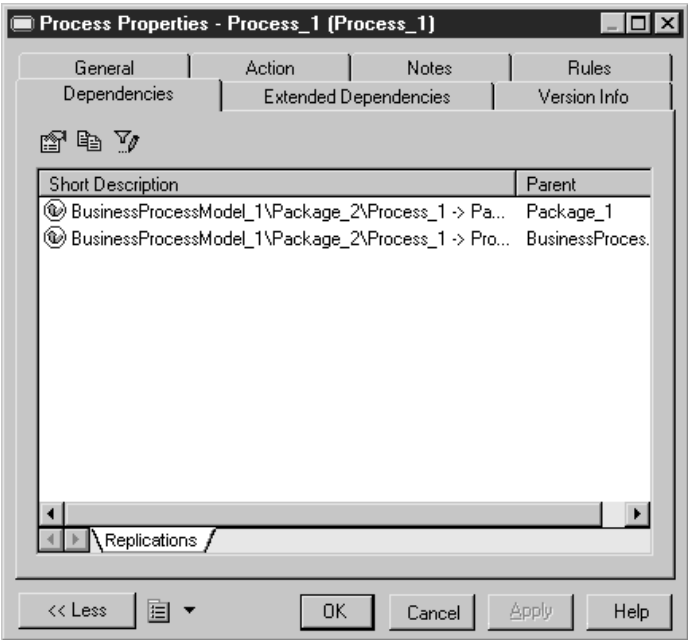
Displaying all the replicas for a particular type of object

You can display all the replicas for a particular type of object in the object lists.

❖ **To display all the replicas of an object type**

1. Select Model ➤ *Object* to open the List of *Objects* .

2. Click the Customize Columns and Filter tool, select Replica in the list of filter options, and then click OK.

   When the R[eplica] column is selected, this indicates that the object is a replica. Any property that is synchronized (and thus not editable) is grayed in the list.
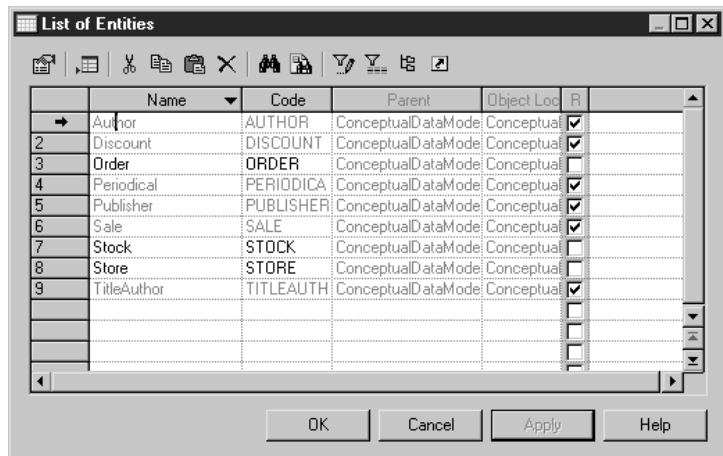
You can select a replica, and then click the Properties tool in order to open its property sheet, and desynchronize any of its properties to make them editable. When a property has been desynchronized, it is no longer greed in the object list, and can be edited.

Displaying replicas of all packages in an object type list
If the Include Sub-packages tools is enabled in the selected list, all the replicas of the current package and those of the sub-packages appear.

Displaying all the replications in the model

You can display all the replications in the model in the List of Replications.

Creating new replications
You cannot create new replications from the List of Replications. You must
use drag and drop or select Edit ➤ Replicate Objects

❖ **To display the List of Replications**

1. Select Model ➤ Replications to open the List of Replications.

    All the replications in the model are displayed grayed out in the list. You
    cannot modify a replication directly in this list, but you can select one and
    click the Properties tool to display its property sheet.
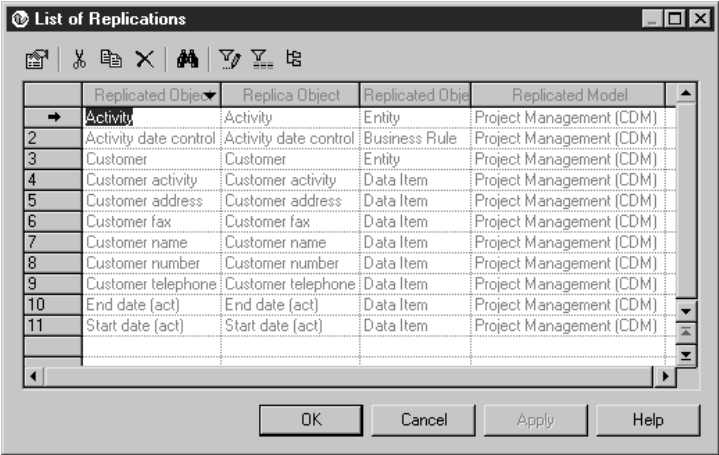
Displaying replications of all packages in the List of Replications
You can display the replications of all packages by clicking the Include
Sub-Packages tool.

Comparing and merging replicas

You can compare and merge the properties of a replica with those of the
original object.

When merging models containing replications, you must merge both replicas
and their associated replications in order to merge a complete replication.
Otherwise they will be merged as ordinary objects and not as replicas.
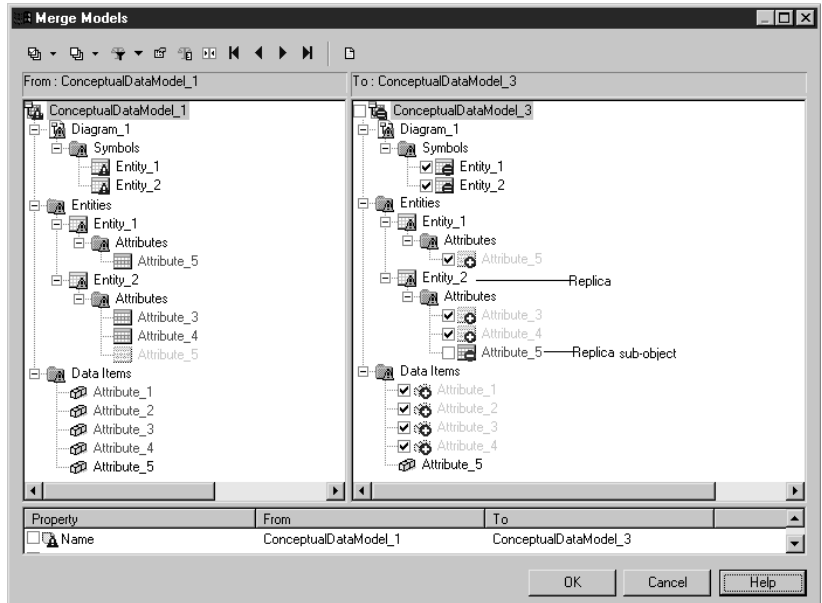
☞ For more information on comparison and merging see the "Comparing and Merging Models" chapter.

Generating Replications

You can generate a replica in another type of model and preserve the link with its original object through generation. You will then be able to continue the synchronization in the generated model.

This requires first generating all original object models to the destination model type so that these new objects can be referenced as the original objects for the generated replicas.

You specify original object models to be used during generation through the Target Models page located in the model type Generation Options dialog box

Before you generate a replication, you must:

♦ Select the Generate check box in the replication property sheet otherwise the replica will be generated as an ordinary object and not as a replica

♦ Select generated models in the Target Model page located in the *model type* Generation Options dialog box in order to retrieve the corresponding original object of each replica in the generated model

The Target Model page also allows you to select generated model for shortcuts.

☞ For more information on the Target Model page, see "Generating Shortcuts" in the "Managing Shortcuts" chapter.

Example of an inter-model generation with replications

The following example shows the generation of a PDM from an OOM:
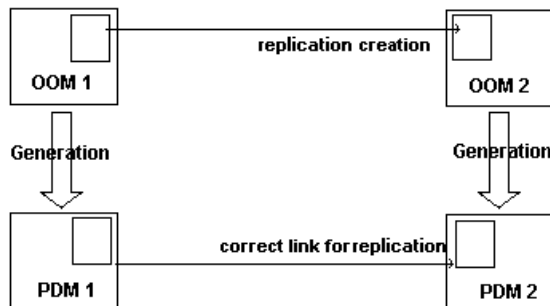
Here is the proper sequence of events for replication generation:

♦ OOM 1 is the target model of a replica in OOM 2

♦ OOM 1 is generated to PDM 1

♦ OOM 2 is the referencing model that contains the replica

318

♦ OOM 2 is generated to PDM 2 while preserving the link between the replica in PDM2 and the original object in PDM 1

Selecting a type of generation for a replication

You can choose to generate a replication in another type of model and preserve the link with its original object through generation. The list of replications allows you to perform a multiple selection.

☞ For more information on the generation of replications, see the ""Generating Replications" on page 318" section.

❖ **To select a type of generation for a replication**

1. Select Model ➤ Replications to open the list of replications.

2. Select a replication in the list.

3. Click the Customize Columns and Filter tool in the list toolbar, select the Generate check box from the list of filter options that is displayed, and click OK.

   You return to the list of replications.

4. Click the Generate column for the replication you want to generate during inter-model generation.

5. Click OK.

   The replication will be automatically included among the objects generated from the model when you will launch the generation process.

CHAPTER 10

Comparing and Merging Models

About this chapter

This chapter describes the comparison and merge features of models in PowerDesigner.

Contents

Topic:	page

Comparing Models

You can compare the content of two PowerDesigner models or two resource files of the same type in order to:

♦ Follow up evolutions in models or resources manipulated by different development teams

♦ Evaluate the differences that exist between the models or resources before merging them

The comparison window displays the objects contained within the models or resource files in a tree format, and highlights the differences between them.

For more information about the comparison of resource files, see "Comparing and merging resource files" in "The Resource Editor" chapter of the *Advanced User's Guide* .

Comparing two models

You can launch a comparison of your model with another model of the same type at any time.

❖ **To compare two models**

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.
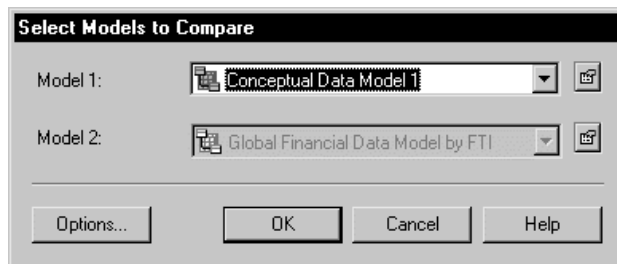
3. [optional] Click the Options button to open the Comparison Options window and specify which objects and properties you want to include in the comparison. For more information, see "Comparison options" on page 323.

4. Click OK.

The Compare Models window opens. See .

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:
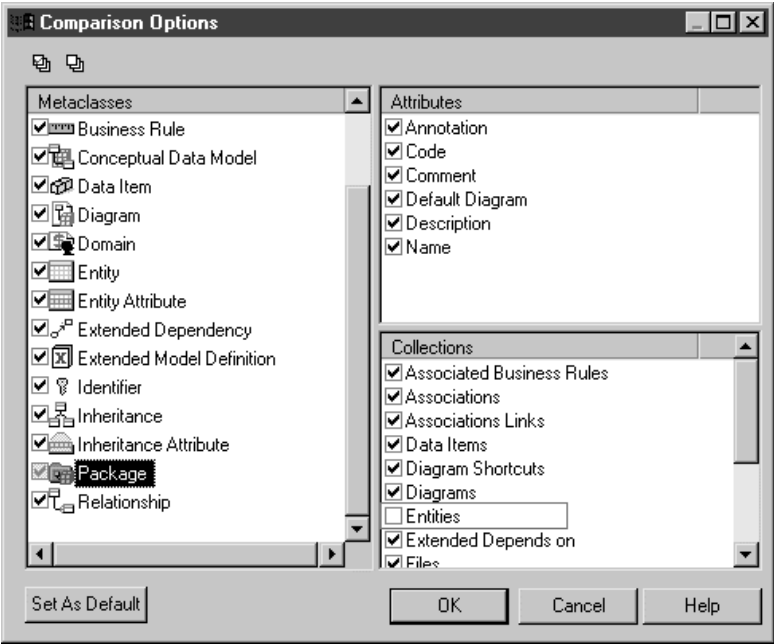
Default comparison options

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 metaclass check box	Select or deselect all its attributes and collections at once
Right-click a meta-class attribute or collection	Select or deselect the attribute or collection for all metaclasses at once from the contextual menu
Click the Select All or Deselect 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:
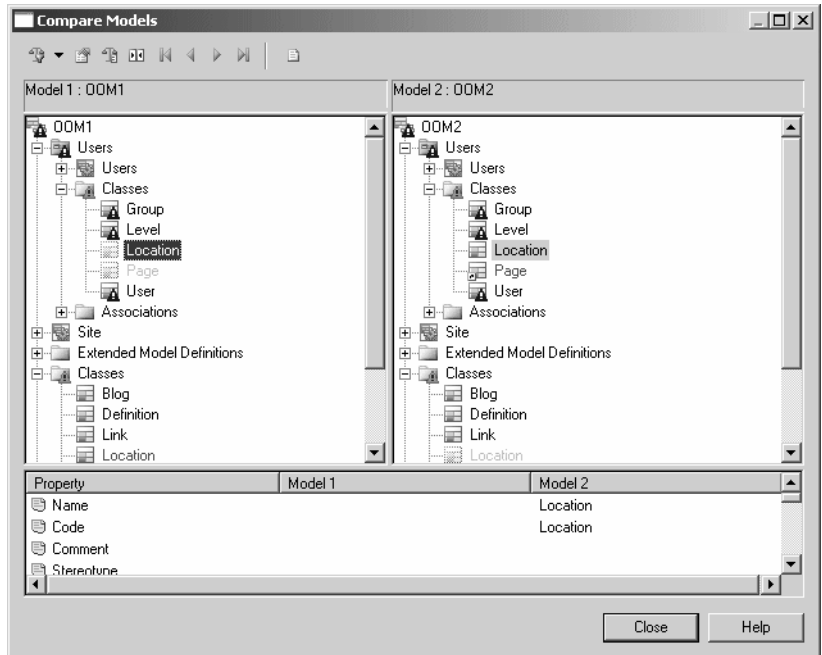
The window is divided in two parts:

♦ 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:

• Property - displays the name of the selected property

• Model 1 - displays the properties of the object belonging to the model on the left of the dialog box

- Model 2 - displays the properties of the object belonging to the model on the right of the dialog box

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.

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

> **Opening the property sheet of an object in read-only mode**
> You can open the property sheet of an object in read-only mode by double-clicking the object node from the object comparison part. 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
▶	Goes to the next difference in the tree view
▶▶	Goes to the last difference in the tree view

> **Hot keys for expanding and collapsing all nodes at once**
> 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 (-).

Analyzing differences in the selected object's properties

Differences between objects are displayed with more details in the properties comparison part of the comparison window.

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
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

Filtering the changes to compare

You can filter the objects that are shown in the comparison window:

❖ **To define a comparison filter**

1. In the Compare Models window, click the arrow next to the Change Filter tool to display the list of possible filters.
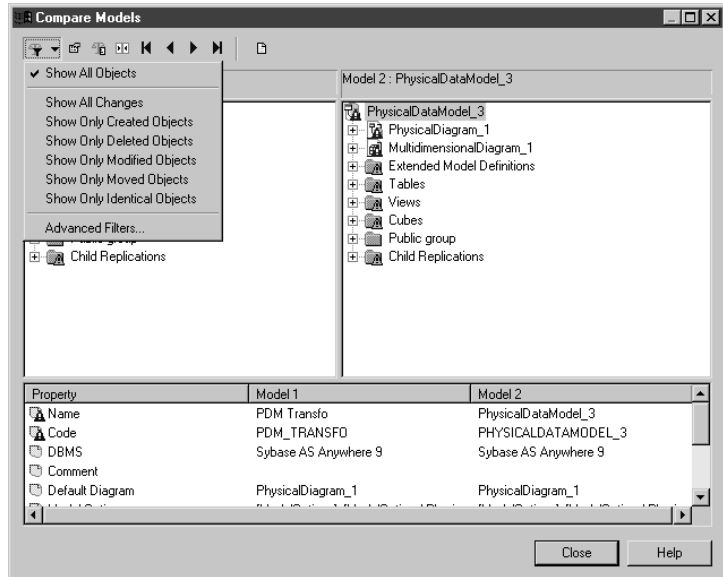
2. Select a filter from the list. You can filter by:

   ◆ All objects

   ◆ All changes

   ◆ Only created objects

   ◆ Only deleted objects

   ◆ Only modified objects

   ◆ Only moved objects

   ◆ Only identical objects

   ◆ Advanced Filters – Opens the Comparison Filters dialog box, which
     allows you to combine several filters.

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).

Once you have selected all the appropriate criteria, click OK.
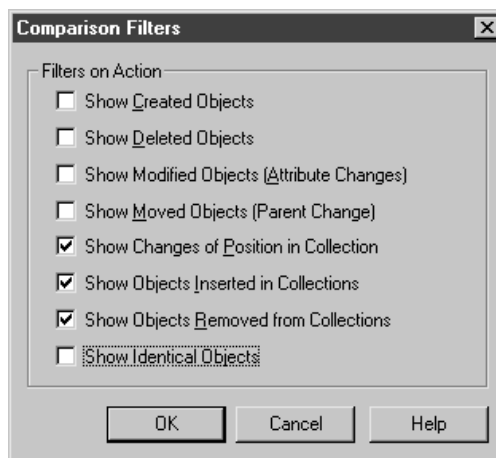
The filter is applied to the Compare Models window, and only objects complying with the filter criteria are displayed.

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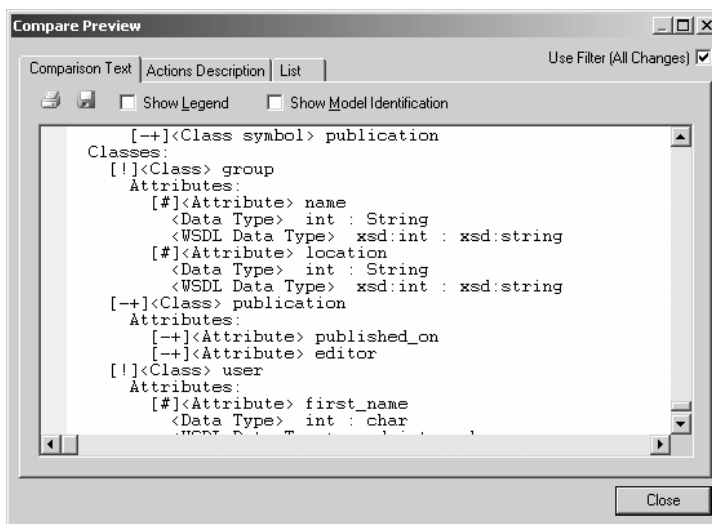
♦ **Actions Description** – a list of the actions that would need to be taken to render the models the same. This will often be the most readable format as it represents each change in the form of a sentence:
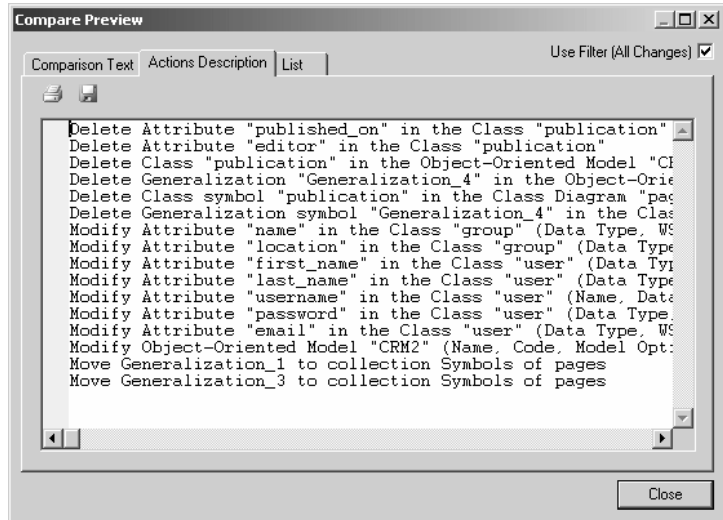
♦ **List** – a columnar representation of the required actions. This format is the most easily manipulated. You can sort by any of the columns, and export the list in a variety of formats to allow for various kinds of transformation:
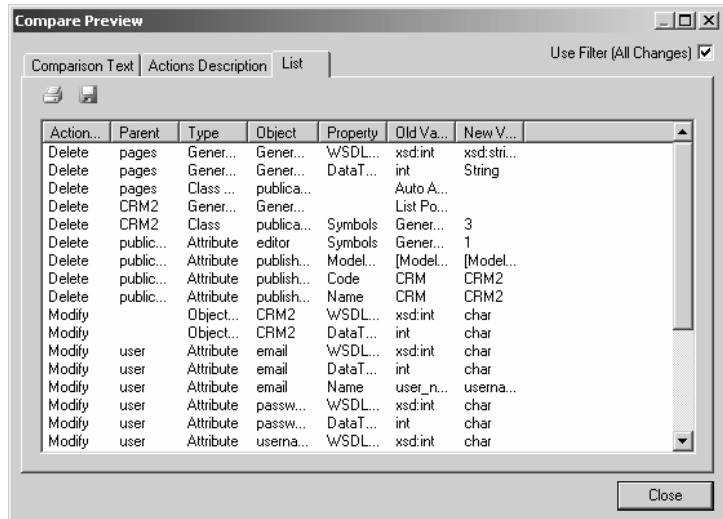
The following tools are available on each of the tabs:

Tool	Description
Use Filter	Filters the changes using the filter specified in the main window (see "Filtering the changes to compare" on page 328). 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.

The merge window displays the objects contained within the models or resource files in a tree format, and highlights the differences between them.

For more information about the merging of resource files, see "Comparing and merging resource files" in "The Resource Editor" chapter of the *Advanced User's Guide* .

Merging two models

You can launch a merge of your model with another model of the same type at any time.

❖ **To merge two models**

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 compare in the From list. Only models of the same type in the workspace are available for merging.
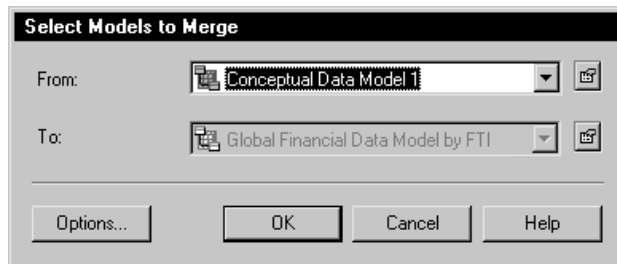
   DBMS and Language
   If you want to merge physical data models, you have to select two models with the same DBMS. If you want to merge object oriented models, you have to select two models with the same language.

   ---

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. For more information, see "Comparison options" on page 323.

4. Click OK.

The Merge Models window opens.

Analyzing differences in the Merge Models window

The Merge Models window allows you to compare the contents of two models:
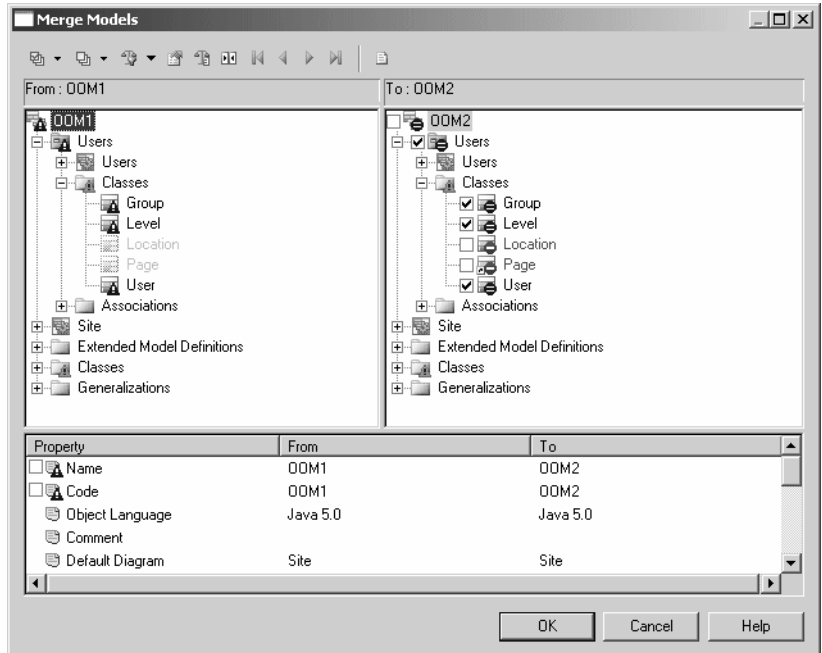
The window is divided in two parts:

♦ 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. Any eventual merge will be effected on the "To" model in the right-hand pane.

♦ 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:

• Property - displays the name of the selected property

• From - displays the properties of the object belonging to the model on the left of the dialog box

• To - displays the properties of the object belonging to the model on the right of the dialog box

Merge is performed from left to right, the model to be merged is compared to the model on the left pane, differences are highlighted in this model, and merge actions are applied in the model on the right pane.

Navigating among differences

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

Contextual menu
You can right-click a moved object and select Next or Previous occurrence to go to its next or previous occurrence.

Hot keys for expanding and collapsing all nodes at once
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 (-).

Understanding merge action icons

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
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

Moving an object from a package to another

When you merge models in one of which you have moved an object from a package to another and modified its properties, the object is displayed in both packages of the model to be merged, in the object comparison part.

In the following example, Entt_5 in Package 2 was modified and moved into Package 1:
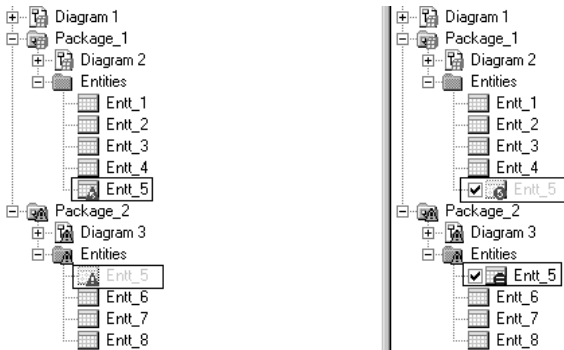
Icon	Description
☑🖹	One occurrence remains in the package on the left pane and the opposite icon is displayed at the bottom right corner of the object to indicate that the object properties were modified
☑🔟	The second occurrence is displayed grayed out in the package to be merged and the opposite icon is displayed at the bottom right corner of the object to indicate that the object was moved but does not yet exist in that location

You are invited to synchronize the models on the right and left panes by selecting both check boxes.

Check box synchronization
When you select or clear a merge action check box for a moved object, the check box of the other occurrences are automatically selected or cleared.

When you move an object that is replaced with a shortcut, the merge action suggested is the move with no indication of shortcut replacement. The unique indication of the object replacement with a shortcut is displayed with the following icon in the model on the left pane of the Merge Models dialog box:

In the following example, Entt_2 in the main diagram was moved into Package 2:
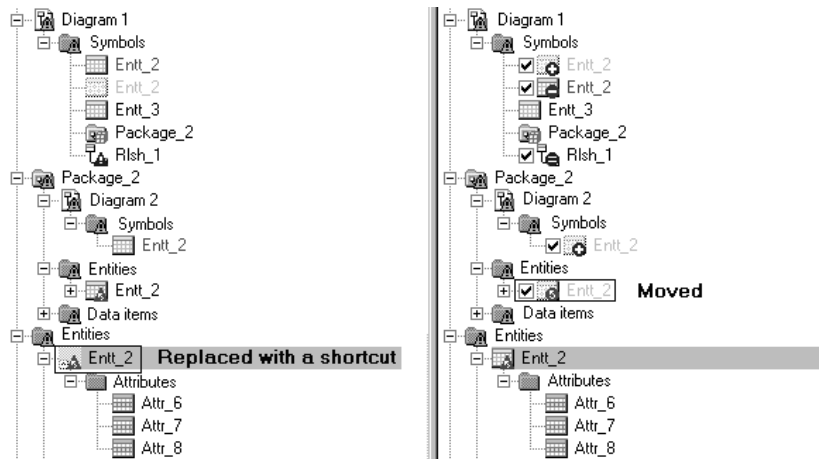
Left model	Right model	Description
🔳	☑ 🔘	Move target object
🔺	🔳	Replace target object with a shortcut

Move and namespace
When you merge models, move is not a suggested merge action for objects which are in different namespaces but have identical name. In that case creation or deletion of the object are proposed.

☞ For more information on namespace in models, see "Managing the namespace in models" in the "Managing Models" chapter.

User permissions in the repository

When a user does not have the appropriate rights and permissions in the repository, merge actions appear with a grayed out check box and cannot be selected.

However, some actions can be applied to objects in packages on which the user does not have the required permission, in order to be consistent with a permitted action.

In the following example, the model on the left pane corresponds to the local model and the model on the right pane corresponds to the repository model. The user has a Read permission on Package 1 and a Write permission on Package 2.
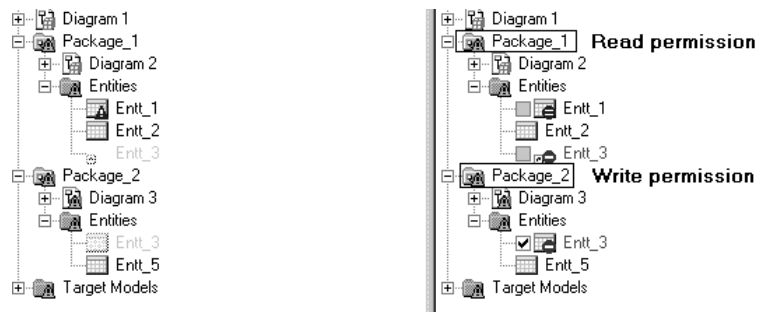
Model	Package	What happens...
Local model (left pane)	Package 2 (Write permission)	Entt_3 was deleted
	Package 1 (Read permission)	Entt_1 properties were modified
		Entt_3 shortcut was deleted because of the deletion of Entt_3 in Package 2
Repository model (right pane)	Package 2 (Write permission)	The proposed merge action for Entt_3 is delete. The user has a Write permission and can select this action
	Package 1 (Read permission)	Entt_1 check box cannot be selected (grayed) because the user does not have Write permission
		The check box for Entt_3 shortcut cannot be selected (grayed) but the shortcut will be deleted anyway to be consistent with the merge action for Entt_3 selected in Package 2

☞ For more information on user permission, see the *Repository User's Guide* .

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.

Consolidated and generated models
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.

Comparison of property values
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 $<<$.

❖ **To select properties to merge**

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.

❖ **To synchronize objects manually**

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:
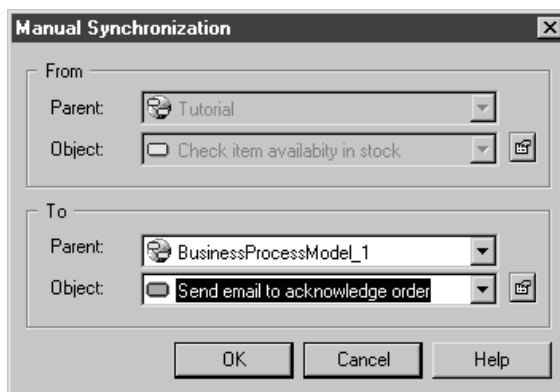
Depending on the merge action proposed, the From or the To box is editable.

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

Filtering the changes to merge

You can filter the objects that are shown in the merge window.

❖ **To define a merge filter**

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

2. Select a filter from the list. You can filter by:
   ♦ All objects
   ♦ All changes
   ♦ Only created objects
   ♦ Only deleted objects
   ♦ Only modified objects
   ♦ Only moved objects
   ♦ Only identical objects
   ♦ Only selected objects
   ♦ Show only conflicts - displays conflicts between models with common origin such as local model and consolidated model or origin model and generated model. A merge conflict occurs when corresponding objects have been modified in both the origin and generated model or the local and Repository model. Such modifications can be a change in object properties, a change in a list of objects, a change in the order of a list or the deletion of an object.
   ♦ Hide all preservations - 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, and also to filter on "object state":
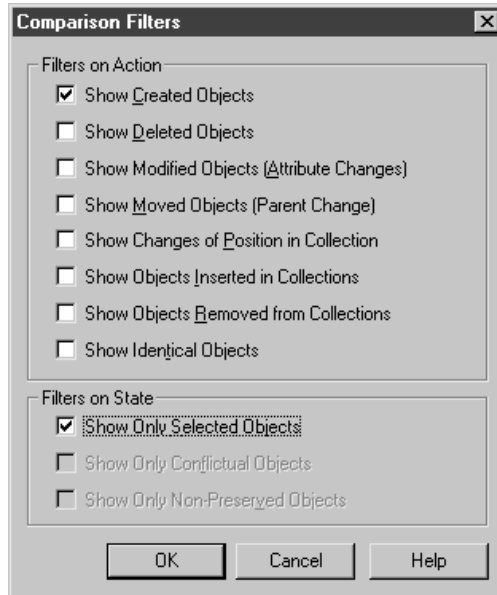
- Show only selected objects - Displays only selected merge actions
- Show only conflictual objects - Displays objects that have been changed simultaneously by two different users. This filter is only available for merge during consolidation
- Show only non-preserved objects - Displays objects modified 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).

Once you have selected all the appropriate criteria, click OK.

The filter is applied to the Merge Models window, and only objects complying with the filter criteria are displayed.

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

Starting the merge process

After selecting merge actions, you can start merging the models.

❖ **To start the merge process**

1. Click OK in the Merge Models dialog box.

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

CHAPTER 11

Impact Analysis

About this chapter

This document explains how to use the impact analysis feature in PowerDesigner.

Contents

Introduction to PowerDesigner Impact Analysis

When you want to perform a change on an object, you need to know the impact that this will have on dependant objects in the current model or other models. Most of the time, the impact is a change or a deletion, but it can also be a user-defined event.

You can use impact analysis to evaluate the consequences of changes before performing them or before consolidating a list of changes in the repository.

The following are examples of situations in which impact analysis may be used:

♦ Changing a requirement when development has already started - You want to know which objects will be impacted and which tests this implies. If the requirement was associated with a use case associated with diagrams and analysis classes implemented by components, impact analysis helps you identify which analysis, design and implementation objects will be impacted by the change

♦ Deleting an object - Impact analysis shows which objects will be impacted by the deletion, it also indicates if this object is used by other objects or in other models, if the deletion implies the deletion of other objects that will themselves impact other objects

♦ Consolidating a model - Impact analysis used with the extract dependencies feature, shows which other models will be impacted by the changes you made

♦ Changing the data type of a primary key column - Impact analysis shows which other tables and columns are impacted by the change

♦ Changing the signature of a class operation - Impact analysis shows which child classes override this operation

PowerDesigner helps in these situations, by allowing you to:

♦ **Analyze** object dependencies and customize the way a change is propagated on depending objects on the Dependencies tab of the Impact Analysis dialog box

♦ **Filter** the list of changes that you want to review on the List tab

♦ **Print or Save** the impact analysis in order to review it on the Preview tab

Opening an impact analysis

You can open an Impact Analysis window in any of the following ways:

♦ Right-click an object entry in the Browser and select Impact Analysis from the contextual menu.

♦ Right-click an object symbol in the diagram window and select Edit ➤ Impact Analysis from the contextual menu.

♦ From the Dependencies tab of an object's property sheet, click the Impact Analysis button.

♦ When deleting an object, click the Impact button from the Confirm Deletion dialog box.

Analyzing the Impact of a Change to a Model

You can analyze the impact of a change to your model from the Impact Analysis window.

The Dependencies page of the Impact Analysis dialog box displays all the objects that depend on the object.

All dependant objects, including external shortcuts to other models are listed, with the external model name appended to the item as in the following example:
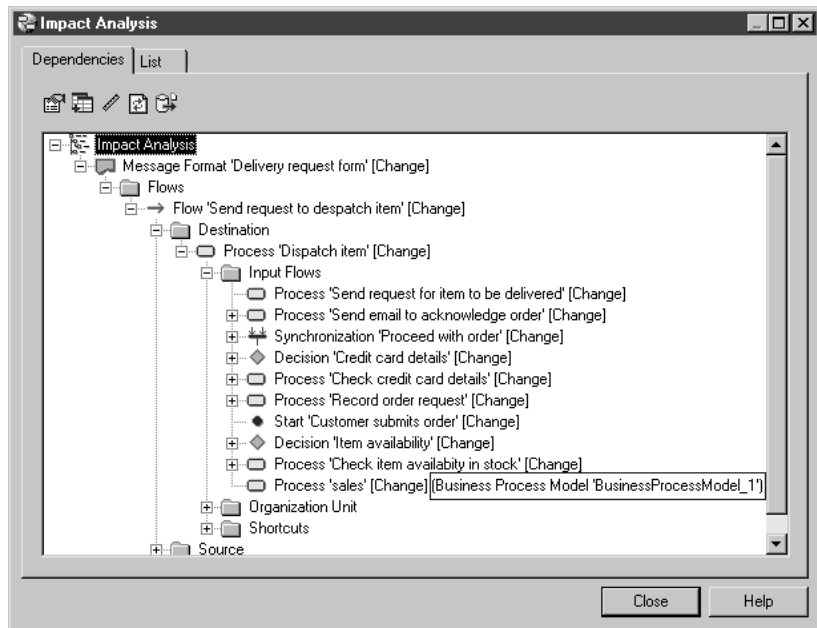
Each impact item in the list displays an "event", which designates the modifications that it will receive. You can modify the default event on an object, for more information see "Customizing events" on page 356.
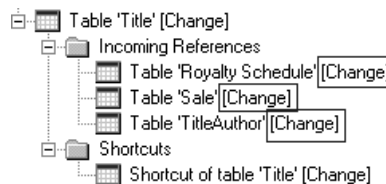
Each event is propagated through propagation rules and may, in turn, provoke other events

☞ For information about how you can customize propagation rules and create your own, see "Creating a user-defined event" on page 356.

Objects that are multiply dependant

If an object is dependant multiple times then, for performance reasons, only the first impact is displayed in detail in the Dependencies tree. You can select the Go to First Occurrence command in the contextual menu of subsequent undeveloped entries to jump to this first occurrence.

The following tools are available on the Dependencies tab:

Tool	Description
	Properties - Opens the property sheet of the item selected in the tree.
	Add Objects - Opens an object selection dialog to let you add objects to be analyzed in the current tree.
	Edit Default Propagation Rule – opens a dialog to let you customize default propagation rules and define new rules.
	Reset - Recalculates the dependencies of the selected node using the propagation rules. Reverses any changes, such as adding or removing collections from the analysis. Since this tool applies only to the selected node, you must select the root object to reset the entire tree.
	Extract Dependencies - If the current model contains links to other models consolidated in the repository, extracts these dependencies. For more information, see "Extracting cross-model dependencies from the repository" on page 353.

In addition, there are many actions available from the contextual menu.

Understanding dependencies

An impact analysis report provides a list of standard object dependencies. You can further refine the analysis and drill down the dependencies tree using propagation rules.

All models open in the workspace are displayed in the dependencies graph. You can also extract and visualize any dependencies in models closed in the workspace but consolidated in the repository.

A dependency is the general term used for a semantic link between objects within the same model, or between different models. Dependencies can

belong to different types of **collections**:

♦ Direct collection - Objects directly related to the object. This link is displayed as an association in the metamodel, and the objects usually appear in a list in the object property sheet. For example, attributes in an entity, or data in a process.

♦ Inverse collection - Objects inversely related to the current object and displayed on the Dependencies tab. For example, diagrams where an object is displayed, or references using a table.

♦ Calculated collection – A user-defined collection created to display a list of associated objects on the Dependencies tab. For example, in a model where columns and domains can diverge, you can create a calculated collection on the domain metaclass that lists all the columns that use the domain and have identical data type

♦ Extended collection – A user-defined collection used to define an additional link between two selected metaclasses or stereotypes, and displayed on the Dependencies tab. For example, you can define an extended collection in the package metaclass and define FileObject as the target metaclass in order to attach documents containing use case specifications to the different packages of a model

☞ You create calculated and extended collections in a resource file attached to the model. For more information, see the "Managing Profiles" chapter in the *Advanced User Documentation* .

In the Dependencies page of the Impact Analysis dialog box, each object is displayed as a node with several sub-categories corresponding to the different collections depending on this object.

```
☐·▦ Table 'AUTHOR' [Delete]
   ☐·🗀 Columns
      ⊞·▦ Column AUTHOR.AU_ID [Delete]
      ⊞·▦ Column AUTHOR.AU_LNAME [Delete]
      ⊞·▦ Column AUTHOR.AU_FNAME [Delete]
      ⊞·▦ Column AUTHOR.AU_BIOGRAPH [Delete]
      ⊞·▦ Column AUTHOR.AU_ADVANCE [Delete]
      ⊞·▦ Column AUTHOR.AU_ADDRESS [Delete]
      ⊞·▦ Column AUTHOR.CITY [Delete]
         ▦ Column AUTHOR.STATE [Delete]
         ▦ Column AUTHOR.POSTALCODE [Delete]
         ▦ Column AUTHOR.AU_PHONE [Delete]
   ☐·🗀 Incoming References
      ⊞·▦ Table 'PICTURE' [Change]
      ⊞·▦ Table 'TITLEAUTHOR' [Change]
   ☐·🗀 Indexes
      ⊞·🗐 Index AUTHOR.AU_LNAME_IDX [Delete]
   ☐·🗀 Keys
      ⊞·🔑 Key AUTHOR.Key_1 [Delete]
```

Collections

Extracting cross-model dependencies from the repository

When you generate a model to another model or create an external shortcut, you create cross-model dependencies. These dependencies are automatically taken into account during impact analysis if both the source and target models are open in the workspace.

If the models are not open in the workspace, but are consolidated in the repository, you can extract these consolidated dependencies.

❖ **To extract dependencies from models consolidated in the repository**

1. Click the Extract Dependencies tool on the Dependencies tab

Extracted dependencies carry a red dot icon on their symbols in the list. To further analyze impact on objects depending on these distant objects you have to open their models.

In the following example, the table Sale is present as a shortcut in another model, and these dependencies are listed under the Incoming References folder:
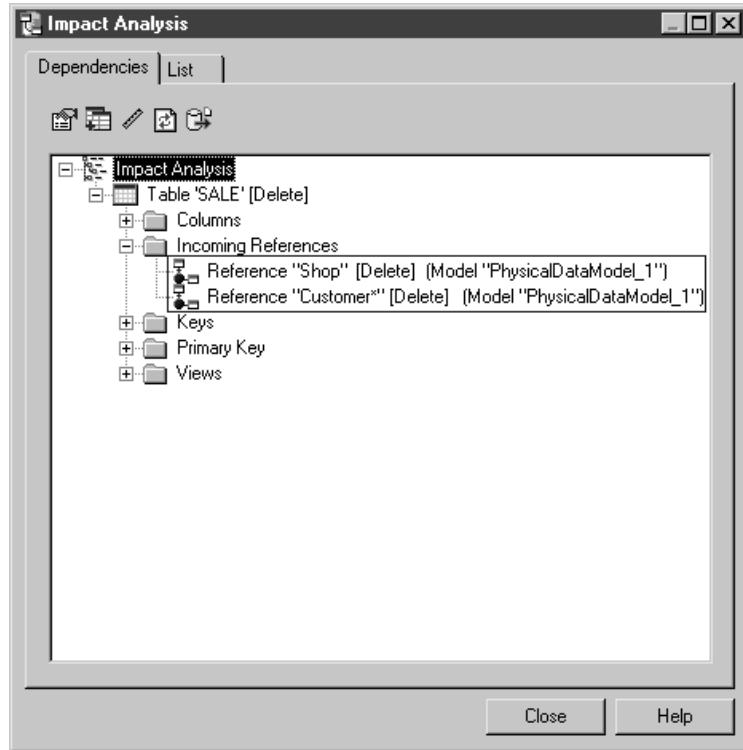
Changing the collections displayed beneath an object

You can add or remove the collections displayed beneath a selected object in order to narrow or widen the analysis.

❖ **To change the default collections of an object**

1. Right-click an object in the Dependencies tree, and select Change
    Propagation Collections from the contextual menu in order to open the
    Object Propagation Collections dialog box.

    This dialog lists all the collections available for the object:

S	Name	Type	L	R	
☐	Articles	Dependency	☐	☐	
☐	Databases	Dependency	☐	☐	
☐	Extended Dependent Objects	Dependency	☐	☐	
☐	Extended Inverse Collections	Dependency	☐	☐	
☑	Incoming References	Dependency	☑	☐	
☐	Incoming View References	Dependency	☐	☐	
☐	Mappings	Dependency	☐	☐	
☐	Replications	Dependency	☐	☐	
☑	Shortcuts	Dependency	☐	☐	
☐	Storages	Dependency	☐	☐	
☐	Synonyms	Dependency	☐	☐	
☐	Tablespaces	Dependency	☐	☐	
☐	Views	Dependency	☐	☐	

Object Propagation Collections dialog box with OK, Cancel, Help buttons.

2. Select or deselect any of the following checkboxes in order to modify the
    display of one or more collections:

    ♦ S [Selected] - displays the corresponding collection under the object
      category in the Dependencies list

    ♦ L [Replace link object by extremity] - If the object is a link, hides the it
      to simplify the display and visualize only the link extremity

    ♦ R [Recursive] - displays recursive collections in a single list

3. [optional] Click the Edit Default Propagation Rules tool to open the
    Propagation Rules Definition dialog and associate additional collections
    with the object. For more information, see "Creating a user-defined
    event" on page 356.

4. Click OK to return to the Dependencies tab.

    The collections appear under the selected object in the tree.

> **Adding user-defined collections**
>
> You can define your own collections of objects and display them in the Impact Analysis graph. For example, in a model where columns diverge from domains, if you create a calculated collection on the domain metaclass that lists all the columns that use the domain and have identical data type, it becomes much easier to evaluate the impact of a change on the domain. For more information, see the "Managing Profiles" chapter in the *Advanced User Documentation* .

Customizing events

The predicted effect on an object in the dependencies list is called an event. You can modify these events using the following commands available from the contextual menu

♦ Change Event to [Delete]

♦ Change Event to [Change]

♦ Change Event Description – opens the Event Description dialog, which allows you additionally to specify you own (previously defined) events

Creating a user-defined event

You create a user-defined event when you need to analyze the consequences of a specific event. For example, you can create an event ModifyDataType on the Domain metaclass. Then you can define a new propagation rule for each collection of the domain likely to be impacted by a data type change in the domain.

❖ **To create a user-defined event**

1. Right-click an object in the Dependencies tree and select Change Event Description in the contextual menu to open the Event Description dialog box.

2. Type an event in the User-defined Events box. Type each event in a single word, you can enter several events using semicolons as separators.

Event Description

Predefined Events:

☐ <u>D</u>elete

☐ <u>C</u>hange

User-defined Events:

ModifyDataType

| OK | Cancel | <u>H</u>elp |
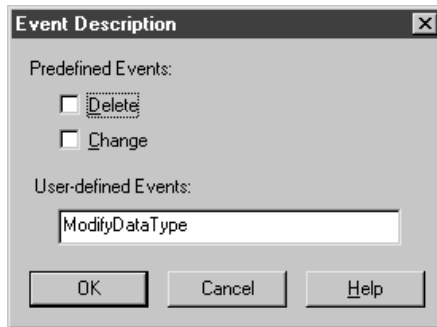
3. Click OK in the Event Description dialog box to return to the Dependencies tree.

The tree is empty since there is no matching event in the default propagation collections.
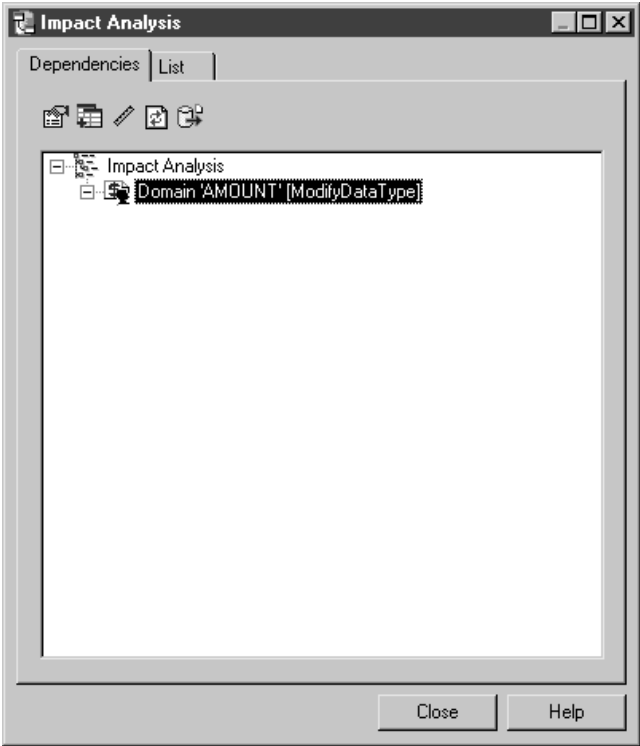
4. Click the Edit Default Propagation Rules tool to open the Propagation Rules Definition dialog box.
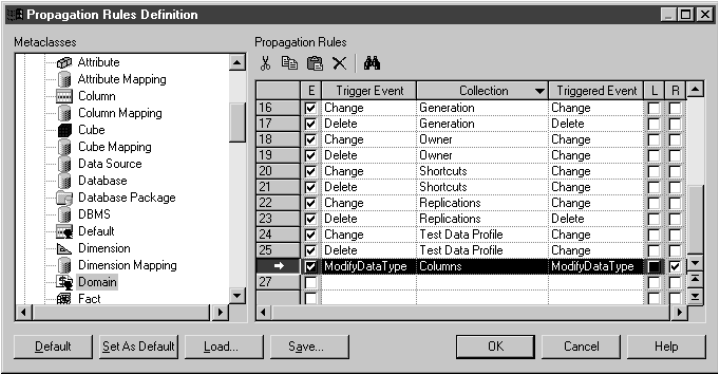
The focus is on the metaclass corresponding to the selected object.

5. Click the Add a Row tool to insert a line in the list.

6. Enter the name of the user-defined event in the Trigger Event zone.

7. Select a collection from the list in the Collection column.

8. Enter the name of the user-defined event or select an existing event in the Triggered Event list. The user-defined event name is case sensitive, you must use the same case as typed in the Event Description dialog box.

9. Click OK to return to the Dependencies list.

   The collection is displayed with the user-defined event in the dependencies tree.
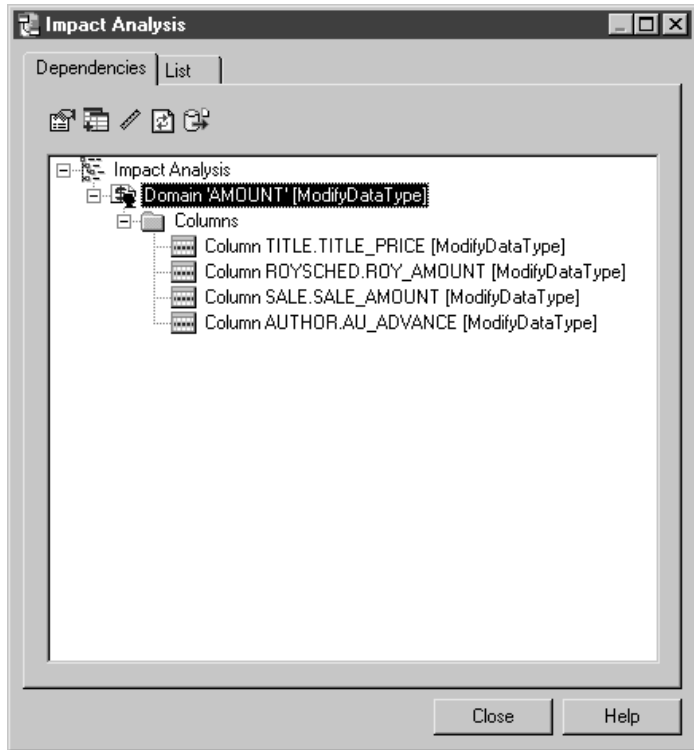
Propagation Rules Definition dialog box

The Propagation Rules Definition dialog box lets you customize default propagation rules and define new propagation rules.

This dialog box is divided in two panes:

♦ Metaclasses – In this tree, the top level lists PowerDesigner object librairies:

  • PdCommon - contains shared metaclasses, such as business rules or files.

  • PdBPM - contains Business Process Model metaclasses.

- PdCDM - contains Conceptual Data Model metaclasses
- PdILM - contains Information Liquidity Model metaclasses
- PdFRM - contains Free Model metaclasses
- PdOOM - contains Object Oriented Model metaclasses
- PdPDM - contains Physical Data Model metaclasses
- PdRQM - contains Requirements Model metaclasses
- PdXSM - contains XML Model metaclasses

You can expand any of these libraries to select a metaclass in order to display its propagation rules.

☞ For more information on PowerDesigner libraries and metaclasses, see the "Resource Files and the Public Metamodel" chapter in the *Advanced User Documentation* .

♦ Propagation Rules - list of propagation rules for a selected metaclass. You can define the following options in this list:

- E [Enable] - enables a propagation rule for a specific collection of the metaclass
- Trigger Event – specifies the event that happens to the metaclass and that triggers the propagation rule
- Collection – Specifies the name of the metaclass collection for which you define a propagation rule
- Triggered Event – Specifies the type of event propagated to objects in the collection
- L [Replace link object by extremity] - Hides the link in the propagation tree in order to simplify the display and visualize only link extremity
- R [Recursive] - Propagates an event recursively to the depending objects of the current metaclass

In the following example, the Entity metaclass is selected from within the list of CDM metaclasses. The first propagation rule in the list specifies that if an entity is changed, then associations related to this entity are also changed. The second rule specifies that if an entity is deleted then associations related to this entity are changed:
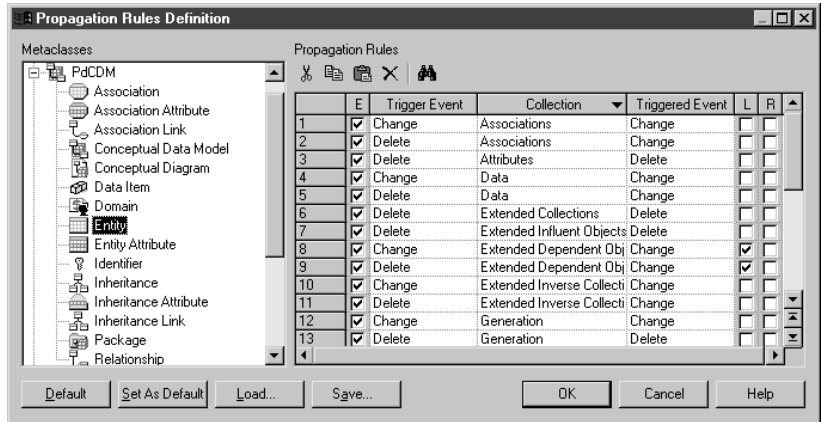
You can delete a propagation rule by selecting it and clicking the Delete tool or pressing the DEL key. To disable a propagation rule without deleting it, clear its checkbox in the E column of the list.

You can save your changes to the propagation rules by clicking the Save tool. The changes are saved in a rule file with the .reg extension. You cannot save changes individually on a metaclass basis.

You can load an existing rule file using the **Load** button or double-clicking a .reg rule file.

To cancel any changes and revert to the default rules, click the **Default** button.

To set your changes as the default and save them to the registry, click the **Set as Default** button . Any change not saved or Set as Default will be lost when you close PowerDesigner.

Managing and Printing the Impact Analysis

The List page of the Impact Analysis dialog box allows you to define the content of the impact analysis report using the Filter tool. You can select exactly which attributes of impacted objects you need to visualize in order to have the best impact overview.

You can print the content of the list using the Print tool.

The list displays standard management tools:

Tool	Description
	Displays the property sheet of the selected item in the list
	Saves the report in RTF (for MS Word) or CSV (for MS Excel) format
	Copies the report in RTF or CSV format
	Prints the report
	Opens the Find dialog box to let you search the list for a specific item
	Opens the Customize Columns and Filter dialog box to let you select attributes to display or remove, and also filter the display using expressions. You can select attributes specific to impact analysis like "Event" that indicates the impact analysis event, or "Propagation path" that provides the list of influent objects and collection names. For more information, see "Defining a filter on a list" in the "Using the PowerDesigner Interface" chapter.
	Enables or disables the filter

Creating Mappings

This document describes how to create mappings between model objects using the PowerDesigner Mapping Editor or the Mapping tab in the property sheet of individual objects.

Contents

Introduction to Model Object Mapping

Object mapping allows you to establish connections between objects belonging to heterogeneous models and diagrams. You create mappings between objects to setup a structure for data movement and transformation or simply to express a simple correspondence between model objects. When data movement is involved, data originates in a data source and is loaded to another target database or model.

You can create mappings in either of the following ways:

♦ From the Mapping Editor that gives you a global view of all the mappings to objects in the current (target) model.

♦ From the Mapping tab in the property sheet of model objects when you are working on a specific object.

The following table lists the supported types of model-to-model mapping in PowerDesigner:

Source	Target	Mapping type
PDM	OOM	**O/R (Object-Relational)** to associate classes to tables to store OOM objects into a relational database
CDM	OOM	**O/R (Object-Conceptual)** to associate classes to entities and associations to associations or relationships to clarify object relationships
PDM	PDM	**R/R (Relational-Relational)**: to associate tables and views to other tables of another database
		Multidimensional-Relational –: to associate cubes, facts and dimensions to tables to populate OLAP cubes from relational databases
PDM	XSM	**XML/R (XML-Relational)**: to associate elements and complex types to PDM tables, views and abstract data types
OOM	XSM	**XML/O (XML-Object)**: to associate elements and complex types to OOM classes
XSM	XSM	**XML/XML (XML-XML)**: to associate elements and complex types from one XML Model to another in order to define how an XML document can be transformed into another (usually, using XSLT or Xquery)

Source	Target	Mapping type
CDM	CDM	**CDM/CDM (Conceptual-Conceptual)**: to associate entities, associations, inheritances and data items from one CDM to another in order to clarify object relationships
OOM	CDM	**CDM/O (Conceptual-Object)**: to associate entities and associations to OOM classes and associations to clarify object relationships

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. In certain cases, you can further define the mapping by associating sub-objects in the target and source models.

Objects that can be mapped

The following table lists the objects and sub-objects that can be mapped for each type of model:

Model	Objects	Sub-objects
PDM	Table	Table column
	View	View column
	Dimension	Attribute
	Fact	Measure
	Cube	Measure
	Reference	N/A
OOM	Class	Class attribute, operation, association
XSM	Element	Element attribute
	Complex type	Complex type attribute
CDM	Entity	Entity attribute
	Association/relationship	Association attribute
	Inheritance	Inheritance attribute
	Data item	N/A

Accessing inter-model mappings in the ILM

The Information Liquidity Model (ILM) provides a graphical view of various databases (PDM) and formats, such as XML document (XSM), data application (OOM), and conceptual data (CDM) together with the different transformations that operate between them. You can easily visualize or further define the correspondence between the various elements by right-clicking the data access link between these elements and open the Mapping Editor:
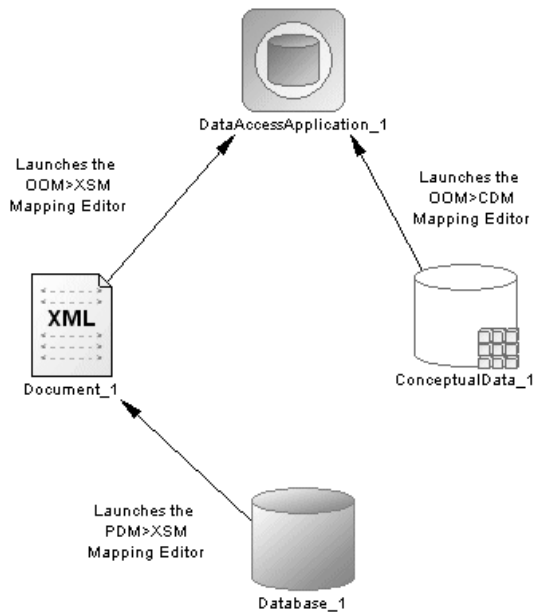
☞ For more information, see the "Data Access Links" section in the "Building Information Liquidity Diagrams" chapter in the ILM User's Guide.

Creating replications

The Mapping Editor also allows you to visualize and manage replications defined for RepServer and Mobilink in the ILM. It is available from the replication process contextual menu:
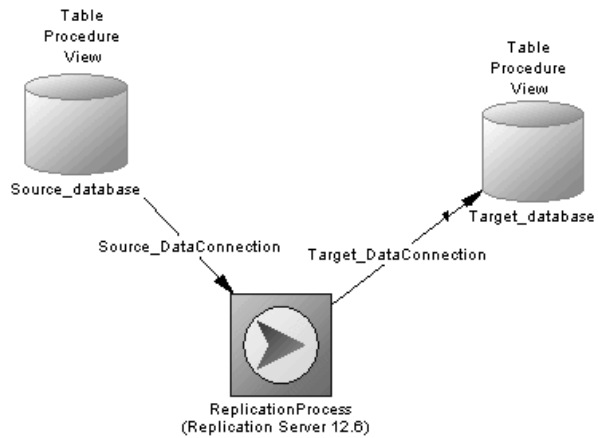
This ILM Mapping Editor is slightly different to the standard Mapping Editor:
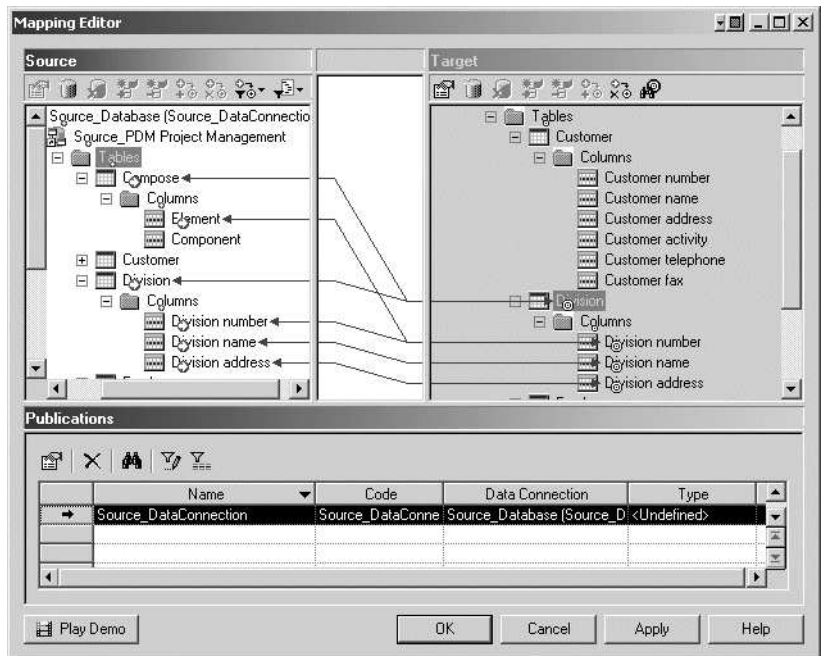
☞ For more information, see the "Using the Mapping Editor for Replications" chapter in the *ILM Users Guide* .

Automatic mapping

When you perform an inter-model generation, and provided you have

367

selected the corresponding mapping check box in the Generation Options dialog box, the objects of both the source and target models are automatically mapped and a data source is created in the target model. This process is time saving to initialize target objects with a simple default mapping.

You can then fine tune the links between source and target model objects by defining object links manually. This process provides great flexibility for customizing the mapping between objects and guarantee a correct storage of objects in the target model.

☞ For more information about inter-model generation, see the corresponding sections in the "Generating Other Models" chapter in the different user's guides.

Object Oriented Model mappings

In an Object Oriented Model (OOM), you can create the following types of mappings:

♦ **Object to relational (O/R)** mapping between OOM classes and PDM tables to manage the persistence of objects in a relational database. See "Object to Relational (O/R) mapping" on page 368.

♦ **Object to conceptual** mapping between OOM classes and CDM entities to express a simple correspondence between objects in a Conceptual Data Model. See "Object to conceptual mapping" on page 371.

The following table shows all the allowed mappings with OOM objects:

Target OOM object	Source PDM object	Source CDM object
Class	Table	Entity
Class attribute	Table column	Entity attribute
Operation	SELECT or UPDATE queries associated with a table	—
Association	Reference, view reference, table or view	Association, relationship

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:
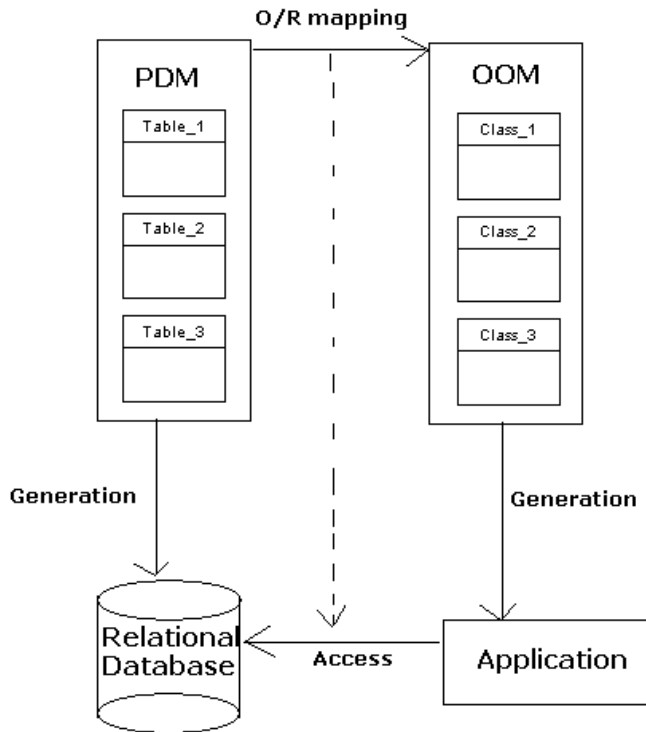
Object persistence implies to store and extract objects in a relational database.

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:
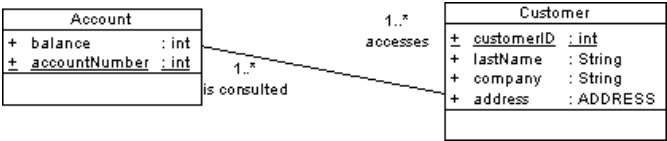
In a relational database, this association is mapped to the following associative table:
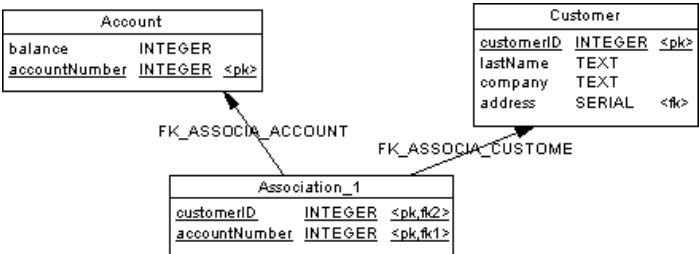
SQL queries

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

Object to conceptual mapping

If you have various types of conceptual models including Conceptual Data Models (CDMs) and Object-Oriented Models (OOMs), you may want to link equivalent concepts between these various models.

Mapping CDM objects with OOM objects will allow you to further define the links between CDM and OOM objects.

Physical Data Model mappings

In a Physical Data Model (PDM), you can link physical and multidimensional objects for the following purposes:

♦ **Relational to relational** mapping between physical objects to generate extraction scripts to populate a data warehouse with operational data

♦ **Relational to multidimensional** mapping between physical and multidimensional objects to generate text files containing cube data to fill the OLAP database

You create a mapping between PDM objects to express a simple correspondence between objects or to setup a structure for data movement and transformation. Data comes from a **data source** and is loaded in another

database.

In PowerDesigner, the data source and the destination database can be designed in physical data models.

Data is taken from a database and sent to another database. The transfer can be from operational to data warehouse or OLAP databases, or from data warehouse to OLAP databases. Operational and data warehouse data are designed in physical diagrams and OLAP data are designed in multidimensional diagrams.

Relational to relational mapping

Data warehousing requires to extract, transform, and load data from operational systems to a data warehouse database. You can map PDM objects to create a relational to relational mapping between operational and data warehouse data.

Operational and data warehouse data can be designed in physical data models.

This kind of mapping is used to clearly identify the data sources i.e. the operational databases, of the data warehouse database.

Moreover, once this mapping is defined, it can be used to generate extraction scripts to populate a data warehouse with operational data.
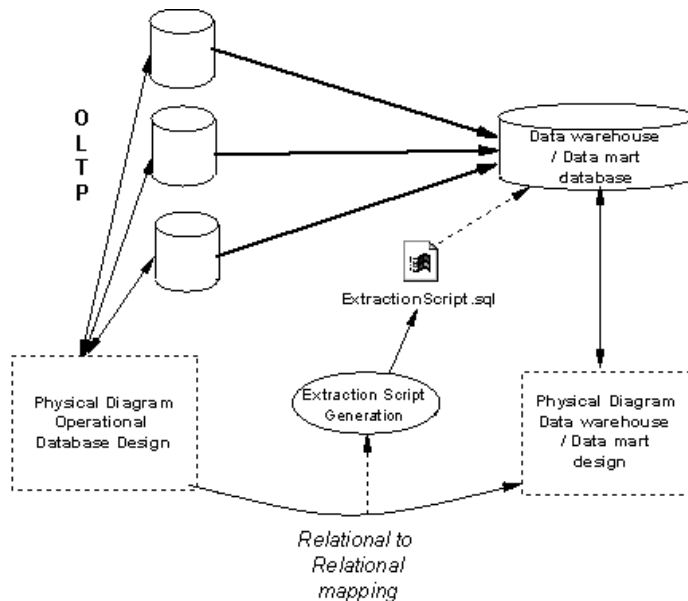
The following table lists object mapping in a relational to relational mapping:

Operational object	Data warehouse object
Table	Table (Fact or Dimension type)
Column	Column

Relational to multidimensional mapping

OLAP cubes are usually filled with data from a data warehouse database.

You can link PDM objects to create a relational to multidimensional mapping between physical objects and OLAP objects. With this link, it will be possible to generate text files containing data warehouse data. These text files can be used to populate OLAP cubes.
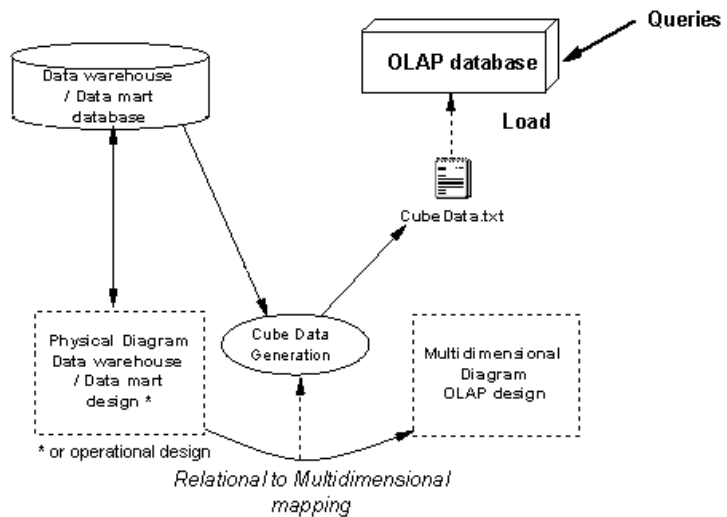
Automatic mapping
When you use the Rebuild Cubes feature to create cubes and dimensions from fact and dimension tables, the mapping between source tables and OLAP objects is automatically performed. For more information about the Rebuild Cubes feature, see section Rebuilding Cubes in chapter Building Multidimensional Diagrams in the *PDM User's Guide* .

The relational to multidimensional mapping is used to generate cube data in

text files to be loaded by OLAP engines.

Mapping a physical object to multidimensional objects	The mapping for a physical object (dimension or fact) is used to supply data for the cube dimensions or cube measures in OLAP databases. You can identify the tables or views used to populate OLAP cubes. The tables in the data source need not be of dimension or fact type. Once the source tables or views are identified, you can define mappings between attributes or measures and table columns.
Select sub-tab	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.

☞ For more information on the generation of cube data, see the "Generating cube data" section in the "Working with PDMs" chapter in the *PDM User's Guide* .

XML Model mappings

In an XML Model (XSM), you can create the following types of mappings to express generation dependencies or data transformation movement between model objects:

♦ **XML-to relational** mapping between XML elements or complex types and PDM tables.

♦ **XML to object** mapping between XML elements or complex types and OOM classes

♦ **XML to XML** mapping between XML elements or complex types and elements of another XSM

The following table shows all the allowed mappings with XML Model (XSM) objects:

Target XSM object	Source PDM object	Source OOM object	Source XSM object
Element	Table, view, table column, view column	Class, class attribute	Element, complex type, element attribute
Element attribute	Column, view column	Class attribute	Element attribute, element
Complex type	Abstract data type	Class	Complex type, element
Complex type attribute	Abstract data type attribute	Class attribute	Complex type attribute, element attribute

Mapping elements

XML elements can also be mapped to the sub-objects of source objects, as an XML element can correspond to a simple type containing only an elementary value (for example <Name>Bill</Name>).

Conceptual Data Model mappings

In a Conceptual Data Model (CDM), you can create the following types of mappings to express a simple correspondence between model objects:

♦ **Conceptual to object mapping between CDM entities and OOM classes**

♦ **Conceptual to conceptual mapping between CDM entities and the entities of another CDM**

Both mapping types help to make object relationships clearer.

The following table shows all the allowed mappings with CDM objects:

Target CDM object	Source OOM object	Source CDM object
Entity	Class	Entity, association, inheritance
Entity attribute	Class attribute	Entity attribute, association attribute, inheritance attribute
Association/relationship	Association	Entity, association, inheritance, relationship
Association attribute	Association attribute	Association attribute, entity attribute, inheritance attribute
Inheritance	Class	Entity, association, inheritance
Inheritance attribute	Class attribute	Inheritance attribute, entity attribute, association attribute
Data item	—	Data item

Mapping inheritances

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.

☞ For more information about inheritances, see section "Inheritances" in the Building a Conceptual Data Model chapter in the *CDM User's Guide* .

Creating Mappings from the Mapping Editor

The Mapping Editor provides a global view of all the mappings defined for a given model, allowing you to quickly identify mapped and unmapped objects.

♦ You first need to create a data source, and then proceed to create mappings.

❖ **To launch the Mapping Editor**

1. Select Tools ➤ Mapping Editor in your target model (OOM, PDM, CDM or XSM) to open the Mapping Editor.

   The Mapping Editor opens and automatically launches the Data Source Creation Wizard if no data source is defined for the model. For detailed information about this wizard, see "Creating a data source" on page 387.

2. Specify a name, an access type and a model type (if required) for the source from which you will draw your data, and then click Next.

3. Select one or more models from the list.

4. Click Next and clear the Create default mappings check box if you do not want to create default mappings between source and target objects with the same name then click Finish.

   or

   Click Finish if you want to create default mappings.

   The wizard closes and the data source is created in the Source pane of the Mapping Editor showing default mappings if any.

> **Mapping Editor video**
> 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.

Understanding the Mapping Editor interface

The Mapping Editor window is divided into three panes:
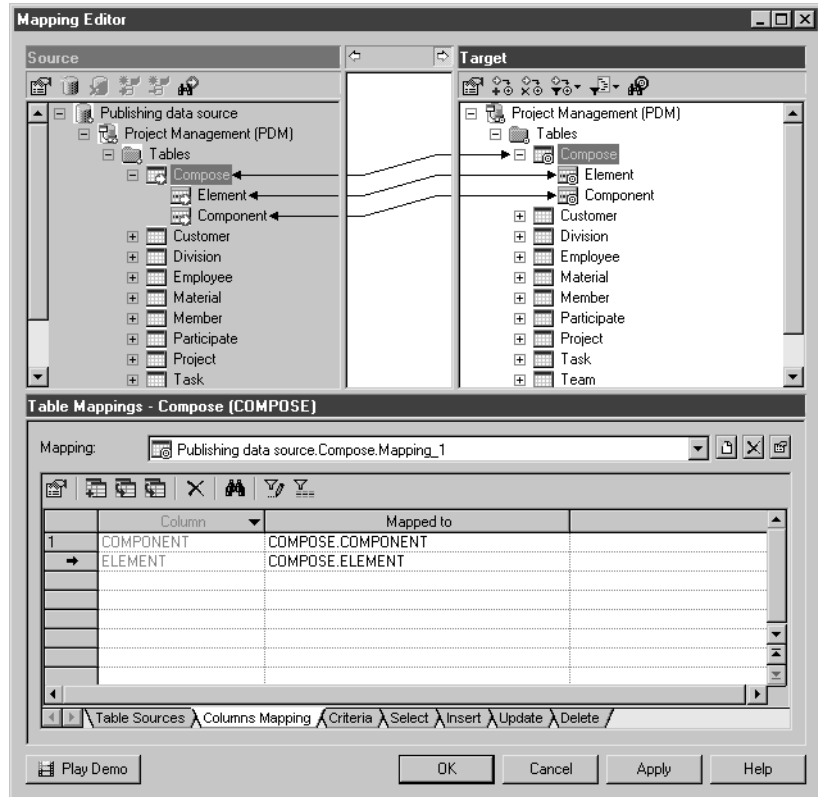
- ◆ The Source pane – has a gray background and is, by default, on the left of the window. For more information, see "Mapping Editor Source pane" on page 394.

- ◆ 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 395.

- ◆ 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 the ""Mapping Editor Mappings pane" on page 397" section.

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 a mapping from the Mapping Editor

You can create a mapping in the Mapping Editor in any of the following ways:

♦ Drag an object from one pane and drop it on an object in the other.

♦ Select an object in each of the target and source panes, and then click the Create Mapping tool.

♦ Select an object in each of the target and source panes, and then click the Create Mapping tool in the Mappings pane.

♦ Select an object in each of the target and source panes, right-click one, and select Create Mapping.

♦ Right click a data source and select Generate Default Mappings.

Differences between the mapping methods
When dragging and dropping, you can either create a mapping if no mapping is defined for an object icon or reuse and complement an existing mapping. When using the Create Mapping tools or contextual menu command, you can create several different mappings for the same target object. Each mapping object is added to the Mapping list and you can access its property sheet.

Generate Default Mappings	The Generate Default Mapping feature allows you to automatically create mappings between source and target objects having the same name. When an object in the target model is mapped to an object in the source model, their sub-objects are automatically mapped when their name and code match.
Automatic mapping	When objects in the Source and Target pane are mapped, their sub-objects are automatically mapped if the current model (target) contains sub-objects whose name and code match those of the sub-objects in the source models.

You can force the mapping using the Generate Mapping tool in the *Sub-Object* sub-tab of the Mappings pane to automatically generate a mapping between sub-objects with same name or code in the source and target models.

> **Multi-selection allowed**
> 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.

❖ **To create a mapping using the drag and drop feature**

1. 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.

> **Re-use of existing mappings**
> 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 attributes 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 attributes mappings in the Sub-Object mapping sub-tab.

♦ Sub-object owned by a different object ➤ object [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 is 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 in the Mapping Editor toolbar

You can create a mapping using the Create Mapping tool on the Mapping Editor toolbar.

❖ **To create a mapping using the Create Mapping tool from 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:
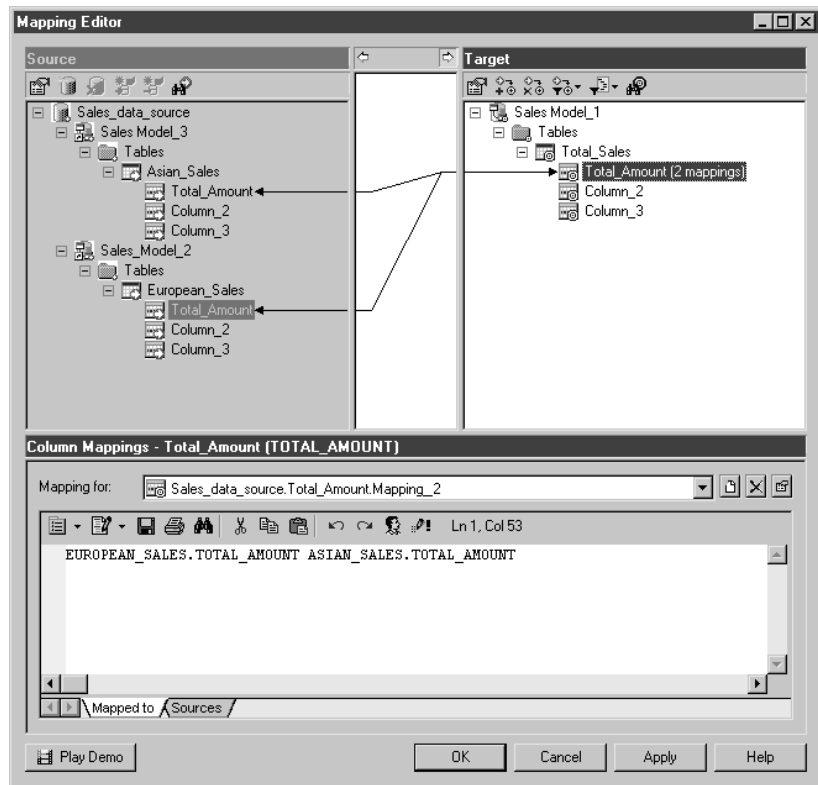
Creating a mapping with the Create Mapping tool in the Mappings pane

You can create a mapping using the Create Mapping tool from the Mappings pane.

❖ **To 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.

❖ **To create a mapping using the Create Mapping command in an object's contextual menu**

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

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:
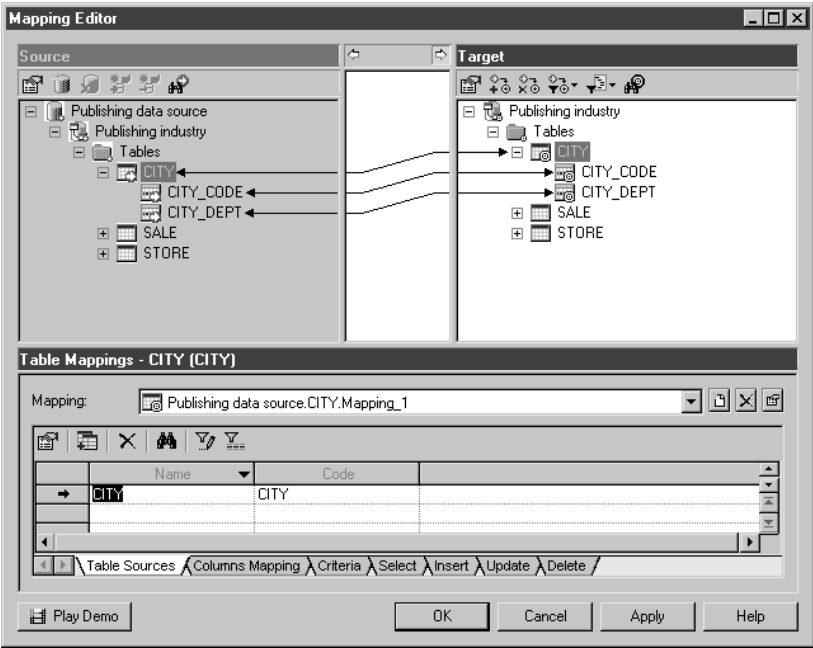
| Multiple sources mapping | A mapping with multiple sources can be designed as follows. The CITY table in the Target pane has two sources (CITY and SALE tables): |
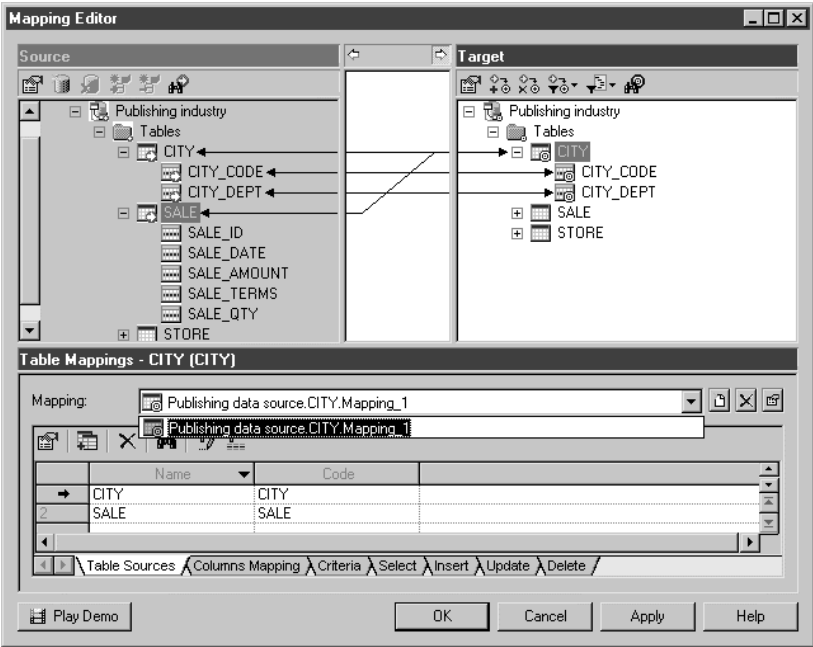
Different mappings for the same target

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:
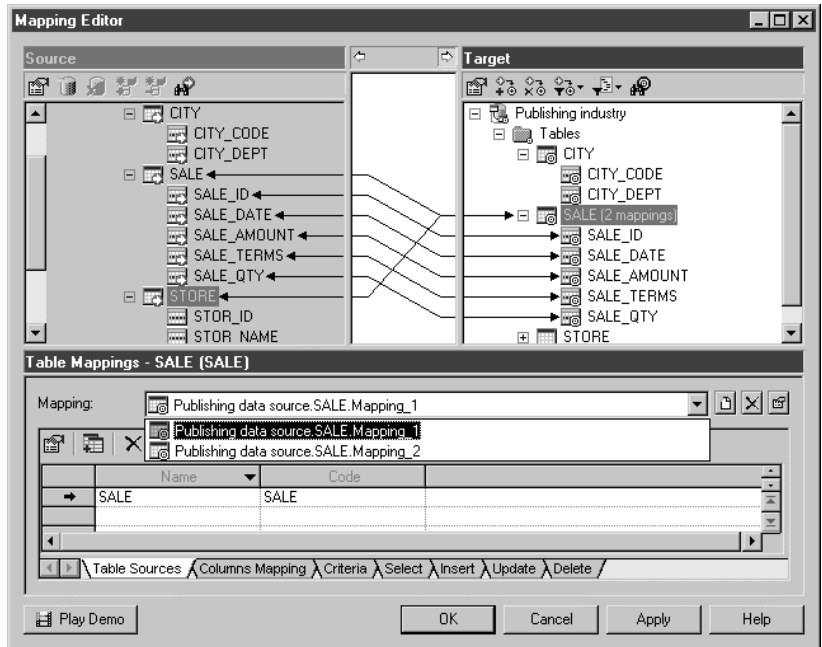
Defining data sources

You create a data source to define where (in which model) data should be extracted and transferred to another model.

When you define a data source, you have to declare available models in the list of data source models. A data source can contain several models, you can select the source models among a list of models open in the current workspace. All the selected models represent the same data source.

A model can contain several data sources.

Data source properties

To display a data source property sheet, double-click its name or its icon in the Browser tree view.

General

The General tab of a data source property sheet displays the following properties:

Property	Description
Name	Name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	Technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Descriptive label for the data source
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 387".
Model Type	Type of model(s) being used as data source. See "Introduction to Model Object Mapping" on page 364

A data source also includes the following properties:

Models

The Models tab of a data source property sheet displays an empty list of models. You have to define the list of models used as source of data. You use the Add Models tools to open a selection dialog box. You must select among models open in the current workspace.

Database connection
(PDM data source only)

For a PDM data source, the Database Connection tab lets you define the following parameters to connect to a database:

Parameter	Description
Using a data source	Data source associated with the database
Using a connection string	Allows you to connect to a database. A connection string is a series of semicolon delimited arguments that define parameters such as the data source provider and the location of the data source. For example, the connection string for jConnect is: `jdbc:sybase:Tds:machine-name:port-number`
User ID	User login
Password	User password

Selecting a data source

The Select a Data Source dialog box lets you define the following parameters:

Parameter	Description
Machine data source	This 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
File data source	This 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
Setup	Displays the Configuration dialog box for the selected DBMS
Add	Displays the Data Source Administrator dialog box for creating a data source

Creating a data source

You can create a data source in any of the following ways:

♦ Select Model ➤ Data Sources to access the List of Data Sources, and click the Add a Row tool. The List of Data Sources allows you to create several data sources at once.

♦ Click the Mapping tab of an object property sheet. When you create a mapping from the Mapping tab, you are prompted to create a data source if no data source is defined for the model. The Data Source Creation Wizard opens automatically. See .

♦ Select Tools ➤ Mapping Editor. You are prompted to create a data source if no data source is defined for the model. The Data Source Creation Wizard opens automatically.

The Data Source Creation Wizard allows you to create a data source in your model. It opens automatically when no data source is defined for the model either when you open the Mapping Editor or when you create a mapping from the Mapping tab of an object's property sheet.

You can create additional data sources at any time.

From the Mapping Editor, the Data Source Creation Wizard automatically creates default mappings between source and target objects with the same name. If you do not want the wizard to create these default mappings, you have to deselect the Create default mappings check box on the Options page. Note that you cannot create default mappings between source and target objects with the same name when you open the Data Source Creation Wizard from the Mapping tab of an object property sheet.

❖ **To create a data source using the Data Source Creation Wizard**

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 390.

   ♦ 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. [If required] Select the type of the model you want to use as schema for the data source.
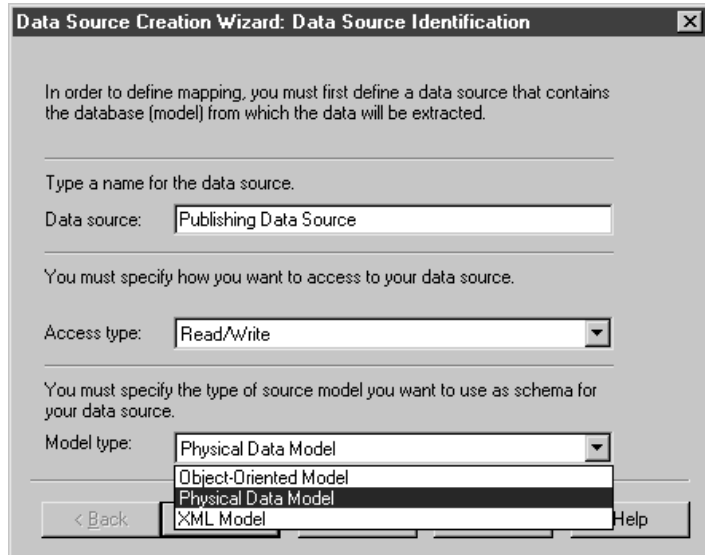
3. Click Next to go to the Source Model Selection page.
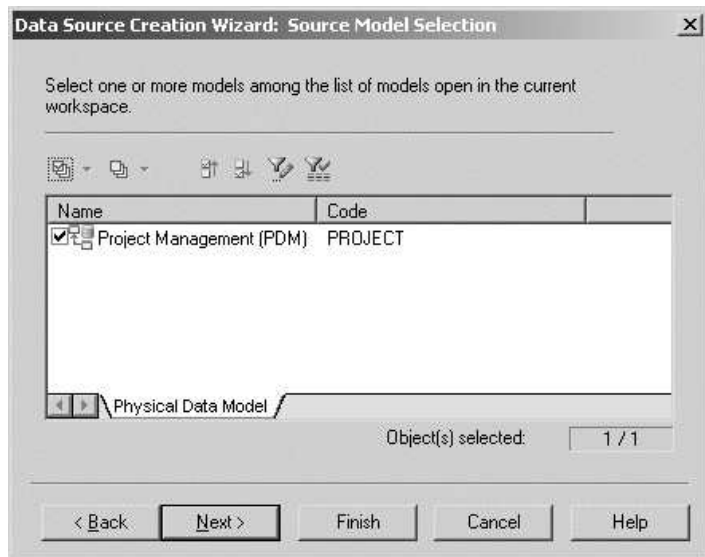
4. Select one or more models from the list of open models in the workspace

5. Click Next to go to the Options page and clear the Create default mappings check box if you do not want to create default mappings between source and target objects with the same name then click Finish.

   or

Click Finish if you want to create default mappings.

The wizard closes and the data source is created in the Source pane of the Mapping Editor. The models you have selected are displayed beneath showing default mappings if any. If you need to create other data sources, you can click the Create Data Source tool in the Source pane toolbar. For more information about the Mapping Editor interface, see "Understanding the Mapping Editor interface" on page 377.

You can add additional models to a data source.

❖ **To add additional models to a data source**

1. From the Mapping Editor, click the Add Model to Selected Data Source tool, select a model from the selection box, and then click OK.

   If the data source is a PDM, you can click the Database Connection tab to define parameters to connect to a database. See "Data source properties" on page 385.

You can create additional data sources.

❖ **To create additional data sources**

1. From the Mapping Editor, click the Create Data Source tool to launch the Data Source Creation Wizard.

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:
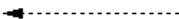
You can also create **reverse mappings, which** define how to populate the objects of the source models with those of the current model. To do this, click the left-hand arrow in the middle area. What was originally the source model is renamed to the target:

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 387) 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.

❖ **To modify the default mapping syntax using the Object Expression Editor**

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.

393
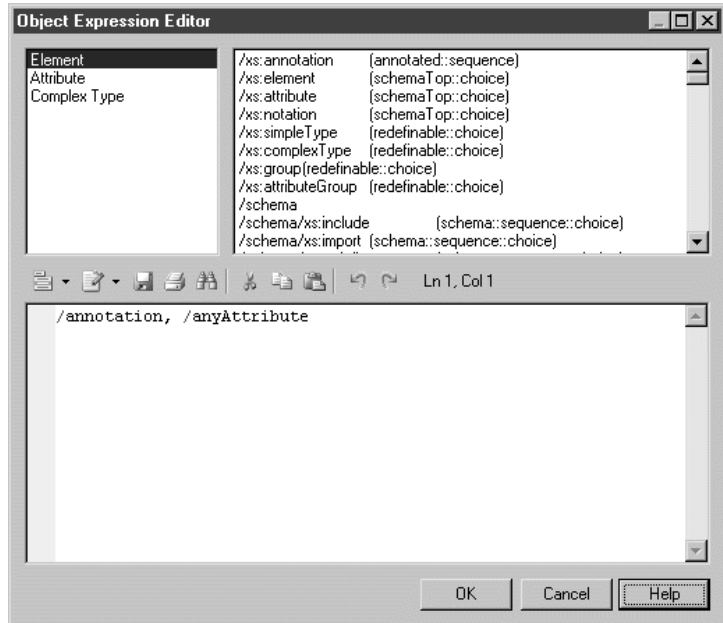
Mapping Editor Interface Reference

The following sections provide detailed information about the Mapping Editor interface.

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 need to create at least one data source 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. You can create several data sources.

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 always displays a gray background ; it is primarily displayed on the left in the Mapping Editor. However, the Source pane is displayed on the right pane in case of reverse mapping editing.

☞ For more information about reverse mapping, see the ""Creating Mappings from the Mapping Editor" on page 377" section.

The following toolbar helps you manage sources:

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 387.
	Delete Data Source - Deletes the selected data source. Related mappings, if any, are automatically deleted.
	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 deletion and object deletion in the Source pane

The following table lists how you can delete mappings and objects in the Source pane.

Type of deletion	Short keys
Mapping deletion	DEL key on a mapped object in the Source pane to delete the mapping with its target object in the Target pane
	Contextual menu: Delete Mapping
Object deletion	SHIFT+DEL combination key to delete a selected object in the Source pane
	Contextual menu: Delete

Generate Default Mappings

You can right click a data source and select Generate Default Mappings. The Generate Default Mapping feature allows you to automatically create mappings between source and target objects having the same name. When an object in the target model is mapped to an object in the source model, their sub-objects are automatically mapped when their name and code match.

Mapping Editor Target pane

The Target pane displays the model where data is extracted. It is primarily displayed on the right in the Mapping Editor. However, the Target pane is displayed on the left pane in case of reverse mapping.

For more information about reverse mapping,, see .

The following toolbar helps you manage mappings in the target model:

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.
	Delete Mappings - Deletes all the mappings for the selected target object.
	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.

Create Mapping tool behavior

Depending on the selected objects in the Source and Target panes, using the Create Mapping tool can lead to diverse behaviors:

Selected source >target objects	What happens...
Folder, model or sub-object > object	A selection dialog box opens to let you select an appropriate object in the source model.
Object > object Sub-object > sub-object	Objects are automatically mapped together with their respective sub-objects if any when their name and code match.
Object > sub-object	A selection dialog box opens to let you select an appropriate sub-object from the selected object in the source model.

☞ For more information about objects and their sub-objects, see .

Mapping deletion and object deletion in the Target pane

The following table lists how you can delete mappings and objects in the Target pane.

Type of deletion	Short keys
Mapping deletion	DEL key on a mapped object in the Target pane to delete the mapping with its source object in the Source pane
	Contextual menu: Delete Mapping
Object deletion	SHIFT+DEL combination key to delete a selected object in the Target pane
	Contextual menu: Delete

For mapping deletion, you can also use the Delete Mappings tool from the Target pane toolbar.

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.
Object. See "Creating a mapping from the Mapping Editor" on page 379	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. See "Creating a mapping from the Mapping Editor" on page 379	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**.
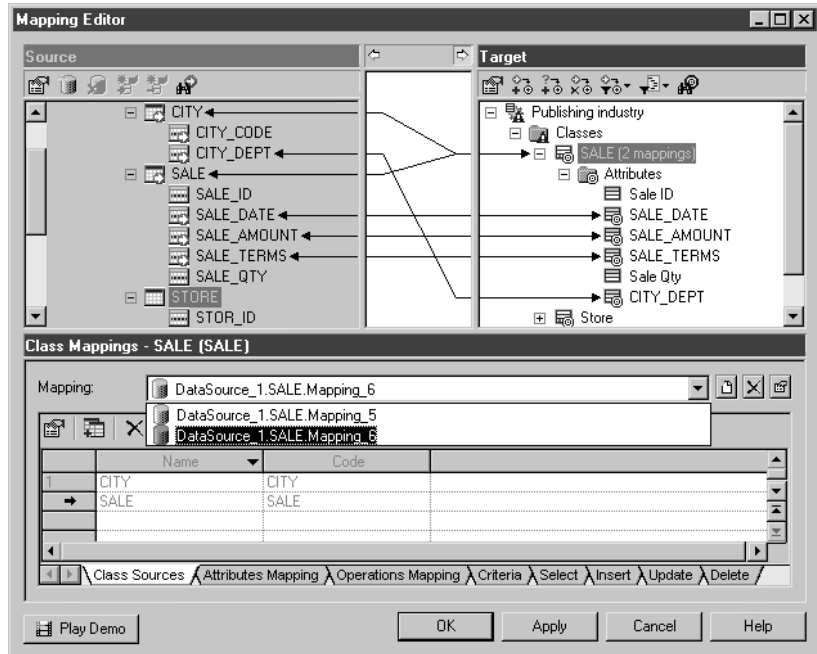
All the mappings for a selected object in the Target pane are displayed in the Mapping list located in the lower part of the Mapping Editor dialog box. You can select one and modify it using:

♦ The Properties tool beside the Mapping list to open its property sheet.

♦ The different queries sub-tabs displayed in the lower part of the Mappings pane. For more information, see the ""Object mappings property sheet Query tab" on page 401" section.

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.

Object mappings properties

When you map an object or a sub-object, its mapping details appear in the Mappings pane. The mapping has the following properties that you access using the Properties tool beside the Mapping list or using the different sub-tabs displayed in the lower part of the Mappings pane.

Object mappings property sheet General tab

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	Mapping expression. You can customize the content of the "Mapped to" box by inserting comments manually. You can also click the Edit tool and use the SQL Editor (for PDM data source) or Object Expression Editor (for other data sources) to modify the default mapping syntax. You can recover the computed expression by clicking the User-Defined tool, this will remove any non calculated expression from the box. For more information about the use of the different editors, see the Defining queries with the SQL Editor section in the Physical Data Model Basics chapter in the *PDM User's Guide* or .

Object mappings property sheet Sources tab

The Sources tab allows you to associate one or several objects or sub-objects in the data source to the object or sub-object in the target model.

You can use the Add Sources tool to select objects or sub-objects from the appropriate models open in the current workspace.

Object mappings property sheet Sub-Object Mappings tab

Each sub-object in the source object can be associated with a sub-object of the target object in order to specify where and how attribute values are stored in the database or model. The sub-object of the source object is displayed in the Selection dialog box in order to let you create a mapping with these inherited sub-objects. You can use the following tools to define sub-object mappings:

Tool	Description
▦	Add Mapping - Allows you to select the sub-objects in the target object that will be mapped to sub-objects in the source object. Once you have selected the sub-objects, you can use the list in the Source column to select the corresponding sub-objects in the source object.
▦	Create from Sources - Allows you to copy sub-objects from the object in the source model to the target 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 - Allows you to automatically generate a mapping between sub-objects with same name or code in the source and target models.

The Ellipsis button in the Source column opens the SQL Editor or the Object Expression Editor dialog box to let you customize the source expression for the sub-objects.

Object mappings property sheet Operation Mappings tab

The Operations Mapping tab allows you to associate a SQL query to an operation of a class.

☞ For more information about operation mappings, see the "Mapping an operation" section in chapter "Working with OOMs" in the *Object-Oriented Model User's Guide.*

Object mappings property sheet Criteria tab

The Criteria tab allows you to specify join criteria between source objects. For example: EMPLOYEE.ID < 100

Object mappings property sheet Query tab

This query is automatically computed to figure out how to retrieve or update data in the database.

The Select query comes from the forward mapping, the Insert, Update and Delete queries come from the reverse mapping. Information contained in the Sources tab, in the Sub-Object Mapping tab and in the Criteria sub-tab are combined in these queries.

Modifying a query   If you modify the SQL query, it becomes user-defined and can no longer be automatically calculated, even when you update the mapping. You can see that a statement is user-defined because the User-Defined tool is pressed in the toolbar.

To modify the query, you should modify the expression of the Mapped to column and / or the content of the Criteria sub-tab.

You can recover the computed expression by clicking the User-Defined tool, this will remove any non calculated expression from the tab.

Select tab

For example, the following SQL queries are automatically computed to figure out how data of class instances are retrieved from or stored into the database:

♦ The **Select** tab displays a statement that allows to retrieve attribute values of class instances from the database using the class identifying attributes. For example:

```
select
      ADDRESS.LINE1 "LINE1",
      ADDRESS.LINE2 "LINE2",
      ADDRESS.CITY "CITY",
      ADDRESS.ZIPCODE "ZIPCODE",
      ADDRESS.COUNTRY "COUNTRY"
from  ADDRESS
```

Insert tab

♦ The **Insert** tab displays a statement that allows you to create an instance of the class and save it with its attributes. For example:

```
insert into ADDRESS(
      ADDRESS.LINE1,
      ADDRESS.LINE2,
      ADDRESS.CITY,
      ADDRESS.ZIPCODE,
      ADDRESS.COUNTRY)
values (
      %LINE1%,
      %LINE2%,
      %CITY%,
      %ZIPCODE%,
      %COUNTRY%)
```

Update tab

♦ The **Update** tab displays a statement that allows you to update attribute values of class instances apart from identifying attributes. For example:

```
update ADDRESS
set   ADDRESS.LINE1 = %LINE1%,
      ADDRESS.LINE2 = %LINE2%,
      ADDRESS.CITY = %CITY%,
      ADDRESS.ZIPCODE = %ZIPCODE%,
      ADDRESS.COUNTRY = %COUNTRY%
```

Delete tab

♦ 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 ADDRESS
```

SQL clauses   You can insert SQL clauses such as GROUP BY, WHERE or HAVING in
the Criteria sub-tab. You can also edit them using the Select, Insert, Update
and Delete query sub-tabs but in that case they become user-defined and are
no longer automatically calculated.

Operation object mappings properties

When you map an operation, its mapping details appear in the Mappings
pane. The mapping has the following properties that you access using the
Properties tool beside the Mapping list or using the different sub-tabs
displayed in the lower part of the Mappings pane.

Operation mappings property sheet General tab

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	Specifies the name of the parent object.
Comment	Descriptive comment for the mapping.

Operation mappings property sheet Criteria tab

The Criteria tab allows you to specify join criteria between source objects.
For example: EMPLOYEE.ID < 100

Operation mappings property sheet Query tab

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.

This query is automatically computed to figure out how to retrieve or update
data in the database. If you modify the SQL query, it becomes user-defined

and can no longer be automatically calculated, even when you update the mapping. You can tell that a statement is user-defined because the User-Defined tool is pressed in the toolbar.

You can recover the computed expression by clicking the User-Defined tool, this will remove any non calculated expression from the page.

Creating Mappings from Object's Property Sheet

You can create mappings individually for a given object from its property sheet. This way of working provides you a great flexibility for customizing the mapping between objects and guarantee a correct storage of objects in the target model.

You use the Mapping tab in the property sheet of an object to define its mappings. The Mapping tab is equivalent to the Mappings pane in the Mapping Editor.

Mapping tab toolbar

The following tools help you manage mappings for a data source in the Mapping tab of an object property sheet:

Tool	Description
	Create Mapping - See "Mapping Editor Mappings pane" on page 397.
	Delete Current Mapping - See "Mapping Editor Mappings pane" on page 397.
	Properties - See "Mapping Editor Mappings pane" on page 397.
	Launch Mapping Editor - Launches the Mapping Editor dialog box, so that you can graphically visualize the mappings of the current object together with all the mappings defined throughout a given model.

Mapping tab sub-tabs

The Mapping tab has the following properties that you access using the Properties tool beside the Mapping list or using the different sub-tabs displayed in the lower part of the Mapping tab.

You generally map objects and further define the mapping by associating their sub-objects.

Depending on the source objects you want to map with the target model objects, the following sub-tabs are displayed:

♦ Object sources sub-tab. See ""Object mappings property sheet Sources tab" on page 400"

♦ Mapped to sub-tab See ""Object mappings property sheet General tab" on page 399".

♦ Object Attributes Mapping sub-tab. See ""Object mappings property sheet Sub-Object Mappings tab" on page 400".

♦ Operations Mapping sub-tab. See ""Object mappings property sheet Operation Mappings tab" on page 401".

♦ Criteria sub-tab. ""Object mappings property sheet Criteria tab" on page 401"

♦ Query sub-tabs (Select, Insert, Update and Delete). See the ""Object mappings property sheet Query tab" on page 401".

☞ For more information about creating a mapping from the Mapping tab, see "Creating a mapping for an object" on page 406.

Modify query template

The syntax of SQL queries follows a predefined template. You can customize the query syntax by modifying the **SelectStatement**, **InsertStatement**, **UpdateStatement**, and **DeleteStatement** entries in the Profile category of the object language or extended model definition.

☞ For more information about customizing templates, see section Defining templates, in chapter Generation Reference Guide in the *Advanced User Documentation* .

Creating a mapping for an object

You generally map objects and further define the mapping by associating their sub-objects. For more information about objects and sub-objects, see the table in the ""Introduction to Model Object Mapping" on page 364" section.

❖ **To create a mapping for an object**

1. Double-click an object to display its property sheet, then click the Mapping tab.

2. Click the Create Mapping tool beside the Mapping list to create a mapping for the target object. If several data sources are available, you have to select one or several data sources in the selection list and click OK. If no data source is available, the Data Source Creation Wizard opens and allows you to specify a name, an access type and a model type (if required) for the source from which you will draw your data. Click Next and select one or more models from the list then click Finish.

   The Select an object dialog box opens to let you select a source object , then click OK. The name and code of the object are displayed in the Object Sources sub-tab and a mapping is created in the Mapping list.

3. Click the Add Objects tool to select more source object to map with the target object and click OK.
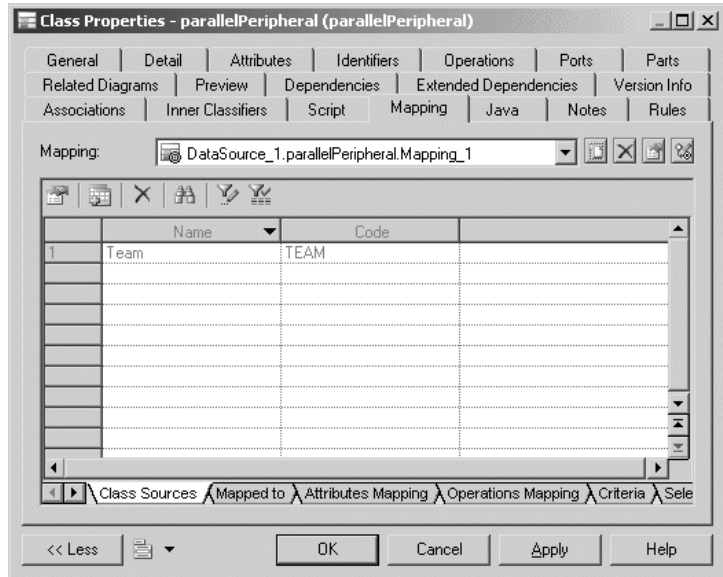
4. Click the Attributes Mapping sub-tab and click the Add Mapping tool in the toolbar.

> **Generate mapping**
> You can click the Generate Mapping tool to automatically create the mapping between sub-objects with identical name or code in the data source and the target model.

5. Select the attributes that will be mapped to sub-objects in the source objects and click OK.

   The selected objects appear in the Attribute column of the list.

6. Click the down arrow in the "Mapped to" column to select the sub-object in the data source object with which the selected sub-object will be mapped.

   You have to repeat this operation for each sub-object you want to map with the data source.
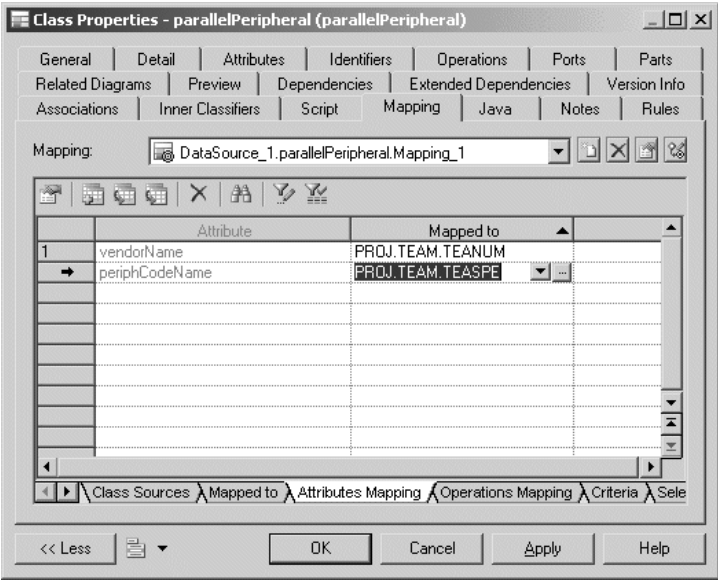
7. [If required] Click the Operations Mapping sub-tab

8. [If required] Click the Criteria sub-tab if you want to specify join criteria between the source objects.

9. [If required] Select the different queries sub-tabs to visualize or modify the SQL statements.

10. Click OK.

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.

❖ **To modify the mapping of a sub-object**

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.
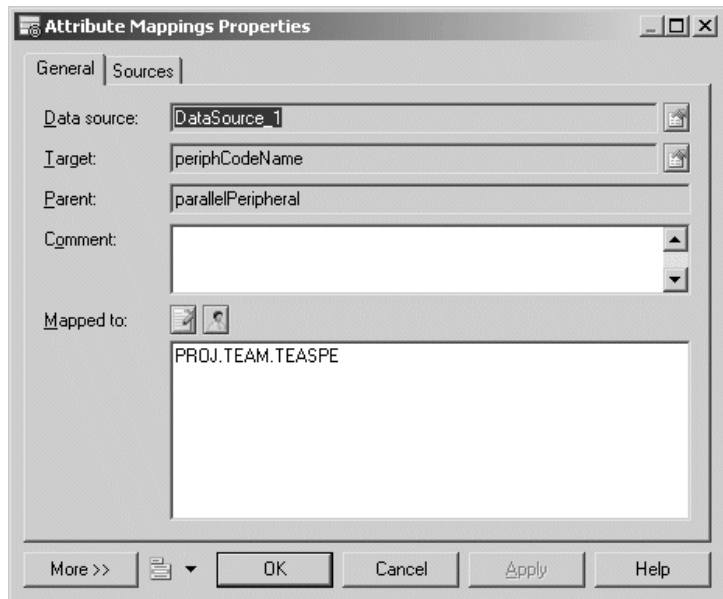
5. Click OK.

CHAPTER 13

The Generation Links Viewer

About this chapter   This document describes the PowerDesigner Generation Links Viewer
(GLV) used to visualize the generation links of models that have origin or
derived models.

Contents

Introduction to Generation Links

You can generate a model from another model of the same type or from another type. The links in the following figure show the different generations allowed among PowerDesigner models:
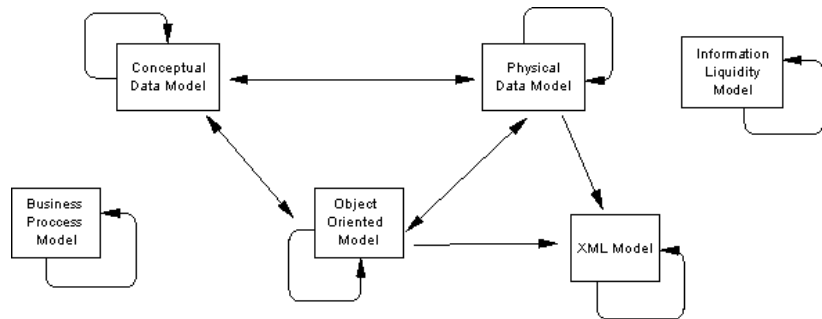
☞ For more information about intermodel generation, see the *Linking and Synchronizing Models in PowerDesigner* chapter, and also the *Generating other Models* chapter in the appropriate models user's guide.

ILM specifics

The Information Liquidity Model (ILM) also provides a graphical view of various databases (PDM) and formats, such as XML documents (XSM) and applications (OOM) together with the transformations between them. You can recover the ILM diagram corresponding to the intermodel generation by selecting Complete Generation Link or Complete All Generation Links (for recursive completeness) in the contextual menu of a data store (database, XML document or data access application). You can visualize the generation dependencies between the elements by right clicking the generation link and opening the Generation Links Viewer:
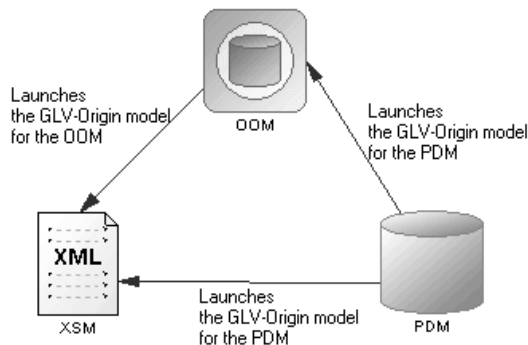
☞ For more information, see the "Defining Generation Links" section in the "Building Information Liquidity Diagrams" chapter in the *ILM User's Guide* .

Using the Generation Links Viewer

The Generation Links Viewer allows you to view all the generation links of a given model. You can identify the origin of each generated or derived object of the model. Note that you cannot edit any of these links.

> **Availability of the Generation Links Viewer**
> Links are created between models during model generation only if you select the **Save Generation Dependencies** option in the Detail tab of the generation dialog box. This option creates the link and allows tracing between the source and target objects in the Generation Links Viewer.

❖ **To view links to the model's origin model**

1. Select Tools ➤ Generation Links ➤ Origin Model.

   The Generation Links Viewer with the current model in the Target pane and the origin model in the Source pane.
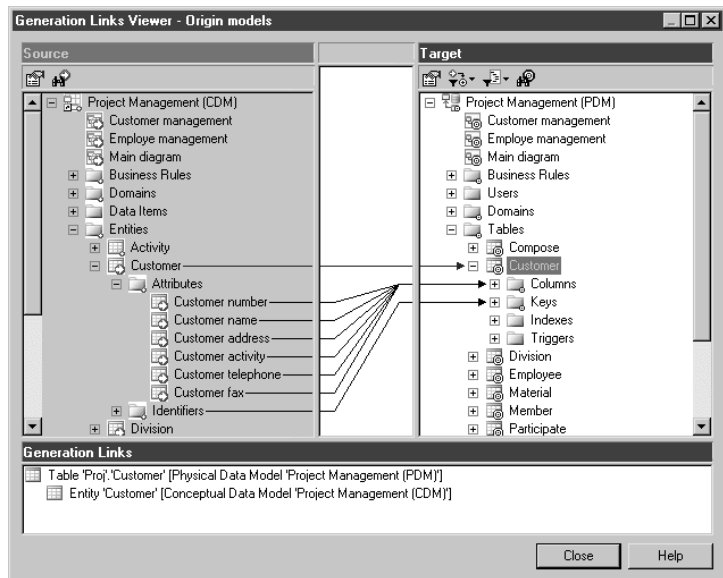
❖ **To view links to the model's derived models**

1. Select Tools ➤ Generation Links ➤ Derived Models.

   The Generation Links Viewer with the current model in the Source pane and the derived models in the Target pane.
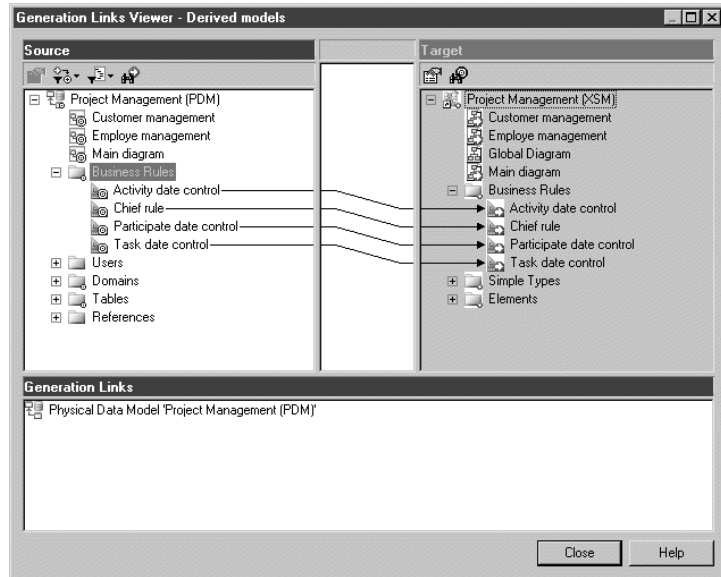
The Generation Links Viewer interface

Whatever the type of generation links you want to visualize, the Generation Links Viewer dialog box is divided in three parts:

The current model owns the generation links details.

♦ 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 the details of the generation link

Links and symbols   A non-editable link line is drawn between linked objects. 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 link details in the Generation Links 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	Linked object icon	Parent hierarchy icon
Source (green)		
Target (red)		

Generation Links Viewer toolbars

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

Generation Links pane

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 either the source or target object from the current model, and the generation link name is composed of:

♦ The object type

♦ The display name (name or code)

♦ The display name of the extremities (link object only)

♦ The object location

It uses the following format:

```
<type> '<name>' (<ext1> - <ext2>) [<location>]
```

For example:
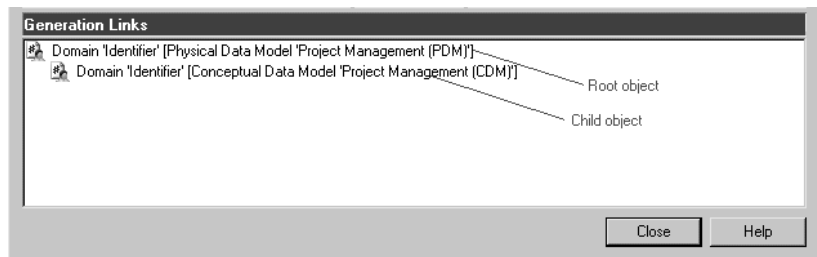
You can double click a generation link in this pane to view the properties of the source or target object.

CHAPTER 14

Model Graphics

About this chapter

This chapter describes how to manipulate the graphical display of models.

Contents

Working with Symbols

You can modify the display of individual symbols in a model including object symbols, links, graphic shapes, and lines.

Selecting symbols

You can select symbols in a PowerDesigner diagram using standard techniques.

Selected symbols display handles. For example, the illustration below shows the selection of the Member table.
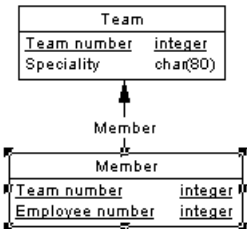
The following table lists the various ways of selecting symbols in a diagram:

To Select	Tool	How to use
One symbol		Click the Pointer tool, and then click the symbol.
Several Symbols		Click the Pointer tool. Click the first symbol to select, and then press the SHIFT key while you click additional symbols.
All symbols within a certain area		Click the pointer tool. Click and hold while drawing a rectangle around the area containing the symbols to select.
All symbols connected to a selected symbol		Click the pointer tool. Click the first symbol to select, and then select Edit ➤ Select Connected Symbols.
All symbols		Click the Grabber tool. Alternatively, select Edit ➤ Select All, or type Ctrl+A

Resizing symbols

You can resize diagram symbols:

♦ Individually – Select the symbol and then click and drag on one of its handles

♦ All at once - Click the Grabber and then click and drag on one of the handles

Bending and straightening link symbols

You can add and remove corners to and from link symbols.

❖ **To add and remove corners to and from link symbols**

1. Draw a link between two objects:
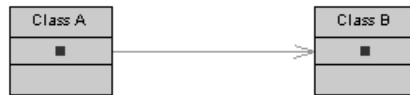
2. Press CTRL while you click a point on the line to create a handle where you want to insert a corner :

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:

Dragging a link symbol from one object to another

You can drag a link symbol from one object to another.

❖ **To drag a link symbol to another object**

1. Click a link symbol in the diagram.

2. Drag one of its end handles to a different object.

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.

❖ **To create a graphical link between any two symbols**

1. Select the Polyline tool in the Palette.

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.

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 430.
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.
Shadow [Ctrl+W]	Applies the standard shadow effect to the selected symbols. See also "Symbol format properties Shadow tab" on page 435.

Menu Item	Function
Adjust to Text [Ctrl+J]	Expands (or shrinks) the width of the selected objects to fit the length of their names. See also
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 symbols in a diagram in horizontal rows from left to right, according to the way they are currently positioned in the diagram.
	If you have selected one or more symbols in the diagram before using the auto-layout function, you will be prompted to choose whether to layout the selected symbols or all symbols.
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 422.
Disposi-tion	Automatically arranges all (or selected) symbols in the diagram. There are various forms of disposition:

For the Disposition item:

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 and 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.

Aligning selected symbols

The following tools are available from the Symbol ➤ Align submenu. Each tool acts on the symbols presently selected in the diagram:

Align submenu item	Tool	Action
Left		Aligns left borders of selected symbols with left-most 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 at least three 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 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 at least three symbols on a vertical axis

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.

❖ **To find a symbol in the diagram from an objects list**

1. Select Model ➤ *Objet type* submenu.

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.

Finding symbols from the Browser

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.

Displaying the sub-diagram of a decomposed object within its symbol

You can display the sub-diagram of a decomposed object (package, process, class, interface, node etc.) within its symbol in the diagram to have a global view of the whole sub-diagram content. To do so, you have to use the Composite View command in the contextual menu of the decomposed object. Shortcuts of a decomposed object can also have a composite view. However, external shortcuts symbol, which target model is not opened in the workspace, display an empty sub-diagram.

The decomposed icon is not displayed in Composite view mode.

Process symbol in Composite View mode:
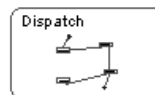
Activity symbol in Composite View mode:
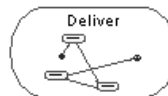
Class and interface symbols in Composite View mode:
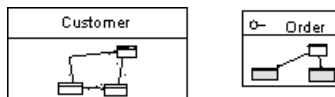
As for the package symbol in Composite View mode, its name is isolated in

the top-left header:
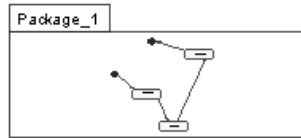
If you double click the composite view, you automatically open the decomposed object sub-diagram. However, if you double-click the name of the decomposed object, you automatically open its property sheet.

It is not possible to edit the sub-diagram in Composite View mode.

❖ **To display the sub-diagram of a decomposed object within its symbol**

1. Right-click the decomposed object symbol, and select Composite View from the contextual menu.

   The decomposed object symbol is expanded in order to display the whole diagram content. You may have to resize the decomposed object symbol to visualize its content. You can re-select the Composite View command to come back to the previous view of the decomposed object symbol.

   ☞ For more information about using the composite view in the different module, see each appropriate module user's guide.

Displaying a set of associated sub-objects within a parent symbol

A sub-object is an object associated with a parent object. It has no symbol in the diagram but displays in its parent symbol. In the following example, the *Team number* and the *Employee number* columns have no symbol in the diagram but display in their parent *Proj.Team* table symbol:

You define a sub-object from its parent property sheet. All the created sub-objects display by default in the parent symbol with a default font defined in the display preferences.

☞ For more information on setting a default font, see "Modifying Global Display Preferences" on page 448.

You can decide to remove sub-objects from the symbol by clearing their corresponding check box in the display preferences for the parent object.

☞ For more information on setting display preferences for an object, see

the different module user's guides.

Sub-Objects tab

However, you can also decide to only display a set of sub-objects using the Sub-Objects tab of the Symbol Format dialog box. It also allows you to define a specific font for each sub-object.

This tab is accessible from the symbol contextual menu or from the Symbol menu.

It is not available from the Symbol Format dialog box that you access from the display preferences. It is neither available from the Symbol Format dialog box that you access from the Custom Symbol of the Profile editor, same when you select more than one symbol in the diagram.

Moreover, this tab is never available when the Comment option is selected in the display preferences to allow the display of the comment in symbols instead of the sub-objects. For more information see, "Object comment" in the Managing Objects chapter.

The Sub-Objects tab contains a sub-tab for each sub-object type selected in the display preferences. Each sub-tab content reflects the corresponding tab in the parent property sheet. By default, all the sub-objects D (for Displayed) check box is selected and the Specific Font column is empty as all sub-objects have the default font.

The following example shows the Sub-Objects tab in the Symbol Format dialog box for the table *Proj.Team* as *Columns* and *Indexes* were selected in the Display Preferences dialog box.
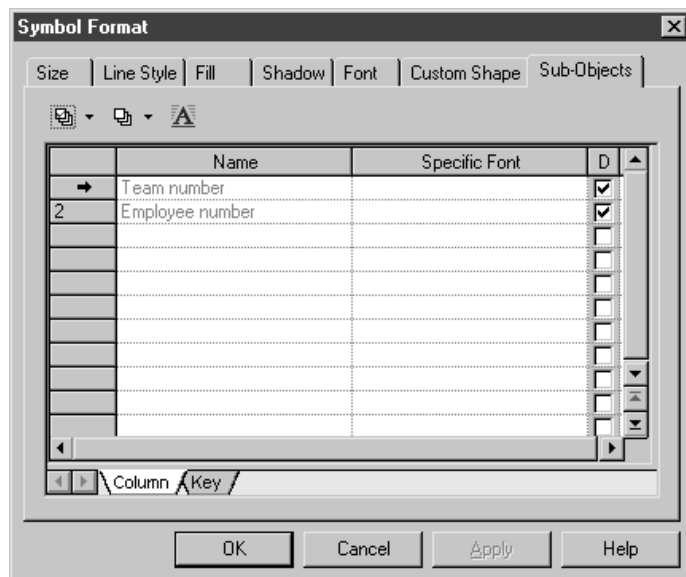
You can use the following tools to manage the display of sub-objects in the symbol:

Tool	Description
▣ ▾	Selects the D (Displayed) check box for all sub-objects in the active page or in all pages at once.
▢ ▾	Clears the D (Displayed) check box for all sub-objects in the active page or in all pages at once.
A	Opens the Font dialog box to allow you to define a specific font for the selected sub-object.

For each sub-object in a sub-tab, you can decide:

♦ To display it or not in the parent symbol.

♦ To apply to it a specific font that displays in the parent symbol.

❖ **To select a sub-object to display in the parent symbol**

1. Right-click a symbol in the diagram and select Sub-Objects Format.

2. Select a sub-object in the list and click the D check box.

   The sub-object you have selected is displayed in the parent symbol when you click Apply.

   When only some sub-objects in a sub-tab are selected for display in the list, ellipses are displayed in the parent symbol to indicate that more data can be displayed.

❖ **To define a specific font for a sub-object**

1. Right-click a symbol in the diagram and select Sub-Objects Format.

2. Select a sub-object in the list and click the Select Font tool to open the Font dialog box and define a font.

3. Click OK to close the Font dialog box.

   The font settings are displayed in the Specific Font column beside the selected sub-object. If the D check box is selected for the sub-object, the font settings are applied to it in the diagram when you click Apply.

   You can also open the Font dialog box by clicking the Ellipses button in the Specific Font column beside the selected sub-object or even directly define a font for a sub-object in the Specific Font column. Beware that if

you type wrong font settings, they are replaced with the display preferences default font.

You can also multi-select sub-objects in the list to apply to them the same specific font.

In the following example, both columns have a specific font but only the *Team number* will be displayed in the parent symbol:
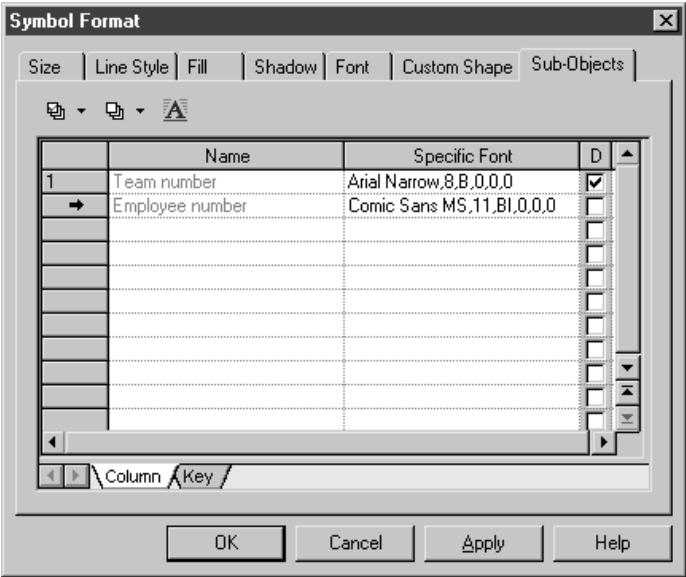
In the *Proj.Member* parent symbol, only the *Team number* column is displayed and ellipses indicate that more data can be displayed:
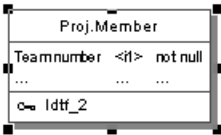
All and Limit options in display preferences

For objects that have "All" and "Limit" options in their display preferences (i.e. CDM entity for attributes or OOM state for actions), the rule is to preserve your selections in the display preferences together with taking into account the sub-objects you have selected to be displayed in the parent symbol.

For example, if 7 attributes are selected to be displayed in the entity Sub-Objects tab whereas the display preferences for entity attributes has a limitation at 5, only the 5 first attributes selected in the Sub-Objects tab are displayed in the entity symbol.

☞ For more information on All and Limit options in display preferences,

see the PDM, OOM, CDM and XML user's guides.

Symbol Format Properties

You can change many aspects of the display of one or more symbols, by changing the properties in its Symbol Format dialog box.

You can open the Symbol format window in any of the following ways:

♦ Right-click the symbol and select Format from the contextual menu

♦ Select one or more symbols and then press CTRL+T

♦ Select one or more symbols and then select Symbol ➤ Format

The Symbol Format window allows you to specify the display format of object symbols. From this window, you can modify the display format of:

♦ All symbols - see "Modifying Global Display Preferences" on page 448

♦ Individual symbols – see "Working with Symbols" on page 418

The following sections list the tabs and properties available in this window.

Symbol format properties Size tab
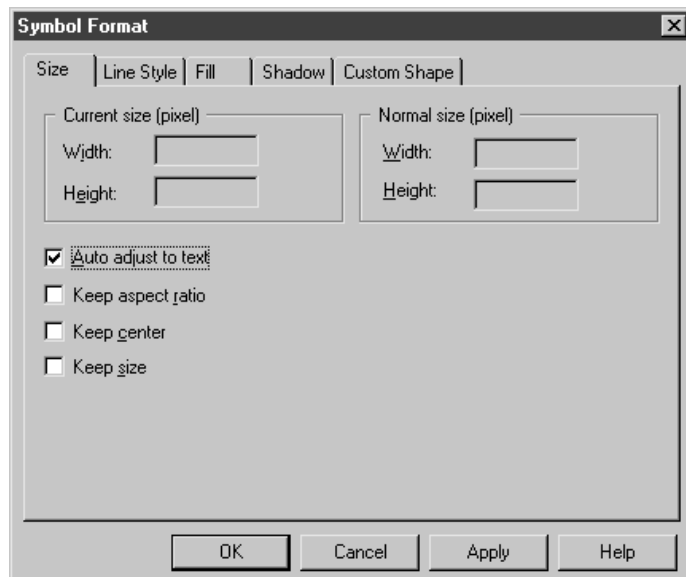
The Size tab controls the size of the symbol and how the size can be manipulated. You can specify the following properties:

Property	Description
Current size: Width	Specifies an exact width (in pixels) for the symbol.
Current size: Height	Specifies an exact height (in pixels) for the symbol.
Auto-adjust to text [CTRL+J]	[Default] Specifies that the width of the symbol varies automatically to display its full name, from the default width up to 254 characters (mutually exclusive with Keep size).
	Adjusting symbols to text resizes symbols so that they display all the text they contain. For example, a rectangle adjusts its size to fit closely around its text. A table symbol adjusts its size to display full column names, or column codes, up to the truncation or word wrap length. You can define default adjustment to all symbols. However, if you manually resize a symbol, the Adjust to Text functionality is automatically disabled for that symbol.
Keep aspect ratio	Maintains the ratio of height to width when resizing the symbol.
Keep center	Maintains the center position of the symbol when resizing it.
Keep size	Prevents the resizing of symbols both manually and through Auto-adjust to text (mutually exclusive with Auto-adjust to text).

Symbol format properties Line Style tab
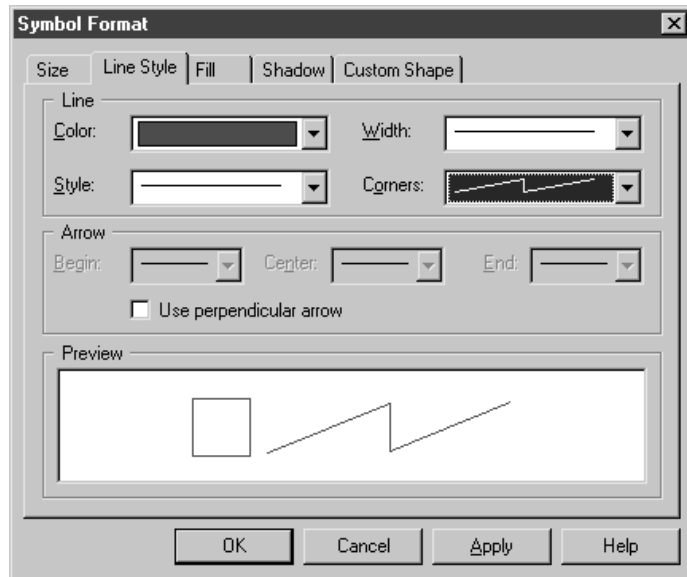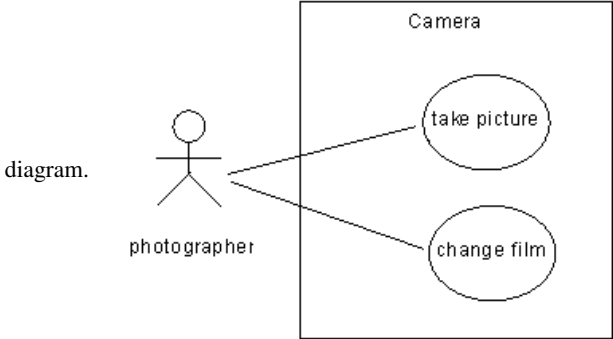
The Line Style tab controls the color, size and format of lines (for link and other one-dimensional symbols) and borders (for two-dimensional symbols, such as classes or tables). You can modify the line style of any symbol in the model. However, in case of fundamental changes in the design semantics of a link symbol (such as a PDM reference for example), it automatically reapplies the default format for the link. You can specify the following properties:

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.

Symbol format properties Fill tab
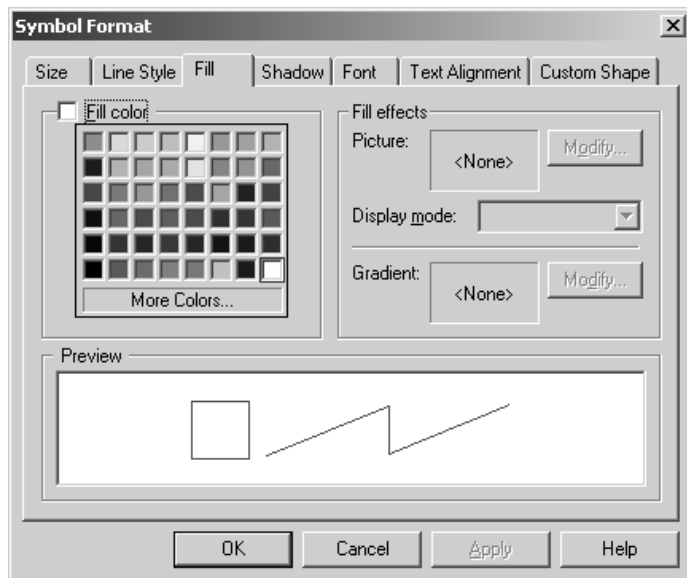
The Fill tab controls the color, content, and effects for symbol filling. You can specify the following properties:

Fill effects	Description
Fill color	The Fill color check box is displayed when you want to select fill display preferences to the following object symbols: ♦ Free symbols ♦ Packages ♦ Interaction fragments ♦ Swimlanes When cleared, this check box allows you to: ♦ Create a transparent symbol ♦ Prevent the selection of the object symbol in its center (only borders can be selected) This feature can be useful when you wish to use an "in-box" UML representation, for packages for example. You create a transparent symbol by clearing the Fill color check box in the Fill tab of the Format dialog box. This check box is only displayed for the above mentioned objects. You can apply transparency to graphic shapes. Transparent shapes become a sort of skeleton as they take the color of the diagram background and can only be selected by clicking on their borders. You apply transparency to graphic shapes by clearing the Fill color check box in the Fill tab of the Format dialog box. You can apply transparency to the rectangle you draw to represent the system in the use case diagram.
Picture	Allows you to select a graphic file to display within the symbol when you click the Modify button. The Display Mode list is automatically enabled

Fill effects	Description
Display Mode	Allows you to select a location for the picture displayed within the symbol (Center, Top Left, Right etc.)
Gradient	Allows you to select gradient fill options for the symbol background if you click the Modify button. You can choose among he following options: Start and End colors, End Color Luminosity, Shading Style and Gradient Mode

Symbol format properties Shadow tab
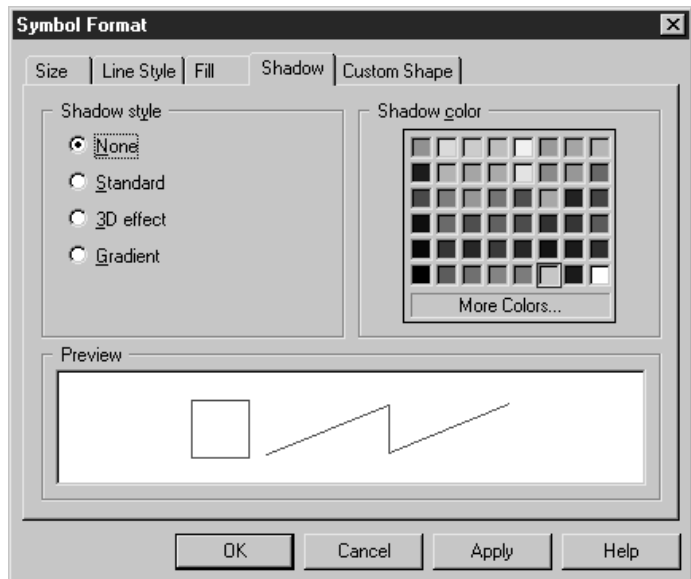
You can add a standard shadow, a 3D effect shadow or a gradient shadow to objects in a diagram.

Symbol format properties Font tab
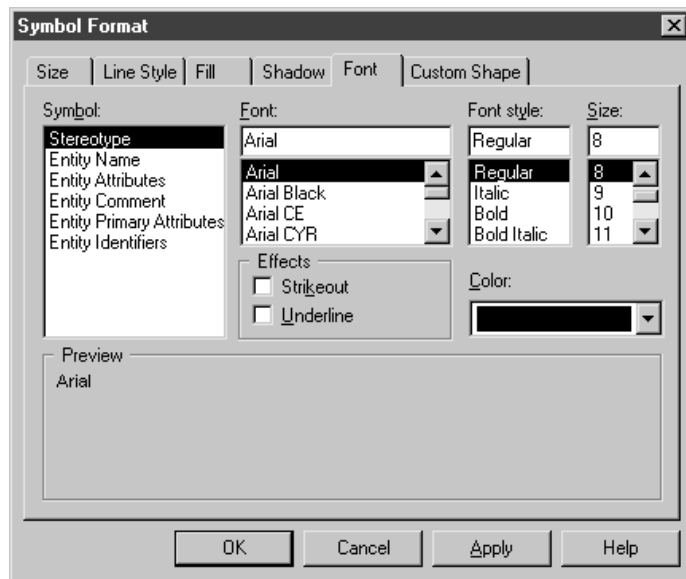
You can define the display preferences for the font, size, style, and color of text associated with symbols in the model. When you modify font preferences, they apply to all existing and new symbols.

Symbol format properties Text Alignment tab
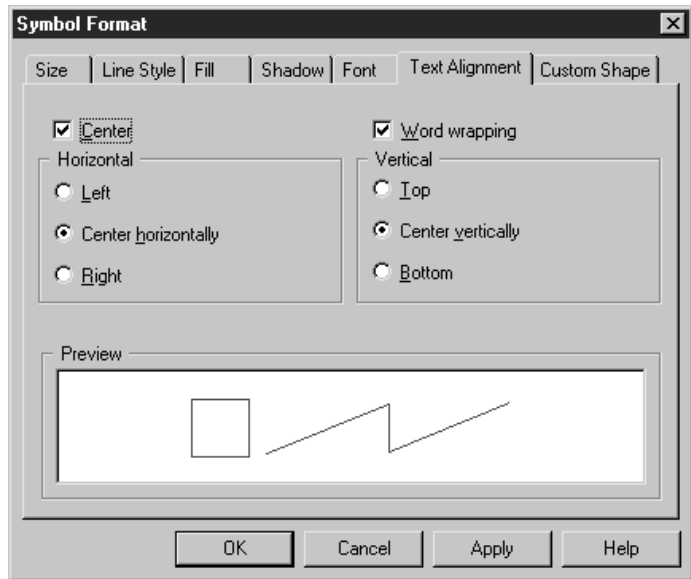
You can define the alignment of text in the following graphic shapes: rectangles, ellipses, rounded rectangles, and polygons.

You can align free text vertically and horizontally. If the text is in a shape, it aligns with the borders of the shape. If the text is not in a shape, it aligns with text handles.

You cannot change the alignment of text associated with an object in the model.

For each graphic shape, you can define the following text alignment parameters:

Parameter	Description
Center	Centers the text horizontally and vertically
Word wrapping	Displays text in the space taken by the graphic shape
Horizontal left	Aligns text to the left according to a horizontal axis
Horizontal right	Aligns text to the right according to a horizontal axis

Parameter	Description
Horizontal center	Centers text according to a horizontal axis
Vertical top	Aligns text to the top according to a vertical axis
Vertical bottom	Aligns text to the bottom according to a vertical axis
Vertical center	Centers text according to a vertical axis

When using the RTF text mode for free text, note that the Text Alignment format (except for the Vertical option) is disabled in the Symbol Format dialog box.

Symbol format properties Custom Shape tab
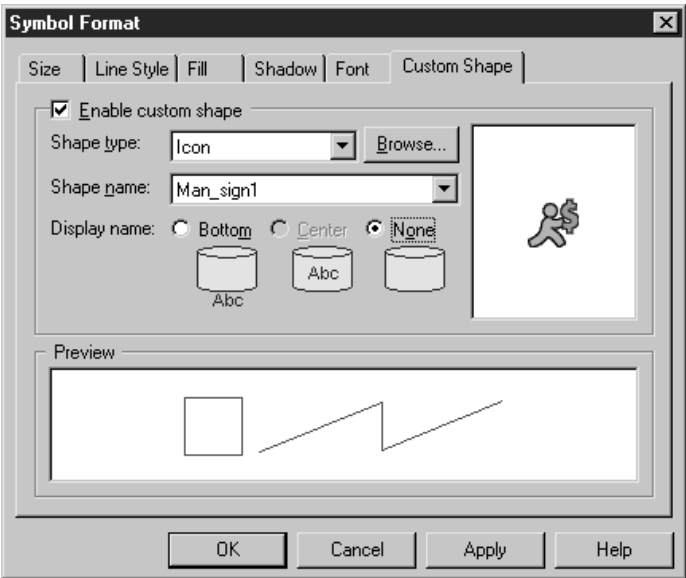
The Custom Shape tab allows you to define a new symbol shape for most non-link symbols. You can specify the following properties:

Property	Description
Enable custom shape	Enables or disables the customization of a symbol shape

Property	Description
Shape Type	Specifies the type of shape to be used. You can choose between: ♦ 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. Use the Browse button to select shapes.
Display name	Allows you to define where the symbol name should appear (Center option is not available for icon shapes)

Working with Decorative Symbols

Decorative symbols have no technical meaning in your diagram, but help make it more readable. You can use them to surround parts of a model, for example, to distinguish domains of activity.

❖ **To draw a rectangle**

1. Select the Rectangle tool in the Palette.

2. Click at the point in the diagram where you want to insert one corner of the rectangle, hold the mouse button, and drag to where you want to place the alternate corner.

3. Release the mouse button to create the rectangle.

4. [optional] Click on and drag a handle to resize the rectangle.

You can create the following kinds of shapes:

Symbol	Tool	Additional Instructions
Line	╲	[none]
Arc	╮	[none]
Rectangle	▢	[none]
Square	▢	Press CTRL while drawing.
Ellipse	◯	[none]
Circle	◯	Press CTRL while drawing.
Rounded Rectangle	▢	[none]

Sym- bol	Tool	Additional Instructions
Rounded Square		Press CTRL while drawing.
Poly-line		Release the mouse button at each point where you want to create a corner. Right click to finish.
Poly-gon		Release the mouse button at each point where you want to create a corner. Right click to finish and close the polygon.
Title Box		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.

Modifying the type of corners

After creating certain shapes, you can modify the type of corner by right-clicking and selecting Format ➤ Line Style and selecting a type in the Corners list. You can also define a default style for corners in the display preferences of graphic shapes.

Working with Text

Free text is text that is not associated with an object in the model. For example, a note that you type in a rectangle is free text, (while a column name is not). You can insert free text in your model independently of any shape, and can select and move it like any symbol.

❖ **To create free text**

1. Click the Text tool in the palette, and then click in the diagram where you want to insert the text. A symbol saying <Default text> is created.

<div align="center">

<Default
text>

</div>

2. Double-click the text to open a text input window.
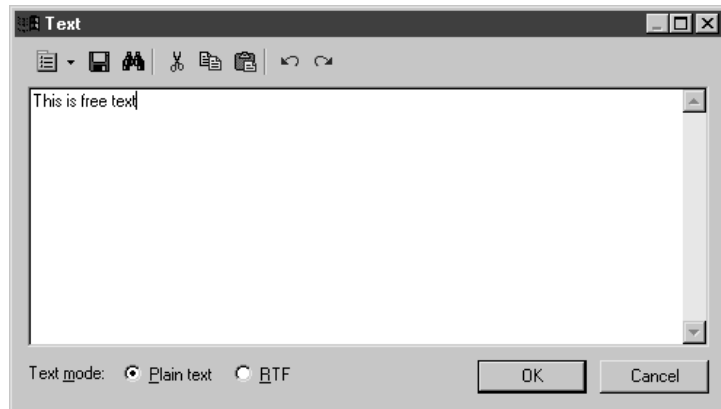
3. Type text in the window, and then click OK.

Inserting text into shapes

You can insert text inside any graphic shape. This text is attached to the shape and will move with it. If you change the fill color of the shape, the text also changes.

❖ **To insert text in a shape**

1. Double-click a shape in the diagram to open a text input window.

2. Type text in the window, and then click OK.

Assigning text to free links

You can assign text to lines and polylines in the diagram to document links between object symbols. The text is attached to the line. If you move the line, the text moves with it.

You can format the text attached to free links using the Symbol Format dialog box available from the contextual menu of the link. The same format is applied to each text zone.

❖ **To assign text to free links**

1. Double-click a line or a polyline in the diagram to open the Link Symbol Text dialog box to the Center Text tab. There are also tabs that allow you to add text to the Source and Destination ends of the line.

2. Type the appropriate text in the appropriate tabs and then click OK

Formatting free text

Free text and text in graphic shapes support two modes. These are selectable at the bottom of the text input window:

♦ Plain text - text formatting is controlled by the display preferences for Free Symbols

♦ RTF (Rich Text Format) – text formatting can be controlled directly in the text input window

❖ **To work with RTF**

1. Create a free text object and double-click it to open a text input window

2. Select the RTF radio button at the bottom of text input window to enter RTF editing mode.

3. Use the RTF tools listed below, or click the Format tool to open the Text Format window. This window has three tabs:
   ♦ Font – including style, size, effects, color and background
   ♦ Paragraph- including indentation, spacing, and alignment
   ♦ Tabs – including positioning and alignment

4. When complete, click OK to return to the diagram

Tool	Function
	Editor Menu
	Launch External Editor
	Save
	Print
	Find
	Cut
	Copy
	Paste
	Undo
	Redo
	Open Format Menu
	Bold
	Italic
	Underline
	Align Left

Tool	Function
	Align Center
	Align Right
	Bullets

Viewing Diagrams

This section describes methods for viewing your diagram.

Change View	Tool	Key	How to use
Zoom In		F6	Select the Zoom In tool and click anywhere in the diagram. The point clicked on is centered. Alternatively: ♦ Select View ➤ Zoom In ♦ Hold down Ctrl and turn your mouse scroll wheel away from you
Zoom Out		F7	Select the Zoom Out tool and click anywhere in the diagram. The point clicked on is centered. Alternatively: ♦ Select View ➤ Zoom Out ♦ Hold down Ctrl and turn your mouse scroll wheel towards you
Zoom In to a particular area		[none]	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.
View the whole diagram		F8	Double-click the Global View tool or select View ➤ Global View.
View actual size	[none]	F5	Select View ➤ Actual Size.
View the current (printable) page	[none]	Ctrl+F10	Select View ➤ Page View ➤ Current Page.
View all pages that contain symbols	[none]	F10	Select View ➤ Page View ➤ Used Pages.

Change View	Tool	Key	How to use
View all pages in the diagram	[none]	[none]	Select View ➤ Page View ➤ All Pages.
Center on selected symbols	[none]	[none]	Select View ➤ View Selection.
Return to previous view	[none]	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.
Go to next view	[none]	Shift+F9	Select View ➤ Next View.
Refresh View		Shift+F5	Select View ➤ Redisplay.

Modifying Global Display Preferences

You can set default display preferences for the symbols in your model. Using the Display Preferences dialog box you can:

♦ Define preferences for the current diagram

♦ Apply display preferences to a selection of diagrams

♦ Restore default display preferences

♦ Set new default display preferences

❖ **To set diagram symbol display preferences**

1. Open the diagram for which you want to change the display preferences. Select Tools ➤ Display Preferences to open the Display Preferences dialog box:
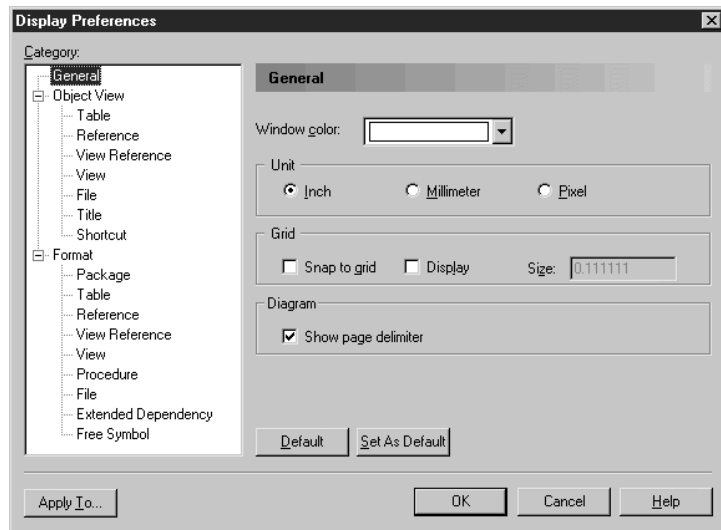
2. Select categories in the left hand panel and set preferences in the right-hand panel. There are three broad categories of display preferences:
   ♦ General display preferences – which control the format of the diagram itself (see "General display preferences" on page 450)
   ♦ Object view display preferences – which control what information is displayed on an object's symbol (see "Object view display preferences" on page 452)
   ♦ Format display preferences – which control the look (color, line style, font, and others) of the symbol (see "Format display preferences" on page 453)

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.

3. [optional] Click the Apply To button to open the Select Diagrams dialog box:
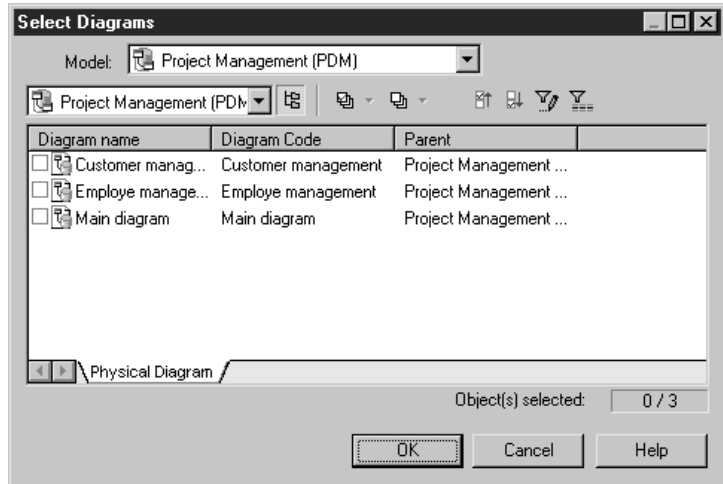
Select the checkboxes of the diagrams to which you want to apply your changes and click OK. You will be asked whether you want to apply your changes to all the symbols in the selected diagrams. If you click:

♦ Yes – All existing and new symbols will reflect your changes to the Format, General and Object View display preferences.

♦ No – New symbols only will reflect your changes. Only the General and Object View 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 box.

4. On the Display Preferences dialog box, click OK. The Changing Formats dialog box will open:
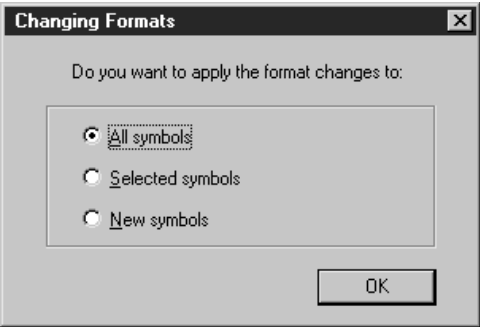
The choices in this dialog apply only to the present diagram and will not override your choices made in the Select Diagrams dialog box.
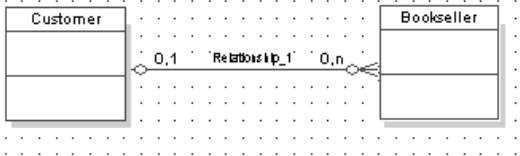
5. Click OK to apply your changes

General display preferences
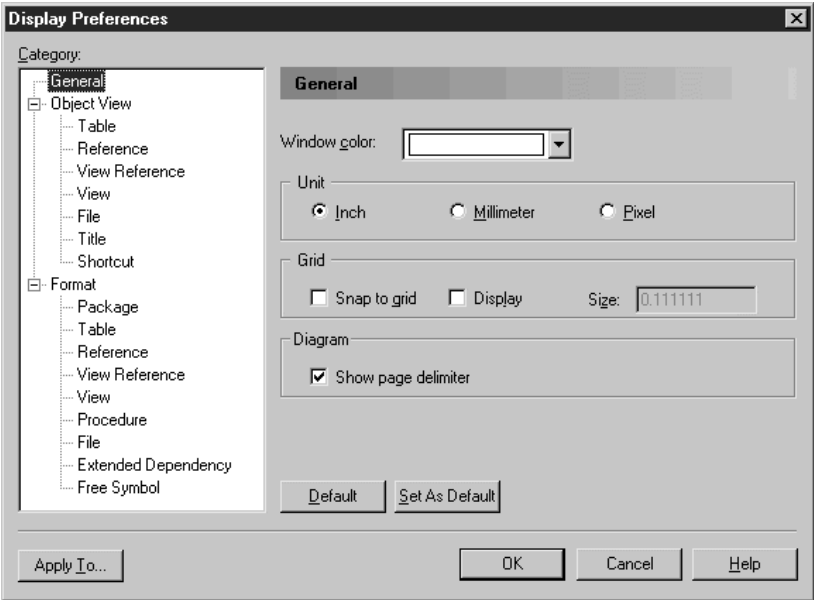
The General category allows you to control the general look of the diagram. You can set the following preferences:

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.

Property	Description
Snap to grid	Automatically aligns all new objects that you create in the diagram window to the anchor points of a grid. You can specify the size and visibility of the grid with the Display and Size options. All existing objects are aligned automatically when you move them within the diagram.
Display	Activates a grid in the background of the diagram.
Size	Determines the number of anchor points per square inch in the grid.
Show page delimiter	Shows limit of pages on the diagram background
Organization unit swimlane	[BPM only] Specifies that organization units will be displayed as swimlanes. When this options is selected, you may also specify the direction of the swimlanes, in choosing between Horizontal and Vertical.
Orientation	[XSM only] Specifies the direction in which the content of global objects will expand. You can choose between Horizontal and Vertical.

Object view display preferences
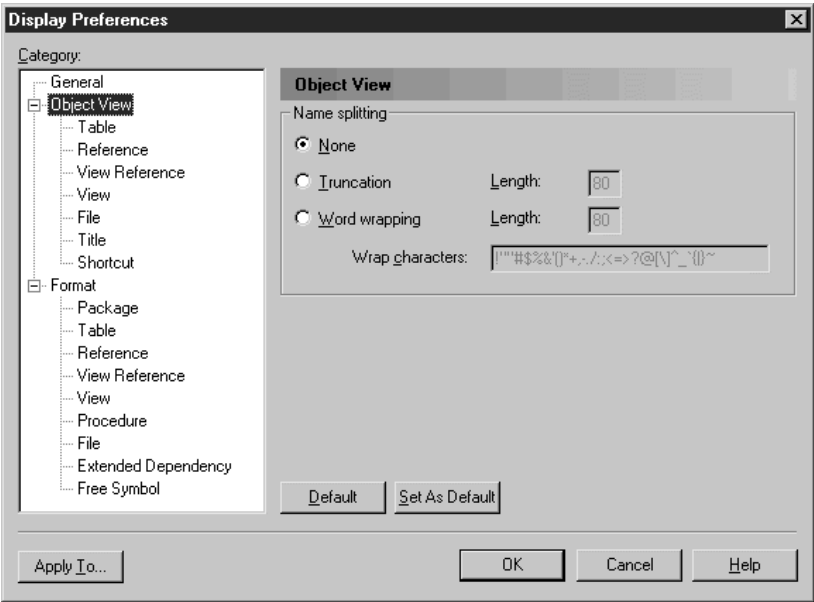
The Object View category allows you to control what information is displayed on the symbol. By clicking on the category itself, you can set the following preferences:

Preference	Description
Name Splitting	Controls the display of names on symbols. 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 (up to the indicated length)
Wrap characters	Set of characters after which word wrapping is possible.

You can modify the information displayed on particular kinds of objects by clicking on the object in the list. For details of the information specific to a particular object, see the relevant model user's guide.

Format display preferences
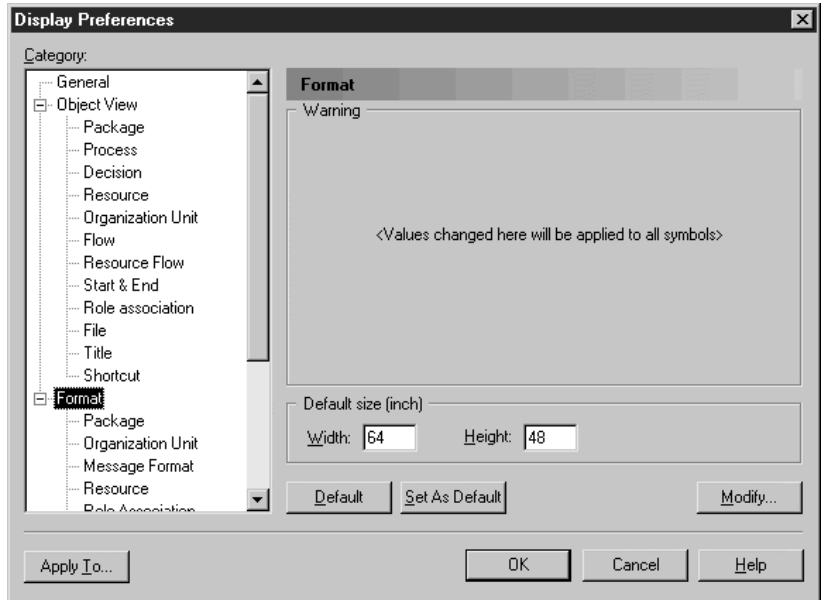
The Format category allows you to control the look of object symbols. By clicking on the category itself, you can set the following preferences:

Prefer-ence	Description
Default Size	Horizontal and vertical size of a symbol containing text (measured in units of 1/7200 of an inch)

Format and preview
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

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.

When you print a diagram, you do not print detailed information about the model objects. To do this, you need to create a model report.

❖ To print a diagram

1. [optional] Select certain 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.
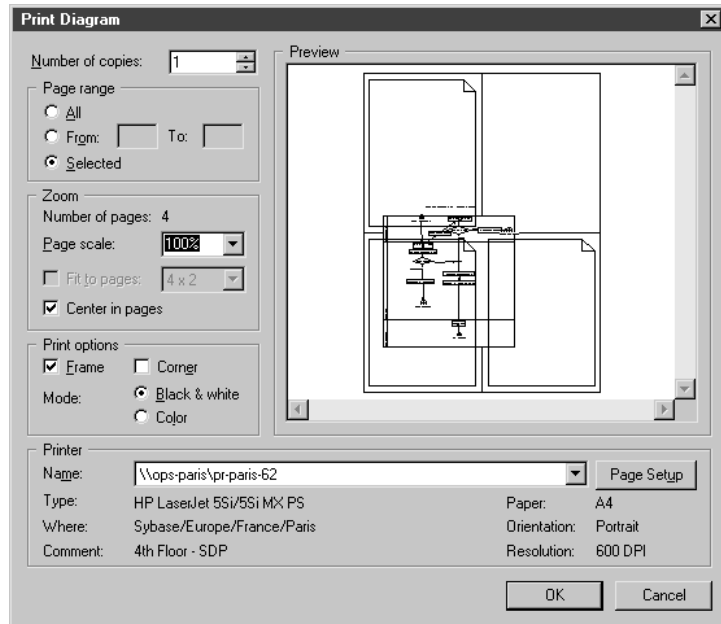
3. [optional] Specify the pages to print in the Page range groupbox or by clicking in the Preview pane (see "Print Diagram options" on page 455). Only pages with an overlaid page frame will be printed.

4. [optional] Specify a page scale or set of pages to fit to (see "Print Diagram options" on page 455). 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 456).

6. Click OK to start printing.

Print Diagram options

When you print a diagram, you can select the following print options from the Print diagram dialog box. It also displays a preview of the selection you want to print:

Option	Description
Number of copies	Number of copies you want to print. You can type it or use the arrows.
Page range	Specifies which pages to print. You can choose between: ♦ All - Prints all the pages of the diagram. ♦ From / To - Prints a range of pages covered by the diagram. The corresponding pages are framed in the Preview window. ♦ Selected - Prints the pages you select in the Preview window.
Number of pages	Displays the number of pages required based on your zoom options.
Page scale	Specifies the scale at which the diagram will be printed.
Fit to pages	Reduces the scale of the diagram to print it on the number of pages specified, for example 1x3. The Number of pages option is automatically updated and the Center in pages option is unavailable. These parameters are not saved and your diagram stays unchanged in the diagram window. If you want to apply these changes to your diagram, you should use the Symbol ➤ Fit to Page command from the menu bar and the File ➤ Page Setup for paper orientation.
Center in pages	Centers the diagram in the pages that it covers. The Fit to pages option is automatically unavailable.
Frame	Solid line border around graphic on all pages.
Corner	Specifies the printing of "crop marks" in each corner to help align multiple pages.

Option	Description
Mode:	Specifies whether the diagram will be printed in Black & White or Color.
Printer: Name	Name of the printer. Select a printer from the Name list. Click the Page Setup button to modify the current printer parameters.

Page Setup options

The Page Setup dialog box allows you to:

♦ Modify standard printer parameters, such as Paper, Orientation and Margins.

♦ Add various kinds of information to the Header, Footer and Page of your printed diagram.

❖ **To customize your printed diagram footer**

1. Open the Print Diagram dialog box (see "Printing Diagrams" on page 454) and click the Page Setup button.

2. Click the arrow tool to the right of the Footer field and select Current Page from the list. Click the tool again, and select Last Page from the list. The field will now contain the following two PowerDesigner variables:

   %PAGE% %TOTALPAGE%

1. Edit the text in the field by adding some additional, hard-coded text, such as the following:

   Page: %PAGE% of %TOTALPAGE%

1. [optional] Click the Apply To button to apply page setup options to other diagrams in the current model or in other models opened in the workspace:

2. Click OK to return to the Print Diagram dialog box, and then click OK. Your footers will print as follows:

   Page: 1 of 16
   Page: 2 of 16

etc

Importing and Exporting Model Graphics

A model can incorporate external graphic files in one of several formats. The imported images are stored in the model when it is saved, and are present the next time you open the model.

❖ **To import an image into the model**

1. Select Edit ➤ Import Image to open Windows Open dialog box.

2. Select a file type and file and then click OK to display the image in the diagram

You can import the following types of graphic files:

File type	Extension
Windows MetaLayout format	WMF
Enhanced metafile	EMF
Bitmap	BMP, DIB, RLE
JPEG Compliant	JPG, JPEG
Portable Network Graphic	PNG

You can also export model graphics.

❖ **To export model graphics**

1. Select one or more symbols to export.

2. [optional] Select Edit ➤ Export in Color in order to retain color when exporting.

3. Select Edit ➤ Export Image to open a Windows Open dialog box.

4. Select a format in the Files of Type listbox, type 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
Scalable Vector Graphics	SVG

Copying to the Clipboard in color
If you cut or copy selected symbols to the Clipboard, you must select Edit
➤ Export in Color to copy the symbols in color.

Index

M