SYBASE®

Conceptual Data Model

Sybase® PowerDesigner®

12.5

Windows

Part number: DC38084-01-1250-01
Last modified: April 2007

Contents

About This Manual **v**

1 Conceptual Data Model Basics **1**
What is a CDM? 2
Creating a CDM 4
Defining packages in a CDM 6
Customizing the CDM Environment 8

2 Building Conceptual Data Models **21**
Conceptual Diagram Basics 22
Data Items (CDM) 24
Domains (CDM) 28
Entities (CDM) 36
Attributes (CDM) 39
Identifiers (CDM) 44
Relationships (CDM) 47
Associations and Association Links (CDM) 62
Inheritances (CDM) 68
Check Parameters (CDM) 77

3 Working with Conceptual Data Models **81**
Checking a CDM 82
Opening a PAM into a CDM 94

4 Generating from a Conceptual Data Model **97**
Generation basics 98
Generating a Conceptual Data Model from a Conceptual Data Model 99
Generating a Physical Data Model from a Conceptual Data Model 102
Generating an Object Oriented Model from a Conceptual Data Model 116

Index **121**

About This Manual

Subject

This book describes the PowerDesigner Conceptual Data Model data modeling environment. It shows you how to do the following:

- Build a Conceptual Data Model (CDM)
- Create and use business rules and other model objects
- Verify the model and import an ERwin model
- Generate other models from the CDM

Audience

This book is for anyone who will be building data models with the PowerDesigner Conceptual Data Model. Although it does not assume you have knowledge about any particular topic, having some familiarity with relational databases, SQL, and design methodology is helpful. For more information, see the Bibliography section at the end of this chapter.

Documentation primer

The PowerDesigner modeling environment supports several types of models:

- **Conceptual Data Model (CDM)** to model the overall logical structure of a database, 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 Conceptual Data Model. For information on other models or aspects of PowerDesigner, consult the following books:

General Features Guide To get familiar with the PowerDesigner interface before learning how to use any of the models.

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.

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

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

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.

Bibliography

Data Modeling Essentials

Graeme Simsion, Van Nostrand Reinhold, 1994, 310 pages; paperbound; ISBN 1850328773

Information Engineering

James Martin, Prentice Hall, 1990, three volumes of 178, 497, and 625 pages respectively; clothbound, ISBN 0-13-464462-X (vol. 1), 0-13-464885-4 (vol. 2), and 0-13-465501-X (vol. 3).

Celko95

Joe Celko, Joe Celko's SQL for Smarties (Morgan Kaufmann Publishers, Inc., 1995), 467 pages; paperbound; ISBN 1-55860-323-9.

CHAPTER 1

Conceptual Data Model Basics

About this chapter

This chapter presents the Conceptual Data Model (CDM) and explains the role of conceptual modeling in the design process.

Contents

Topic:	page
What is a CDM?	2
Creating a CDM	4
Defining packages in a CDM	6
Customizing the CDM Environment	8

What is a CDM?

A Conceptual Data Model (CDM) represents the overall logical structure of a database, which is independent of any software or data storage structure. It gives a formal representation of the data needed to run an enterprise or a business activity, and will often contain data objects not yet implemented in the physical database.

When designing a database, the design process normally starts at the conceptual level, where you do not need to consider the details of actual physical implementation.

A CDM allows you to:

- Represent the organization of data in a graphic format to create Entity Relationship Diagrams (ERD)
- Verify the validity of data design
- Generate a Physical Data Model (PDM), which specifies the physical implementation of the database
- Generate an Object-Oriented Model (OOM), which specifies an object representation of the CDM using the UML standard
- Generate another CDM, to create another model version in order to represent different design stages

☞ For more information on ERD, see James Martin's *Information Engineering* , Prentice Hall, 1990 (three volumes).

Logical model

A logical model allows you to design the database structure and perform some database denormalization actions.

In PowerDesigner, you design a logical model using a Physical data Model (PDM), and specifying the "Logical Model" DBMS. The "Logical Model" DBMS is a PDM containing standard database objects, but without DBMS-specific physical options and generation capabilities.

You can use a logical model as an intermediary step in the database design process between the conceptual and physical designs:

- Start with a CDM containing entities, attributes, relationships, domains, data items and business rules
- Generate a logical model (PDM with the <Logical Model> DBMS). Create indexes and specify FK column names and other common features

- Generate a set of PDMs, each targeted to a specific DBMS implementation

If need be, you may develop the CDM in several design steps starting from a high level model to a low level CDM.

This design process allows you to keep everything consistent in a large development effort.

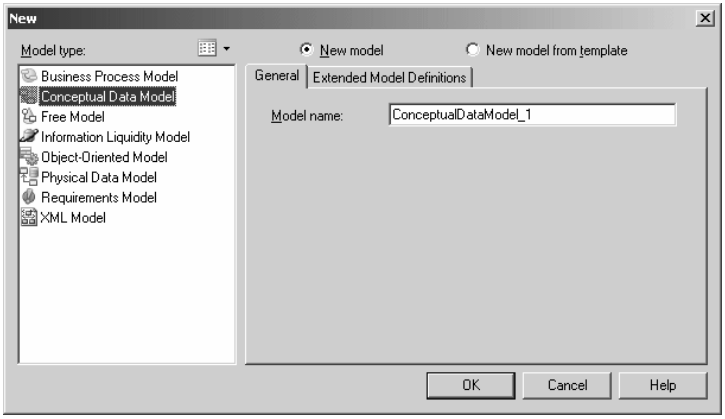
☞ For more information on the physical data model, see the *PDM User's Guide* .

Creating a CDM

You can create a new CDM from scratch, by importing a Process Analyst Model (.PAM) or an ERwin model (.ERX), or by generating it from a CDM, PDM, or OOM.

❖ To create a new CDM

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



2. Select Conceptual Data Model in the list of model types.
3. Select one of the following radio buttons:
   - New model – Creates a new, standard, model.
   - New model from template – Creates a model from a model template. 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.

     ☞ For more information about model templates, see the “Creating a model” section in the “Managing Models” chapter in the *General Features Guide* .
4. Type a model name in the Model name box. 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.
5. [optional] If you want to attach one or more extended model definitions to your model, click the Extended Model Definitions tab, and select the extended model definitions of your choice.

☞ For more information on attaching extended model definition to a model, see the “Extended Model Definitions Reference Guide” chapter in the *Advanced User Documentation* .

6. Click OK to create the new CDM in the current Workspace.

Demo example
A sample CDM is available in the Examples directory.

CDM properties

The Model property sheet displays the definition of the current model. From this property sheet you can modify the model definition.

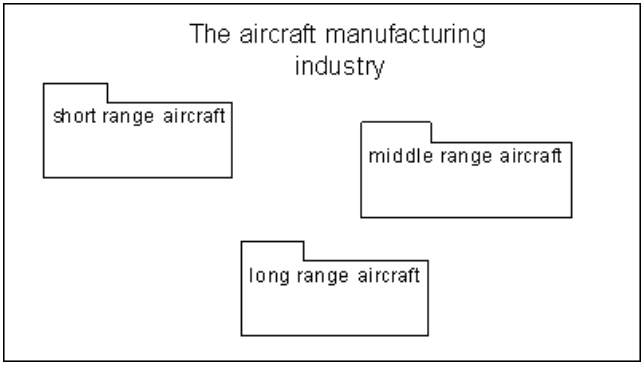
A CDM has the following model properties:

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 model
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 Title page of the model display preferences
Default diagram	Diagram displayed by default when you open the model

Defining packages in a CDM

A package is a piece of a model.

☞ When working with a large model, you can split the model into smaller subdivisions to avoid manipulating the entire set of model objects. Packages can be useful to assign portions of a model, representing different tasks and subject areas, to different development teams.



Package hierarchy

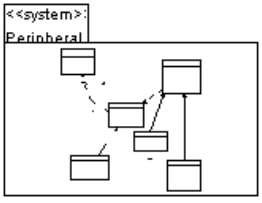
You can create several packages at the same hierarchical level within a model or decompose a package into other packages and continue this process without limitation in decomposition depth. Each package is displayed with a default diagram window. At each level of decomposition you can create several diagrams.

☞ For more information on packages, see the *General Features Guide* .

Composite view

You can expand a package to have a global view of the whole diagram content.

To do so, you have to right-click a package and select Composite View from the contextual menu. You must resize the composite package symbol to visualize its content:



To return to the package normal view, re-select the Composite View command in the contextual menu of the expanded package.

Note that if you double-click the composite view, you automatically open the sub-package diagram.

☞ For more information on the composite view feature, see section Expanding the symbol of a composite object in chapter Model Graphics in the General Features Guide.

CDM 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	Optional label that describes a package and provides additional information
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Default diagram	Diagram displayed by default when you open the package
Use Parent Namespace	Defines the package as being the area in which the name of an object must be unique in order to be used.

Modifying package display preference

You can modify the following display preference for packages using the Tools ➤ Display Preferences command:

Preference	Description
Stereotype	Displays the stereotype of the package
Comment	Displays the comment of the package

Customizing the CDM Environment

The CDM environment includes a set of parameters and configuration options that define various aspects of the model content and behavior. You can set these parameters:

- At model creation
- After creating a model with default options and parameters
- When creating a model template

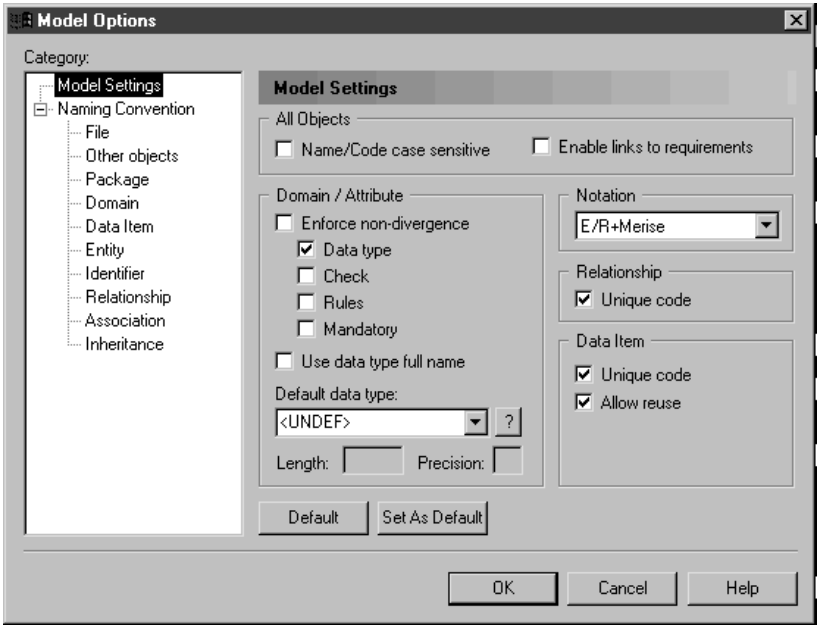
Setting CDM model options

This section explains how to set global options for the objects in your CDM. These options apply only to the current CDM.

For information about controlling the naming conventions of your models, see the chapter Managing Models in the *General Features Guide* .

Setting Model Settings

To set Model Settings, select Tools ➤ Model Options or right-click the diagram background and select Model Options from the contextual menu.



The options on this tab affect all the objects in the model, including those

already created, while changes to the object-specific options on the sub-category tabs only affect objects created subsequently.

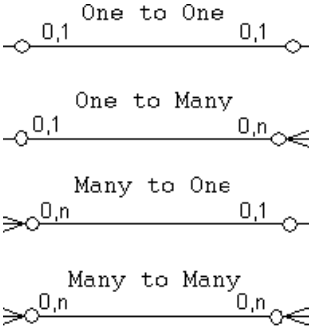
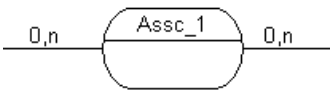
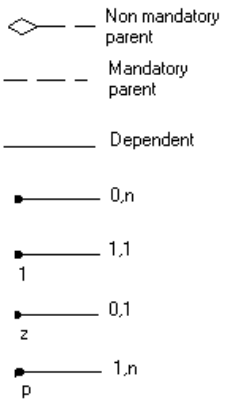
You can set the following options on this tab:

All Objects

Option	Description
Name/Code case sensitive	You can define the case sensitivity of names and codes for all objects in the current model. When this check box is selected, it implies that you can have two objects with identical name or code but different case in the same namespace. Unlike other model options, you can modify the name and code case sensitivity during the design process. However, if you do so, make sure you run the check model feature to verify if the model does not contain any duplicate object.
Enable links to requirements	Requirements are descriptions of customer needs that must be satisfied during development processes. You can enable links to requirements for all objects in the current model. When this check box is selected, it implies that the **Requirements** tab is displayed in the objects property sheet. The Requirements page allows you to attach requirements to objects; these requirements are defined in the Requirements models open in the workspace. Attached requirements and Requirements models are synchronized. For more information on requirements, see the *Requirements Model User's Guide* .

Notation

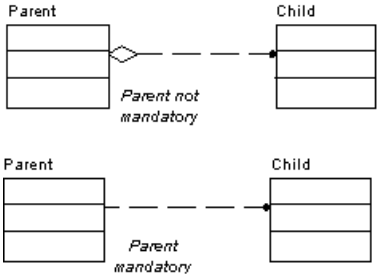
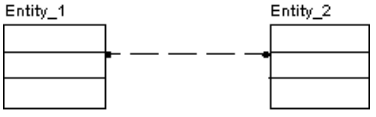
You can choose to use one or both of the following notation types in the current model:

Option	Description
Entity / Relationships	[Default – used throughout this manual] Entity/relationship notation connects entities with links representing one of four relationships between them. These relationships have properties that apply to both entities involved in the relationship 
Merise	Merise notation uses associations instead of relationships 
E/R + Merise	Both entity/relationship and Merise are used in the same model
IDEF1X	Data modeling notation for relationships and entities. In this notation, each set of relationship symbols describes a combination of the optionality and cardinality of the entity next to it 

When you change notation, all symbols in all diagrams are updated accordingly.

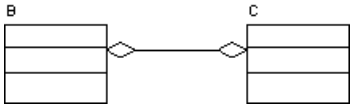
For the IDEF1X notation:

- **Entities** display rounded rectangles when they depend on another entity either through an inheritance link or when the relationship has the dependent property selected.
- **Relationships**: unlike the other notations, relationship symbols cannot be parsed in terms of optionality and cardinality independently. Each set of symbols describes a combination of the optionality and cardinality of the entity next to it.

Cardinality	Representation
One - Many	
One - One or Many - Many	
One - One dependent displays a continuous line and entity with rounded angles	

PowerDesigner does not support the following notation:

or



Sub-type or inheritance symbol changes according to the Complete property:

Complete	Symbol
Yes	
No	

Data Item

Option	Description
Unique code	Requires that data items have unique codes
Allow reuse	Makes it possible to reuse one data item as an attribute for more than one entity provided the attributes have same name and data type and do not belong to a primary key. When deselected or when the attribute belongs to a primary key, the data item cannot be reused. In this case, if the Unique code check box is selected, a new data item with identical name but different code is created, otherwise a new data item with identical name and code is created

When you delete an entity or entity attributes, these options determine whether or not the corresponding data items are also deleted, as follows:

Options set	Result of deleting an entity attribute
Both	Deletes the entity attribute.
Unique Code only	Deletes the entity attribute.
Allow Reuse only	Deletes the entity attribute and the corresponding data item (if it is not used by another entity).
None	Deletes the entity attribute and the corresponding data item.

Relationship

Option	Description
Unique code	Requires that relationships have unique codes.

Domain / Attribute

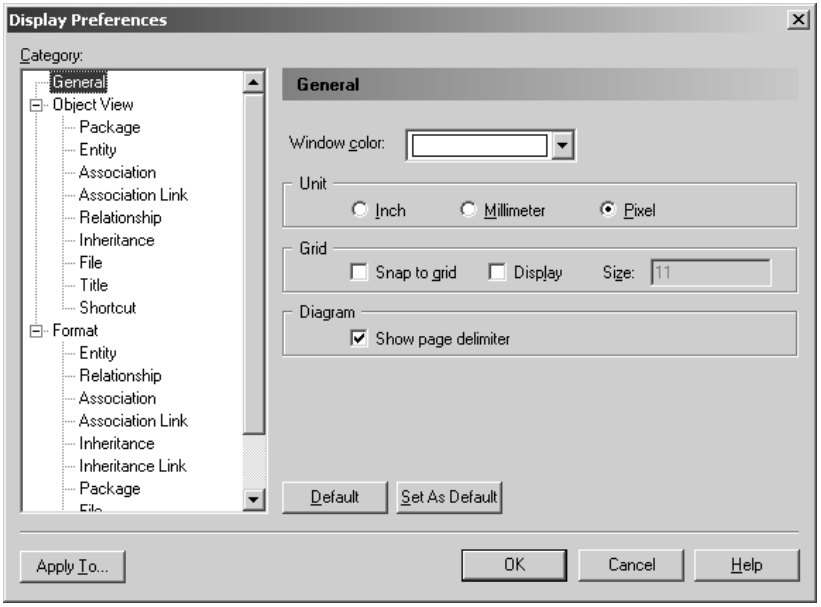
Option	Description
Enforce non-divergence	Enforces non-divergence between a domain definition and the attributes using the domain. You can select any or all of the following attribute properties: ♦ Data type - Data type, length, and precision ♦ Check - Check parameters, such as minimum and maximum values ♦ Rules - Business rules ♦ Mandatory - Entity, association and inheritance attribute mandatory property **When you apply the Enforce non-divergence options** You are asked if you want to apply domain properties to attributes currently attached to the domain. If you click **OK**, the attribute properties are modified in order to be consistent with the properties of the domain to which they belong. **When you modify the properties of a domain** The properties of the attributes attached to the domain are updated provided these properties are selected in the Model Options dialog box. **Attributes cannot be modified** When you select an attribute property under Enforce non-divergence, each instance of that attribute property in the lists of attributes and the property sheets of attributes is displayed as unavailable and can not be modified. If you want to modify an attribute property that is defined as non-divergent, you must detach the attribute from its domain, or clear the appropriate Enforce non-divergence model option.
Use data type full name	Specifies that the complete data type is displayed in entity symbols.
Default data type	Specifies a default data type to apply to domains and attributes if none is selected for them.

Setting CDM Display Preferences

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

To set CDM display preferences, select Tools ➤ Display Preferences or right-click the diagram background and select Display Preferences from the

contextual menu.

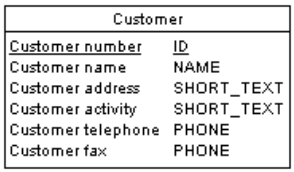
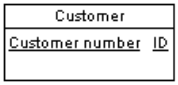
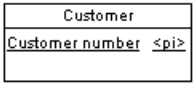
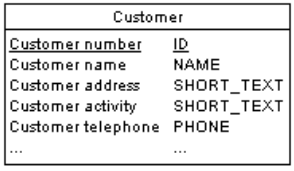
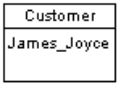
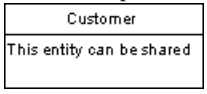


For information about changing the format of symbols, see the “Selecting format display preferences” section in the “Model Graphics” chapter in the *General Features Guide* . The following sections list the options available to customize the information displayed on CDM object symbols.

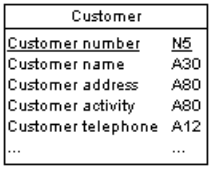
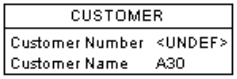
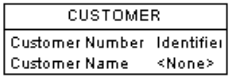
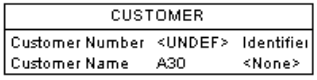
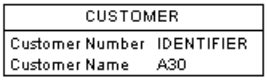
Entity display preferences

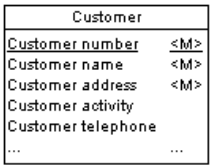
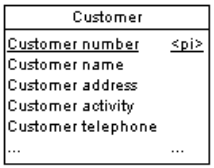
To set display preferences for entities, select Tools ➤ Display Preferences, and select the Entity sub-category in the left-hand Category pane.

Entity

Preference	Display description
Attributes	Specifies whether Attributes are displayed on entity symbols. If selected, you can choose between displaying: ♦ All attributes - All attributes including all identifying attributes:  ♦ Primary attributes - Only entity attributes that are primary identifier attributes:  ♦ Identifying attributes - All attributes that are defined as identifier attributes for the entity, including primary identifier attributes:  ♦ Display limit - Number of entity attributes shown depending on defined value. For example, if set to 5: 
Identifiers	All identifier attributes for the entity are listed at the bottom of the entity symbol: 
Stereotype	Stereotype of the entity.
Comment	Comment of the entity. When selected, all other check boxes are deselected, except for Stereotype: 

Entity attributes

Preference	Display description
Data types	Data type for each entity attribute: 
Replace by Domains	Domain for each entity attribute. You can only display domains when the Data type check box is selected.
Domains	Domain of an attribute in an entity. This display option interacts with the selection for Data types. As a result, there are four display options: ♦ Data types selected - Displays only the data type, if it exists:  ♦ Domains selected - Displays only the domain, if it exists:  ♦ Data types and Domain selected - Displays both data type and domain, if they exist  ♦ Data types and Replace by domains selected - Displays either data type or domain, if they exist, and domain if both exist: 

Preference	Display description
Mandatory	<*M*> indicators are displayed next to each mandatory entity attribute: 
Identifier indicators	<*pi*> indicators are displayed next to primary identifier attributes and <*ai*> indicators next to non-primary identifier attributes: 
Stereotype	Stereotype of the entity attributes

Entity identifiers

Preference	Display description
Stereotype	Stereotype of the entity identifiers

Association display preferences

To set display preferences for associations, select Tools ➤ Display Preferences, and select the Association sub-category in the left-hand Category pane.

Association

Preference	Display description
Attributes	Displays the name of the association.
Display Limit	Controls the maximum number of attributes displayed in symbol.
Stereotype	Displays the stereotype of the association.
Comment	Displays the comment of the association. When selected, all other check boxes are deselected, except for Stereotype.

Association attributes

Preference	Display description
Data types	Displays the data types of association attributes.
Replace by domains	Displays the domains of association attributes. You can only display domains when the Data types check box is selected.
Domains	Displays the domain of association attributes. For details of how this and the Data types preferences interact, see "Entity display preferences" on page 14.
Mandatory	Displays the letter *M* beside mandatory association attributes.
Stereotype	Displays the stereotype of association attributes.

Association Link display preferences

To set display preferences for association links, select Tools ➤ Display Preferences, and select the Association Link sub-category in the left-hand Category pane.

Preference	Description
Role	Displays the role of the association link
Cardinality	Displays the cardinality of the association link
Stereotype	Displays the stereotype of the association link

Relationship display preferences

To set display preferences for relationships, select Tools ➤ Display Preferences, and select the Relationship sub-category in the left-hand Category pane.

Preference	Display description
Name	Displays the name of the relationship*
Role	Displays the role of each direction in the relationship
Cardinality	Displays the minimum and maximum number of instances that the first entity can have relative to the second entity
Dominance	Displays the letter *D* on the dominant entity side of the relationship
Stereotype	Displays the stereotype of the relationship

Inheritance display preferences

To set display preferences for inheritances, select Tools ➤ Display Preferences, and select the Inheritance sub-category in the left-hand Category pane.

Preference	Description
Name	Displays the name of the inheritance
Stereotype	Displays the stereotype of the inheritance

Working with CDM extended model definitions

An extended model definition allows you to expand object definitions and complement the generation targets and commands. Extended model definitions are created and saved in files with the XEM extension. You can create or attach one or several extended model definitions to a model.

Extended model definitions may contain:

- Extended attributes for applicable objects in order to further define their properties

Stereotypes to define extended dependencies established between model objects (see "Working with CDM extended dependencies" on page 19).

- Generation targets and commands to complement the generation of an object model, or to perform an extended generation

☞ For more information on extended model definitions, see chapter Extended Model Definitions Reference Guide in the Advanced User Documentation.

Working with CDM extended dependencies

Extended dependencies are links between CDM objects. These links help to make object relationships clearer but are not interpreted and checked by PowerDesigner, as they are meant to be used for documentation purposes only.

You can complement these links by applying stereotypes. Stereotypes are used to define extended dependencies between objects in the CDM.

You can type stereotypes directly in the Stereotype column of the object property sheet or select a value from the list if you have previously defined stereotypes in an embedded or imported extended model definition (.XEM).

☞ For more information on extended dependencies, see section Using Extended Dependencies in chapter Using the PowerDesigner Interface in the *General Features Guide*.

CHAPTER 2

Building Conceptual Data Models

About this chapter

This chapter describes how to build a Conceptual Data Model (CDM). It explains the role of each object in a conceptual model and shows how to modify them in a conceptual diagram.

Contents

Topic:	page
Conceptual Diagram Basics	22
Data Items (CDM)	24
Domains (CDM)	28
Entities (CDM)	36
Attributes (CDM)	39
Identifiers (CDM)	44
Relationships (CDM)	47
Associations and Association Links (CDM)	62
Inheritances (CDM)	68
Check Parameters (CDM)	77

Conceptual Diagram Basics

A conceptual diagram allows you to define the logical structure of a database.

You can create the following objects in a conceptual diagram:

Object	Tool	Symbol	Description
Domain	[none]	[none]	Set of values for which a data item is valid. See "Domains (CDM)" on page 28.
Data Item	[none]	[none]	Elementary piece of information. See "Data Items (CDM)" on page 24.
Entity			Person, place, thing, or concept that has characteristics of interest to the enterprise and about which you want to store information. See "Entities (CDM)" on page 36.
Entity Attribute	[none]	[none]	Elementary piece of information attached to an entity. See "Attributes (CDM)" on page 39.
Identifier	[none]	[none]	Entity attribute, or a combination of entity attributes, whose values uniquely identify each occurrence of the entity. See "Identifiers (CDM)" on page 44.
Relationship			Named connection or relation between entities (Entity Relationship (ER) modeling methodology). See "Relationships (CDM)" on page 47.
Inheritance			Special relationship that defines an entity as a special case of a more general entity. See "Inheritances (CDM)" on page 68.

Object	Tool	Symbol	Description
Association			Named connection or association between entities (Merise modeling methodology). See "Associations and Association Links (CDM)" on page 62.
Association Link			Link that connects an association to an entity and on which you define the cardinality an entity has relative to another. See "Associations and Association Links (CDM)" on page 62.

Creating a conceptual diagram

You can create a conceptual diagram in an existing CDM in any of the following ways:

- Right-click the model in the Browser and select New ➤ Conceptual Diagram from the contextual menu
- Right-click the background of any diagram and select Diagram ➤ New Diagram ➤ Conceptual Diagram from the contextual menu.

To create a new CDM with a conceptual diagram, select File ➤ New, choose Conceptual Data Model from the Model type list, and click OK.

Data Items (CDM)

A **data item** is an elementary piece of information in a model. It represents a fact or a definition in an information system, which may or may not have any eventual existence as a modeled object.

You can attach a data item to an entity. It then becomes an entity attribute of that entity. When a physical data model (PDM) is generated from a CDM, an entity attribute generates a column in a table.

You can also define a data item, but not attach it to an entity. It remains defined in the model and can be attached to an entity at any time.

Depending on its defined model options, a data item can be unique in the model, or unique only for a particular entity. In the latter case, a data item is equivalent to an entity attribute.

Example

In the information system for a publishing company, the last names for authors and customers are both important pieces of business information. The data item LAST NAME is created to represent this information. It is attached to the entities AUTHOR and CUSTOMER, and becomes entity attributes of those entities.

Another piece of information is the date of birth of each author. The data item BIRTH DATE is created in the CDM, but as there is no immediate need for this information in the model, it remains in the CDM, but is not attached to any entity.

Creating a data item

You can create a data item in any of the following ways:

- Select Model ➤ Data Items to access the List of Data Items, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select New ➤ Data Item.

☞ For general information about creating objects, see the chapter “Getting Started with PowerDesigner” in the *General Features Guide* .

Data item properties

You can modify an object’s properties from its property sheet. To open a data item property sheet, double-click its Browser entry in the Data Items 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 data item
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Data type	Code indicating the data format, such as N for numeric or A for alphanumeric, followed by the number of characters
Length	Maximum number of characters
Precision	Number of places after the decimal point, for data values that can take a decimal point
Domain	Name of the associated domain

The following tabs are also commonly used:

- Standard Checks - contains checks which control the values permitted for the data item. For more information, see "Check Parameters (CDM)" on page 77.
- Additional Checks - allows you to specify additional constraints (not defined by standard check parameters) for the data item.
- Rules - lists the business rules associated with the data item. For more information, see the "Using Business Rules in a CDM" chapter.

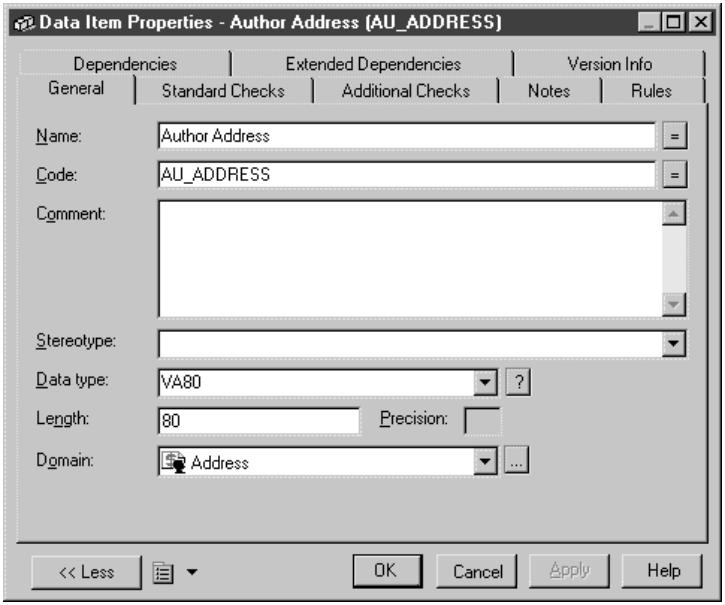
Specifying a data type for a data item

You can specify a data type for a data item in either of the following ways.

- Attach the data item to a domain - The domain dictates a standard data type, a length, and a level of precision, as well as optional check parameters. See
- Manually select a data type - You select a standard data type along with a length, a level of precision, and optional check parameters

❖ To select a data type for a data item

1. Open the data item property sheet by double-clicking its Browser entry or from the List of Data Items.



2. Select a data type from the Data Type list (or click the Question mark button to the right of the list to display and select from a list of standard data types).

> **Undefined data type**
> If you do not want to select a data type immediately, you can choose the <Undefined> data type.

3. [optional] For data types (such as Characters) that may have different lengths, type the maximum number of characters to be permitted in the Length box.
4. [optional] For data types (such as Decimal) that take a decimal point, type the number of places permitted after the decimal point in the Precision box.
5. Click OK to close the data item property sheet and return to the model.

Attaching a data item to a domain

If you attach a data item to a domain, the domain supplies a data type to the data item, and can also apply length, decimal precision, and check parameters.

For information about creating domains, see "Domains (CDM)" on page 28.

❖ To attach a data item to a domain

1. Open the data item property sheet by double-clicking its Browser entry or from the List of Data Items.
2. Select a domain from the Domain list (or click the ellipsis button to the right of the list to display and select from the List of Domains).
3. Click OK to close the data item property sheet and return to the model.

Controlling uniqueness and reuse of data items

The following model options allow you to control naming restraints and reuse for data items:

Option	When selected	When cleared
Unique code	Each data item must have a unique code.	Multiple data items can have the same code.
Allow reuse	One data item can be an entity attribute for multiple entities.	Each data item can be an entity attribute for only one entity

If you do not select Unique Code, two data items can have the same code, and you differentiate them by the entities that use them. The entities are listed in the Used By column of the list of data items.

> **Item not visible in list**
> To make an item visible in a list, click the Customize Columns and Filter tool in the list toolbar, select the appropriate check box from the list of filter options that is displayed, and click OK.

❖ To define code and reuse options for data items

1. Select Tools ➤ Model Options to open the Model Options dialog box:
2. Select or clear the Unique Code and Allow Reuse check boxes in the Data Item groupbox, and then click OK to return to the model.

Error message

The following error message is displayed if you select the Unique Code option, when data items are already sharing a name in the CDM:

Error message	Solution
Unique Code option could not be selected because two data items have the same code: *data_item_code*.	Assign unique codes to all data items

Domains (CDM)

Domains help you identify the types of information in your model. Applying domains to data items makes it easier to standardize data characteristics for entity attributes in different entities.

In a CDM, you can associate the following information with the domain:

- Data type, length, and precision
- Check parameters
- Business rules
- Mandatory

Creating a domain

You can create a domain as follows:

- Select Model ➤ Domains to access the List of Domains, and click the Add a Row tool
- Right-click the model or package in the Browser, and select New ➤ Domain

☞ For general information about creating objects, see the chapter "Getting Started with PowerDesigner" in the *General Features Guide* .

Domain properties

You can modify an object's properties from its property sheet. To open a domain property sheet, double-click its Browser entry in the Domains 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 domain
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined

Property	Description
Data type	Form of the data corresponding to the domain, such as numeric, alphanumeric, boolean, or others
Length	Maximum number of characters
Precision	Number of places after the decimal point, for data values that can take a decimal point
Mandatory	Domain values are mandatory for all entity attributes using that domain

The following tabs are also commonly used:

- Standard Checks - contains checks which control the values permitted for the domain. For more information, see "Check Parameters (CDM)" on page 77.
- Additional Checks - allows you to specify additional constraints (not defined by standard check parameters) for the domain.
- Rules - lists the business rules associated with the domain. For more information, see the "Using Business Rules in a CDM" chapter.

Specifying a domain's data type, length, and precision

Length and precision

The properties length and precision do not apply to all data types. Depending on data type, length may indicate a maximum or a fixed number of characters.

In the list of available data types, a variable indicates where you have to type a length or precision, as follows:

Variable	Replace with
%n	Length
%s	Length with precision
%p	Decimal precision

For example, if you are using Sybase Adaptive Server Anywhere and you choose the data type `char(%n)`, you can choose a length of ten by typing `char(10)`.

Undefined data type

All target DBMS allow you to select the <undefined> data type. The <undefined> data type indicates which domains remain without data types.

If an <undefined> data type is present when you generate your database, it is replaced by the default data type for your database.

❖ **To specify a data type for a domain from the list of domains**

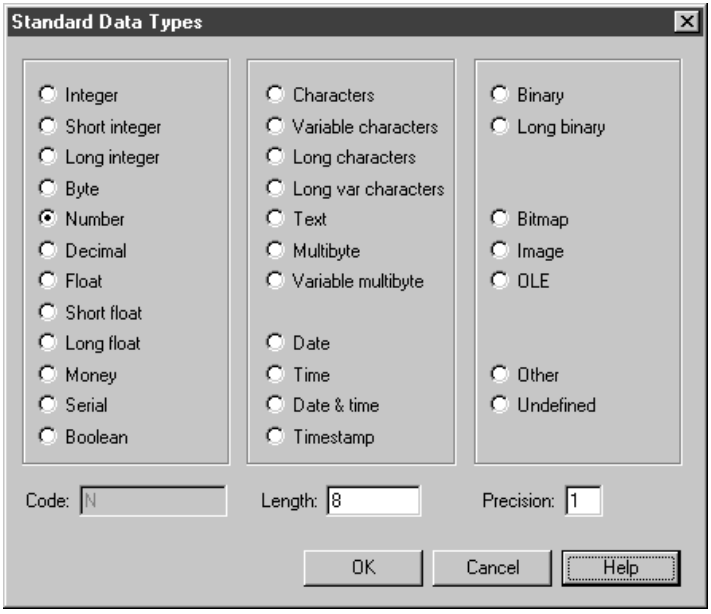
1. Select Model ➤ Domains to open the List of Domains.
2. Click the domain that you want to define.
3. Select a data type from the list in the Data Type column and, if necessary, enter a length and precision.
4. Click OK.

❖ **To select a data type for a domain from its property sheet**

1. Select Model ➤ Domains to display the List of Domains.
2. Click the domain that you want to define and then click the Properties tool to open its property sheet.
3. Select a data type from the Data Type list or click the question mark button to the right of this list to open the list of standard data types.
4. If necessary, enter values in the Length and Precision boxes.
5. Click OK.

List of standard data types

You can open the list of Standard Data Types by clicking the question mark button to the left of the list of Data Types on the General Tab of a domain property sheet.



This is the same list that is available in the Physical Data Model, and any . data type selected in this list will be automatically mapped to a conceptual data type.

Numeric data types

The following numeric data types are available:

Conceptual data type	DBMS-specific physical data type	Content	Length
Integer	int / INTEGER	32-bit integer	—
Short Integer	smallint / SMALL-INT	16-bit integer	—
Long Integer	int / INTEGER	32-bit integer	—
Byte	tinyint / SMALLINT	256 values	—
Number	numeric / NUMBER	Numbers with a fixed decimal point	Fixed
Decimal	decimal / NUMBER	Numbers with a fixed decimal point	Fixed
Float	float / FLOAT	32-bit floating point numbers	Fixed

Conceptual data type	DBMS-specific physical data type	Content	Length
Short Float	real / FLOAT	Less than 32-bit point decimal number	♦ —
Long Float	double precision / BINARY DOUBLE	64-bit floating point numbers	—
Money	money / NUMBER	Numbers with a fixed decimal point	Fixed
Serial	numeric / NUMBER	Automatically incremented numbers	Fixed
Boolean	bit / SMALLINT	Two opposing values (true/false; yes/no; 1/0)	—

Character data types

The following character data types are available:

Conceptual data type	DBMS-specific physical data type	Content	Length
Characters	char / CHAR	Character strings	Fixed
Variable Characters	varchar / VARCHAR2	Character strings	Maximum
Long Characters	varchar / CLOB	Character strings	Maximum
Long Var Characters	text / CLOB	Character strings	Maximum
Text	text / CLOB	Character strings	Maximum
Multibyte	nchar / NCHAR	Multibyte character strings	Fixed
Variable Multibyte	nvarchar / NVARCHAR2	Multibyte character strings	Maximum

Time data types

The following time data types are available:

Conceptual data type	DBMS-specific physical data type	Content	Length
Date	date / DATE	Day, month, year	
Time	time / DATE	Hour, minute, and second	
Date & Time	datetime / DATE	Date and time	
Timestamp	timestamp / TIMESTAMP	System date and time	

Other data types

The following other data types are available:

Conceptual data type	DBMS-specific physical data type	Content	Length
Binary	binary / RAW	Binary strings	Maximum
Long Binary	image / BLOB	Binary strings	Maximum
Bitmap	image / BLOB	Images in bitmap format (BMP)	Maximum
Image	image / BLOB	Images	Maximum
OLE	image / BLOB	OLE links	Maximum
Other	—	User-defined data type	—
Undefined	undefined	Not yet defined data type	—

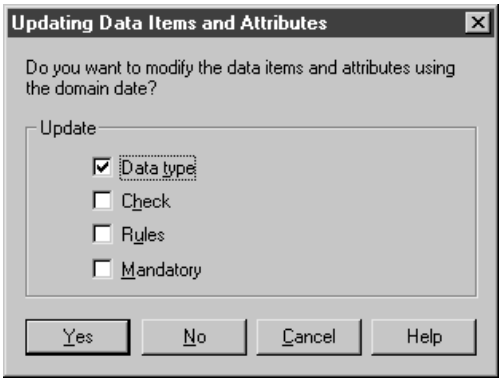
Cascading updates to columns associated with the domain

When you modify data types associated with a domain, an update confirmation box is displayed asking if you want to modify the columns currently using the domain.

❖ To modify domain properties

1. Open the property sheet of a domain and edit its properties as required.
2. Click OK.

   If the domain is used by one or more columns, an update confirmation box is displayed asking if you want to modify domain properties for the columns using the domain.



The Data Type check box is selected or not according to the options set to enforce non-divergence from a domain (see “Enforcing non-divergence from domains in a CDM” on page 34).

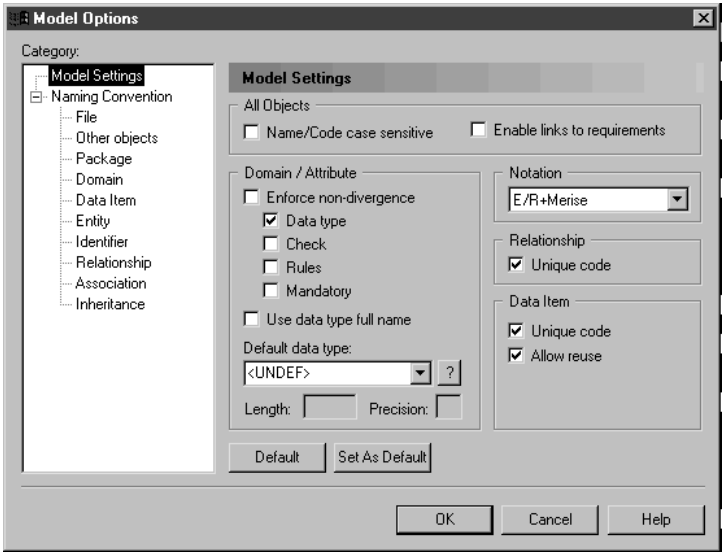
1. Select any other properties that you want to update (Check, Rules, Mandatory, Profile) for all columns using the domain.
2. Click one of the following buttons:
   - Yes - The columns currently using the domain are modified according to the update
   - No - The columns currently using the domain are not modified according to the update but the current modification is accepted if domain divergence is allowed in the model options (see).
   - Cancel - The update is cancelled and nothing is changed

Enforcing non-divergence from domains in a CDM

You can enforce non-divergence between a domain and the columns that use the domain.

❖ To enforce domain non-divergence in a CDM

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



2. Select the check boxes of the column properties that are not permitted to diverge from the domain definition. You can specify any or all of:
   - Data type - data type, length, and precision
   - Check - check parameters such as minimum and maximum values
   - Rules – business rules
   - Mandatory – mandatory property of the column

   If you subsequently modify in your domain any of the properties specified as non-divergent here, then the corresponding properties of the entity attributes attached to that domain are automatically updated.

   Attribute properties specified as non-divergent appear dimmed and are non-editable in Attribute property sheets. If you want to modify a non-divergent attribute property, you must detach the attribute from its domain.

3. Click OK to close the Model Options dialog box.

4. When you set the Enforce non-divergence options, you are asked if you want to apply domain properties to attributes currently attached to the domain. If you click **OK**, the attribute properties are modified in order to be consistent with the properties of the domain to which they belong.

Entities (CDM)

An entity represents an object defined within the information system about which you want to store information. For example, in a model concerning employees and divisions, the entities are Employee and Division.

An occurrence of an entity is an individual element belonging to the entity. For example, the employee Martin is one occurrence of the entity Employee.

Creating an entity

You can create an entity in any of the following ways:

- Use the Entity tool in the diagram Palette.
- Select Model ➤ Entities to access the List of Entities, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select New ➤ Entity.

☞ For general information about creating objects, see the chapter “Getting Started with PowerDesigner” in the *General Features Guide* .

Entity properties

You can modify an object’s properties from its property sheet. To open an entity property sheet, double-click its diagram symbol or its Browser entry in the Entities 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 entity
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Number	Estimated number of occurrences in the physical database for the entity (the number of records)
Generate	Indicates if the entity will generate a table in a PDM

The following tabs are also commonly used:

- Attributes - lists the attributes associated with the entity. For more information, see "Attributes (CDM)" on page 39.
- Identifiers - lists the attributes associated with the entity. For more information, see "Identifiers (CDM)" on page 44.
- Rules - lists the business rules associated with the entity. For more information, see the "Using Business Rules in a CDM" chapter.

Copying an entity

You can make a copy of an entity within the same model or between models.

The following rules apply to copied entities. The indicated selections for Unique code and Allow reuse apply to the model that receives the copied entity:

Data item options selected	Result of copying an entity
Unique Code	New entity with new name and code
Allow Reuse	New identifier with new name code
	Reuses other entity 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

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

❖ To copy an entity within a model

1. Select an entity in the CDM, and then select Edit ➤ Copy and Edit ➤ Paste.
2. [alternatively] Press CTRL and drag the entity to a new position in the diagram.

   The entity is copied and the new entity is displayed in the Browser and diagram.

❖ To copy an entity to a different model

1. Select an entity in the CDM, and then select Edit ➤ Copy
2. Select the new diagram or model and then select Edit ➤ Paste.

   The entity is copied and the new entity is displayed in the Browser and diagram.

Attributes (CDM)

Attributes are data items attached to an entity, association, or inheritance. When you generate a Physical Data Model (PDM) from a Conceptual Data Model (CDM), entity attributes are generated as columns in tables (entities).

Creating an attribute

You can create an entity attribute using any of the following tools, available on the Attributes tab in the property sheet of an entity, association, or inheritance:

Tool	Description
	Add a Row – Creates a new attribute and associated data item (with the same name and code). If you have enabled the Allow Reuse model option, the new data item can be used as an attribute for other objects. If you have enabled the Allow Reuse and Unique Code model options and you type the name of an existing data item, it will be automatically reused.
	Add Data Item - Opens a Selection window listing all the data items available in the model. Select one or more data items in the list and then click OK to make them attributes to the object. If the data item has not yet been used, it will be linked to the object. If the data item has already been used, it will be copied (with a modified name if you have enabled the Unique code model option) and the copy attached to the object.
	Reuse Data Item – [only available if the Allow Reuse model option is enabled] Opens a Selection window listing all the data items available in the model and not currently associated with the object. Select one or more data items in the list and then click OK to make them attributes to the object. You can view all the objects to which the data item is associated on the Dependencies tab of its property sheet.

☞ For general information about creating objects, see the chapter “Getting Started with PowerDesigner” in the *General Features Guide* .

Attribute properties

You can modify an object’s properties from its property sheet. To open an

attribute property sheet, double-click its Browser entry in the Attributes folder within an entity, association, or inheritance.

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 attribute
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
[parent]	[read-only] Specifies the parent entity, association, or inheritance. Click the tool to the right of the field to open its property sheet.
Data Item	[read-only] Specifies the related data item. Click the tool to the right of the field to open its property sheet.
Data type	Form of the data corresponding to the attribute, such as numeric, alphanumeric, boolean, or others
Length	Maximum length of the data type
Precision	Maximum number of places after the decimal point
Domain	Name of the associated domain
Primary Identifier	[entity attributes only] Indicates whether or not the attribute is the primary identifier of the current entity
Displayed	[entity and association attributes only] Displays the attribute in the object symbol.
Mandatory	Specifies that the attribute must be assigned a value

The following tabs are also commonly used:

- Standard Checks - contains checks which control the values permitted for the attribute. For more information, see "Check Parameters (CDM)" on page 77.

- Additional Checks - allows you to specify additional constraints (not defined by standard check parameters) for the attribute.
- Rules - lists the business rules associated with the attribute. For more information, see “Business Rules” in the Managing Objects chapter of the *General Features Guide* .

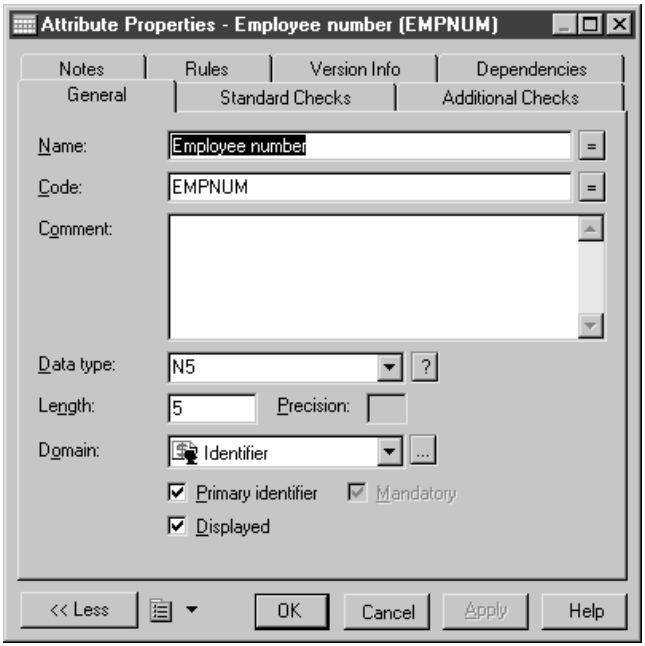
Specifying a data type for an attribute

You can specify a data type for an attribute in either of the following ways.

- Attach the attribute to a domain - The domain dictates a standard data type, a length, and a level of precision, as well as optional check parameters (see “Attaching an attribute to a domain” on page 42).
- Manually select a data type - You select a standard data type along with a length, a level of precision, and optional check parameters

❖ To select a data type for an attribute

1. Open the attribute property sheet by double-clicking its Browser entry or from the Attributes tab of its parent object.



2. Select a data type from the Data Type list (or click the Question mark button to the right of the list to display and select from a list of standard data types).

Undefined data type
If you do not want to select a data type immediately, you can choose the <Undefined> data type.

3. [optional] For data types (such as Characters) that may have different lengths, type the maximum number of characters to be permitted in the Length box.
4. [optional] For data types (such as Decimal) that take a decimal point, type the number of places permitted after the decimal point in the Precision box.
5. Click OK to close the attribute property sheet and return to the model.

Attaching an attribute to a domain

If you attach an attribute to a domain, the domain supplies a data type to the attribute, and can also apply length, decimal precision, and check parameters.

For information about creating domains, see "Domains (CDM)" on page 28.

❖ To attach an attribute to a domain

1. Open the attribute property sheet by double-clicking its Browser entry or from the Attributes tab of its parent object.
2. Select a domain from the Domain list (or click the ellipsis button to the right of the list to display and select from the List of Domains).
3. Click OK to close the attribute property sheet and return to the model.

Specifying an attribute as mandatory

If an attribute is mandatory, every object occurrence must have a value for that attribute. Identifiers (see "Identifiers (CDM)" on page 44) are always mandatory.

❖ To define an attribute as mandatory

1. Open the attribute's property sheet and click the Attributes tab.
2. Select the Mandatory check box at the bottom of the dialog box and click Ok to return to the model.

Deleting attributes

When you delete an attribute, model options determine whether or not the corresponding data items are also deleted:

Model options selected	Result of deleting an attribute
Unique Code Allow Reuse	Does not delete corresponding data item
Unique Code only	Does not delete corresponding data item
Allow Reuse only	Deletes corresponding data item if it is not used by another entity
None	Deletes corresponding data item

Arranging a list of attributes

Each entity, association, and inheritance has its own list of attributes.

In a list of attributes, you can arrange items in any order. When you generate a PDM, columns will be generated in this order.

❖ To arrange a list of attributes

1. Open the property sheet of an entity, association, or inheritance and click the Attributes tab.
2. Select one or more attributes in the list and then use the arrow buttons at the bottom left corner of the list to move the items in the list.
3. Click OK to return to the model.

Identifiers (CDM)

An **identifier** is an entity attribute, or a combination of entity attributes, whose values uniquely identify each occurrence of the entity. An identifier is the CDM equivalent of a primary key or an alternate key in a physical data model (PDM).

Each entity must have at least one identifier. If an entity has only one identifier, it is designated by default as the primary identifier for the entity. A primary identifier is the main identifier for an entity.

Creating an identifier

You can create an entity in any of the following ways:

- Open the Identifiers tab in the property sheet of an entity, select one or more attributes, and click the Create Identifier tool. The selected attributes are associated with the identifier and are listed on the attributes tab of its property sheet.
- Open the Identifiers tab in the property sheet of an entity, and click the Add a Row tool.

☞ For general information about creating objects, see the chapter “Getting Started with PowerDesigner” in the *General Features Guide* .

Identifier properties

You can modify an object’s properties from its property sheet. To open an identifier property sheet, double-click its Browser entry in the Identifiers folder beneath an entity. The following sections detail the property sheet tabs that contain the properties most commonly entered for identifiers.

Identifier property sheet General tab

The General tab contains the following properties:

Property	Description
Name	The name of the identifier which should be clear and meaningful, and should convey its purpose to non-technical users
Code	The technical name of the identifier used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Descriptive label for the identifier
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Entity	The name of the entity to which the identifier belong.s
Primary identifier	Specifies that the identifier is a primary identifier.

Identifier property sheet Attributes tab

The Attributes tab lists the attributes associated with the identifier: Click the Add Attributes tool to add an attribute.

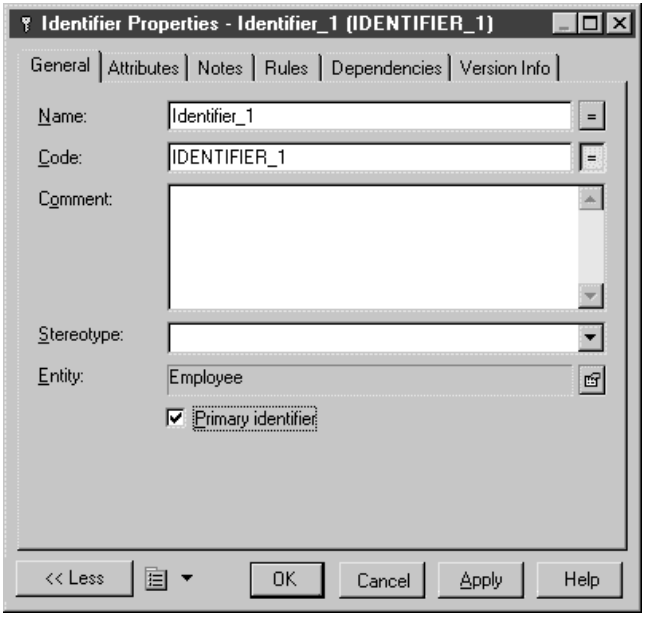
For more information, see "Attributes (CDM)" on page 39.

Designating a primary identifier

A **primary identifier** is an identifier that has been designated as the main identifier for the entity. It generates a primary key in a PDM.

❖ To designate a primary identifier

1. Open the property sheet of an entity and click the Attributes tab.
2. Select an identifier from the list and then click the Properties tool to open the identifier property sheet..



3. Select the Primary Identifier check box and click OK to return to the entity property sheet. The P column is selected for the new primary identifier.

> **Item not visible in list**
> To make an item visible in a list, click the Customize Columns and Filter tool in the list toolbar, select the appropriate check box from the list of filter options that is displayed, and click OK.

Relationships (CDM)

A relationship is a link between entities. For example, in a CDM that manages human resources, the relationship Member links the entities Employee and Team, because employees can be members of teams. This relationship expresses that each employee works in a team and that each team has employees.

An occurrence of a relationship corresponds to one instance of each of the two entities involved in the relationship. For example, *the employee Martin working in the Marketing team* is one occurrence of the relationship Member.

Relationships and association

Relationships are used in the Entity Relationship (ER) and IDEF1X modeling methodologies. In PowerDesigner you can also use associations used in the Merise methodology to link entities. You can use either relationships or associations exclusively, or combine the two methodologies in the same model.

☞ For more information on using associations, see "Associations and Association Links (CDM)" on page 62.

This section analyzes relationships in the Entity Relationship methodology, for more information on IDEF1X, see the CDM Model Options in chapter Conceptual Data Model Basics.

Creating a relationship

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

- Use the Relationship tool in the diagram Palette. Click inside the first entity to be linked and, while continuing to hold down the mouse button, drag the cursor to the second entity. Release the mouse button inside the second entity.
- Select Model ➤ Relationships to access the List of Relationships, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select New ➤ Relationship.

☞ For general information about creating objects, see the chapter "Getting Started with PowerDesigner" in the *General Features Guide* .

Relationship properties

You can modify an object's properties from its property sheet. To open a relationship property sheet, double-click its diagram symbol or its Browser

entry in the Relationships folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for relationships.

Relationship property sheet General tab

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 relationship
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Entity1 Entity2	The two entities linked by the relationship. You can use the tools to the right of the lists to create an object, browse the complete tree of available objects or view the properties of the currently selected object
Generate	Indicates to generate a reference in PDM
Cardinalities	Contains data about cardinality as the number of instances of one entity in relation to another entity

Relationship property sheet Cardinalities tab

The Cardinalities tab contains the following properties:

Property	Description
Cardinality	Specifies the number of instances (none, one, or many) of an entity in relation to another entity. You can choose from the following values: ♦ One-to-one (symbol: <1..1>) - One instance of the first entity can correspond to only one instance of the second entity ♦ One-to-many (symbol: <1..n>) - One instance of the first entity can correspond to more than one instance of the second entity ♦ Many-to-one (symbol: <n..1>) - More than one instance of the first entity can correspond to the same one instance of the second entity ♦ Many-to-many (symbol: <n..n>) - More than one instance of the first entity can correspond to more than one instance of the second entity

In addition, this tab contains a groupbox for both directions of the relationship, each of which contains the following properties:

Property	Description
Dominant role	In a One to One relationship, indicates to generate a reference in the PDM for this direction of the relationship only.
Role name	Text that describes the relationship of EntityA to EntityB.
Dependent	Indicates that each instance of the EntityA is identified by an instance of EntityB.
Mandatory	Indicates that each instance of the EntityA requires an instance of the EntityB.

Property	Description
Cardinality	Specifies the maximum and minimum number of instances of EntityA in relation to EntityB (if mandatory, at least 1). You can choose from the following values: ♦ 0..1 – Zero to one instances ♦ 0..n – Zero to many instances ♦ 1..1 – Exactly one instance ♦ 1..n – one to many instances

Termination points

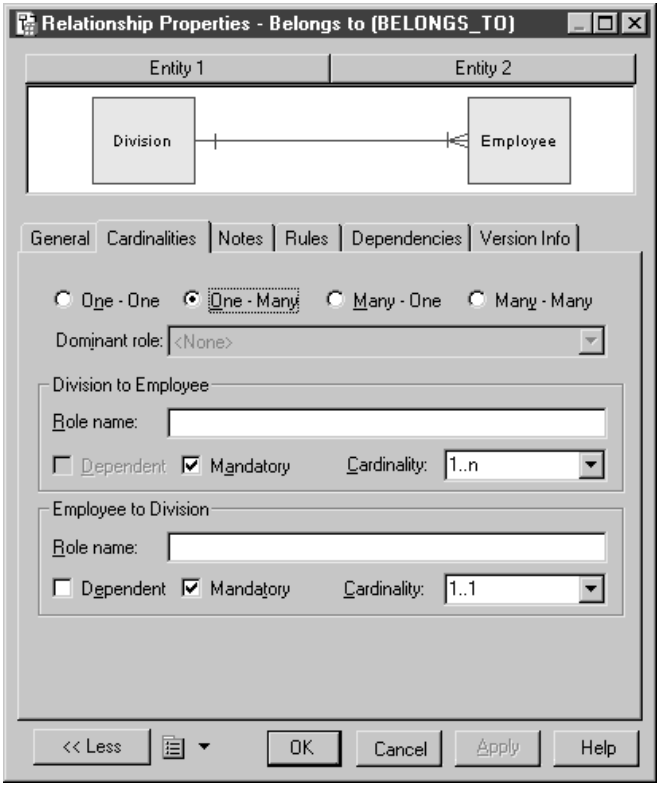
In a diagram, termination points indicate cardinality at each end of a relationship. A termination with a single contact point denotes a cardinality of one. A termination with three contact points denotes a cardinality of many.

Termination point	Cardinality
	One
	Many

☞ For more information on IDEF1X cardinality notation, see section CDM options in chapter Conceptual Data Model Basics.

Example

The Cardinalities page below indicates that an employee must belong to one and only one division, while a division must include one or more employees.

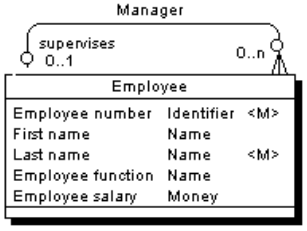


Creating a reflexive relationship

A reflexive relationship is a relationship between an entity and itself.

Example

The reflexive relationship *Supervise* expresses that an employee (Manager) can supervise other employees.



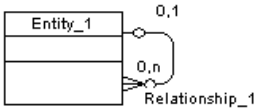
Getting neat relationship lines

To obtain clean lines with rounded corners when you create a reflexive relationship, select Display Preferences ➤ Format ➤ Relationship and modify the Line Style with the appropriate type from the Corners list.

❖ To create a reflexive relationship

1. Click the Relationship tool in the Palette.
2. Click inside the entity symbol and, while continuing to hold down the mouse button, drag the cursor a short distance within the symbol, before releasing the button.

   A relationship symbol loops back to the same entity.



Entity dependencies
In the Dependencies page of the entity, you can see two identical occurrences of the relationship, this is to indicate that the relationship is reflexive and serves as origin and destination for the link

Defining a code option for relationships

The following model option allows you to control naming restraints for relationships:

Option	When selected	Result when cleared
Unique code	Each relationship must have a unique code.	Multiple relationships can have the same code.

If you do not select Unique Code, two relationships can have the same code, and you differentiate them by the entities they link.

Error message

The following error message is displayed when the option you choose is incompatible with the current CDM:

Error message	Solution
Unique Code option could not be selected because at least two relationships have the same code: *relationship_code*.	Change the code of one relationship

❖ To define the Unique Code model option for relationships

1. Select Tools ➤ Model Options to open the Model Options dialog box:
2. Select or clear the Unique Code check box in the Relationship groupbox, and then click OK to return to the model.

Defining a mandatory relationship

The property Mandatory indicates whether the relationship between entities is mandatory or optional. You define options from the point of view of both entities in the relationship.

Termination points

In the resulting relationship symbol, a bar represents a mandatory relationship. A circle represents an optional relationship.

Termination point	Existence	Cardinality	Description
	Mandatory	One	One and only one must exist
	Mandatory	Many	One or more must exist
	Optional	One	None or one may exist
	Optional	Many	None or more may exist

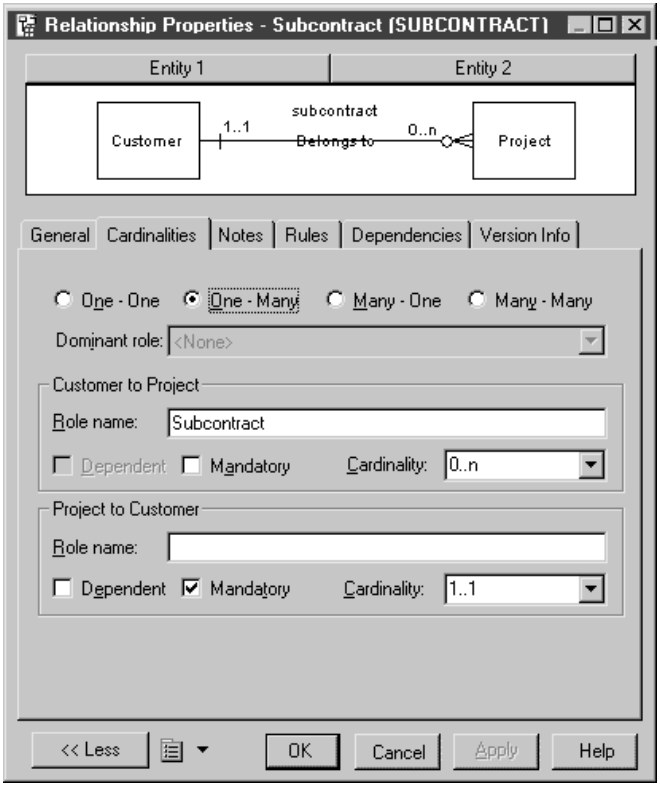
☞ For more information on IDEF1X existence notation, see the CDM options section in the Conceptual Data Model Basics chapter.

❖ To define a mandatory relationship

1. Open the relationship property sheet and click the Cardinalities tab.
2. Select the Mandatory check box in one or both of the groupboxes corresponding to a relationship direction.
3. Click OK to return to the model.

Example

The Cardinalities page below indicates that the subcontract relationship is optional from customer to project, and mandatory from project to customer. Each project must have a customer, while each customer does not have to have a project.



Defining a dependent relationship

In a dependent relationship, one entity is partially identified by another. Each entity must have an identifier. In some cases, however, the attributes of an entity are not sufficient to identify an occurrence of the entity. For these entities, their identifiers incorporate the identifier of another entity with which they have a dependent relationship.

Example

An entity named Task has two entity attributes, TASK NAME and TASK COST. A task may be performed in many different projects and the task cost will vary with each project. To identify each occurrence of TASK COST the unique Task entity identifier is the compound of its *Task name* entity attribute and the *Project number* identifier from the Project entity.

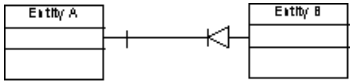
> **Many to many relationship**
> A many-to-many relationship cannot be a dependent relationship.

Termination points

In the resulting model, a triangle just before the termination point of the relationship indicates a dependent relationship. The triangle points away

from the dependent entity.

For example, here Entity B depends on Entity A.



The bar indicates that the relationship in not only dependent, but also mandatory. The single contact point indicates the one-to-one cardinality of the relationship. In this case, for each occurrence of Entity B there must be only one occurrence of Entity A.

The tip of the triangle shown above has a bar to note a mandatory relationship: every occurrence of Entity A requires an occurrence of Entity B.

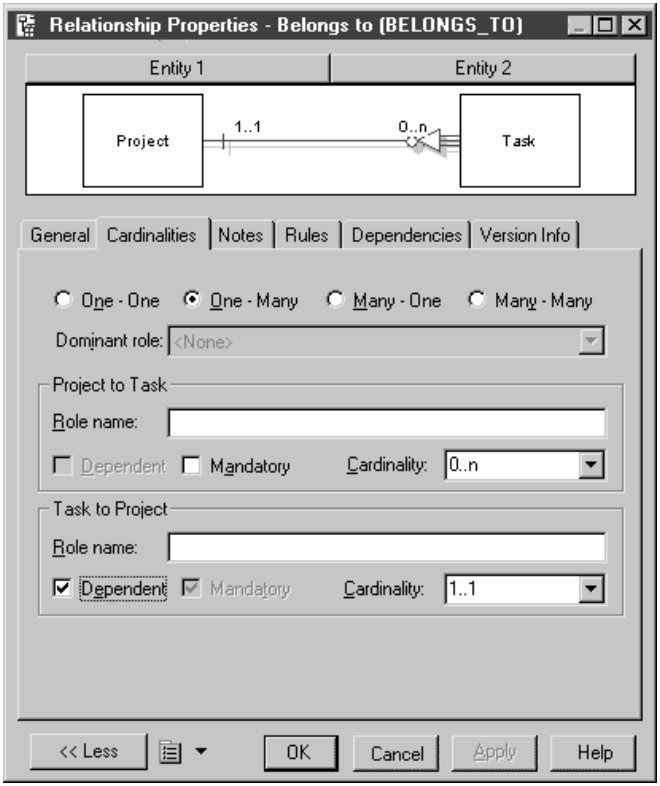
☞ For more information on IDEF1X notation, see section CDM options in chapter Conceptual Data Model Basics.

❖ To define a dependent relationship

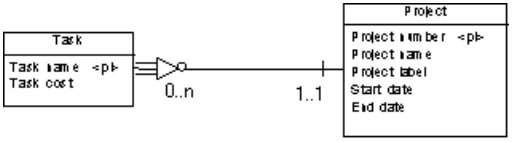
1. Open the relationship property sheet and click the Cardinalities tab.
2. Select or clear the Dependent check box in each groupbox corresponding to a relationship direction.
3. Click OK to return to the model.

Example

The property sheet below shows that the Task entity is dependent on the Project entity.



The relationship pictured here expresses this dependency.



The circle at the tip of the triangle indicates that occurrences of the Project entity do not require an occurrence of the Task entity. But an occurrence of the Task entity requires an occurrence of the Project entity on which it depends.

Defining a dominant relationship

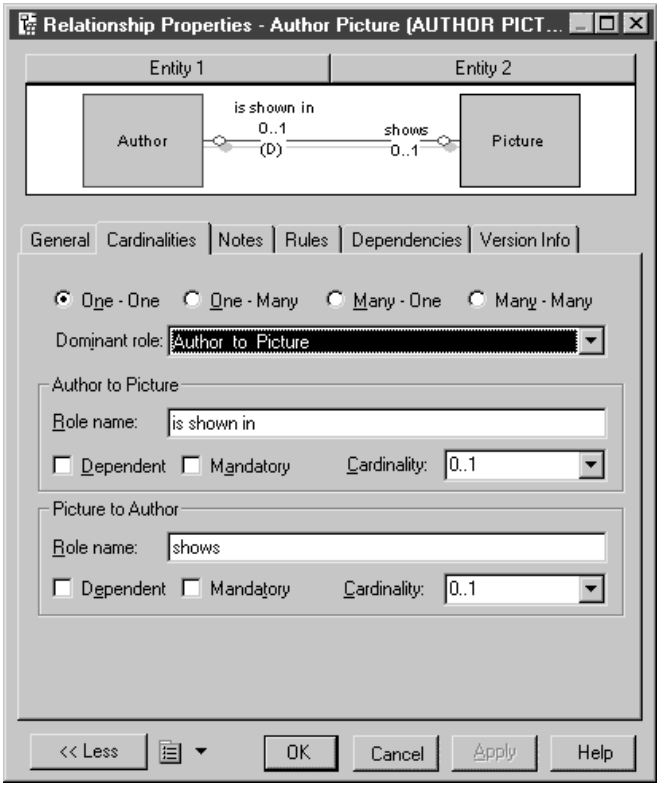
In a one-to-one relationship, you can define one direction of the relationship as dominant. If you define a dominant direction, the one-to-one relationship generates one reference in the PDM. The dominant entity becomes the parent table. If you do not define a dominant direction, the one-to-one relationship generates two references.

❖ To define a dominant relationship

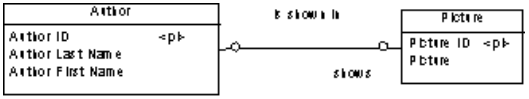
1. Open the relationship property sheet and click the Cardinalities tab.
2. Select the One to One check box.
3. Select the direction for the dominance from the Dominant Role list.
4. Click OK to return to the model.

Example

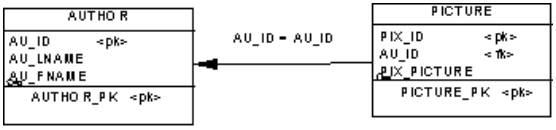
For example, in the property sheet below, the Author to Picture direction is dominant in the relationship.



The relationship pictured here shows the one-to-one relationship.



In a PDM, this relationship generates the following reference: Author is the parent table, and its primary key migrates to the Picture table as foreign key.



Changing a relationship into an associative entity

You can transform a relationship into an associative entity linked by two relationships, and then attach entity attributes to the associative entity, that you could not attach to the relationship.

The associative entity retains the name and code of the relationship, and the two new relationships handle cardinality properties.

❖ To change a relationship directly into an associative entity

1. Right-click a relationship symbol and select Change to Entity ➤ Standard from the contextual menu.

   An associative entity with two relationships replaces the relationship. The associative entity takes the name of the original relationship.

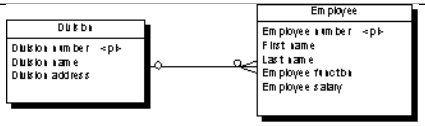
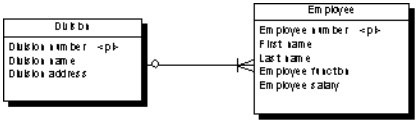
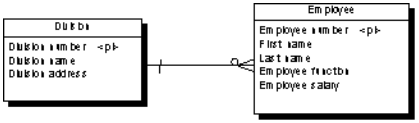
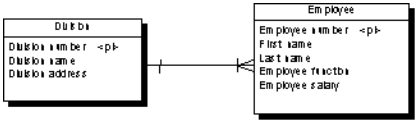
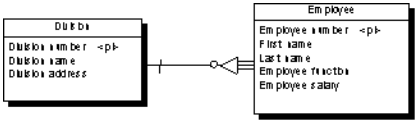
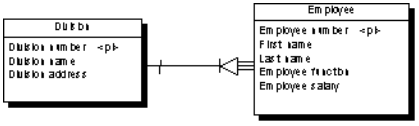
❖ To change a relationship into an associative entity using the Change to Entity Wizard

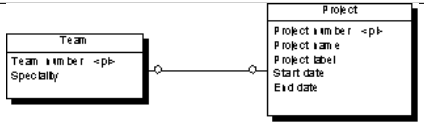
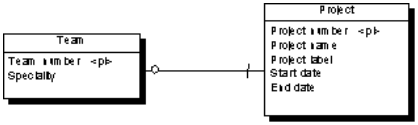
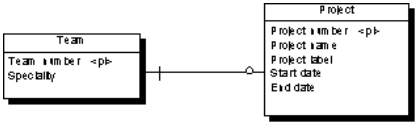
1. Right-click a relationship symbol and select Change to Entity ➤ Wizard from the contextual menu to open the Change to Entity Wizard.
2. On the Customizing Entity page, type an entity name and code, and then click Next.
3. On the first Customizing Relationship page, complete the details for the relationship that will be created between the first entity and the new entity, and then click Next.
4. On the second Customizing Relationship page, complete the details for the relationship that will be created between the new entity and the second entity, and then click Finish.

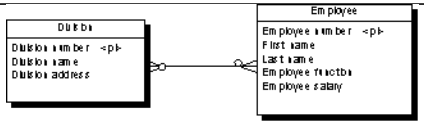
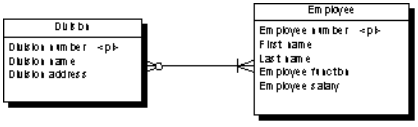
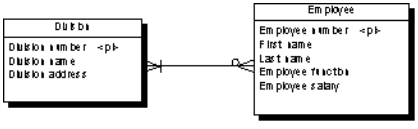
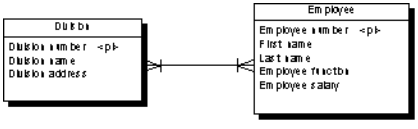
   The associative entity with two relationships replaces the relationship.

Relationship examples

This section shows the graphic representation of various relationship properties.

One-to-many relationship	Description
	Each division may have zero or more employees Each employee may belong to zero or one division
	Each division must have one or more employees Each employee may belong to zero or one division
	Each division may have zero or more employees Each employee must belong to one and only one division
	Each division must have one or more employees Each employee must belong to one and only one division
	Each division may have zero or more employees Each employee must belong to one and only one division Each employee is identified uniquely by division number and employee number
	Each division must have one or more employees Each employee must belong to one and only one division Each employee is identified uniquely by division number and employee number

One-to-one relationship	Description
	Each team works on zero or one project Each project is managed by zero or one team
	Each team works on one and one project only Each project is managed by zero or one team
	Each team works on zero or one project Each project is managed by one and one team only

Many-to-many relationship	Description
	Each division may have zero or more employees Each employee may belong to zero or more divisions
	Each division must have one or more employees Each employee may belong to zero or more divisions
	Each division may have zero or more employees Each employee must belong to one or more divisions
	Each division must have one or more employees Each employee must belong to one or more divisions

Modifying a relationship graphically

From the CDM graphic, you can modify a relationship as follows:

❖ To bend a relationship symbol

1. Press CTRL while you click a point on the relationship to insert an angle.
   The point becomes a handle.
2. Release CTRL.
3. Drag and drop the handle to create the bend.

❖ To straighten a relationship symbol

1. Click a relationship symbol that has angles.
   Handles appear on the relationship symbol.
2. Press CTRL while you click a handle.
   The handle and its angle disappear.

❖ To drag a relationship to a different entity

1. Click a relationship and drag one of its attach points to a different entity.

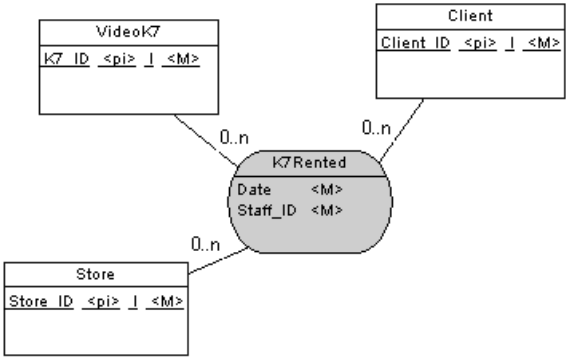
Associations and Association Links (CDM)

An **association** is a connection between entities. In the Merise modeling methodology an association is used to connect several entities that each represents clearly defined objects, but are linked by an event, which may not be so clearly represented by another entity.

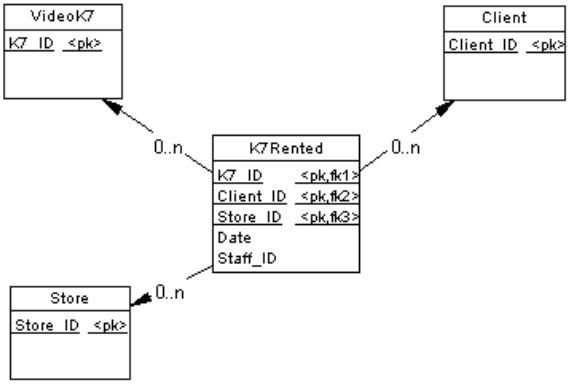
Each instance of an association corresponds to an instance of each entity linked to the association.

Example

Three entities VIDEOK7, CLIENT, and STORE contain video cassette, client, and store information. They are linked by an association which represents a video cassette rental (K7RENTAL). The K7RENTAL association also contains the attributes DATE and STAFF_ID, which give the date of the rental, and the identity of the staff member who rented out the video cassette.



When you generate a PDM, K7RENTED is generated as a table with five columns, STORE_ID,K7_ID, CLIENT_ID, DATE, and STAFF_ID.



In PowerDesigner you can use associations exclusively in your CDM, or use both associations and relationships in the same model.

Association links

An association is connected to an entity by an association link. An **Association Link** symbolizes the role and the cardinality between an association and an entity.

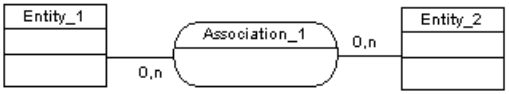
Creating an association with links

The easiest way to create an association between entities is to use the Association Link tool, which will create the association and the necessary links as well.

❖ To create an association with links

1. Click the Association Link tool in the Palette.
2. Click inside the first entity and while continuing to hold down the mouse button, drag the cursor to a second entity. Release the mouse button.

   An association symbol is created between the two entities.



Creating an association without links

You can create an association without links in any of the following ways:

- Use the Association tool in the diagram Palette.
- Select Model ➤ Associations to access the List of Associations, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select New ➤ Association.

Once you have created the association, you can link it to the relevant entities by using the Association Link tool.

☞ For general information about creating objects, see the chapter “Getting Started with PowerDesigner” in the *General Features Guide* .

Association properties

You can modify an object’s properties from its property sheet. To open an association property sheet, double-click its diagram symbol or its Browser entry in the Associations 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 association
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Number	Estimated number of occurrences in the physical database for the association (the number of records)
Generate	Indicates if the association will generate a table in a PDM
Attributes	Data item attached to an association
Rules	Business rules associated with the association

Association link properties

You can modify an object's properties from its property sheet. To open an association link property sheet, double-click its diagram symbol or its Browser entry in the Association Links folder.

The General tab contains the following properties:

Property	Description
Entity	Entity connected by the association link. You can use the tools to the right of the list to create an object, browse the complete tree of available objects or view the properties of the currently selected object
Association	Association connected by the association link
Role	Label indicating the role of the association link
Identifier	Indicates if the entity is dependent on the other entity

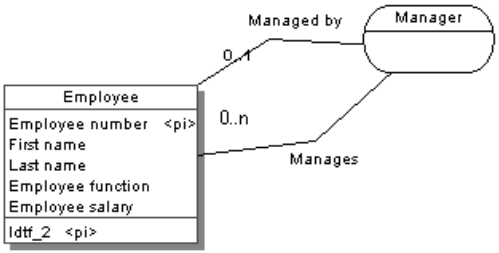
Property	Description
Cardinality	Cardinality indicates the number of occurrences (one or many) that one entity has relative to another. You define the cardinality for each association link between the association and the entity. You can choose between: ♦ 0,1 - There can be zero or one occurrence of the association in relation to one instance of the entity. The association is not mandatory ♦ 0,n - There can be zero or many occurrences of the association in relation to one instance of the entity. The association is not mandatory ♦ 1,1 - One occurrence of the entity can be related to only one occurrence of the association. The association is mandatory ♦ 1,n - One occurrence of the entity can be related to one or many occurrences of the association. The association is mandatory You can change the default format of cardinalities from the registry: `HKEY_CURRENT_USER\Software\Sybase\ PowerDesigner <version>\ ModelOptions\Conceptual Options CardinalityNotation=1 (0..1) or 2 (0,1)`
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined

Creating a reflexive association in a CDM

A reflexive association is a relationship between an entity and itself.

❖ To create a reflexive association

1. Click the Association Link tool in the Palette.
2. Click inside the entity symbol and, while continuing to hold down the mouse button, drag the cursor a short distance within the symbol, before releasing the button.
3. Drag the resulting association symbol away from entity to make clear its two links to the entity:



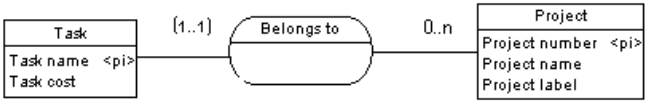
In the example above, the reflexive association Manager expresses that an employee (Manager) can manage other employees.

Defining a dependent association in a CDM

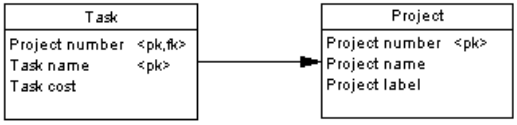
In a dependent association, one entity is partially identified by another. Each entity must have an identifier. In some cases, however, the attributes of an entity are not sufficient to identify an occurrence of the entity. For these entities, their identifiers incorporate the identifier of another entity with which they have a dependent association.

Example

An entity named Task has two entity attributes, TASK NAME and TASK COST. A task may be performed in many different projects and the task cost will vary with each project. To identify each occurrence of TASK COST the unique Task entity identifier is the compound of its *Task name* entity attribute and the *Project number* identifier from the Project entity.



When you generate a PDM, the TASK table contains the PROJECT NUMBER column as a foreign key, which is also a primary key column. The primary key therefore consists of both PROJECT NUMBER and TASK NAME columns.



Association link identifiers and associations
The same association can not have two identifier association links.

❖ To define a dependent association

1. Double-click an association link symbol to display the association link property sheet.
2. Select the Identifier check box and then click OK to return to the model..

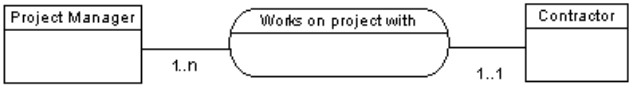
   The cardinality of the association link is enclosed in parenthesis to indicate that the association link is an identifier.

Changing an association into an associative entity

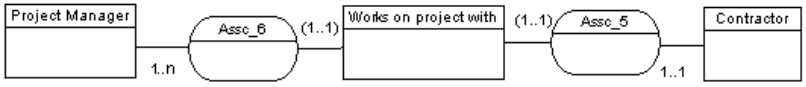
You can transform an association into an associative entity linked by two associations. The associative entity gets the name and code of the association. The two new associations handle cardinality properties.

Example

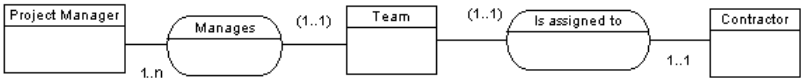
Two entities PROJECT MANAGER and CONTRACTOR are linked by the association WORKS ON PROJECT WITH:



You can represent this association with an associative entity:



The two new associations can be represented as follows:



❖ To change an association into an associative entity

1. Right-click an association symbol, and select Change to Entity from the contextual menu.

   An associative entity that is linked to two associations replaces the original association. The associative entity takes the name of the original association.

Creating an association attribute

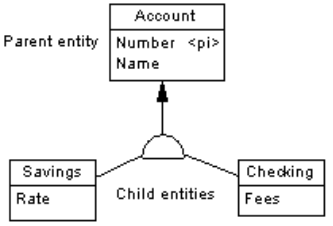
The tools used for creating association attributes on this tab are the same as those for creating entity attributes. For more information, see "Creating an attribute" on page 39.

Inheritances (CDM)

Inheritance allows you to define an entity as a special case of a more general entity. The entities involved in an inheritance have many similar characteristics but are nonetheless different. The general entity is known as a supertype (or parent) entity and contains all of the common characteristics. The special case entity is known as a subtype (or child), entity and contains all of the particular characteristics.

Between entities, it is also possible to define an inheritance link. In an inheritance link, one or more subtype (or child) entities inherit, at the physical level, all or part of the entity attributes carried by one supertype (or parent) entity.

The Account entity below represents all the bank accounts in the information system. It includes two subtypes: checking accounts and savings accounts. The notion of inheritance represents the entities Checking and Savings as specialized types of the parent entity Account.



Inheritance display

Depending on the model notation, the inheritance symbol displays the inheritance status:

IDEF1X	E/R and Merise	Description
		Standard
—		Mutually exclusive inheritance
		Complete inheritance
—		Mutually exclusive and complete inheritance

Creating an inheritance

You can create an inheritance in any of the following ways:

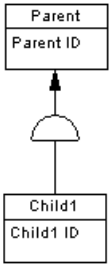
- Use the Inheritance tool in the diagram Palette (see below).
- Select Model ➤ Inheritances to access the List of Inheritances, and click the Add a Row tool. You will be required to specify a parent entity.
- Right-click the model or package in the Browser, and select New ➤ Inheritance. You will be required to specify a parent entity.

☞ For general information about creating objects, see the chapter "Getting Started with PowerDesigner" in the *General Features Guide* .

❖ To create an inheritance link using the Inheritance tool

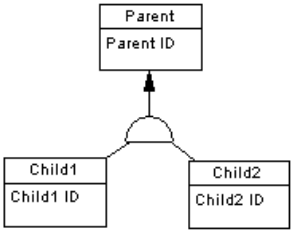
1. Click the Inheritance tool in the Palette.
2. Click the child entity, while continuing to hold down the mouse button, drag the cursor to the parent entity. Release the mouse button inside the child entity.

   The link is displayed between the two entities and has a half-circle in the middle and an arrowhead that points to the parent entity.



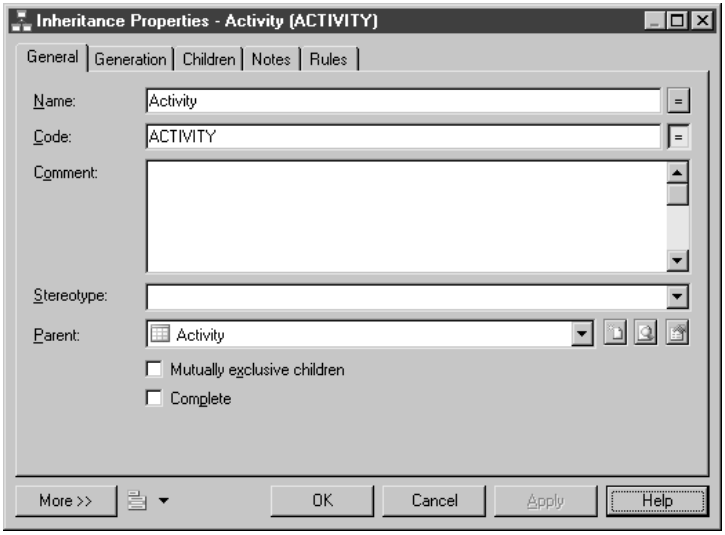
3. If you want to create additional child entities for the same link, drag and drop an inheritance link from the half-circle to the additional child entity.

   The symbol links all the child entities to the parent.



Dragging an inheritance link to a different child entity
You can change the child entity at the end of an inheritance link by clicking the inheritance link and drag one of its attach points to a different entity.

4. Double-click the new link in the diagram to display the inheritance property sheet.
5. Type an inheritance name and an inheritance code.



6. Click OK.

Inheritance properties

You can modify an object's properties from its property sheet. To open an inheritance property sheet, double-click its diagram symbol or its Browser entry in the Inheritances folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for inheritances.

Inheritance property sheet General tab

The General tab contains the following properties:

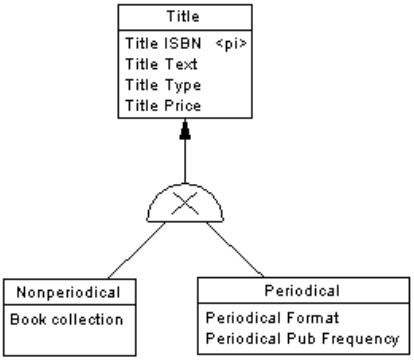
Property	Description
Name	The name of the item which should be clear and meaningful, and should convey the item's purpose to non-technical users
Code	The technical name of the item used for generating code or scripts, which may be abbreviated, and should not generally include spaces
Comment	Descriptive label for the inheritance link
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Parent	Name of the parent entity. You can use the tools to the right of the list to create an object, browse the complete tree of available objects or view the properties of the currently selected object
Mutually exclusive children	Indicates if only one child can exist for one occurrence of the parent entity.
Complete	Indicates that all instances of the parent entity (surtype) must belong to one of the children (subtypes). For example, entity Person has 2 sub-types Male and Female; each instance of entity Person is either a male or a female

Inheritance property sheet Generation tab

The generation mode defines the physical implementation of an inheritance structure. It indicates which entities in an inheritance structure should correspond to tables in the PDM.

The Generation tab contains the following properties:

Property	Description
Generation Mode	Specifies which parts of the inheritance will be generated. You can specify one or both of the following: ♦ Generate parent - Generates a table corresponding to the parent entity, which: • Inherits entity attributes of each child entity • Is affected by child entity relationships • Contains reference to any table which has a many-to-one relationship with child entity ♦ Generate children - Generates a table corresponding to each child entity. The primary key of each child table is the concatenation of the child entity identifier and the parent entity identifier. When this option is selected, you must choose between: • Inherit all attributes – Each table inherits all the entity attributes of the parent entity • Inherit only primary attributes - Each table inherits only the identifier of the parent entity Note that, if you do not select Generate Children, you can control the generation of child tables using the option Generate in the property sheet of each child entity.

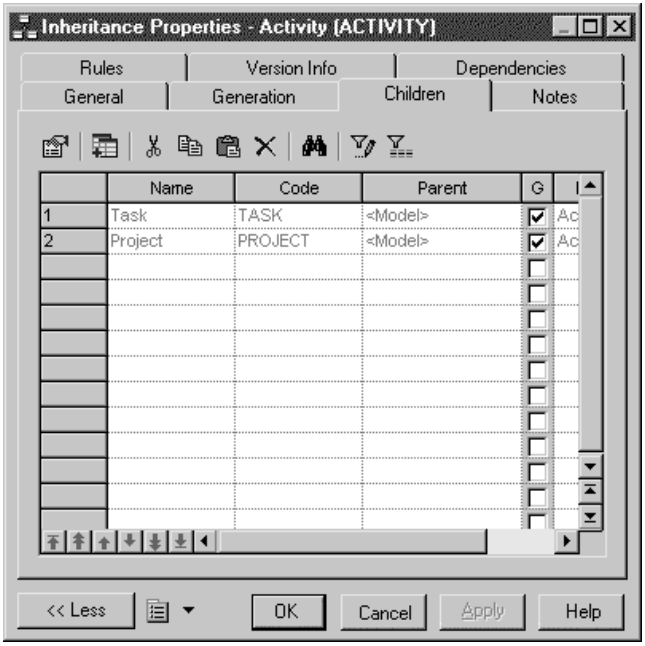
Property	Description
Specifying attributes	In the case of parent-only generation, you can choose to define a **specifying attribute**. A specifying attribute is an entity attribute that is defined for a parent entity which differentiates occurrences of each child. In the example below, the TITLE entity has two children, NONPERIODICAL and PERIODICAL. As only the parent table TITLE will be generated in a PDM, a specifying entity attribute PERIODICAL is defined for the inheritance link to differentiate between the two child entities. In the PDM, each of the child entity attributes will generate columns in the table TITLE, and the specifying entity attribute PERIODICAL will generate a corresponding column PERIODICAL. The values of this column indicate whether an instance of TITLE is a periodical or not.  The tools available on this tab for creating specifying attributes are the same as those for creating entity attributes. For more information, see "Creating an attribute" on page 39.

Inheritance property sheet Children tab

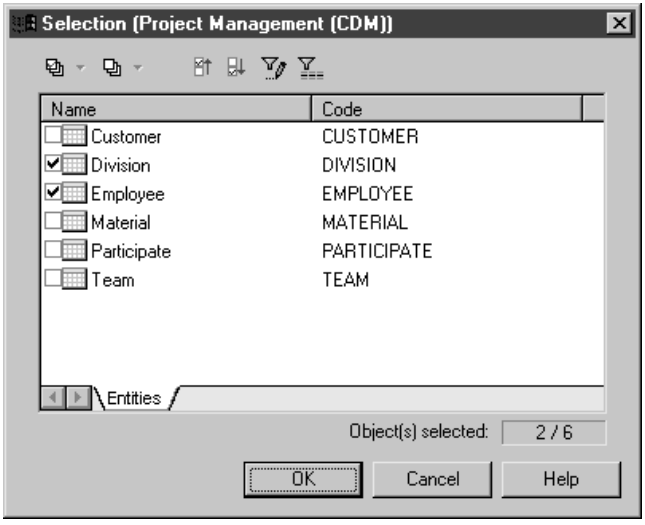
The Children tab list the child entities attached to the inheritance.

❖ To add a child entity to an inheritance link

1. Open an inheritance property sheet and click the Children tab:



2. Click the Add Children tool to open a selection window listing all the entities available in the model:



3. Select one or more entities, and then click OK to return to the inheritance property sheet.

The new entity is added to the list of child entities.

4. Click OK to return to the diagram..

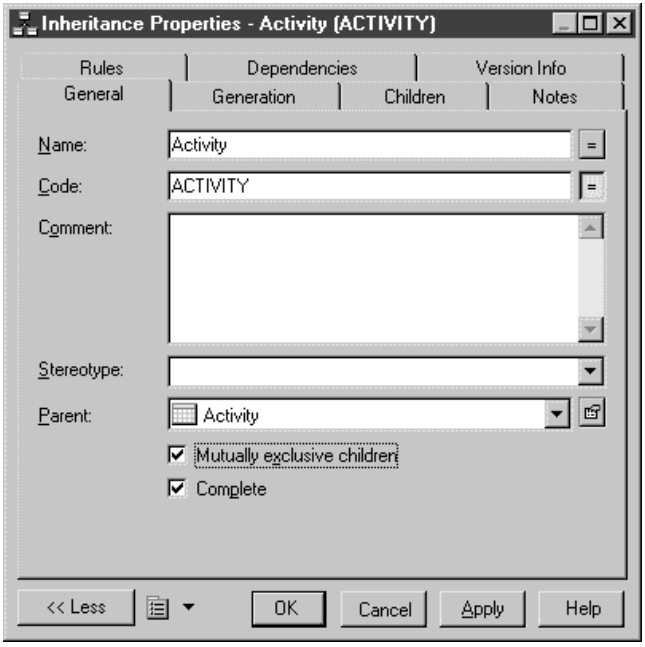
   The new child entity is linked to the inheritance symbol in the diagram.

Making inheritance links mutually exclusive

When an inheritance link is mutually exclusive between children, one occurrence of the parent entity cannot be matched to more than one child entity. This information is for documentation only and has no impact in generating the PDM.

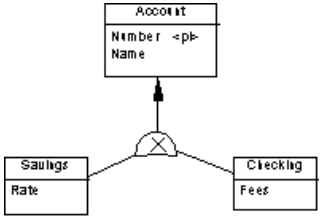
❖ To make an inheritance link mutually exclusive

1. Open an inheritance property sheet:



2. Select the Mutually Exclusive Children check box, and then click OK to return to the diagram.

   The mutually exclusive inheritance links display an X on the half-circle symbol.



In the diagram above, the inheritance link is mutually exclusive, meaning that an account is either checking or savings, but never both.

Check Parameters (CDM)

Check parameters are set of conditions which data must satisfy to remain valid. There are three types of check parameters:

Parameter type	Description	Can be attached to
Standard check parameters	Common data constraints which define a data range. For example minimum and maximum values for a column	Entity attributes Domains
Additional check parameters	SQL expression defining a data constraint using the %MINMAX%, %LISTVAL%, and %RULES% variables that are instantiated with standard parameter values	Entity attributes Domains
Validation rule	Business rule that is defined as a server expression, and is attached to one of the following listed objects	Entities Entity attributes Domains

Setting standard parameters for objects in a CDM

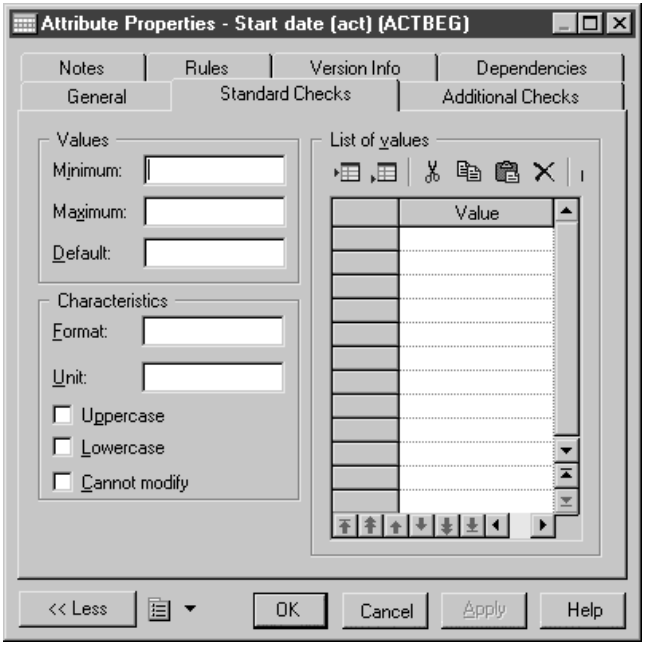
Standard parameters indicate common data constraints. The following table lists standard parameters:

Parameter	Description
Minimum	Lowest acceptable numeric value
Maximum	Highest acceptable numeric value
Default	Value assigned in absence of an expressly entered value
Unit	Standard measure
Format	Data format (for example, 9999.99)
Lowercase	Forces all alphabetical characters to lowercase
Uppercase	Forces all alphabetical characters to uppercase
Cannot Modify	Protects from changes, results in a non-modifiable column in the physical data table
List of Values	Authorized values

Parameter	Description
Label	String that identifies an authorized value in the list

❖ To set standard parameters for objects in a CDM

1. Open the property sheet of a domain or entity attribute, and click the Standard Checks tab:



2. Enter the appropriate standard parameters, and then click OK.

Defining additional check parameters for objects in a CDM

You define additional check parameters for data constraints where standard check parameters are not sufficient.

Example

An entity in a data model for a clothing shop may contain check parameters defined for an entity attribute SIZE, which depend on the check parameters defined for another entity attribute COUNTRY, as clothing size for a skirt in France may be different from the same size in the United States.

In this case an expression is required to create a constraint which uses check parameters defined for both entity attributes.

❖ To define additional check parameters for objects

1. Open the property sheet of a domain or entity attribute and click the Additional Checks tab.

2. Type a SQL expression, which may include any of the following variables:
   - %MINMAX% - Minimum and maximum values defined in Values groupbox on Standard Checks tab
   - %LISTVAL% - Customized values defined in List Values groupbox on Standard Checks tab
   - %RULES% - Validation rule expression defined on Expression tab of the Rules property sheet
3. Click OK to return to the model diagram.

Using a validation rule in check parameters in a CDM

A validation rule is a rule that validates data based on a corresponding business rule. A validation rule can be generated as a check parameter when the following conditions apply:

- Validation rule is attached to an entity, entity attribute, or domain
- Validation rule is defined as a server expression

At generation, validation rule variables are instantiated with the following values:

Variable	Value
%COLUMN%	Code of the column to which the business rule applies
%DOMAIN%	Code of the domain to which the business rule applies
%TABLE%	Code of the table to which the business rule applies
%MINMAX%	Minimum and maximum values for the entity attribute or domain
%LISTVAL%	List values for the entity attribute or domain
%RULES%	Server validation rules for the entity attribute or domain

☞ For more information on defining business rules, see the Using Business Rules in a CDM chapter.

❖ To use a validation rule in check parameters

1. Open the property sheet of a domain, entity, or entity attribute, and click the Rules tab.
2. Click the Add Objects button to open a selection box, and select a business rule in the list.
3. Click OK in each of the dialog boxes.

> **Validation rule expressions**
> You must click the Rules button to modify the expression attached to a validation rule. You can also modify validation rule expressions from the list of business rules, by clicking the Define button.

CHAPTER 3

Working with Conceptual Data Models

About this chapter

This chapter describes how to check the validity of a Conceptual Data Model and how you can import an ERwin model into a CDM. It also gives information on how you can open a ProcessAnalyst Model (PAM) into a CDM.

Contents

Topic:	page
Checking a CDM	82
Opening a PAM into a CDM	94

Checking a CDM

You can use check the validity of your CDM at any time. A valid CDM conforms to the following kinds of rules:

- Each object name must be unique
- Each entity must have at least one attribute
- Each relationship must be attached to at least one entity

> **Check your CDM before generation**
> We recommend that you check your CDM before generating another model from it. If the check encounters errors, generation will be stopped. The Check model option is enabled by default in the Generation dialog box.

❖ To check a CDM

1. Press F4, or
2. Select Tools ➤ Check Model, or
3. Right-click the diagram background and select Check Model from the contextual menu

The Check Model Parameters window opens, which allows you to specify the kinds of checks to perform, and the objects to apply them to. For detailed information about this window and correcting problems reported, see "Checking a Model" in the "Managing Models" chapter of the *General Features Guide* .

The following sections document the checks available by default for a CDM.

Business Rule checks

The following CDM model checks are made on business rules:

Check	Description and Correction
Business rule name and code uniqueness	Business rule names and codes must be unique in the model. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code
Unused business rule	The business rule you have created is not used in the model. Manual correction: Apply the business rule to an object in the model. Automatic correction: None

Package checks

The following CDM model checks are made on packages:

Check	Description and Correction
Circular dependency	A circular dependency occurs when an entity depends on another and so on until a dependency loop is created between entities. A package cannot contain circular dependencies. Manual correction: Clear the Dependent check box for the link or delete an inheritance link. Automatic correction: None.
Circularity with mandatory links	A circular dependency occurs when an entity depends on another and so on until a dependency loop is created between entities through mandatory links. Manual correction: Clear the Mandatory parent check box or delete a dependency on a relationship. Automatic correction: None.
Shortcut code uniqueness	Two shortcuts with the same code cannot be in the same namespace. Manual correction: Change the code of one of the shortcuts. Automatic correction: None.

Check	Description and Correction
Shortcut potentially generated as child table of a reference	The package should not contain associations or relationships with an external shortcut as child entity. Although this can be tolerated in the CDM, the association or relationship will not be generated in a PDM if the external shortcut is generated as a shortcut. Manual correction: Modify the design of your model in order to create the association or relationship in the package where the child entity is defined. Automatic correction: None.

Domain checks

The following CDM model checks are made on domains:

Check	Description and Correction
Domain name and code uniqueness	Domain names and codes must be unique in the model. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Inconsistency in check parameters	The values entered in the check parameters page are inconsistent for numeric and string data types: default does not respect minimum and maximum values, or default does not belong to list of values, or values in list are not included in minimum and maximum values, or minimum is greater than maximum value. Check parameters must be defined consistently. Manual correction: Modify default, minimum, maximum or list of values in the check parameters page. Automatic correction: None.
Data type precision and length	The data type precision should not be greater than or equal to the length. Manual correction: Make the data type length greater than or equal to the precision. Automatic correction: None.

Check	Description and Correction
Undefined data type	Undefined data types for domains exist. To be complete, a model should have all its domain data types defined. Manual correction: While undefined data types are tolerated, you must select data types for currently undefined data types before you can generate a PDM. Automatic correction: None.
Non-valid data type	Invalid data types for domains exist. To be complete, a model should have all its domain data types correctly defined. Manual correction: While tolerated, you must select valid data types for currently non-valid data types to generate the PDM. Automatic correction: None.

Data item checks

The following CDM model checks are made on data items:

Check	Description and Correction
Data item name and code uniqueness	Data item names and codes must be unique in the model. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Unused data items	There are unused data items. These are useless for PDM generation. Manual correction: To use a data item, add it to an entity. If you do not need an unused data item, delete it to allow PDM generation. Automatic correction: None.
Re-used data items	There are entities using the same data items. This can be tolerated if you defined this check as a warning. Manual correction: Take care to ensure consistency when defining data item properties. Automatic correction: None.

Check	Description and Correction
Data item and domain divergence	There is a divergence between data items and associated domains. This can be tolerated if you defined this check as a warning. Manual correction: Ensure consistency when defining data item properties Automatic correction: Restores divergent attributes from domain to data items (domain values overwrite data item values).
Inconsistency in check parameters	The values entered in the check parameters page are inconsistent for numeric and string data types: default does not respect minimum and maximum values, or default does not belong to list of values, or values in list are not included in minimum and maximum values, or minimum is greater than maximum value. Check parameters must be defined consistently. Manual correction: Modify default, minimum, maximum or list of values in the check parameters page Automatic correction: None.
Data type precision and length	The data type precision should not be greater than or equal to the length. Manual correction: Make the data type length greater than or equal to the precision. Automatic correction: None.
Undefined data type	Undefined data types for data items exist. To be complete, a model should have all its data items data types defined. Manual correction: While undefined data types are tolerated, you must select data types for currently undefined data types before you can generate a PDM. Automatic correction: None.
Non-valid data type	Invalid data types for data items exist. To be complete, a model should have all its data types for data items correctly defined. Manual correction: While tolerated, you must select valid data types for currently non-valid data types to generate the PDM. Automatic correction: None.

Entity checks

The following CDM model checks are made on entities:

Check	Description and Correction
Entity name and code uniqueness	Entity names and codes must be unique in the namespace. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Entity name and code length	The entity name and code length are limited to a maximum length of 254 characters specified in the naming conventions of the model options. Manual correction: Modify the entity name/code length to meet this requirement. Automatic correction: Truncates the entity name/code length to the maximum length specified in the naming conventions.
Existence of attributes	An entity must always contain at least one attribute. Manual correction: Add an attribute to the entity or delete the entity. Automatic correction: None.
Number of serial types greater than 1	An entity cannot have more than one serial type attribute. Serial types are automatically calculated values. Manual correction: Change the types of the appropriate entity attributes to have only one serial type attribute. Automatic correction: None.
Existence of identifiers	An entity must contain at least one identifier. Manual correction: Add an identifier to the entity or delete the entity. Automatic correction: None.
Existence of relationship or association link	An entity must have at least one relationship or association link. Manual correction: Add a relationship or an association link to the entity or delete the entity. Automatic correction: None.

Check	Description and Correction
Redundant inheritance	An entity inherits from another entity more than once. This is redundant and adds nothing to the model. Manual correction: Delete redundant inheritances Automatic correction: None.
Multiple inheritance	An entity has multiple inheritance. This is unusual but can be tolerated if you defined this check as a warning. Manual correction: Make sure that the multiple inheritance is necessary in your model. Automatic correction: None.
Parent of several inheritances	An entity is the parent of multiple inheritances. This is unusual but can be tolerated if you defined this check as a warning. Manual correction: Verify if the multiple inheritances could not be merged. Automatic correction: None.

Entity attribute checks

The following CDM model checks are made on entity attributes:

Check	Description and Correction
Entity attribute name and code uniqueness	Attribute names and codes must be unique in the entity. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.

Entity identifier checks

The following CDM model checks are made on entity identifiers:

Check	Description and Correction
Entity identifier name and code uniqueness	Entity identifier names and codes must be unique in the namespace. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.

Check	Description and Correction
Existence of an attribute	At least one attribute must exist for an entity identifier. Manual correction: Add an attribute to the entity identifier or delete the identifier. Automatic correction: None.
Identifier inclusion	An identifier cannot include another one. Manual correction: Delete the identifier that includes an existing identifier. Automatic correction: None.

Relationship checks

The following CDM model checks are made on relationships:

Check	Description and Correction
Relationship name and code uniqueness	Relationship names and codes must be unique in the model. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Reflexive dependency	A dependency means that one entity is defined through a relationship with another. A dependent relationship cannot therefore be reflexive. Manual correction: Change or delete the reflexive dependency. Automatic correction: None.
Reflexive mandatory relationship	A reflexive mandatory relationship exists. Manual correction: Deselect the Mandatory check boxes for the relationship to be non-mandatory. Automatic correction: None.
Bijective relationship between two entities	There is a bijective relationship between two entities when there is a two-way one to one relationship between the entities. This is equivalent to a merge of two entities. Manual correction: Merge the entities or modify the relationship. Automatic correction: None.

Check	Description and Correction
Name uniqueness constraint for many to many relationships and entities	A many-to-many relationship and an entity cannot have the same name or code. Manual correction: Change the name or code of the many-to-many relationship or the name or code of the entity. If you do not, PDM generation will rename the generated table. Automatic correction: None.
Consistency between dominant and dependent relationships	A dependent relationship between entities cannot also be a dominant relationship. Manual correction: Select the Dominant check box on the other (correct) side of the relationship. Automatic correction: None.

Association checks

The following CDM model checks are made on associations:

Check	Description and Correction
Association name and code uniqueness	Association names and codes must be unique in the namespace. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Association has at least two links	An association is isolated and therefore does not define a relationship between entities. Manual correction: Define at least two links between the isolated association and one or several entities. Automatic correction: None.
Association has two links with identifier link	An identifier link introduces a dependency between two entities. An association with this type of link must be binary. Manual correction: Delete the unnecessary links or clear the Identifier check box for a link. Automatic correction: None.

Check	Description and Correction
Association has no more than one identifier link	An identifier link introduces a dependency between two entities. There can only be one identifier link between two entities otherwise a circular dependency is created. Manual correction: Clear the Identifier check box for one of the links. Automatic correction: None.
Absence of properties with identifier links	An association with an identifier link cannot have any properties. Manual correction: Move the association properties into the dependent entity (the one linked to the association with an identifier link). Automatic correction: None.
Bijective association between two entities	There are bijective associations between two entities when a two-way one to one association between the entities exist. This is equivalent to a merge of two entities. Manual correction: Merge the entities or modify the cardinality links. Automatic correction: None.
Maximum cardinality links	An association with more than two links can only have links with a maximum cardinality greater than one. Manual correction: Change the maximum cardinality of such links to be greater than 1. Automatic correction: None.
Reflexive identifier links	An identifier link introduces a dependency between two entities. An association with this type of link cannot therefore be reflexive. Manual correction: Change the relationship between the entities or clear the Identifier check box for a link. Automatic correction: None.
Name unicity constraint between many-to-many associations and entities	A many-to-many association and an entity cannot have the same name or code. Manual correction: Change the name or code of the many-to-many association or the name or code of the entity. If you do not, PDM generation will rename the generated table. Automatic correction: None.

Inheritance checks

The following CDM model checks are made on inheritances:

Check	Description and Correction
Inheritance name and code uniqueness	Inheritance names and codes must be unique in the model. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.
Existence of inheritance link	An inheritance must have at least one inheritance link, from the inheritance to the parent entity. Manual correction: Define the inheritance link or delete the inheritance. Automatic correction: None.

File object checks

The following CDM model checks are made on file objects:

Check	Description and Correction
Embedded file name uniqueness	Embedded file names must be unique in the model. Manual correction: Modify the duplicate name Automatic correction: Appends a number to the duplicate name.
Existence of external file location	External file objects must have a valid location path. Manual correction: Define a valid path location. Automatic correction: None.

Replication checks

The following CDM model checks are made on replications:

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

Extended object/link checks

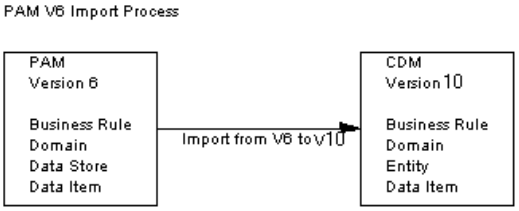
The following CDM model checks are made on object/link:

Check	Description and Correction
Extended object/link name and code uniqueness	Extended object and link names and codes must be unique in the namespace (model or package). Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code.

Opening a PAM into a CDM

You can recover process model information into a conceptual model by opening a PAM into a CDM.

You will then retrieve PAM objects into your CDM.



The open process translates PAM objects into CDM objects as follows:

Object in a PAM	Imported object in a CDM
Business rule	Business rule
Domain	Domain
Data store	Entity
Data item	Data item

When opening a PAM, you are asked to choose between opening into a BPM or opening into a CDM knowing that opening a PAM into a BPM allows you to recover processes.

☞ For more information on opening a PAM into a BPM, see the *BPM User's Guide* .

❖ To open a PAM into a CDM

1. Select File ➤ Open.

   An open file dialog box is displayed.

2. Select or browse to the folder that contains the PAM file.

3. Select ProcessAnalyst Model (*.pam) file from the Files As Type dropdown list box to display only PAM files.

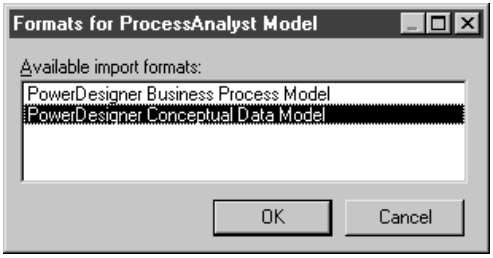
   The available PAM files are listed.

4. Select a file.

5. Click Open.

   The Formats for ProcessAnalyst Model window is displayed.

6. Select PowerDesigner Conceptual Data Model and click OK.



The Output window displays a message following the successful import and the default diagram of the model is displayed. Each imported object type can be viewed from its respective list.

CHAPTER 4

Generating from a Conceptual Data Model

About this chapter

This chapter describes how you can generate a CDM, a PDM, and an OOM from a Conceptual Data Model.

Contents

Topic:	page
Generation basics	98
Generating a Conceptual Data Model from a Conceptual Data Model	99
Generating a Physical Data Model from a Conceptual Data Model	102
Generating an Object Oriented Model from a Conceptual Data Model	116

Generation basics

You can generate a CDM, a PDM, or an OOM from a CDM.

☞ For more information on model to model or inter-model generation common features, see the “Linking and Synchronizing Models in PowerDesigner” chapter in the *General Features Guide* .

Generating a Conceptual Data Model from a Conceptual Data Model

This section explains how to generate a Conceptual Data Model from a Conceptual Data Model.

Why generate a CDM into a CDM?

You can generate a CDM into a CDM when you need to keep two models synchronized during the design process.

This kind of generation allows you to create a copy of a given model and define generation links between objects in the source CDM and their equivalent in the generated CDM. When changes are made to the source model, they can then be easily propagated to the generated models using the Update Existing Model generation mode.

The generated model is the one that usually contains more information.

Generating and updating a CDM

To generate a CDM, you must choose between one of the following methods:

- Generate new Conceptual Data Model
- Update existing Conceptual Data Model

You must indicate the following parameters when you generate a new CDM:

Parameter	Description
Name	File name for the resulting CDM
Code	Reference code for the resulting CDM
Configure Model Options	Lets you specify model options for the model to be generated. For example links to requirements in the origin model are preserved, by default, in the generated model, unless you clear the Enable Links Requirements checkbox in the Model Options dialog box before starting the generation

☞ For more information, see the “Linking and Synchronizing Models in PowerDesigner” chapter in the *General Features Guide* .

Defining CDM generation options

You can set the following general generation options:

Option	Description
Check Model	When selected, verifies the model before generating the CDM, and stops generation if an error is found
Save Generation Dependencies	When selected, PowerDesigner keeps track of the identity of the origin of each generated object. It is useful when merging two CDM which have been generated from the same CDM. Objects can be compared and recognized as the same object, even if the object has been modified in the merged CDM. If not selected, origin objects have no link with generated objects
CDM/CDM Mapping	When selected, allows to define the current CDM as the data source of the generated CDM and to create object correspondence between source and target CDM. If you select the CDM/CDM mapping option, the Save Generation Dependencies option is automatically selected because CDM/CDM mapping uses object identifying numbers. You cannot deselect the Save Generation Dependencies option as long as the CDM/CDM mapping option is selected
Enable transformations	This button is used to activate transformations during generation. When you click this button, the Pre-generation tab is displayed if the source model contains transformations. You can select the transformations to execute before generation. The Extended Model Definitions tab also is displayed for you to select extended model definition files to attach to the generated model. These files may contain post-generation transformations, in this case, the Post-Generation tab is displayed to let you select the transformations you want to be executed in the generated model. If the generation is an update, and the generated model contains extended model definitions with post-generation transformations, the Post-generation tab automatically is displayed as soon as you click the Enable Transformations button

Check model before generation
If you select the Check Model option, the procedure to generate a CDM starts by checking the validity of the CDM . A CDM results when no errors are found. You can set check options by selecting Tools ➤ Check Model.

Generating a Physical Data Model from a Conceptual Data Model

This section explains how to generate a Physical Data Model from a Conceptual Data Model.

Generating PDM objects

When you generate a physical data model (PDM) from a CDM, PowerDesigner converts CDM objects and data types to PDM objects and data types supported by the current DBMS.

Converting CDM to PDM objects

PDM generation converts conceptual objects into physical objects as follows:

CDM objects	Generated object in a PDM
Entity	Table
Entity attribute	Table column
Primary identifier	Primary or foreign key depending on independent or dependent relationship
Identifier	Alternate key
Relationship	Reference

Changing the name of a column automatically
Two columns in the same table cannot have the same name. If column names conflict due to foreign key migration, PowerDesigner automatically renames the migrated columns. The new name is composed of the first three letters of the original entity name followed by the code of the attribute.

Generating keys from identifiers

Primary identifiers generate primary and foreign keys in the PDM. Identifiers that are not primary identifiers generate alternate keys.

The type of key that is generated in the PDM depends on the cardinality and type of dependency defined for a relationship in the CDM.

A **primary key** is a column or columns whose values uniquely identify a row in a table.

A **foreign key** is a column or columns that depend on and migrate from a primary key column in another table.

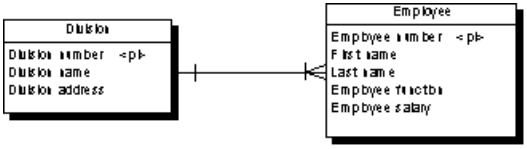
An **alternate key** is a column or columns whose values uniquely identify a row in a table, and is not a primary key.

Independent one-to-many relationships

In independent one-to-many relationships, the primary identifier of the entity on the one side of the relationship becomes a:

- Primary key in the table generated by the entity on the one side of the relationship
- Foreign key in the table generated by the entity on the many side of the relationship

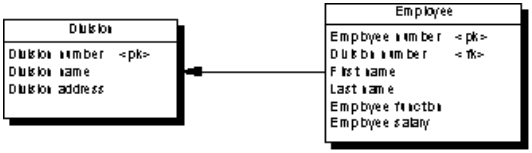
The CDM below shows an independent relationship. Each division contains one or more employees:



The CDM above generates two tables:

Table	Primary key	Foreign key
Division	Division number	—
Employee	Employee number	Division number

The following PDM results from generation:

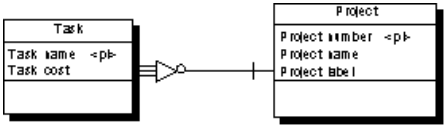


Dependent one-to-many relationships

In dependent relationships, the primary identifier of the nondependent entity becomes a primary/foreign key in the table generated by the dependent entity.

The migrated column is integrated into the primary key if it already exists.

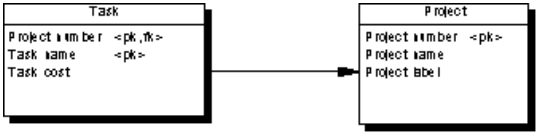
The CDM below shows a dependent relationship. Each task must have a project number.



The CDM above generates two tables:

Table	Primary key	Foreign key
Project	Project number	—
Task	Project number/Task number	Project number

The following PDM results from generation.



Independent many-to-many relationships

In independent many-to-many relationships, the primary identifiers of both entities migrate to a join table as primary/foreign keys. The CDM below shows an independent relationship. Each employee can be a member of one or more teams, and each team can have one or more employees as members.

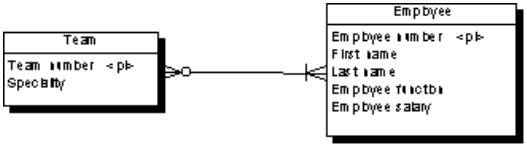
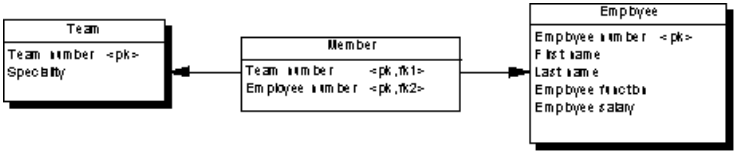


Table	Primary key	Foreign key
Team	Team number	—
Employee	Employee number	—
Member	Team number/Employee number	Team number/Employee number

The following PDM results from generation.



Independent one-to-one relationships

In independent one-to-one relationships, the primary identifier of one entity migrates to the other generated table as a foreign key.

Generating tables from entities with inheritance links

Two properties influence the generation of tables from entities with inheritance links.

Object	Property	When selected generates
Entity	Generate table	Table for the entity (parent or child)
Inheritance	Generation mode	Parent and/or children as indicated

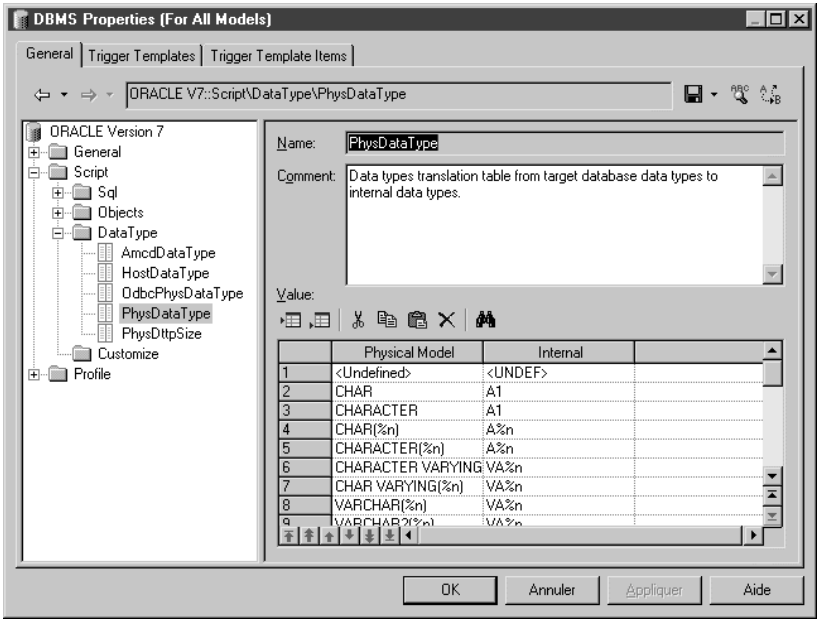
Converting CDM to PDM data types

PowerDesigner supports both conceptual and physical data types. Data types that you select in the CDM are not always supported by the current DBMS. In this case, the data type is converted to a data type supported by the current DBMS when you generate the PDM.

Accessing the DBMS to define data type conversions

Each DBMS contains rules for data type conversion.

The correspondence between the PowerDesigner data type and the physical data type for the current DBMS is listed under the Data Type node, under the Script node in the DBMS tree view.



The Physical Model column lists the data types for the current DBMS and the Internal column lists the PowerDesigner equivalent. You can modify the definition of an existing physical data type by making modifications directly to either of the two columns.

Modifying the DBMS
You should only modify a copy of the DBMS shipped with PowerDesigner. Once a modification has been committed in the DBMS you cannot return to the default value for the data type.

☞ For more information on modifying a DBMS, see chapter DBMS Basics in the Physical Data Model User's Guide.

❖ To access a DBMS

1. Select Tools ➤ Resources ➤ DBMS to display the List of DBMS.
2. Select a DBMS from the list.

   Click the Properties tool.

   or

   Double-click a DBMS name in the list.

   The DBMS property sheet is displayed.
3. From the General page, expand the Script node, then expand the Data Type node.
4. Select the item PhysDataType.

   The values for the DBMS data types and the corresponding PowerDesigner data types appear in the Value pane to the left of the tree view.
5. Click OK.

Converting conceptual data types

The following table lists the conceptual data types to which the DBMS assigns conversions:

Numeric data types

Conceptual data type	Code in DBMS	What it stores	Conversion example for SQL Anywhere
Integer	I	32-bit integer	integer
Short Integer	SI	16-bit integer	smallint

Conceptual data type	Code in DBMS	What it stores	Conversion example for SQL Anywhere
Long Integer	LI	32-bit integer	integer
Byte	BT	256 values	smallint
Number	N	Numbers with a fixed decimal point	numeric
Decimal	DC	Numbers with a fixed decimal point	decimal
Float	F	32-bit floating point decimal numbers	float
Short Float	SF	Less than 32-bit floating point decimal number	real
Long Float	LF	64-bit floating point decimal numbers	double
Money	MN	Numbers with a fixed decimal point	numeric
Serial	NO	Automatically incremented numbers	numeric
Boolean	BL	Two opposing values (true/false; yes/no; 1/0)	numeric(1)

Character data types

Conceptual data type	DBMS	What it stores	Conversion example for SQL Anywhere
Characters	A	Character strings	char
Variable Characters	VA	Character strings	varchar
Long Characters	LA	Character strings	varchar
Long Var Characters	LVA	Character strings	long varchar
Text	TXT	Character strings	long varchar
Multibyte	MB	Multibyte character strings	char
Variable Multibyte	VMB	Multibyte character strings	varchar

Time data types

Conceptual data type	DBMS	What it stores	Conversion example for SQL Anywhere
Date	D	Day, month, year	date
Time	T	Hour, minute, and second	time
Date & Time	DT	Date and time	timestamp
Timestamp	TS	System date and time	timestamp

Other data types

Conceptual data type	DBMS	What it stores	Conversion example for SQL Anywhere
Binary	BIN	Binary strings	binary
Long Binary	LBIN	Binary strings	long binary
Image	PIC	Images	long binary
Bitmap	BMP	Images in bitmap format (BMP)	long binary
OLE	OLE	OLE links	long binary
Other	*	User-defined data type	char(10)
Undefined	<UNDEF>	Not yet defined data type	<undefined>

Generating and updating a PDM

To generate a PDM, you must choose between one of the following methods:

- Generate new Physical Data Model
- Update existing Physical data Model

You must indicate the following parameters when you generate a new PDM:

Parameter	Description
DBMS	Database Management System definition (DBMS) for the resulting PDM
Share	The resulting PDM uses a shared DBMS definition file stored in the DBMS library
Copy	The resulting PDM uses a copy of the DBMS definition file stored in the model
Name	File name for the resulting PDM
Code	Reference code for the resulting PDM
Configure Model Options	Lets you specify model options for the model to be generated. For example links to requirements in the origin model are preserved, by default, in the generated model, unless you clear the Enable Links Requirements checkbox in the Model Options dialog box before starting the generation

☞ For more information, see the "Linking and Synchronizing Models in PowerDesigner" chapter in the *General Features Guide* .

Configuring the generated PDM options

When you configure the options of a PDM to generate, you may define options diverging from the CDM options.

To avoid conflicts, PowerDesigner applies the following rule for default values of PDM options: an option defined for the generated PDM should respect the equivalent option of the CDM.

Equivalent options are model options found in both models. The Enforce non-divergence option exists in both CDM and PDM.

CDM option	PDM option	Result in generated PDM
✔ Enforce non-divergence	—	Enforce non-divergence in model according to CDM options. Columns attached to the domain cannot have divergent definitions
—	✔ Enforce non-divergence	Enforce non-divergence in model according to PDM options defined using the Configure Model Options feature

References unique code

Unique Code for references is not selected by default in the PDM options. However, if you select Unique Code for references in the PDM options, relationships are renamed during the generation of a CDM to a PDM.

Options with no equivalent, like Use Data Type Full Name in the CDM without any corresponding option in a PDM, are generated using default values found in the registry.

Defining PDM generation options

You can set the following general generation options from the Detail page:

Option	Description
Check Model	When selected, checks the model before generating the PDM, and stops generation if an error is found
Save Generation Dependencies	When selected, PowerDesigner keeps track of the identity of the origin of each generated object. This is useful when merging two PDM which have been generated from the same CDM. Objects can be compared and recognized as the same object, even if the object has been modified in the target PDM. If not selected, origin objects have no link with generated objects
Convert Names into Codes	When selected, object codes are generated from names using the corresponding conversion script. This is useful when generating models with very different naming conventions . For example a Java class attribute code may be "customerName" whereas a PDM table column code may be "CUSTOMER_NAME". With this option selected, both objects will have their codes generated from their names. If you do not select this option, generated object codes will be copied from original object codes
Rebuild Triggers	When selected, allows to build triggers with preservation option, after the PDM generation. Rebuild takes place after merge if you are updating an existing PDM
Enable transformations	This button is used to activate transformations during generation. When you click this button, the Pre-generation tab is displayed if the source model contains transformations. You can select the transformations to execute before generation. The Extended Model Definitions tab also is displayed for you to select extended model definition files to attach to the generated model. These files may contain post-generation transformations, in this case, the Post-Generation tab is displayed to let you select the transformations you want to be executed in the generated model. If the generation is an update, and the generated model contains extended model definitions with post-generation transformations, the Post-generation tab automatically is displayed as soon as you click the Enable Transformations button

☞ For more information conversion scripts, see section Using a conversion script in chapter Managing models in the *General Features Guide* .

Check model before generation
If you select the Check Model option, the procedure to generate a PDM starts by checking the validity of the CDM . A PDM results when no errors are found. You can set check options by selecting Tools ➤ Check Model.

Table parameters

The following parameter defines the naming convention for tables:

Parameter	Description
Table Prefix	Prefix for the names of tables generated from entities in the CDM

Reference parameters

The following parameters define default referential integrity constraints:

Parameter	Description
Update Rule	Default update constraint for referential integrity
Delete Rule	Default delete constraint for referential integrity

Foreign key name

By default, the name of the migrated foreign keys generated in a PDM is identical to the name of the primary key. In case of column name conflict, the first 3 letters of the name of the parent table are added before the name of the migrated column.

You can modify this default behavior and use templates for naming the migrated foreign keys. In the Reference groupbox, you can select one of the predefined templates in the FK Column Name Template list. These templates are saved in the registry, you can modify them, but their default value is preserved.

You can also create your own foreign key naming templates by typing a template directly in the FK Column Name Template box using the following variables:

Variable	Value
%REFR%	Name / Code of the reference
%PARENT%	Name / Code of the parent table
%COLUMN%	Name / Code of the parent column
%KEY%	Name / Code of the key constraint attached to the reference
%CONSTNAME%	Name of the key constraint attached to the reference (same as %KEY%)
%PROLE%	Role of the entity that generated the parent table, this variable proceeds from the conceptual environment. If no role is defined on the relationship or association, %PROLE% takes the content of %PARENT% to avoid generating a column with no name

You can use the default PowerDesigner formatting variables in the template as explained in section List of PowerDesigner formatting variables in chapter Variables in PowerDesigner in the *PDM User's Guide* .

You can also use square brackets [ ] for evaluating optional variables as explained in "Optional strings and variables" in the "DBMS Reference Guide" chapter in the *Advanced User Documentation* .

Customized naming templates reappear in the generation dialog box the next time you open it, but are not saved in the registry thus not added to the dropdown list of predefined templates.

For example, if you define the template %PROLE%, the name of the foreign key will be made of the role beside the entity that generated the parent table containing the primary key. If the parent table contains multiple keys, the foreign key names will be automatically renamed.

Other example:

```
[%PROLE%=%PARENT%?%.3:PARENT%_%COLUMN%:%PROLE%]
```

This checks the %PROLE% value; if it is equal to the parent name (which is the replacement value) then the template "%.3:PARENT%_%COLUMN%" will be used, otherwise template "%PROLE%" will be used because the user has entered a parent role for the relationship.

In the Reference groupbox, you can select the Always Use Template radio button to always use the default or customized template for naming FK

columns, even if there is no conflict with an existing column.

Index parameters

The following parameters define naming conventions for indexes:

Parameter	Description
PK index names	Naming convention for primary keys, by default %TABLE%_PK
Key index names	Naming convention for alternate keys, by default %TABLE%_AK
FK index names	Naming convention for foreign keys, by default %REFR%_FK
FK threshold	Minimum number of estimated occurrences of an entity necessary to create an index on a foreign key

If you enter a value for the index threshold, indexes on foreign keys are only generated if the number of estimated occurrences is greater than the threshold. The estimated number of occurrences is an entity property that you can include in the entity definition.

If the entity has no specified number of occurrences, the foreign key indexes are generated by default.

PK index name variables

You can use the following variables in the PK index names and Key Index Names fields:

Variable	Value
%TABLE%	Generated code of the table. This is the table code that is generated in the database. It may or may not be truncated if the code contains characters not supported by the DBMS
%TNAME%	Table name
%TCODE%	Table code
%TLABL%	Table comment

FK index name variables

You can use the following variables in the FK index name field. The generated code of a variable is the code defined in the object property sheet, but which may or may not be truncated when generated if the code contains characters not supported by the DBMS

Variable	Value
%REFR%	Generated code of the reference

Variable	Value
%REFNAME%	Reference name
%REFRCODE%	Reference code
%PARENT%	Generated code of the parent table
%PNAME%	Parent table name
%PCODE%	Parent table code
%CHILD%	Generated code of the child
%CNAME%	Child table name
%CCODE%	Child table code
%PQUALIFIER%	Parent table qualifier
%CQUALIFIER%	Child table qualifier

Preserving a modified primary key
If you modify a primary key in a PDM, then regenerate the PDM from a CDM, the modified primary key is not preserved. If you want to preserve a modified primary key, you need to modify the identifier in the CDM before regeneration.

Generating an Object Oriented Model from a Conceptual Data Model

This section explains how to generate an Object Oriented Model from a Conceptual Data Model.

Generating OOM Objects

When you generate an Object Oriented Model (OOM) from a CDM, PowerDesigner converts CDM objects into specified object language objects as follows:

CDM Objects	Generated object in an OOM
Entity	Class
Attribute	Attribute
Association	Relationship or association
Binary association with attributes	Association class
Inheritance	Generalization

Persistent entities

All entities are generated as persistent classes with the "Generate table" persistence mode.

When the Generate check box of an entity is not selected, the generated class has the "Migrate columns" persistence mode.

OOM naming conventions
If the code of the generated OOM objects does not correspond to the target language naming conventions, you can define a code naming convention script to convert object names into codes. For more information on conversion scripts, see section .convert_name & .convert_code macros in chapter Managing Models in the *General Features Guide*.

Converting CDM to OOM data types

PowerDesigner supports both conceptual and object language data types.

The following table lists the conceptual data types to which the object language definition assigns conversions:

Conceptual data type	Code in CDM	What it stores	Conversion example with Java
Characters	A	Character strings	Char
Boolean	BL	Two opposing values (true/false; yes/no; 1/0)	Boolean
Byte	BT	256 values	Byte
Short Integer	SI	16-bit integer	Short
Long Integer	LI	32 bit integer	Long integer
Integer	I	32 bit integer	Integer
Float	F	32 floating decimal numbers	Float
Number	N	Numbers with a fixed decimal point	Double
Text	TXT	Character strings	*

Generating and updating an OOM

To generate an OOM, you must choose between one of the following methods:

- Generate new Object-Oriented Model
- Update existing Object-Oriented Model

You must indicate the following parameters when you generate a new OOM:

Parameter	Description
Object Language	Target object language
Share	Object language for the resulting OOM uses a shared Object language definition file stored in the Object language library
Copy	Object language for the resulting OOM uses a copy of the Object language definition file stored in the Object language library
Name	File name for the resulting OOM
Code	Reference code for the resulting OOM
Configure Model Options	Lets you specify model options for the model to be generated. For example links to requirements in the origin model are preserved, by default, in the generated model, unless you clear the Enable Links Requirements checkbox in the Model Options dialog box before starting the generation

☞ For more information, see the "Linking and Synchronizing Models in PowerDesigner" chapter in the *General Features Guide* .

Configuring the generated OOM options

When you configure the options of an OOM to generate, you may define options diverging from the CDM options.

To avoid conflicts, PowerDesigner applies the following rule for default values of OOM options: an option defined for the generated OOM should respect the equivalent option of the CDM.

Equivalent options are model options found in both models. The Enforce non-divergence option exists in both CDM and OOM.

CDM option	OOM option	Result in generated OOM
✔ Enforce non-divergence	—	Enforce non-divergence in model according to CDM options
—	✔ Enforce non-divergence	Enforce non-divergence in model according to OOM options defined using the Configure Model Options feature

Options with no equivalent, like Use Data Type Full Name in the CDM

without any corresponding option in an OOM, are generated using default values found in the registry.

Defining OOM generation options

You can set the following general generation options from the Detail page:

Option	Description
Check Model	When selected, checks the model before generating the OOM, and stops generation if an error is found
Save Generation Dependencies	When selected, PowerDesigner keeps track of the identity of the origin of each generated object. It is useful when merging two OOM which have been generated from the same CDM. Objects can be compared and recognized as the same object, even if the object has been modified in the target OOM. If not selected, origin objects have no link with generated objects
OOM/CDM Mapping	When selected, allows to define the current CDM as the data source of the generated OOM and to create object correspondence between source CDM and target OOM. If you select the OOM/CDM mapping option, the Save Generation Dependencies option is automatically selected because OOM/CDM mapping uses object identifying numbers. You cannot deselect the Save Generation Dependencies option as long as the OOM/CDM mapping option is selected
Convert Names into Codes	When selected, object codes are generated from names using the corresponding conversion script. This is useful when generating models with very different naming conventions since conversion macros are used to modify codes. For example a Java class attribute code may be “customerName” whereas a PDM table column may be “CUSTOMER_NAME”. With this option selected, both objects will have their codes generated from their names. If you do not select this option, generated object codes will be copied from original object codes

Option	Description
Enable transformations	This button is used to activate transformations during generation. When you click this button, the Pre-generation tab is displayed if the source model contains transformations. You can select the transformations to execute before generation. The Extended Model Definitions tab also is displayed for you to select extended model definition files to attach to the generated model. These files may contain post-generation transformations, in this case, the Post-Generation tab is displayed to let you select the transformations you want to be executed in the generated model. If the generation is an update, and the generated model contains extended model definitions with post-generation transformations, the Post-generation tab automatically is displayed as soon as you click the Enable Transformations button
Class Prefix	Prefix for a class. It can help identify a class in a model

☞ For more information conversion scripts, see section Using a conversion script in chapter Managing models in the *General Features Guide* .

Check model before generation
If you select the Check Model option, the procedure to generate an OOM starts by checking the validity of the CDM . An OOM results when no errors are found. You can set check options by selecting Tools ➤ Check Model.

Index

A

association 62
  cardinality 65
  change to entity 67
  create 63
  define 67
  dependent 66
  entity 67
  entity attribute 67
  properties 63
  reflexive 65
association (CDM)
  check model 90
association attribute
  create 67
association display preferences 17
association link
  cardinality 65
  define 63
  properties 64
association link display preferences 18
attribute 39
  associate domain 42
  association 67
  create 39
  data type 41
  delete 43
  domain 42
  entity 43
  identifier 43
  list 43
  mandatory 42
  optional 42
  properties 39
  reuse 43
attribute (CDM)
  enforce coherence 12

B

bibliography vii
binary data type 33, 108
bitmap data type 33, 108
boolean data type 31, 106
business rule
  apply 79
  domain 34
  expression 79
  validation rule 79
business rule (CDM)
  check model 82
byte data type 31, 106

C

cardinality
  association 65
  change format 65
  change from the registry 65
  define 49
  define for an association link 65
  many to many 49
  many to one 49
  one to many 49
  one to one 49
  relationship 49, 58
CDM
  check model 82
  create 4
  data type 106
  define 2
  domain 28
  enable transformations 100
  enforce domain 8
  Entity Relationship Diagram 2
  ERD 2
  error 27, 52
  extended dependency 19
  extended model definition 19
  generate 98, 99
  generate links requirements 99, 109, 118
  generation option 100
  generation parameter 99
  logical model 2
  modify 5

- object 102, 116
- open PAM 94
- option 12, 27, 52
- package 6
- property 5
- role 2
- validation 82

CDM environment 8
CDM/CDM Mapping option 100
Change to Entity Wizard 58
check model
- association (CDM) check 90
- business rule (CDM) 82
- CDM 82
- data item check 85
- domain check 84
- entity attribute check 88
- entity check 87
- entity identifier check 88
- extended link 93
- extended object 93
- file object 92
- inheritance check 92
- package (CDM) check 83
- relationship 89
- replication (CDM) check 92

check parameter
- column 77, 79
- data type 25, 41
- define 77
- domain 34, 77, 79
- properties 77
- standard 77
- type 77
- validation rule 77, 79

client
- expression 79

code
- data item 27
- option 52
- package (CDM) 7
- relationship 52
- unique 27, 52

code naming convention 116
column
- check parameter 77, 79
- generate 102
- rename 102
- validation rule 79
- variable 79

composite view 6
conceptual diagram
- association 62
- attribute 39
- create 23
- data item 24
- entity 36
- identifier 44
- inheritance 68
- relationship 47

configure
- OOM options 118
- PDM options 110

convert
- CDM to OOM objects 116
- CDM to PDM objects 102, 105
- data type 105, 116

D

data item 24
- associate domain 26
- association 67
- check model 85
- code 27
- copy 37
- create 24
- data type 25
- delete 43
- domain 26
- error 27
- non-divergence 34
- option 37
- properties 24
- reuse 8, 27, 43
- unique 8
- unique code 27

data type 29, 30
- attribute 41
- binary 33, 108
- bitmap 33, 108
- boolean 31, 106
- byte 31, 106
- character 32
- conceptual 106

conversion in DBMS 105
convert to OOM 116
convert to PDM 105
data item 25
date 32, 108
decimal 31, 106
default 8
define 105
domain 29, 34
float 31, 106
image 33, 108
integer 31, 106
length 29, 30
map to OOM 116
map to PDM 105
money 31, 106
number 31, 106
OLE 33, 108
precision 29, 30
serial 31, 106
show full name 8
specifying 25, 41
time 32, 108
undefined 25, 29, 33, 41, 108
Data Type Full Name option 12
date data type 32, 108
DBMS data type conversion 105
decimal data type 31, 106
Default Data Type option 12
define 47
delete data item 43
dependent association 66
display preference
package (CDM) 7
display preferences
association 17
association link 18
entity 14
inheritance 19
relationship 18
diverge from domain 12
domain
attribute 42
business rule 34
CDM 28
check parameter 34, 77, 79
create 28
data item 26
data type 29, 30, 34
define 28
enforce 34
entity attribute 34
length 29, 30
modify properties 33
non-divergence 34
precision 29, 30
properties 28
use 34
validation rule 79
variable 79
domain (CDM)
check model 84
diverge from 12
enforce coherence 8, 12
Domain and Attribute option 12
dominant relationship 56

E

E/R + Merise 9
enforce domain 34
entity 36
add to inheritance 73
association 67
associative 58
attribute 43
change from association 67
change from relationship 58
check model 87
copy 37
create 36
delete 43
entity attribute 37
generate table 105
identifier 37, 43
inheritance 105
properties 36
wizard 58
entity attribute
add to identifier 45
association 67
check model 88
copy 37
domain 34
identifier 37

inheritance 71
properties 71
relationship 58
specifying entity attribute 71
entity display preferences 14
entity identifier check 88
entity/relationship 9
notation 8
ERD
CDM 2
Entity Relationship Diagram 2
error message
data item 27
relationship 52
unique code 27, 52
expression
business rule 79
client 79
server 79
extended dependency (CDM) 19
extended link
check 93
code uniqueness 93
name uniqueness 93
extended model definition (CDM) 19
extended object
check 93
code uniqueness 93
name uniqueness 93

F

file object
check 92
external 92
name uniqueness 92
float data type 31, 106
foreign key 112
generate 114
generate from identifier 102
index 114
name 114
name template for generation 112

G

general option 27, 52
generate
CDM 98, 100
CDM from CDM 99
column 102
configure OOM options 118
configure PDM options 110
foreign key 102, 114
foreign key name 112
index parameter 114
inheritance 105
links requirements (CDM) 99, 109, 118
new CDM 99
new model 98
new OOM 117
new PDM 109
OOM 117–119
OOM from CDM 116
OOM object 116
OOM object conversion 116
PDM 109, 110
PDM from CDM 102
PDM object 102
PDM object conversion 105
PDM option 110
primary key 102, 114
reference parameter 112
table 105
table parameter 112
update existing CDM 99
update model 98
validation rule 79

I

IDEF1X 9
identifier 44
attribute 43
child 71
copy 37
create 44
delete 43
entity attribute 37, 45
generate key 102
parent 71
primary identifier 45
properties 44
image data type 33, 108
index
foreign key 114
generate 114

primary key 114
inheritance 68
add child entity 73
check model 92
child 71
create 69
entity attribute 71
generate 71
generation mode 71
mutually exclusive 75
parent 71
primary key 71
properties 70, 71
inheritance display preferences 19
integer data type 31, 106

L

list
arrange 43
order 43
location for file object 92
logical model 2

M

map
CDM to OOM 116
CDM to PDM 105
Merise 9
notation 8
model
create 4
OOM options 118
open 94
option 27, 52
PDM options 110
model option
data item 12
data type full name 12
default data type 12
domain and attribute 12
notation 9
relationship 12
money data type 31, 106

N

name
file object 92
foreign key 114
package (CDM) 7
primary key 114
name uniqueness
extended link 93
extended object 93
Name/Code Case Sensitive option 8
namespace for package (CDM) 7
non-divergence
data item 34
domain 34
notation
Entity/Relationship 8
Merise 8
Notation option 9
number data type 31, 106

O

OLE data type 33, 108
OOM
code naming convention 116
enable transformations 119
generate object 116
generation option 119
generation parameter 117
map CDM object 116
object 116
OOM/CDM Mapping 119
option
CDM 27, 52
code 52
configure for generated OOM 118
configure for generated PDM 110
data item 37
general 27, 52
model 27, 52
relationship 52

P

package (CDM)
CDM 6
check model 83
code 7
comment 7
composite view 6
define 6

display preferences 7
name 7
namespace 7
properties 7
shortcut code uniqueness 83
shortcut generated as child table 83
PAM opened into CDM 94
PDM
enable transformations 110
generate 109
generate object 102
generation option 110
generation parameter 109
map CDM object 105
object 102
update 109
precision 29, 30
primary identifier 45
primary key
child 71
generate 114
generate from identifier 102
index 114
inheritance 71
name 114

R

reference
foreign key name 112
generate 112
reflexive association 65
relationship 47
associative entity 58
bend 61
cardinality 49, 58
change to entity 58
check model 89
code 52
create 47, 51
define 54, 56, 58
dependent 54
display 61
dominant 56
drag 61
entity attribute 58
error 52
example 58
mandatory 53
modify 61
option 52
optional 53
properties 47
reflexive 51
termination point 50, 53
unique code 8, 52
relationship display preferences 18
Relationship option 12
replication (CDM) check 92

S

serial data type 31, 106
server
expression 79
shortcut code uniqueness in package 83
shortcut generated as child table 83
specifying entity attribute 71
standard data type list 30
Sybase SQL Anywhere data types 106

T

table generation 105, 112
time data type 32, 108
typographic conventions vi

U

undefined data type 25, 29, 41

V

validation
CDM 82
validation rule 77
apply 79
business rule 79
check parameter 79
column 79
domain 79
generate 79
variable
column 79
domain 79