

SYBASE®

Information Liquidity Modeling
PowerDesigner® 15.1

Windows

DOCUMENT ID: DC00120-01-1510-01

LAST REVISED: June 2009

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

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

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

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

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

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

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

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

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

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

Contents

Getting Started with Information Liquidity Modeling	1
ILM Diagrams	1
Information Liquidity Diagram	1
Data Transformation Diagram	3
Transformation Control Flow Diagram	4
Creating an ILM	4
ILM Properties	5
Building Information Liquidity Diagrams	7
Information Liquidity Diagram Objects	7
Creating an Information Liquidity Diagram	9
Using the Mapping Editor for Replications	9
Understanding the RPME Dialog Box	9
Working with Mappings in the RPME	13
Replicating Objects with the Replication Wizard	16
Databases (ILM)	18
Creating a Database	19
Database Properties	19
Working with the Database Contextual Menu	21
Executing SQL Queries	21
Servers (ILM)	22
Creating a Server	22
Server Properties	22
XML Documents (ILM)	23
Creating an XML Document	23
XML Document Properties	23
Business Process (ILM)	24
Creating a Business Process	24
Business Process Properties	24
Flat Files (ILM)	25
Creating a Flat File	25
Flat File Properties	25
Replication Processes (ILM)	26
Creating a Replication Process	27
Replication Process Properties	27
Publications (ILM)	29
Creating a Publication	29
Publication Properties	30
Articles (ILM)	31
Creating an Article	31
Article Properties	32
Columns (ILM)	33

Creating a Column	33
Column Properties	34
Procedures (ILM)	35
Creating a Procedure	35
Procedure Properties	35
Subscriptions (ILM)	36
Creating a Subscription	37
Subscription Properties	37
Users (ILM)	37
Creating a User	37
User Properties	38
Data Connection Groups (ILM)	38
Creating a Data Connection Group	38
Data Connection Group Properties	38
Event Scripts (ILM)	39
Creating an Event Script	39
Event Script Properties	40
Process Connections (ILM)	40
Creating a Process Connection	41
Process Connection Properties	41
Data Connections (ILM)	41
Creating a Data Connection	42
Data Connection Properties	42
Migrating Deprecated Model Container Objects into a Project	43
Creating a Project	44
Adding Models to a Project Diagram	44
Rebuilding Dependency Links in a Project Diagram	45
Building Data Transformation and Control Flow Diagrams	47
Modeling Data Transformations	48
Creating a Data Transformation with the Transformation Wizard	49
Creating a Data Transformation with the Convert Mappings to ETL Wizard	51
Data Transformation Diagram Basics	54
Data Transformation Diagram Objects	55
Creating a Data Transformation Diagram	55
Data Transformation Task Properties	55
Data Structure Mapping Editor	56
Transformation Control Flow Diagram Basics	57
Transformation Control Flow Diagram Objects	58
Creating a Transformation Control Flow Diagram	58
Transformation Control Flow Properties	59
Transformation Processes (ILM)	59
Creating a Transformation Process	60
Transformation Process Properties	60
Data Inputs (ILM)	61
Creating a Data Input	62

Data Input Properties	62
Actions (ILM)	63
Creating an Action	64
Action Properties	64
Inserting a Script Execution	65
Inserting a Data Query Execution	66
Inserting a Data Aggregation	67
Inserting a Data Sort	68
Inserting a Data Filter	69
Inserting a Data Split	70
Inserting a Data Merge	70
Inserting a Data Join	70
Inserting a Data Lookup	71
Inserting a Data Projection	73
Inserting a Data Calculator	73
Data Outputs (ILM)	73
Creating a Data Output	74
Data Output Properties	75
Data Flows (ILM)	75
Creating a Data Flow	76
Data Flow Properties	76
Data Structure Columns (ILM)	77
Creating a Data Structure Column	77
Data Structure Column Properties	77
Transformation Parameters (ILM)	78
Creating a Transformation Parameter	78
Transformation Parameter Properties	79
Example: Assigning a Parameter to a Data Structure Column	79
Transformation Starts (ILM)	80
Creating a Transformation Start	81
Transformation Start Properties	81
Transformation Task Executions (ILM)	81
Creating a Transformation Task Execution	82
Transformation Task Execution Properties	82
Creating Multiple Transformation Task Executions	83
Transformation Synchronizations (ILM)	83
Creating a Transformation Synchronization	83
Transformation Synchronization Properties	84
Transformation Decisions (ILM)	84
Creating a Transformation Decision	84
Transformation Decision Properties	85
Transformation Ends (ILM)	85
Creating a Transformation End	86
Transformation End Properties	86
Control Flows (ILM)	86

Creating a Control Flow	86
Control Flow Properties	87
Working with Information Liquidity Models	89
Customizing the ILM Environment	89
Setting ILM Model Options	89
Setting ILM Display Preferences	90
Working with ILM Extended Model Definitions	91
Working with ILM Extended Dependencies	92
Generating Other Models from an ILM	92
Checking an ILM	92
Package Checks	93
Database Checks	93
Replication Process Checks	93
Publication Checks	94
Subscription Checks	94
User Checks	95
Article Checks	95
Article Column Checks	95
Procedure Checks	96
Article Event Script Checks	96
Replication Process Event Script Checks	97
XML Document Checks	97
Business Process Checks	98
Flat File Checks	98
Data Connection Checks	99
Process Connection Checks	99
Connection Group Checks	99
Server Checks	99
Transformation Process Checks	99
Data Transformation Task Checks	100
Data Transformation Diagram Checks	100
Data Input Checks	101
Data Output Checks	101
Data Transformation Action Checks	102
Data Flow Checks	103
Transformation Control Flow Checks	104
Transformation Control Flow Diagram Checks	104
Transformation Start/end Checks	104
Transformation Synchronization Checks	105
Transformation Decision Checks	105
Transformation Task Execution Checks	105
Control Flow Checks	105
Comparing and Merging ILMs	105
Generating and Reverse Engineering Replication Objects	107
Generating Replication Objects	107

To Generate for Several Targets:	107
To Generate for a Specific Target:	107
Selecting Targets to Generate in an ILM	107
Selecting Objects to Generate in an ILM	108
Defining Generation Options in an ILM	109
Defining Generation Tasks in an ILM	109
Reverse Engineering Replication Objects	109
Selecting a Data Source	110
Reverse Engineering a Single Replication Process	110
Reverse Engineering Several Replication Processes	110
Working with Replication Server	111
Modeling for Replication Server	111
Creating a Basic Replication Server Environment with the Replication Wizard	112
Completing your Replication Environment	113
Modeling for Heterogeneous Replications with Replication Agents and ECDA	114
Modeling a Warm Standby Application	115
Modeling for Mirror Activator	115
Modeling for RepConnector	115
Modeling Replications to a Sybase IQ Data Warehouse	116
Replication Server Object Properties	118
Generating for Replication Server	132
Reverse Engineering for Replication Server	133
Reverse Engineering a Single Replication Process Object	133
Reverse Engineering Several Replication Processes	134
Working with MobiLink	135
MobiLink Components	135
Modeling a MobiLink Synchronization Definition	136
Creating a MobiLink Synchronization with the Replication Wizard	137
MobiLink Servers	137
Creating a MobiLink Server	137
MobiLink Server Properties	138
MobiLink Consolidated and Remote Databases	138
MobiLink Remote Database Properties	138
MobiLink Articles	139
MobiLink Article Properties	139
MobiLink Subscriptions	140
MobiLink Subscription Properties	140
MobiLink Connection Scripts/Table Scripts	141
MobiLink Connection Script/table Script Properties	141
Generating for MobiLink	142
Generating the Consolidated Database	142
Specifying MobiLink Generation Options	143
Generating MobiLink Synchronization Scripts	143

Registering MobiLink Users143

Generating a Remote Sybase AS Anywhere Database for a Single MobiLink User
.....143

Generating Remote Sybase AS Anywhere Databases for All MobiLink Users144

Generating a Remote UltraLite Database144

Generating All MobiLink Script Files144




Reverse Engineering for MobiLink144

Getting Started with Information Liquidity Modeling

An *information liquidity model (ILM)* provides a global view of the movement of information in your organization. You can analyze and document where your data originates, where it moves to, and how it is transformed on the way, including replications and ETL.




The movement of data is a central concern for the enterprise. Data movement modeling allows you to model where data originates to where it is loaded, and how it is transformed on the way.

The PowerDesigner® ILM provides the following diagrams:

Diagram	Icon	Description
Information liquidity diagram		Shows data movement through: <ul style="list-style-type: none"> • Data replication – One or more source databases are replicated into one or more remote databases via replication engines. You can generate and reverse engineer Replication Server® and MobiLink™ files. • Data transformation – Data from diverse sources are combined to be extracted, transformed, and loaded to output sources via ETL (Extract Transform and Load) and EII (Enterprise Information Integration).
Data transformation diagram		Low level diagram that analyzes a data transformation to show how data is extracted from inputs, transformed by actions and loaded into outputs (see Data transformation diagram on page 3).
Transformation control flow diagram		Low level diagram that analyzes the sequence of execution of a series of data transformations (see Transformation control flow diagram on page 4).

ILM Diagrams

The PowerDesigner ILM provides the following diagrams:

Diagram	Icon	Description
Information liquidity diagram		Shows data movement through: <ul style="list-style-type: none"> • Data replication – One or more source databases are replicated into one or more remote databases via replication engines. • Data transformation – A high-level view of the input and output sources involved in a transformation process.
Data transformation diagram		Low level diagram that analyzes a data transformation to show how data is extracted from inputs, transformed by actions and loaded into outputs (see Data transformation diagram on page 3).
Transformation control flow diagram		Low level diagram that analyzes the sequence of execution of a series of data transformations (see Transformation control flow diagram on page 4).

Information Liquidity Diagram

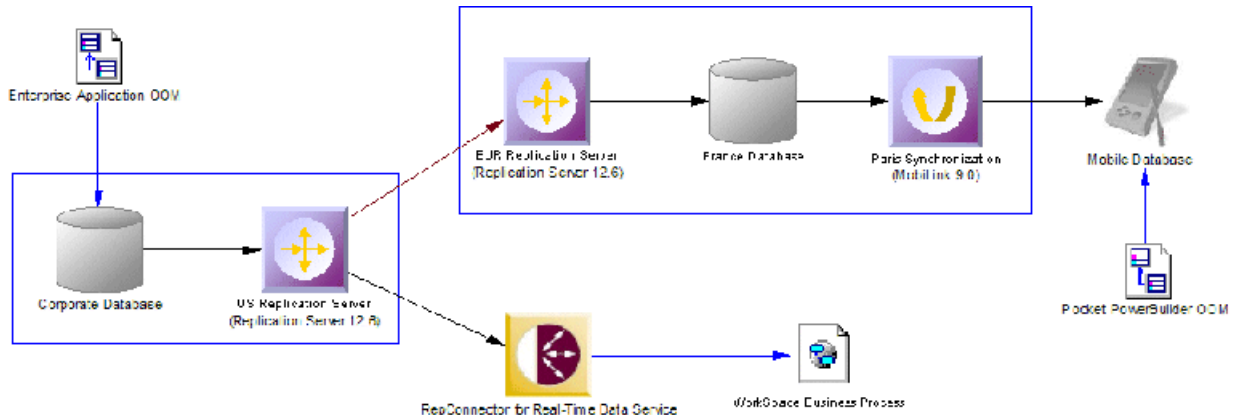
The *information liquidity diagram* is the core diagram of the ILM, which lets you model data replications and views of source and target data involved in data transformations.

Data Replication

The ILM diagram lets you describe and configure replication processes for the Replication Server engine (see [Working with Replication Server](#) on page 111) and the MobiLink engine (see [Working with MobiLink](#) on page 135), where a source database is replicated into one or more target databases.

For detailed information about the objects required to model a replication, see [Building Information Liquidity Diagrams](#) on page 7.

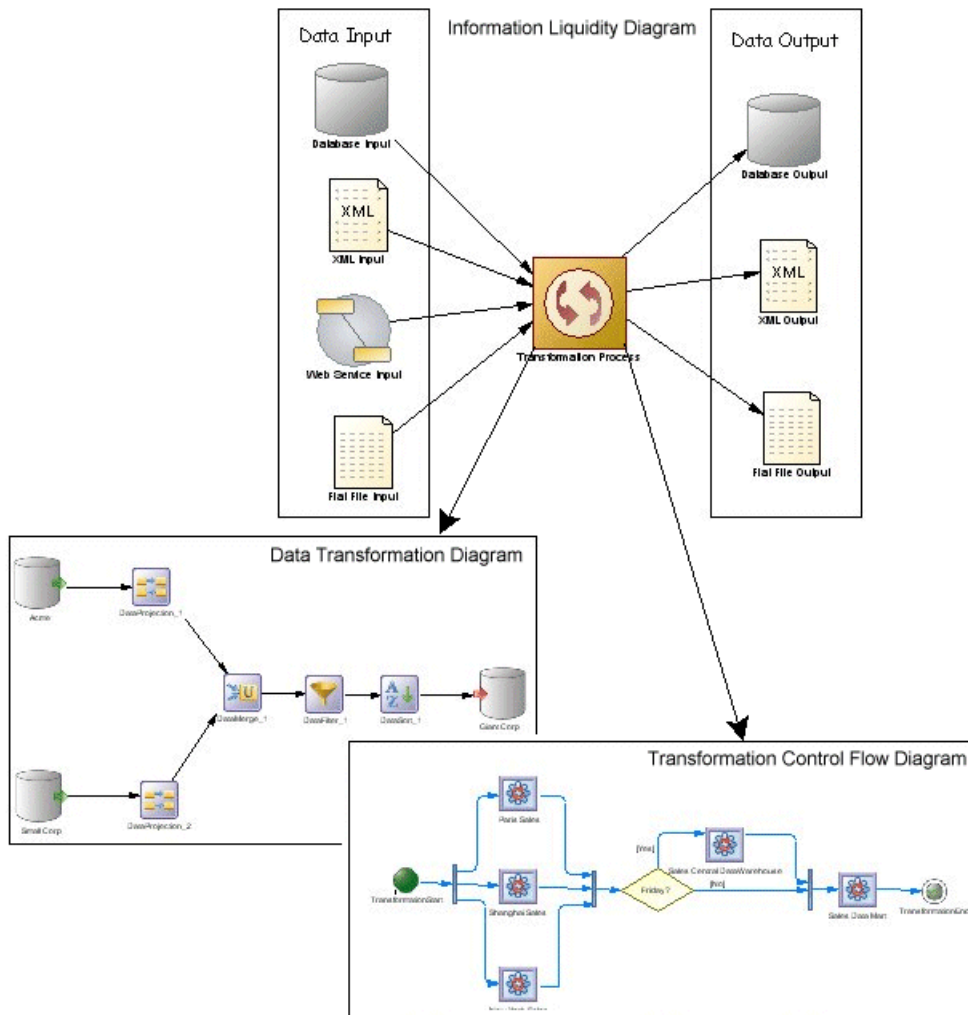
In the following example, the Consolidate Database is replicated into the Mobile Database and the RepConnector database.



Data Transformation

The information liquidity diagram is the entry point to document ETL and EII transformations. This diagram shows a high-level view of the input and output sources involved in a data transformation. These sources are linked to the transformation process, and are further specified in the data transformation diagram (see [Data transformation diagram](#) on page 3) in which they are transformed and loaded to outputs. Data transformations can be sequenced in the Transformation Control Flow diagram (see [Transformation control flow diagram](#) on page 4).

The following example shows how the information liquidity diagrams can interact with the other diagrams when modeling a data transformation. The information liquidity diagram contains the transformation process with its input and output sources. You can build one or more data transformation diagram(s) to detail your transformations, and a transformation control flow diagram to show the order in which these transformation are executed:



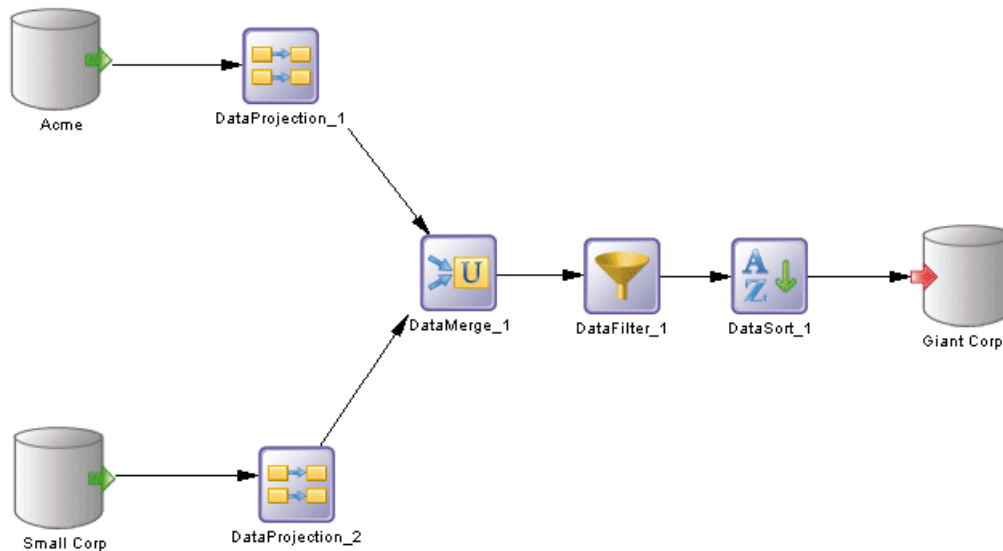
Data Transformation Diagram

A *data transformation diagram* lets you detail how data is extracted from data inputs, transformed by actions and loaded into data outputs.

At each step of the transformation, you can display a mapping between the current data and its parent. Data inputs and outputs are linked to the input and output sources specified in the ILM diagram (see [Information liquidity diagram](#) on page 1).

For detailed information, see [Building Data Transformation Diagrams](#) on page 54.

In the following example, Acme and SmallCorp are inputs which are transformed by various actions, and then loaded in GiantCorp output:



Transformation Control Flow Diagram

A *transformation control flow diagram* lets you represent several data transformations in a sequence.

For detailed information, see [Building Data Transformation Diagrams](#) on page 54.

In the following example, the Merge Employee transformation is executed before the Merge Role transformation. You can press ctrl and double-click a data transformation to display it in the data transformation diagram:

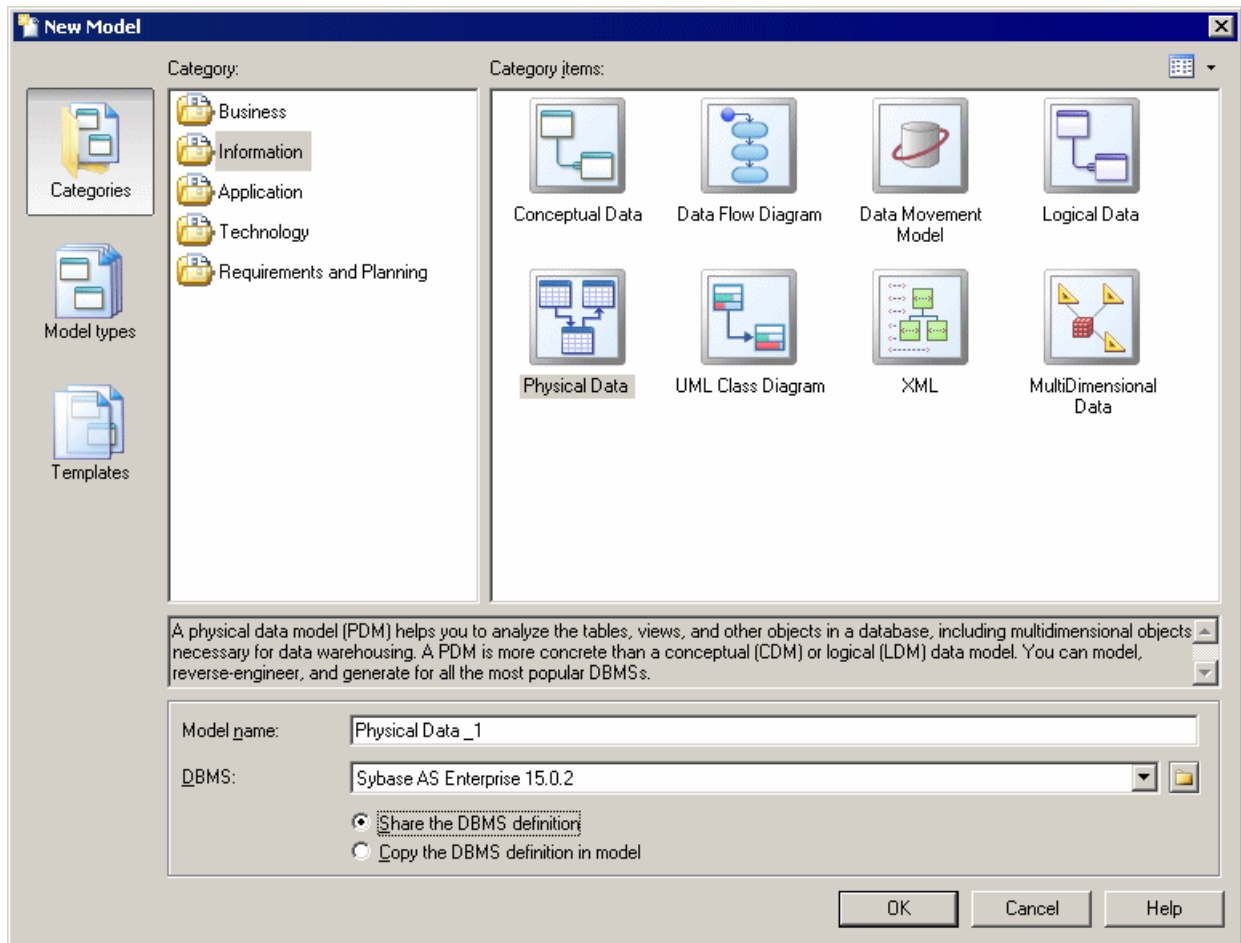


Creating an ILM

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

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

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



1. Select **File > New Model** to open the New Model dialog.
2. Click a button, and then select a category or model type (**Information Liquidity Model**) in the left-hand pane.
3. Select an item in the right-hand pane. Depending on how your New Model dialog is configured, these items may be first diagrams or templates on which to base the creation of your model.

Use the **Views** tool on the upper right hand side of the dialog to control the display of the items.

4. Enter a model name.
The code of the model, which is used for script or code generation, is derived from this name according to the model naming conventions.
5. [optional] Click the **Extensions** button to open the Extended Model Definitions dialog, and attach one or more extensions to your model .
6. Click **OK** to create and open the information liquidity model .

Note: Sample ILMs are available in the Example Directory.

ILM Properties

The Model property sheet displays the definition of the current model.

An ILM has the following model properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.

Property	Description
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the item.
Filename	Specifies the location of the model file. This box is empty if the model has never been saved.
Author	<p>Specifies the author of the model. You can insert a name, a space, or nothing.</p> <p>If you insert a space, the Author field in the title box remains empty.</p> <p>If you intentionally leave the box empty, the Author field in the title box displays the user name from the Version Info tab of the model property sheet.</p>
Version	Specifies the version of the model. You can use this box to display the repository version or a user defined version of the model. This parameter is defined in the display preferences of the Title node.
Default diagram	Diagram displayed by default when you open the model.

Building Information Liquidity Diagrams

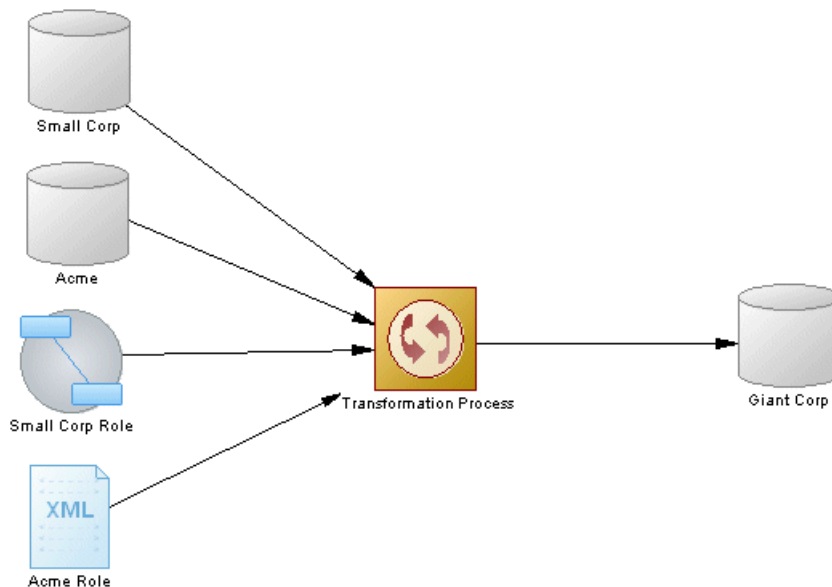
An *information liquidity diagram* provides a high-level graphical view of the liquidity of your information, including data sources, replications, and ETL operations.

You can create the following types of information liquidity diagrams:

- Replication diagram – lets you model the replication architecture of data contained in primary databases to remote databases using one or more replication processes. Replication processes contain publications and article definitions that describe the way in which data are replicated. Scripts can be generated for Replication Server and MobiLink engines. In the following example, data contained in the Consol database is replicated by a MobiLink replication process into the Remote database:




- Transformation diagram – is a high-level view to document ETL and EII transformations. This diagram shows input and output sources involved in a data transformation. The sources are linked to the transformation process, and are further specified in the data transformation diagram in which they are transformed and loaded to outputs. In the following example, data coming from Small Corp and Acme databases, Small Corp Role business process model, and Acme Role XML model are transformed and loaded to the Giant Corp database:




Information Liquidity Diagram Objects

You create the following objects in an information liquidity diagram:

Object	Tool	Description
Database		Data store based on a relational model and stored in one or several Physical Data Models. See Databases (ILM) on page 18.

Object	Tool	Description
Server		Hosts the database that stores the tables gathering information. See Servers (ILM) on page 22.
XML Document		Data store that contains data written under the XML format. Provides access to the XSM model. See XML Documents (ILM) on page 23.
Business Process		Data store based on a process model and stored in one or several Business Process Models. See Business Process (ILM) on page 24.
Transformation process		Data movement process that lets you model and document data transformations using Data Transformation Diagrams and Transformation Control Flow Diagrams. See Data Transformations Basics on page 47.
Flat file		File that contains data and its format. See Flat Files (ILM) on page 25.
Replication process		Server engine which role is to make a copy of data incoming from a given database into one or several databases or replication processes. See Replication Processes (ILM) on page 26.
Replication Server		Specifies near-real-time replication between a relatively small number of databases. This tool only displays when the Replication Server XEM is attached to the ILM. See Working with Replication Server on page 111.
MobiLink		Specifies data replication between a central database and a large number of remote databases. This tool only displays when the MobiLink XEM is attached to the ILM. See Working with MobiLink on page 135.
Publication	N/A	Specifies published data or procedures for a database connection, and allows the identification of a set of data to replicate and from where it is extracted. See Publications (ILM) on page 29.
Article	N/A	Identifies tables or views to replicate. See Articles (ILM) on page 31.
Where clause	N/A	Script that complements the code generation (SQL) to restrict the rows of a table you want to replicate. See Article property sheet Where Clause tab on page 33.
Column	N/A	Data structure that contains an individual data item within a row (record) model equivalent of a database field. See Columns (ILM) on page 33.
Procedure	N/A	Precompiled collection of SQL statements stored under a name in the database and processed as a unit. See Procedures (ILM) on page 35.
Subscription	N/A	Establishes a link between a publication and a database connection. It defines where data published via the publication must be replicated. See Subscriptions (ILM) on page 36.
User	N/A	Name that identifies a person or group working with objects in an information liquidity diagram. See Users (ILM) on page 37.
Connection group	N/A	Set of data connections that provides a relay in case the primary database goes down. See Data Connection Groups (ILM) on page 38.
Event script	N/A	Specifies how events on the replication process or on an article are implemented. See Event Scripts (ILM) on page 39.
Process connection		Relationship that specifies the way data is moved between two replication processes or between two transformation processes. See Process Connections (ILM) on page 40.

Object	Tool	Description
Data connection		Relationship that specifies the way data is moved between a database and a replication process, a data store and a transformation process or two data stores. See Data Connections (ILM) on page 41.

Creating an Information Liquidity Diagram

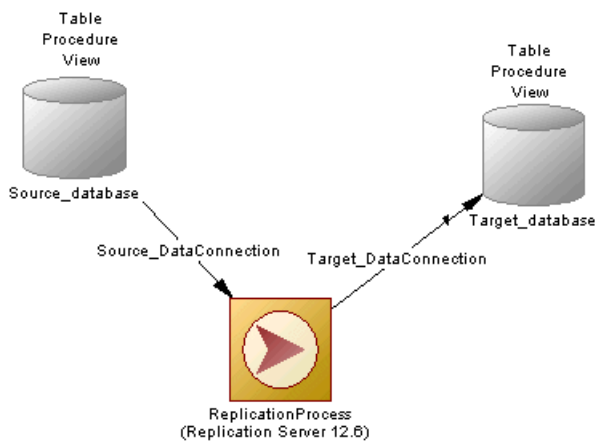
You can create a physical diagram in an existing ILM in any of the following ways:

- Right-click the model in the Browser and select **New > Information Liquidity Diagram**.
- Right-click the background of any information liquidity diagram and select **Diagram > New Diagram > Information Liquidity Diagram**.

Using the Mapping Editor for Replications

You can use the Mapping Editor to create and visualize replications.

A simple data replication can be designed as follows:



Data contained in a source database is transferred to the replication process via a source data connection and published (publication). The published data can be subscribed (subscription) by a target database using a target data connection.

The Mapping Editor accessible from the replication process contextual menu (let us call it the Replication Process Mapping Editor (RPME)) allows you to visualize and create the publication — subscription couple using mappings. A mapping can be a publication, an article, and article column or a procedure depending on the source data you want to replicate.

The mapping is materialized by a graphical link joining source and target.

Understanding the RPME Dialog Box

The Replication Process Mapping Editor (RPME) facilitates the replication of objects from a relational database to another. It only displays source and target objects allowing those who do not know about replication systems to create one without having to deal with the complexity of defining publications and subscriptions.

The RPME is available using the Mapping Editor command from the replication process contextual menu, whatever the type of the replication process.

The RPME dialog box is divided in three parts:

- The Source pane (with a white background).
- The Target pane (with a gray background).
- The properties pane. Depending on the selected object in the Source pane, that pane can either contains a list of publication mappings, article mappings or procedure mappings.

You create a mapping using any of the following ways:

- Drag and Drop feature.
- Create Mapping Between Selected Source and Target Objects tool.

For more information about mapping creation, see [Working with Mappings in the RPME](#) on page 13.

If the Replication Process Has Associated Data Connections

they display in the appropriate pane (Source and/or Target) depending on their access type. If mappings (in terms of publication — subscription) are detected, they display in the properties pane as publications, articles or procedures. You can add additional data connections.

For more information about data connection access type, see [Creating a data connection in the RPME](#) on page 13.





Note: If the replication process has associated data connection groups (or logical connection in RepServer), they are also available in the Replication Process Mapping Editor (RPME). However you cannot create new ones from the RPME. For more information about connection groups, see [Data Connection Groups \(ILM\)](#) on page 38.

If the Replication Process Stands Alone

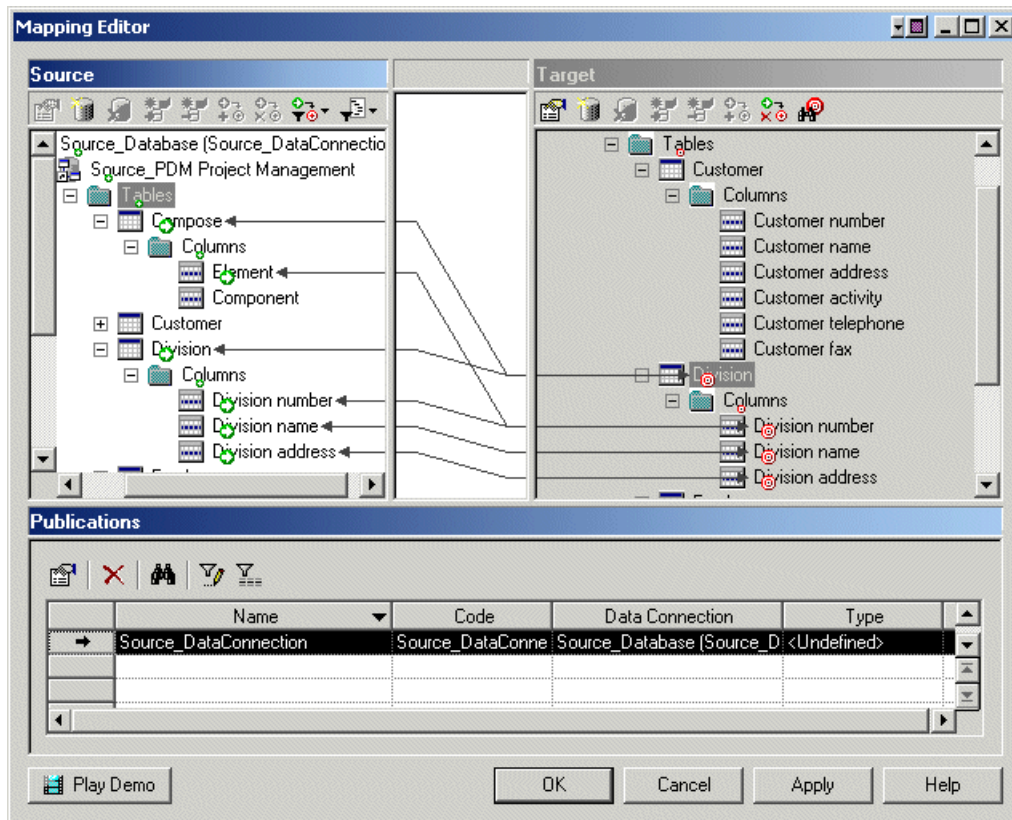
the RPME you open is empty. You must create source and target data connections in order to replicate data by defining mappings between source and target objects.

When a mapping is defined, it is materialized by a non-editable link with an arrow joining mapped object icons, giving you a global graphical view. If you click a link in the section between the Source and the Target pane, the source and the target objects of the mapping are highlighted and the related publication or article is displayed in the properties pane.

In addition, a small symbol is placed on the bottom right-hand corner of the source and target objects icon and also on their respective parent hierarchy icon to ease readability:

Pane	Mapped object icon	Parent hierarchy icon
Source (green)		
Target (red)		

When you define a mapping for a parent (for example a table) a mapping is automatically defined for its child objects (columns in this case):



The mapping is also displayed in the source and target objects Dependencies tab in their property sheet.

Source Pane





The Source pane displays all the data connections where data can be read (or read and written depending on the access type) and their source models where data is extracted showing all the objects that can be replicated.







For more information about the data connection access type, see [Creating a data connection in the RPME](#) on page 13.

You need to create at least one data connection and declare models to define where data should be extracted to be replicated to the model in the Target pane. You can create several data connections.

A data connection can contain several source models, you can select the models among a list of models open in the current workspace.

The following toolbar helps you manage sources:

Tool	Name	Description
	Properties	Opens the property sheet of the selected source object.
	Create Data Connection	Launches the Database Connection Wizard that allows you to select an existing database connection or create one then select one or more PDMs that specify its schema.
	Delete Data Connection	Deletes the selected database connection. Related mappings, if any, are automatically deleted.
	Add Models to Database of Data Connection	Adds one or more models to an existing data database from a selection dialog box.

Tool	Name	Description
	Remove Model from database	Removes the selected source model from the database. Related mappings, if any, are automatically deleted.
	Create Mapping Between Selected Source and Target Objects	Creates a mapping between selected source and target objects. The mapping (in terms of publication — subscription) is materialized by a link and the mapping details appear in the properties pane. This tool is available only when a mapping between the two selected objects is appropriate.
	Delete Mappings	Deletes all the mappings for the selected source object.
	Filter Mappings	Filters mappings to show: All mappings Only mappings of the selected object Only mappings of the selected object and its child objects
	Filter Objects	Filters objects to show: All objects Only objects with mappings Only objects without mappings
	Find Source Object	Finds an object in the Source pane and highlights it.

Target Pane

The Target pane displays all the data connections where data can be written (or read and written depending on the access type) and their target models where data is replicated showing all the objects that can be mapped.

The toolbar that helps you manage mappings in the target model is the same as the one for the source model, except for the Filter tools that are not available as the target model owns the mappings.

For more information about the data connection access type, see [Creating a data connection in the RPME](#) on page 13.

Properties Pane

When you create a mapping, the properties pane displays objects that define the mappings (in terms of publication — subscription) in the lower part of the Replication Process Mapping Editor (RPME) dialog box.

The following PDM parent and child objects can be replicated via mappings in the RPME:

PDM parent object	PDM child object
Table View Procedure	Table column View column

The properties pane content changes automatically depending on the object you select in the Source pane:

Selected object	Properties pane displays...
Data connection, model or folder	List of <i>publications</i> using the selected data connection or model or folder under the data connection.
Parent object (table, view, procedure)	List of <i>articles</i> using the selected parent object (table or view) List of <i>procedures</i> using the selected parent object (procedure)
Child object (table column or view column)	List of <i>article columns</i> using the selected column.

Replication Process Mapping Editor Video

You can click the Play Demo tool in the lower-left corner of the Replication Process Mapping Editor (RPME) dialog box to launch a video that briefly illustrates the main features of the RPME used for replications.

Working with Mappings in the RPME

You need a source and a target data connection to create a mapping in the Replication Process Mapping Editor (RPME).

You can:

- Use existing ones if the replication process already has associated data connections when you launch the RPME.
- Create at least one source and one target data connection.

Creating a Data Connection in the RPME

The data connection allows you to replicate data in the Replication Process Mapping Editor (RPME) using mappings. You need at least one data connection in the Source pane and one in the Target pane in order to proceed with mappings. The source and the target data connection must contain one or more models from which data are extracted or replicated.

The Add Models to Database of Data Connection tool allows you to add source models from a selection list to an existing database.

A data connection can have the following access types that determine the direction of the data flow between the Source and the Target panes:

- Read/Write
- Read-only
- Write-only

The pane in which you create the data connection influences the data connection access type:

Pane	Data connection access type
Source	Read/Write Read-only (Default)
Target	Read/Write Write-only (Default)

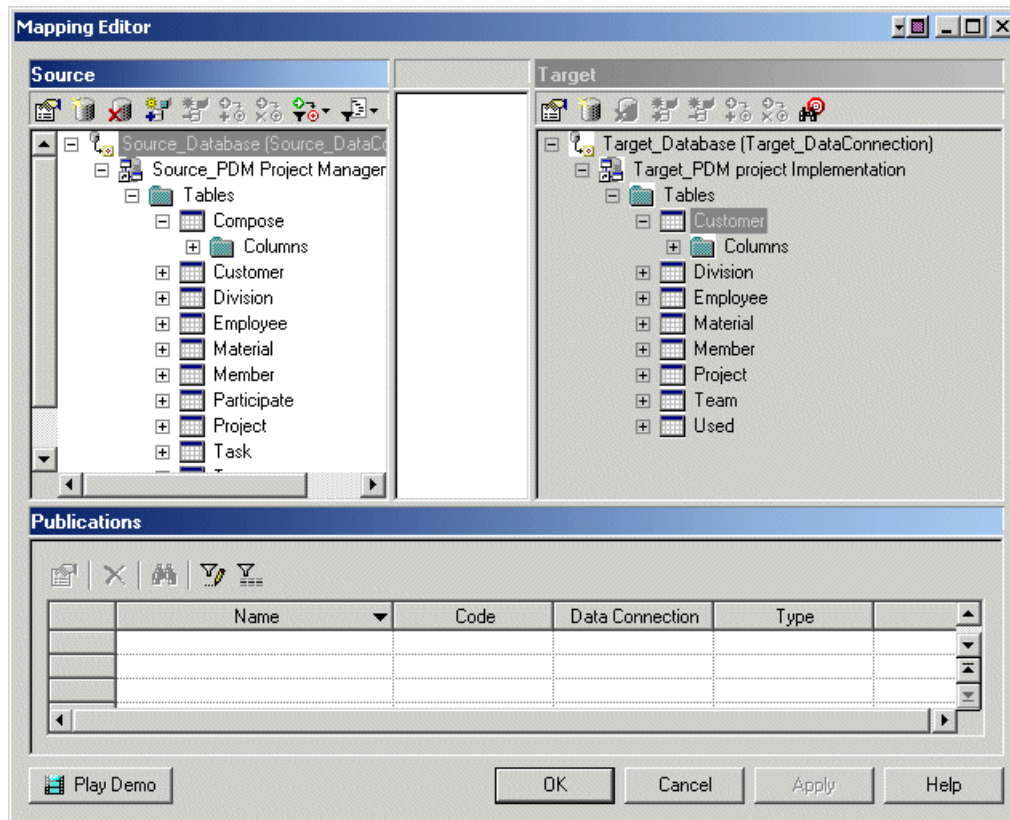
If you modify the access type of a data connection in its property sheet so that it no longer corresponds to a valid access type in the pane where it is displayed, the data connection is automatically moved to the appropriate pane. If you select the Read/Write access type, the data connection is displayed in the Source and in the Target pane.

1. Click the Create Data Connection tool in the source or target toolbar.

The Database Connection Wizard opens to the Database selection page.

2. Select or type a database name from which you want to extract or replicate data and click Next.
3. From the Physical Data Model Selection page, select one or more models among the list of open models in the workspace to declare source or target models and click Finish.

The data connection and its associated models are displayed in the Source or Target pane:



Creating a Mapping in the RPME

You can create a mapping from the Replication Process Mapping Editor (RPME) dialog box using any of the following way:

- Drag and Drop feature.
- Create Mapping Between Selected Source and Target objects tool.

A mapping corresponds to a publication — subscription couple. You can create several different mappings for the same source object. Each mapping is added to the mapping pane that allows you to parameter the mapping.

As soon as you create a mapping (in terms of publication — subscription) a publication for the data connection you use is created and displayed in the properties pane. Besides, an article or article column corresponding to the object to replicate is displayed in the properties pane. If you delete the replicated object you do not delete the associated publication. But if you delete a publication, all the replicated objects that use the publication are deleted.

Automatic Mapping

When parent objects in the Source and Target pane are mapped, their child objects are automatically mapped if the current model (source) contains child objects whose name and code match those of the child objects in the target models.

Creating a Mapping Using the Drag and Drop Feature

You can create a mapping by dragging an object icon from the Source pane and dropping it to the Target pane.

When the drop position is unauthorized a barred circle is displayed.

For a given mapping, the following results occur depending on the type of object you drag and where you drop it:

Source > Destination	Result
Parent object > parent object	Both parent objects are mapped together with their child objects if their name and code match. In the properties pane, an article is created for the mapping of the parents and also an article for each one of the child objects that matched.
Child object > child object	Both child object are mapped together with their parent and this mapping (article) is added to the properties pane.
Child object or parent object > folder or model	The child object or the parent object with its child objects are created in the appropriate folder in the target model and are mapped with their respective source objects. The corresponding mappings (publication and article) are added to the properties pane.

Drag an object icon from the Source pane and drop it onto the appropriate object icon in the Target pane.

A link is automatically created between the two objects to materialize the mapping and a small symbol is displayed on the source and the target objects icon. Besides, the mapping (publication, article or procedure) displays in the properties pane.

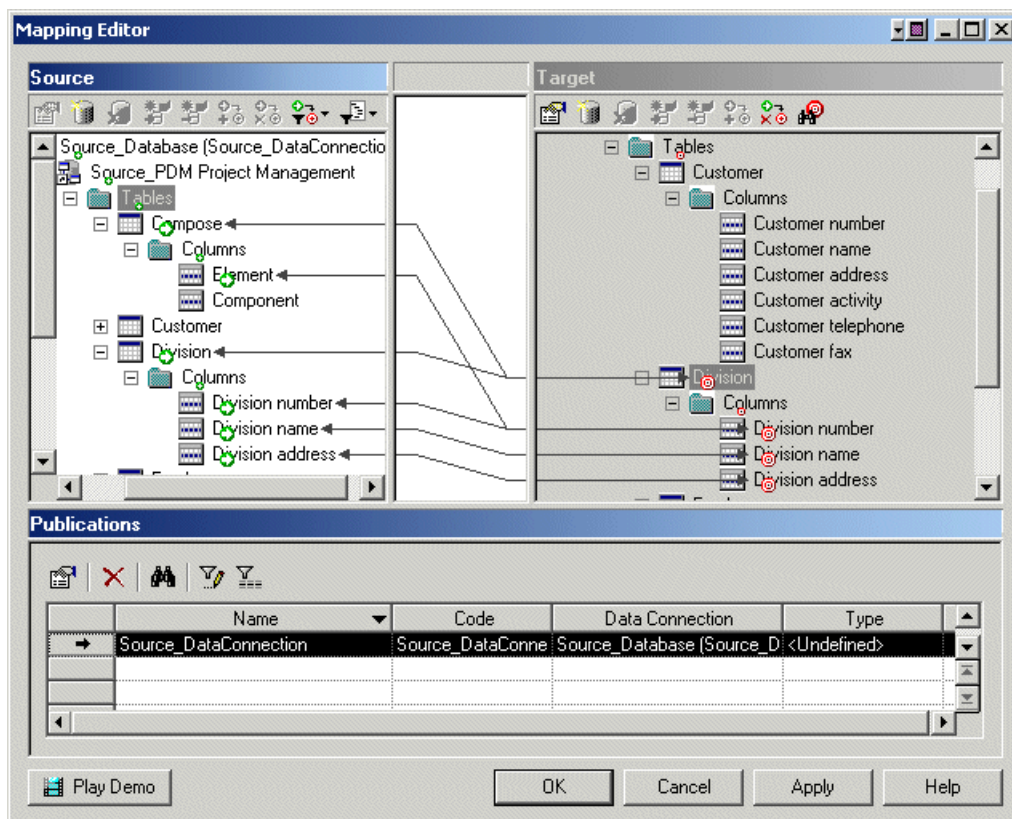
Creating a Mapping Using the Create Mapping Between Source and Target Objects Tool

You can create a mapping using the Create Mapping between source and target objects tool from the RPME toolbar. To do so you first need to select an object icon in the Source and Target panes.

1. Select an object icon in the Source pane and select an object icon in the Target pane.
2. Click the Create Mapping between source and target objects tool in the Source or Target pane toolbar.

A link is automatically created between both objects to materialize the mapping and a small symbol is displayed on the source and target object icons. Besides, the mapping (publication, article or procedure) displays in the properties pane.

In the following example, the *Compose* table and its *Element* column together with the *Division* table and all its columns in the Source pane are replicated in the *Division* table and all its columns in the Target pane:



Modifying a Publication in the RPME

A publication defines published data or procedures for a data connection. You can modify publication properties directly from the properties pane in the Replication Process Mapping Editor (RPME) using the:

- Data Connection column that allows you to select a connection for which data or procedures are published.
- Type column that allows you to select a type for the publication when the replication process has a Replication Server type. The publication type is <Undefined> by default. Note that the <Database> type is not graphically created by the RPME.

You can display other columns using the Customize Columns and Filters tool from the properties pane toolbar.

You can also modify a publication from its property sheet using the Properties tool.

Graphical changes are immediately displayed in the RPME.

For more information about publications, see [Publications \(ILM\)](#) on page 29.

Modifying an Article in the RPME

An article corresponds to a table or a view to replicate.

You can modify article properties directly from the properties pane in the Replication Process Mapping Editor (RPME) using the:

- Source Table column or Source View column that allows you to select the source table or view to which the article corresponds.
- Remote Table column or Remote View column that allows you to select the target table or view to which the article corresponds.

You can display other columns using the Customize Columns and Filters tool from the properties pane toolbar.

You can also modify an article from its property sheet using the Properties tool.

Graphical changes are immediately displayed in the RPME.

You modify article columns the same way.

For more information about articles, see [Articles \(ILM\)](#) on page 31.

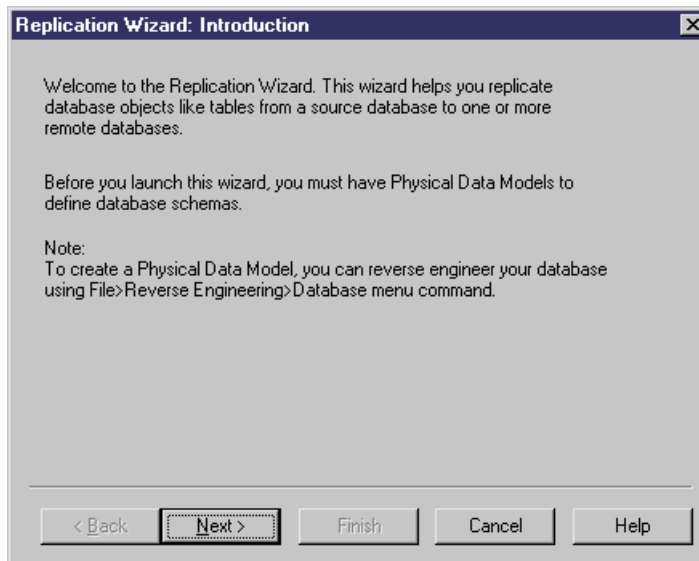
Note: A *procedure* corresponds to a stored procedure from the source database. You modify a procedure the same way you modify an article.

Replicating Objects with the Replication Wizard

The Replication Wizard is an easy way to replicate database objects (like tables, views, etc.) from a source database to one or more remote databases using a replication engine.

1. Launch the Replication Wizard in any of the following ways, and then click Next to go to the next step:

- Select **Tools > Replication Wizard**.
- Right-click a replication process, and select Replication Wizard.
- Right-click a source database, and select Replication Wizard.



2. The Source Database Selection [not available if launched from the source database] lets you specify the location of the data you need to replicate. You can:
 - Select an existing data connection.
 - Enter an existing database name. You can use the Select a Database tool next to the list to search for an existing database. This action assumes that no data connection to your replication process exists at that time. By doing so, you automatically create the missing data connection.
 - Enter a new database name. This action assumes that both the database and the data connection to your replication process do not exist at that time. By doing so, you automatically create the missing database and data connection.

Make your selection, and then click Next.

3. The Source Physical Data Model Selection page [not available if launched from a source database with an associated PDM] lets you specify the schema of the source database. You can:
 - Select one or more existing PDMs.
 - Create a new PDM. Select a DBMS, click the Share or Copy radio button, select or clear the Reverse engineer the database using a data source option to reverse engineer the schema for the selected database, and specify a data source, and connection parameters.

Make your selection, and then click Next.

4. The Replication Process Selection page [not available if launched from a replication process] lets you select or create the replication process to use for the replication of the source database tables. You can:
 - Select an existing replication process if the source database is linked to one or more replication process or click the Browse button to select a replication process that is not linked to the source database. The Type list is automatically filled with the type of the selected replication process. If the replication process is "Undefined", select a type (MobiLink or Replication Server) to proceed with the wizard. If you select a replication process that is not already linked to the source database, you automatically create the data connection between both objects in the diagram.
 - Create a replication process, and select a type (MobiLink or Replication Server). By doing so, you automatically create the corresponding replication process linked to the source database via a data connection in the diagram.

Make your selection, and then click Next.

5. The Publication Selection page lets you create or select publications for the replication of tables. You can:
 - Create or select an existing publication to gather all the replicated tables, and simplify the replication when several remote databases must duplicate the same set of tables.
 - Create a publication for each replicated table, and thus manage each table replication individually.

- [only available for a replication process with a Replication Server type] Select a publication type (Undefined, Publication, Replication Definitions) that will influence the generation code for the Replication Server (see [Working with Replication Server](#) on page 111).

Make your selection, and then click Next.

6. The Replicated Object Selection page lets you select tables, procedures or views to replicate. If you have selected an existing publication in the previous page, only the database objects that are not yet selected for this publication are displayed in the list and can be selected.

Make your selection, and then click Next.

7. The Remote Database Connection Selection page lets you specify the connections used to access the remote database(s) where you need to replicate the selected database objects. You can:

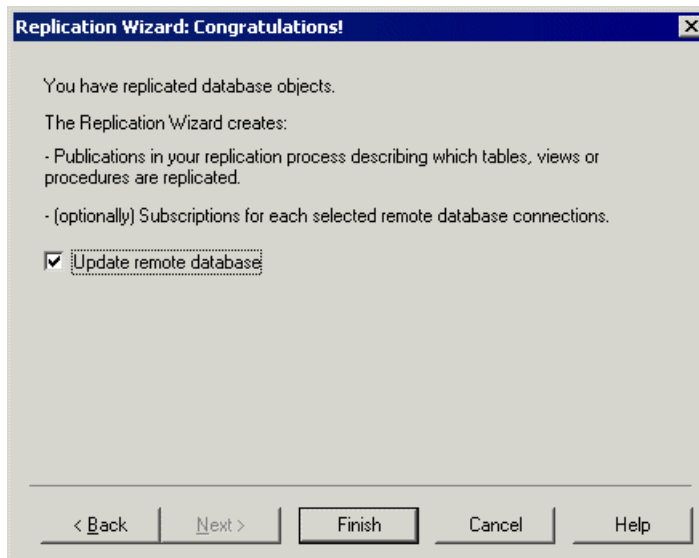
- Enter an existing database name. You can use the Select a Database tool next to the field to search for an existing database. By doing so, you automatically create the missing data connection.
- Enter a new database name. By doing so, you automatically create the missing data connection and database. Note that the database name must respect the following syntax: <server name>.<database name> (server name is optional).
- Select one or more existing data connections in the lower part of the page.

Make your selection, and then click Next.

8. The last page of the wizard summarizes the objects that will be created. You can choose to update the remote database with the new tables created in the source database.

When you click Finish, the wizard creates:

- A source database linked to a replication process, which is in turn linked to a remote database via data connections in the information liquidity diagram.
- One or more publications in the replication process describing which objects (tables, views, and procedures) are replicated.
- [Optionally] One or more subscriptions for each selected remote database connections.



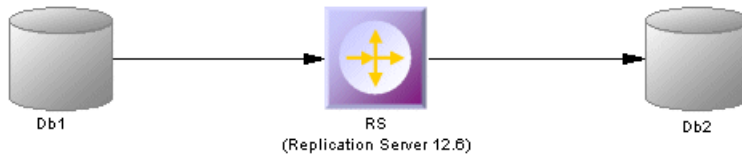
Databases (ILM)

A *database* is a data store based on one or more Physical Data Models (PDM), and which can be linked to:

- A replication process via a data connection to replicate data from a database to another. See [Replication Processes \(ILM\)](#) on page 26.

- A transformation process via a data connection. The database is a data input or output in the modeling of data transformations. For more information about data transformations, see [Building Data Transformation Diagrams](#) on page 54.
- Any other diagram objects via dependency links to show dependencies between these objects and the PDM associated with the database.

In the following example, the Db1 database is linked, via a database connection, to the RS replication process, which is in turn linked, via another database connection, to the Db2 database to indicate how the data is copied from a database to another:



Some PDM commands, in the database contextual menu, can help you build a PDM, if you do not already have one. You can use:

- Reverse Engineer Database – lets you generate a PDM from an existing database schema. You first need to fill data source connection information to proceed to the reverse, and the reverse automatically creates and attaches the PDM to the database.
- All database generation features. See [Working with the database contextual menu](#) on page 21.

Creating a Database

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

- Use the Database tool in the diagram Palette.
- Select **Model > Databases** to access the List of Databases, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Database**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Database Properties

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




The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Server	Specifies the name of the server on which the database is defined. A server hosts the database that stores the tables gathering information. 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. For more information about servers, see Servers (ILM) on page 22.

Property	Description
Type	<p>Specifies the database type, which influences the generation. Types are defined in the extended model definitions (XEM) attached to the model. You can view in the Preview tab the generated code according to the type you selected. You can choose one of the following types:</p> <ul style="list-style-type: none"> • Undefined – to represent any standard relational database. • RepConnector – to replicate data on mobile devices (PDA, Pocket PC etc.), between persons wherever they are. • UltraLite – to capture database changes in real time and deliver them in XML to message queues, that can be used by any supported message queuing system.

Database Type

The following database types are available. The database symbol changes according to the selected type:

Type	Symbol
Undefined	
UltraLite	
RepConnector	

Database Property Sheet Physical Data Models Tab

This tab lets you associate one or more Physical Data Models (PDMs) with the database in order to describe its content. PDMs must be open in the workspace to create associations. You can also delete associations you no longer want.

1. Open a database property sheet, and click the Physical Data Models tab.
2. Click the Add Physical Data Models tool in the toolbar to open a selection list, select one or more PDMs and click OK. You can click the Open Model tool in the toolbar to open the associated PDM.
3. Click OK to close the property sheet and return to the model diagram.

Database Property Sheet Database Connection Tab

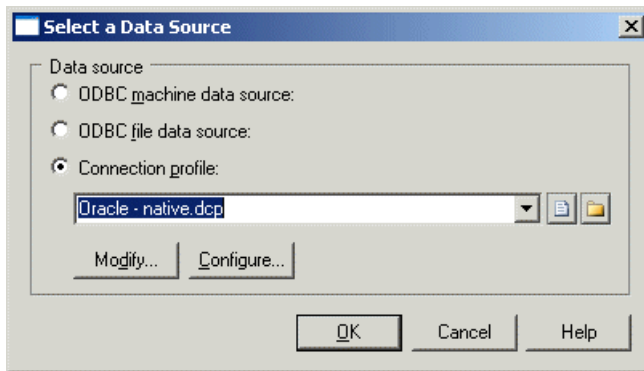
This tab lets you create a data source to define the connection parameters between the client and the database on the server.

The following connection parameters are available:

Parameter	Description
Data source	Specifies the connection profile that is used to connect to your database.
Login	Specifies the name of the user ID with which you log on to the database.
Password	Specifies the password for the connection.

For detailed information about creating, configuring, and using connection profiles, see "Connecting to a Database" in the Models chapter of the *Core Features Guide*.

1. Open a database property sheet, and click the Database Connection tab.
2. Click the Select a Data Source tool to open the Select a Data Source dialog box.



3. Select one of the following radio buttons, depending on your chosen method for connecting to your data source:

- ODBC machine data source
- ODBC file data source
- Connection profile

You can use the tools to the right of the data source field to browse for a new connection profile file or directory, and the Modify and Configure buttons to modify or configure your data source connection.

4. Click OK.

Working with the Database Contextual Menu

The contextual menu of a database symbol contains the following commands:

Command	Description
Generate Scripts	Lets you generate a script to prepare the replication of tables. This script is to be executed on a connected database at a later time
Generate and Execute Scripts	Lets you directly execute a script on a connected database to replicate tables.
Generate Database	[PDM command] Lets you generate the database corresponding to the PDM attached to the current model.
Modify Database	[PDM command] Lets you generate a database by synchronizing a modified PDM with one of the following existing database schema: Archived model Live database connection Generation script file Model from the repository
Reverse Engineer Database	[PDM command] Lets you reverse engineer an existing database into a new PDM. The data source can either be from script files or from a data source.
Connect	Lets you connect to a data source before executing an .SQL file.
Execute SQL	Lets you execute the .SQL file you have generated using the Generate Scripts command.

For more information about the PDM commands, see the *Data Modeling* guide.

Executing SQL Queries

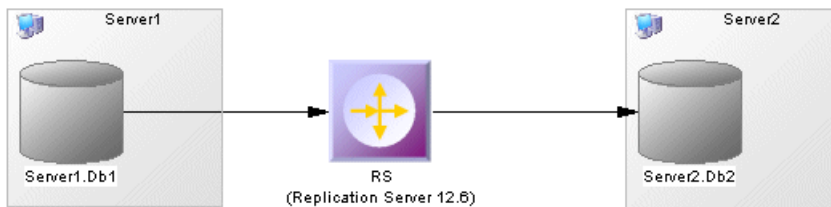
You can send SQL queries from a database or a replication process to a connected data source, and display the results.

1. Right-click a database or replication process in the diagram or Browser, and select **Execute SQL**. If you are not already connected to the data source, the **Connect to Data Source** window opens. Choose your connection profile, and click **Connect** to open the **Execute SQL Query** dialog box.
2. Enter one or more SQL statements in the dialog box, and click the **Run** button to apply them to the data source, and display the query results in the **Results** window.

Servers (ILM)

A server identifies a machine where a database, a replication process or a transformation process is deployed.

In the following example, Db1 is deployed on Server1 and Db2 is deployed on Server2:



To graphically attach one or more databases or processes to a server, drag and drop the database or process onto the server symbol. To detach the database or process from the server symbol, drag it outside the server symbol.

Creating a Server

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

- Use the **Server** tool in the diagram Palette.
- Open a database or a replication process property sheet, and click the **Create** tool.
- Select **Model > Servers** to access the **List of Servers**, and click the **Add a Row** tool.
- Right-click the model or package in the **Browser**, and select **New > Server**.

For general information about creating objects, see the **Objects** chapter in the *Core Features Guide*.

Server Properties

You can modify an object's properties from its property sheet. To open a server property sheet, double-click its diagram symbol or its **Browser** entry in the **Servers** folder.

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

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

XML Documents (ILM)

An *XML document* is a data store, which contains data written under the XML format, and which can be associated with one or more XML Models (XSM) to describe its contents. The XML document can be linked to:

- A transformation process via a data connection. The XML document is a data input or output in the modeling of data transformations. For more information about data transformations, see [Building Data Transformation Diagrams](#) on page 54.
- Any other diagram objects via dependency links to show dependencies between these objects and the XSM associated with the XML document.

In the following example, Acme Role is linked to TransformationProcess, which is in turn, linked to GiantCorp.



Creating an XML Document

You can create an XML document in any of the following ways:

- Use the XML Document tool in the diagram Palette.
- Select **Model > XML Documents** to access the List of XML Documents, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > XML Document**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

XML Document Properties

You can modify an object's properties from its property sheet. To open an XML document property sheet, double-click its diagram symbol or its Browser entry in the XML Documents folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
XML file path	Specifies the location of the file that contains the XML data. You can enter a file path or click the Select File tool to the right of the field to select a file.
XSD file path	Specifies the location of the file that contains the XML schema. You can enter a file path or click the Select File tool to the right of the field to select a file.

Property	Description
Source model	<p>Specifies the XML model (XSM) associated with the current object from the XML models open in the workspace. You have access to the following XSM-specific commands from the XML document contextual menu:</p> <ul style="list-style-type: none"> • Generate Schema – to generate a schema file to describe the structure of the XML document. • Reverse Engineer Schema – to reverse engineer a schema file and create an XSM.

Business Process (ILM)

A *business process* is a data store, which contains data under a specific format, such as a web service for example, and which can be associated with one or more Business Process Model (BPMs) to describe its content. The business process can be linked to:

- A transformation process via a data connection. The business process is a data input in the modeling of data transformations. For more information about data transformations, see [Building Data Transformation Diagrams](#) on page 54.
- Any other diagram objects via dependency links to show dependencies between these objects and the BPM associated with the business process.

In the following example, Small Corp Role is linked to TransformationProcess via a data connection, which is in turn linked to Giant Corp via another data connection. The Corp Role business process is a data input in the data transformation diagram:



Creating a Business Process

You can create a business process in any of the following ways:

- Use the Business Process tool in the diagram Palette.
- Select **Model > Business Process** to access the List of Business Processes, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Business Process**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Business Process Properties

You can modify an object's properties from its property sheet. To open a business process property sheet, double-click its diagram symbol or its Browser entry in the Business Processes folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.

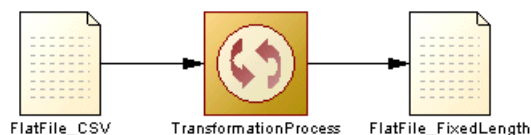
Property	Description
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Source model	Specifies the Business Process Model (BPM) associated with the current object from the BPMs open in the workspace. You have access to the following BPM commands from the business process contextual menu: <ul style="list-style-type: none"> • Generate Business Language – to generate objects from a BPM. • Reverse Engineer Business Language – to reverse engineer business process language files into a BPM. <p>You open a BPM associated with a business process by right-clicking its symbol, and select Open Model.</p>

Flat Files (ILM)

A *flat file* contains data and the format of data. Data format can be specified by a custom separator, a fixed length, or by columns that you define with a name and a data type.

A flat file can be linked to a transformation process via a data connection in order to model and document data transformations. For more information about data transformations, see [Building Data Transformation Diagrams](#) on page 54.

In the following example, FlatFile_CSV is linked to TransformationProcess via a data connection, which is turned linked to FlatFile_FixedLength via another data connection:



Creating a Flat File

You can create a flat file in any of the following ways:

- Use the Flat File tool in the diagram Palette.
- Select **Model > Flat Files** to access the List of Flat Files, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Flat File**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Flat File Properties

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

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Separator	Specifies a column separator. You must select the Custom Delimiter mode in order to select a predefined value or enter a new one.
Row delimiter	Specifies a row delimiter, which separates records. You must select the Custom Delimiter mode in order to select a predefined value or enter a new one.
Header	Specifies whether you want to retrieve the column headers of the file whose path is specified in the Path field.
Mode	<p>You can choose one of the following values:</p> <ul style="list-style-type: none"> • CSV (Comma Separated Value) – Specifies a file which contains tabular data, and which uses a comma to separate values. When you select this option, the Separator field and the Row Delimiter field are not available. • Custom Delimiter – Lets you specify a column separator and a row delimiter to separate records. • Fixed Length – Lets you specify a row delimiter to separate records. When you select this option, the Separator field is not available.
Path	Specifies the path to the file containing data. Click the Select File tool to the right of the field to browse for a file.
Type	Type of the transformation process, when defined in a XEM attached to the model.

Flat File Property Sheet Columns Tab

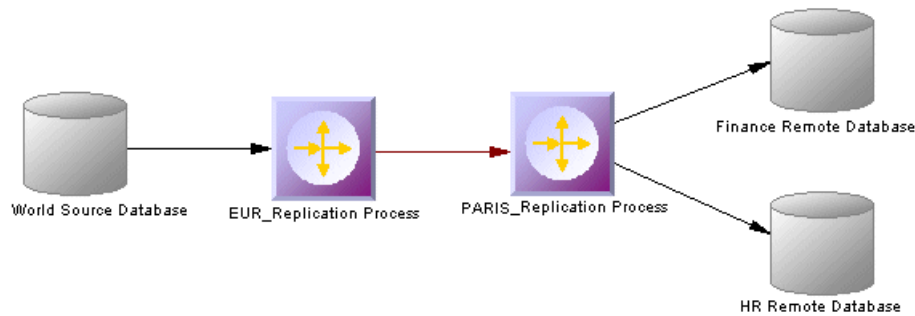
This tab lets you create, edit, or delete data structure columns. Click the Retrieve Columns by Parsing File Header tool, if you want to retrieve columns by parsing the header of the file, whose path is specified in the Path field. The list of columns can be ordered.

For more information, see [Data Structure Columns \(ILM\)](#) on page 77.

Replication Processes (ILM)

A *replication process* is a server engine that copies data from a given database into one or more databases or replication processes.

In the following example, EUR_Replication Process copies data contained in World Source Database into Paris_Replication Process, which in turn copies data into Finance Remote Database and HR Remote Database:



You can right-click a replication process and select Replication Wizard to replicate database objects from a source database to one or more remote databases. See [Replicating Objects with the Replication Wizard](#) on page 16.

The replication process can contain the following main items:

- **Publications** – Specifies a set of data to replicate, and from where it was extracted. Publications contain articles that correspond to the tables of the replicated database and, which point to the tables of the source database. Articles contain columns that give details on replicated columns. For more information, see [Publications \(ILM\)](#) on page 29.
- **Subscriptions** – Specifies where the data is replicated by indicating which replicated database subscribes to which publication. For more information, see [Subscriptions \(ILM\)](#) on page 36.
- **Users** – Specifies who is involved in the replication process. For more information, see [Users \(ILM\)](#) on page 37.
- **Connection groups** – Specifies a logical group for database connections. For more information, see [Data Connection Groups \(ILM\)](#) on page 38.
- **Event scripts** – Specifies how events on the process are implemented. For more information, see [Event Scripts \(ILM\)](#) on page 39.

Creating a Replication Process

You can create a replication process in any of the following ways:

- Use the Replication Process tool in the diagram Palette.
- Select **Model > Replication Processes** to access the List of Replication Processes, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Replication Process**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Replication Process Properties

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

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.

Property	Description
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Server	Specifies the name of the server on which the replication process is deployed. 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. For more information, see Servers (ILM) on page 22.
Type	<p>Specifies the type of the replication process depending on the extended model definition (XEM) attached to your model. The type influences the generation and the display of additional information within specific tabs and in tabs displaying extended attributes. Click the Preview tab to see the generated code. You can choose one of the following values:</p> <ul style="list-style-type: none"> Undefined – Specifies a replication server with no type. MobiLink – Specifies data replication between a central database and a large number of remote databases. For more information about the MobiLink replication process, see Working with MobiLink on page 135. Replication Server – Specifies near-real-time replication between a relatively small number of databases. For more information about the Replication Server, see Working with Replication Server on page 111.

Replication Server Type

The following replication server types are available. The replication server symbol changes according to the selected type:

Type	Symbol
Undefined	
MobiLink	
Replication Server	

Replication Process Property Sheet Publications Tab

This tab lets you create, delete or add a publication selection for data connections.

A *publication* specifies published data or procedures for a database connection. You must specify a database connection incoming on its parent replication process before you can create a publication.

Replication Process Property Sheet Subscriptions Tab

This tab lets you create or delete subscriptions. A *subscription* establishes a link between a publication and a database connection.

Replication Process Property Sheet Users Tab

This tab lets you create or delete users. A *user* corresponds to a person who is involved in the replication process.

Replication Process Property Sheet Connection Groups Tab

This tab lets you create or delete connection groups. A *connection group* is a set of data connections that can alternatively play the role of the source database. One of them is the primary database, the others are secondary databases and will be used in case of failure of the primary database.

Replication Process Property Sheet Event Scripts Tab

This tab lets you add event scripts using the Add Event Scripts tool in the toolbar or delete some. An *event script* is the implementation of an event on a replication process.

Replication Process Property Sheet Database Connection Tab

This tab lets you select a data source to send orders to the replication process.

For more information about how to define a data source, see [Database property sheet Database Connection tab](#) on page 20.

Publications (ILM)

A *publication* specifies published data or procedures for a data connection. You must specify a database connection incoming on its parent replication process before you can create a publication. When a data connection is deleted, all publications attached to it are deleted.

A publication can contain the following items:

- *Articles* – Specifies tables to replicate. For more information, see [Articles \(ILM\)](#) on page 31.
- *Procedures* – Specifies stored procedures to replicate. For more information, see [Procedures \(ILM\)](#) on page 35.
- *Subscriptions* – Specifies which data is replicated and where. This list is also available from the replication process property sheet. For more information, see [Subscriptions \(ILM\)](#) on page 36.

Publications have no symbol in the diagram.

Creating a Publication

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

- Select the Publications tab in a replication process property sheet.
- Right-click a replication process in the Browser, and select **New > Publication**.

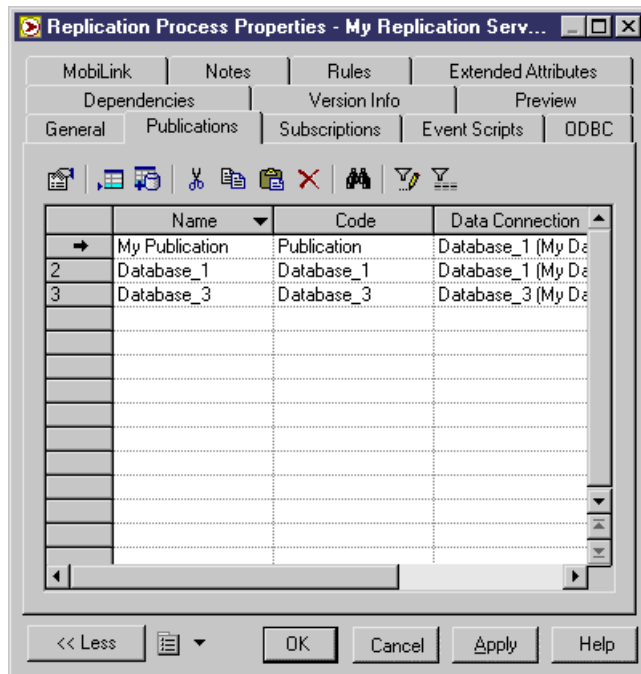
For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

1. Open a replication process property sheet and click the Publications tab.
2. Click the Add a Row tool, and then enter a name and code for the publication, and select a data connection from the Data Connection list. You must specify a database connection incoming on its parent replication process before you can create a publication.

or

Click the Add Publications for Data Connections tool, select one or more data connections from the selection dialog box, and then click OK. The selection list contains data connections incoming to the replication process to which the publication belongs

In the following dialog box, *My Publication* was created in the list and *Database_1* and *Database_3* were added from a selection list.



3. Click OK to close the property sheet and return to the model diagram.

Publication Properties

You can modify an object's properties from its property sheet. To open a publication property sheet, double-click its Browser entry in the Publications folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for publications.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Process	Specifies the replication process to which the publication belongs.
Data Connection	Specifies the connection for which data or procedures are published. Select an incoming database connection available from the list for the parent replication process.
Type	<p>[Only available with Replication Server parent] Specifies the publication type. Click the Preview tab to see the generated code. You can choose one of the following values:</p> <ul style="list-style-type: none"> • Database – to generate the entire database. • Publication – to generate publications, replication definitions and articles. • Replication Definition – to generate replication definitions only.

Publication Property Sheet Articles Tab

This tab lets you create, delete or add a tables or views selection taken from the database of the source or remote connection. An *article* is a PDM table or view to replicate. Before you can add tables or views from the remote database connection, you must define a subscription for the publication.

For more information about creating articles, see [Creating an article](#) on page 31.

Publication Property Sheet Procedures Tab

This tab lets you create, delete or add a stored procedure selection taken from the database of the source or remote connection. A *procedure* is a stored procedure from the source database. Before you can add stored procedures from the remote database connection, you must define a subscription for the publication.

Publication Property Sheet Subscriptions Tab

This tab lets you add a selection of subscriptions taken from a selection of outgoing data connections. A *subscription* establishes a link between a publication and a database connection to define where data published via the publication must be replicated.

Articles (ILM)

An *article* is a table or a view to replicate, and can contain the following items:

- *Columns* – Specifies the columns of the table to replicate.
- *Scripts* – Specifies the implementation for an event on an article.
- *Where Clause* – Lets you complement the code generation.

Articles have no symbol in the diagram.

Creating an Article

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

- Select the Articles tab in a publication property sheet.
- Right-click a publication in the Browser, and select **New > Article**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

1. Open a publication property sheet and click the Articles tab.
2. Click the Add a Row tool and enter a name and a code for the article.

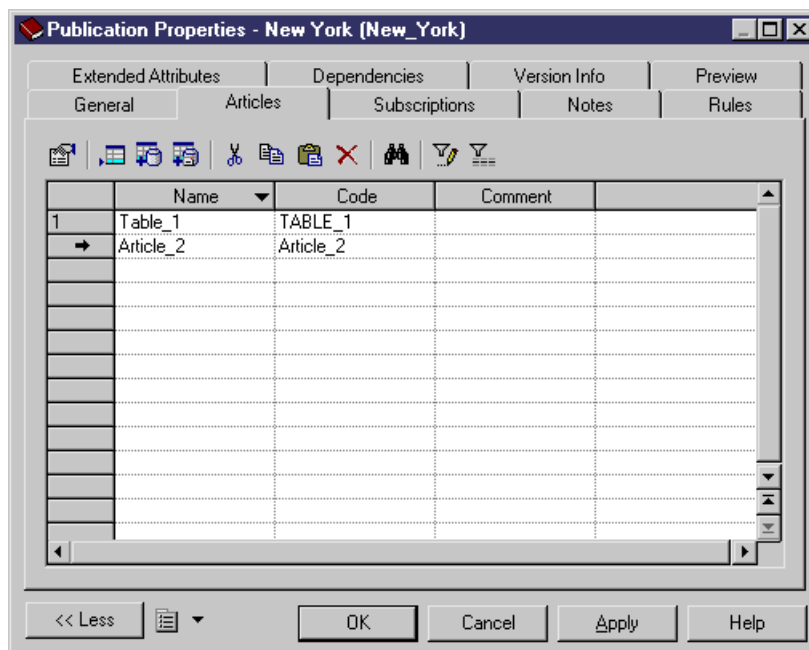
or

[PDM associated with the source database] Click the Add Article from Source Database tool, select one or more tables or views from the selection dialog box, and then click OK.

or

[PDM associated with at least one remote database] Click the Add Article from Remote Databases tool, select one or more tables or views from the selection dialog box, and then click OK.

In the following dialog box, *Article_2* was created in the list and *Table_1* was added from a selection list.



- Click OK to close the property sheet and return to the model diagram.

Article Properties

You can modify an object's properties from its property sheet. To open an article property sheet, double-click its Browser entry in the Articles folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for articles.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Publication	Specifies the publication to which the article belongs.
Source table	Specifies the source table or view to which the article corresponds. Click the Properties tool to the right of the field to display the source object property sheet.
Remote table	Specifies the remote table or view to which the article corresponds.

Article Property Sheet Columns Tab

This tab lets you create, delete or add a selection of columns taken from the database of the source or remote connection. An *article column* is a table or view column to replicate.

Article Property Sheet Where Clause Tab

This tab lets you create where clauses in a script editor. A *where clause* specifies a script that complements the SQL code generation to restrict the rows of a table to replicate. For example, you can replicate only the data of table SALES, which amount is > 1000 \$.

Note: You can use the Auto Completion List tool or press ctrl and space bar to display a contextual help for typing the clause. Click inside the clause text to close the list.

Article Property Sheet Event Scripts Tab

This tab lets you select one or more events for an article and specify their corresponding script. An *event script* allows you to define a script that implements an event on an article. This event script belongs to only one article. The list of available events is taken from the extended model definition used for the parent replication process and changes accordingly. Event scripts also apply to replication processes.

Columns (ILM)

An *article column* is a table or view column to replicate. Article columns have no symbol in the diagram.

Creating a Column

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

- Select the Columns tab in an article property sheet.
- Right-click an article in the Browser, and select **New > Article Column**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

1. Open an article property sheet and click the Columns tab.
2. Click the Add a Row tool and enter a name and a code for the article column.

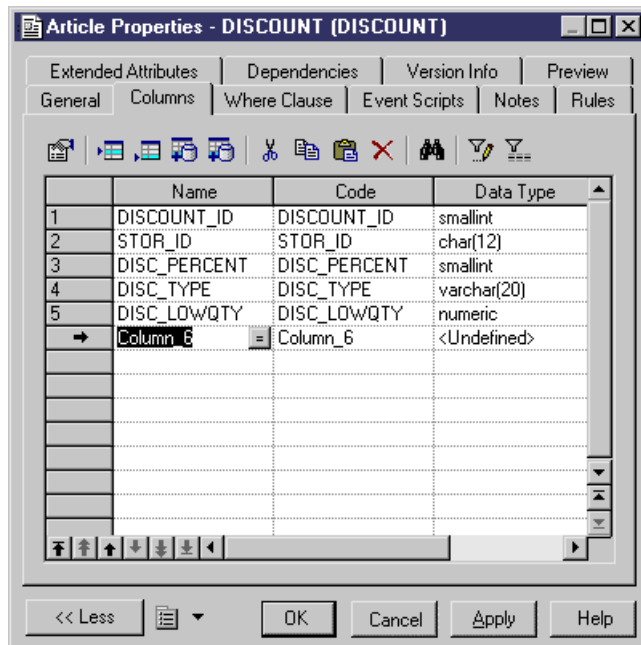
or

[PDM associated with source database] Click the Add Columns from Source Database tool, select one or more table or view columns from the selection dialog box, and then click OK.

or

[PDM associated with at least one remote database] Click the Add Columns from Remote Databases tool, select one or more table or view columns from the selection dialog box, and then click OK.

In the following dialog box, *Column_6* was created in the list and the other columns were automatically added when creating the corresponding article.



3. Click OK to close the property sheet and return to the model diagram.

Column Properties

You can modify an object's properties from its property sheet. To open a column property sheet, double-click its Browser entry in the Columns folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Article	Specifies the article to which the column belongs.
Source column	Specifies the source table or view column to which the article column corresponds. Click the Properties tool to the right of the field to display the source object property sheet.
Remote column	Specifies the remote table or view column to which the article column corresponds
Data type	Specifies the form of the data corresponding to the column, such as numeric, alphanumeric, boolean, or other.
Length	Specifies the maximum length of the data type.
Precision	Specifies the maximum number of places after the decimal point.

Procedures (ILM)

A *procedure* is a stored procedure from the source database. A stored procedure is a precompiled collection of SQL statements stored under a name and processed as a unit.

Procedures have no symbol in the diagram.

Creating a Procedure

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

- Select the Procedures tab in a publication property sheet.
- Right-click a publication in the Browser, and select **New > Procedure**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

1. Open a publication property sheet and click the Procedures tab.
2. Click the Add a Row tool and enter a name and a code for the procedure.

or

[PDM associated with source database] Click the Add Procedure from Source Database tool, select one or more tables or views from the selection dialog box, and then click OK.

or

[PDM associated with at least one remote database] Click the Add Procedure from Remote Databases tool, select one or more tables or views from the selection dialog box, and then click OK.

3. Click OK to close the property sheet and return to the model diagram.

Procedure Properties

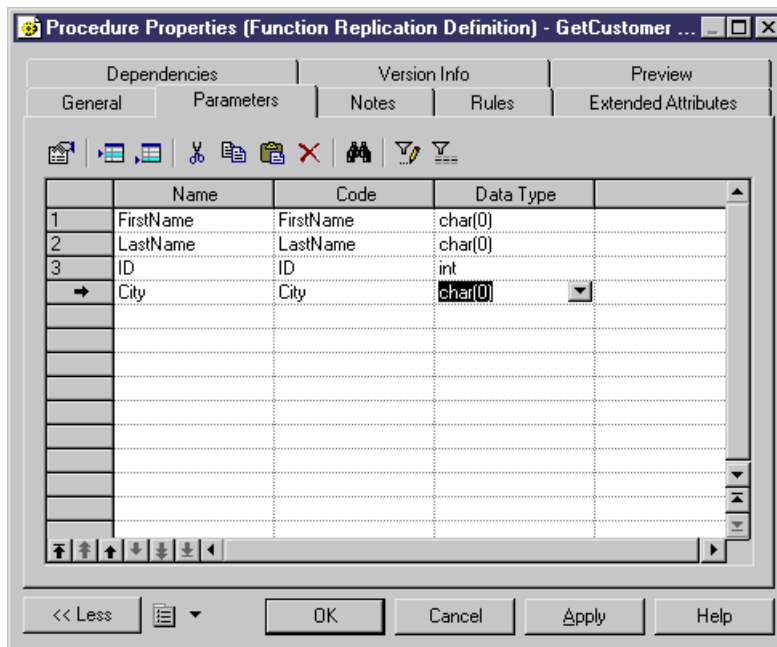
You can modify an object's properties from its property sheet. To open a procedure property sheet, double-click its Browser entry in the Procedures folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for procedures.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Parent	Specifies the publication in which the procedure is defined.
Source procedure	Specifies the procedure in the source database.
Remote procedure	Specifies the procedure in the remote database. By default this name is identical to the source procedure name, but you can modify it.

Procedure Property Sheet Parameters Tab

This tab lets you create or delete the call parameters of the procedure. In the following example, the GetCustomer procedure has a FirstName parameter with a char(0) data type:



Each parameter contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Procedure	Specifies the procedure to which the parameter belongs.
Data type	Specifies the form of the data corresponding to the parameter, such as numeric, alphanumeric, boolean, or other.
Length	Specifies the maximum length of the data type.
Precision	Specifies the maximum number of places after the decimal point.

Subscriptions (ILM)

A *subscription* establishes a link between a publication and a database connection to define where data published via the publication must be replicated.

Subscriptions have no symbol in the diagram.

Creating a Subscription

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

- Select the Subscriptions tab in a replication process property sheet.
- Right-click a replication process in the Browser, and select **New > Subscription**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Subscription Properties

You can modify an object's properties from its property sheet. To open a subscription property sheet, double-click its Browser entry in the Subscriptions folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Process	Specifies the replication process to which the subscription belongs.
Publication	Specifies the publication that publishes the data to subscribe to.
Data Connection	Specifies the connection to the remote database subscribing to the publication. Select an outgoing database connection of the parent replication process from the..

Subscription Property Sheet Where Clause Tab

This tab lets you create where clauses in a script editor. A *where clause* specifies a script that complements the SQL code generation to restrict the rows of a table to subscribe to in the remote databases. For example, you can replicate only the data of table SALES, which amount is > 1000 \$.

Note: You can use the Auto Completion List tool or press ctrl and space bar to display a contextual help for typing the clause. Click inside the clause text to close the list.

Users (ILM)

A *user* is a person involved in the replication process, and who is allowed to connect to the database.

Users have no symbol in the diagram.

Creating a User

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

- Select the Users tab in a replication process property sheet and click the Add a Row tool.
- Right-click a replication process in the Browser, and select **New > User**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

User Properties

You can modify an object's properties from its property sheet. To open a user property sheet, double-click its Browser entry in the Users folder.

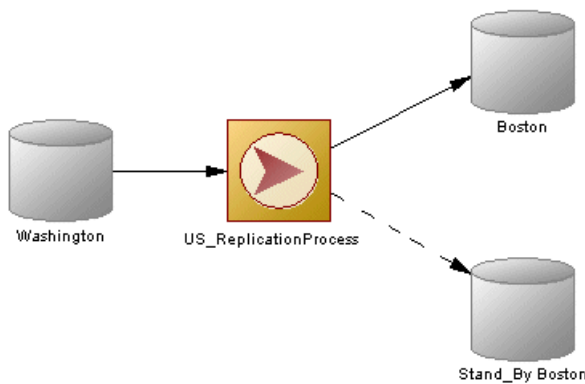
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Data Connection Groups (ILM)

A *data connection group* is a set of data connections that provides a relay in case the primary database goes down. Remote databases continue to get data from either primary or secondary data connections through the replication processes. You switch from a data connection to another in the Data Connection Group property sheet.

In the following example, US_ReplicationProcess reads data in Washington, and writes data in Boston. If the data connection to Boston fails, Stand_By Boston is the relay:



Creating a Data Connection Group

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

- Select the Connection Groups tab in a replication process property sheet.
- Right-click a replication process in the Browser, and select **New > Data Connection Group**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Data Connection Group Properties

You can modify an object's properties from its property sheet. To open a data connection group property sheet, double-click its Browser entry in the Data Connections Group folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for data connections groups.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Process	Specifies the name of the replication process to which the connection group belongs.
Default connection	Specifies the data connection to use during the replication process.

Data Connection Group Property Sheet Connections Tab

This tab lets you add one or more data connections to a connection group. If one data connection fails, you can use another one, without causing the abortion of the whole process.

1. Double-click a connection group in the Connection Groups tab of the replication property sheet and click the Connections tab.
2. Click the Add Data Connections tools from the toolbar to display a selection list.
3. Select one or more data connections and click OK to close the list.
4. Click OK again to close the property sheet, and return to the diagram.

Event Scripts (ILM)

An *event script* allows you to define how events on the replication process or on an article are implemented. Event scripts have no symbol in the diagram.

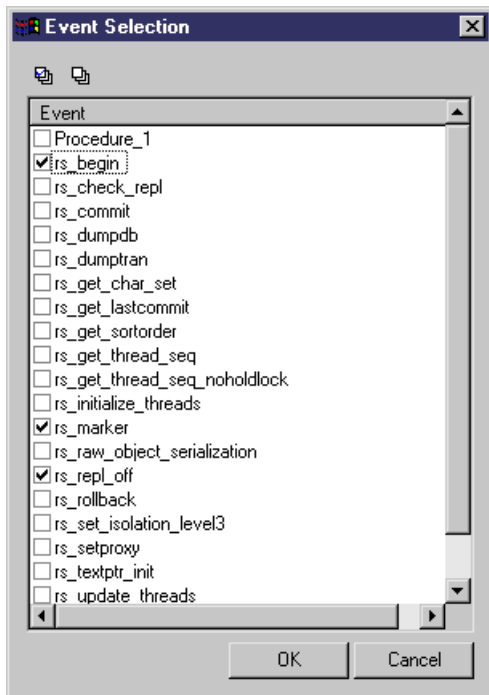
Creating an Event Script

You can create an event script in any of the following ways:

- Select the Event Scripts tab in a replication process property sheet or in an article property sheet and click the Add a Row tool.
- Right-click a replication process or an article in the Browser, and select **New > Event Script**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

1. Open a replication process property sheet and click the Event Scripts tab.
2. Click the Add Event Scripts tool in the toolbar to open the Event Selection dialog box, which contains the event scripts specified in the extended model definition you use for your replication process.
3. Select one or more event script, and click OK to close the dialog box, and return to the Event Scripts tab, which is filled with the events you have selected.



4. Select an event, and click the Properties tool in the toolbar to open its property sheet.
5. Enter a script definition in the edit zone, and click OK in each of the dialog boxes.

Event Script Properties

You can modify an object's properties from its property sheet. To open an event script property sheet, double-click its Browser entry in the Event Scripts folder.

The General tab contains the following properties:

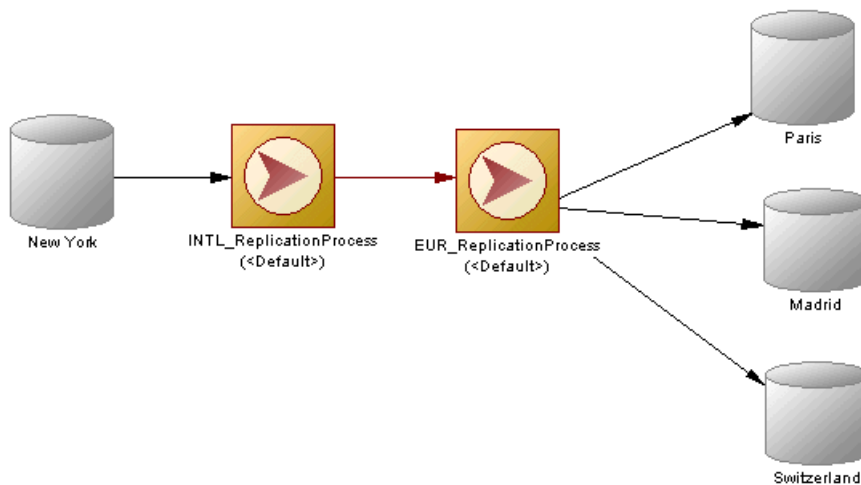
Property	Description
Parent	Specifies the replication process to which the event script belongs.
Event	Specifies the event script.
Comment	Specifies a descriptive comment for the event script.
[Text box]	Specifies the script definition.

Note: You can use the Auto Completion List tool or press ctrl and space bar to display a contextual help for typing the clause. Click inside the clause text to close the list.

Process Connections (ILM)

A *process connection* is a relationship that specifies the way data are moved between two replication processes.

In the following example, INTL_ReplicationProcess reads data contained in New York, and writes data into EUR_ReplicationProcess, which in turn writes data into Paris, Madrid, and Switzerland:



Creating a Process Connection

You can create a process connection in any of the following ways:

- Use the Connection tool in the diagram Palette.
- Select **Model > Process Connections** to access the List of Process Connections, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Process Connection**.

When you create a process connection from the Browser or from the List of Process Connections, you must have previously defined source and destination objects for the process connection.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Process Connection Properties

You can modify an object's properties from its property sheet. To open a process connection property sheet, double-click its diagram symbol or its Browser entry in the Process Connections folder.

The General tab contains the following properties:

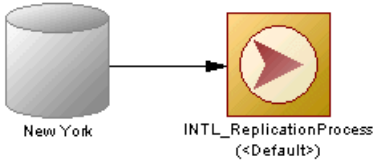
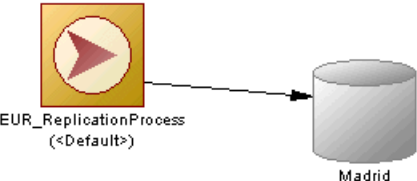
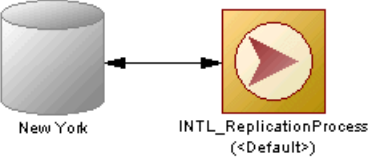
Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Target Process	Specifies the target extremity of the process connection. This can be a replication process or a transformation process. 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.
Source Process	Specifies the source extremity of the process connection. This can be a replication process or a transformation process. You can click the Properties tool to the right of the list to display the object property sheet.

Data Connections (ILM)

A *data connection* is a relationship that specifies the way data are copied between:

- A database and a replication process - to replicate data from a database to another.
- A data store (database, XML document, business process, and flat file) and a transformation process – to model data inputs and outputs in the modeling of data transformations. For more information about data transformations, see [Building Data Transformation Diagrams](#) on page 54.
- Two data stores with associated models – to show the way data are mapped between models.

The way you draw a data connection determines its direction, and the way objects at its extremities use the data:

Type	Description
<p>Read-only</p> 	<p>The replication process reads only (extracts) data from the database.</p>
<p>Write-only</p> 	<p>The replication process writes only (consolidates) the data in the database.</p>
<p>Read/Write:</p> 	<p>The replication process reads and writes data from and in the database.</p>

Creating a Data Connection

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

- Use the Connection tool in the diagram Palette.
- Select **Model > Data Connections** to access the List of Data Connections, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Data Connection**.

When you create a data connection from the Browser or from the List of Data Connections, you must have previously defined source and destination objects for the data connection.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Data Connection Properties

You can modify an object's properties from its property sheet. To open a data connection property sheet, double-click its diagram symbol or its Browser entry in the Data Connections folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the object, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can enter stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
[Process or data store]	Specifies one extremity of the data connection. This can be a replication process, transformation process or a data store. 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.
[Data store]	Specifies the other extremity of the data connection. This can be any data store. 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.
Access Type	Specifies the data connection direction. You can choose one of the following access types to access data in a data store <ul style="list-style-type: none"> • Write-only • Read only • Read/Write

Migrating Deprecated Model Container Objects into a Project

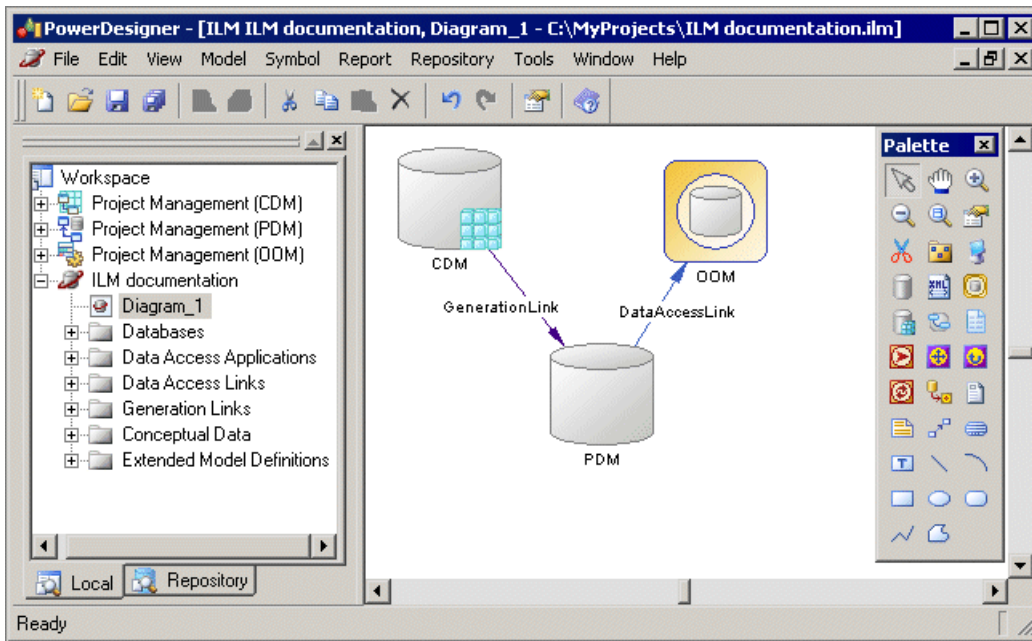
PowerDesigner projects, a new way of organizing and representing the links between models, was introduced in version 15.0, and can represent your former ILM diagrams.

The Information Liquidity Model (ILM) is used to model data movement and replication. Some ILM objects in version 12.5 enabled you to show in a diagram containers for models and the generation and mapping links between them. This was inconsistent with the purpose of the ILM.

The following objects are no longer available in the Information Liquidity Model (ILM):

- Conceptual data – container for conceptual data models.
- Data access application – container for object-oriented models.
- Data access link – relationship that documents the way data are mapped between model containers.
- Generation link – relationship that documents generation dependencies between model containers.

In the following example, the ILM version 12.5 lets you show how a CDM, a PDM, and an OOM are linked by generation and data access links:



PowerDesigner projects enable you to:

- Gather together and display in a diagram any types of PowerDesigner models and other files.
- Display different types of link, such as shortcuts, references, traceability links and so on.
- Benefit from the automatic update of links.
- Check in and out all the models and other files contained within the project from the repository in one operation.

For detailed information about projects, see the Projects and Framework Matrices chapter of the *Core Features Guide*.

Creating a Project

Create a project to contain the models whose links you want to view.

1. Select **File > New Project** to open the New Project dialog box.
2. Select **Empty Project** in the tree, enter a project name and location, and select the **Append Name To Location** check box if you want to add the project name to the root directory.
3. Click **OK** to close the dialog box, and create the project.

The project is created in the Browser, and an empty project diagram opens.

Adding Models to a Project Diagram

Complement a project diagram by adding models whose links you want to view.

- Drag and drop one or more models from the file system to the Browser or from the Browser to the project diagram (see "Adding existing model and file documents to a project" in the Project and Framework Matrices chapter of the *Core Features Guide*), or
- Click the **Add Project Document** tool in the Palette, click in the diagram to open a standard Open dialog box, browse to and select one or more models in your file system, and then click **Open** (see "Creating a project" in the Project and Framework Matrices chapter of the *Core Features Guide*).

In order to maximize the convenience of the project as a container, you should create (or place) all the associated models inside the project directory. However, you can also link to files outside the project directory. Such files are listed under the project node in the Browser, but display small icons on their symbol to indicate that they are located outside the project folder. You can, at any time, right-click a model in the Browser or its symbol in the diagram, and select **Move to Project Directory** to move it inside the project.

Note: We recommend that your models are open when you add them to a project in order to guarantee that their dependency links are correctly rebuilt.

Rebuilding Dependency Links in a Project Diagram

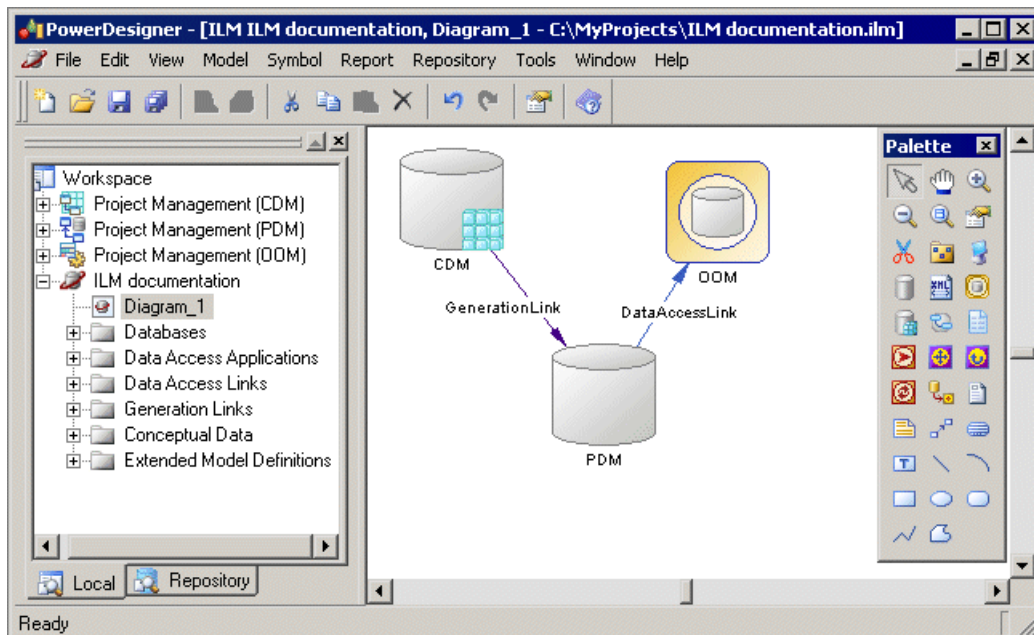
Create dependency links (for example, generation, mappings, shortcuts, and so on) between models in a project diagram in the models themselves. You cannot manually create them. Dependency links are automatically generated in your project diagram when you add linked models to it.

Models that are included in the project, but which are not displayed in the project diagram will not be added nor have their links represented when you rebuild dependency links.

Note: Note Models must be present in the project diagram before you can rebuild their dependency links.

1. Select **Tools > Rebuild Dependency Links** to open the Rebuild Dependency Links dialog box.
2. Select the check boxes that correspond to the dependency links you want to rebuild.
3. Click **OK** to close the dialog box and return to the diagram.
Any missing links are updated in the diagram.

The following example shows the interconnected models in a project diagram. The CDM, PDM, and OOM are linked by generation and mapping links:



You can explore the details of any of the dependency links in your diagram by right-clicking it and selecting **Show Dependencies**. Each type of link has its own viewer:

- **Generation** – displays the generation links between models in the Generation Links Viewer (see "Using the Generation Links Viewer" in the Linking and Synchronizing Models chapter of the *Core Features Guide*).
- **Mapping** – displays the mapping links between models in the Mapping Editor (see the Creating Mappings chapter of the *Core Features Guide*).
- **Reference** – displays the shortcuts and replications between models in the Shortcuts and Replications dialog box.

Building Data Transformation and Control Flow Diagrams

The PowerDesigner Information Liquidity Model (ILM) lets you model and document ETL and EII processes in a rich graphical environment, supported by sophisticated metadata. By modeling your transformations in PowerDesigner, you can benefit from its powerful traceability and impact analysis features.

In most enterprises, information is stored in multiple databases, data warehouses and applications. Such a situation requires the recombination and transformation of data coming from diverse sources into new formats for replication reporting or other consumption.

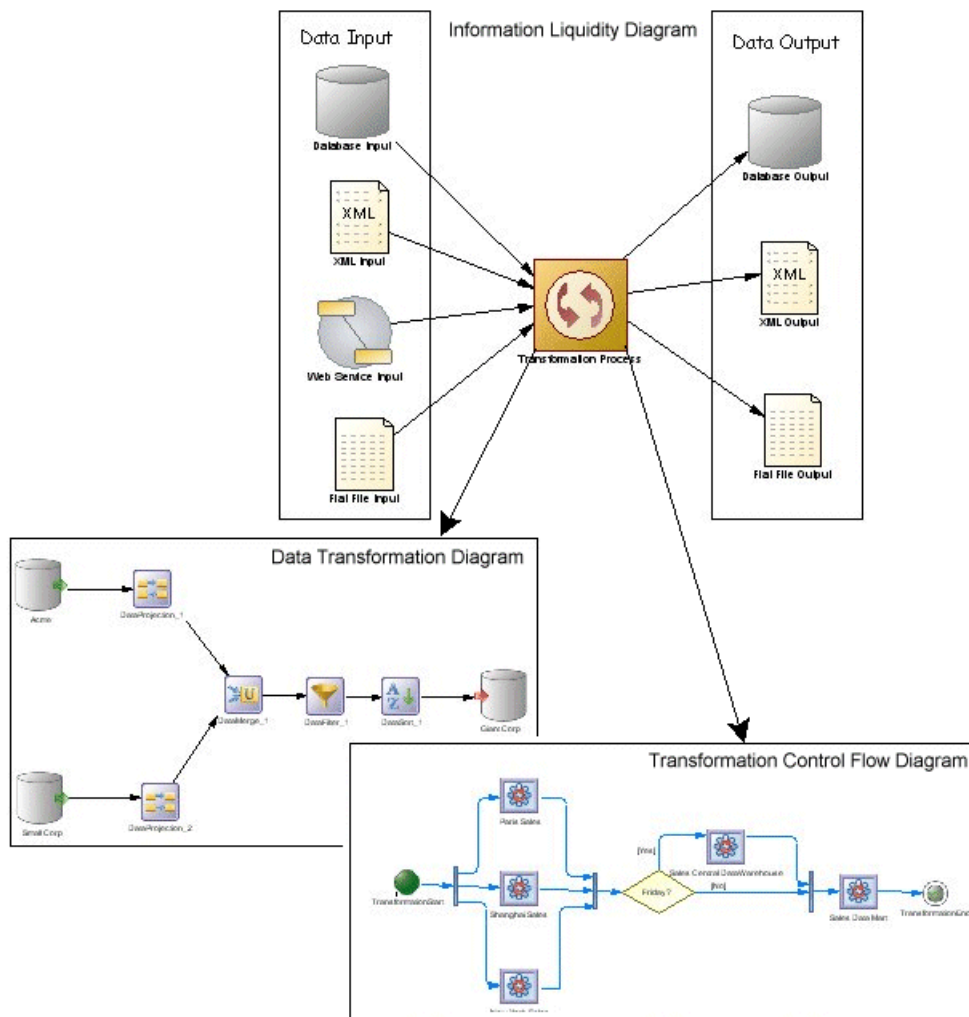
ETL (Extract Transform and Load) and EII (Enterprise Information Integration) are two technologies that address this need:

- *ETL* is a process in data warehousing that refers to three separate functions combined into a single programming tool:
 - Extract – reads data from a specified source system.
 - Transform – manipulates the extracted data to convert it for particular specialized consumption.
 - Load – writes the resulting data to a target.
- *EII* combines raw data by orchestrating and organizing queries to diverse source systems without any previous aggregation of the content. The data is not permanently moved into a new location.

The ILM lets you model and analyze data transformations in high-level and detailed views as follows:

- Information liquidity diagram – high-level diagram, which enables you to model a high-level view of your data transformations by specifying:
 - Input sources – Databases (PDMs), XML documents (XSMs), business processes (BPMs), and flat files such as .CSV and .XLS
 - Transformation process – where the transformations occur. It contains lower level detailed transformation diagrams (see [Transformation Processes](#) on page 59).
 - Output sources – Databases (PDMs), XML documents (XSMs), and flat files such as .CSV and .XLS
- Data transformation diagram – low level diagram, which enables you to model a transformation task by specifying how data is extracted from data inputs, transformed by actions and loaded into data outputs. Data inputs and outputs are linked to the input and output sources specified in the high-level diagram.
- Transformation control flow diagram – low level diagram, which enables you to specify the sequence of execution of a series of tasks.

The following example shows how input and output sources can be linked to a transformation process at a high level, and how the transformation is modeled in the lower level diagrams:



Modeling Data Transformations

You can model a data transformation in PowerDesigner by:

- Creating each transformation object individually (see [Building Data Transformation Diagrams](#) on page 54 and [Building Transformation Control Flow Diagrams](#) on page 57).
- Using a Wizard (see [Creating a data transformation with the Transformation Wizard](#) on page 49 and [Creating a data transformation with the Convert Mappings to ETL Wizard](#) on page 51).

We recommend that you:

1. Identify any input and output sources. These may be existing PDMs, XSMs, BPMs and flat files or live data sources that can be reversed engineered.
2. Create an ILM and launch one of the following wizards to create your basic transformation environment:
 - Transformation Wizard – see [Creating a data transformation with the Transformation Wizard](#) on page 49.
 - Convert Mappings to ETL Wizard [for existing PDM mappings] – see [Creating a data transformation with the Convert Mappings to ETL Wizard](#) on page 51.
3. Open the created data transformation diagram(s), and specify any other necessary transformation objects, such as data query executions, calculators, etc. (see the [Building Data Transformation Diagrams](#) on page 54).
4. [optional] Create a control flow diagram to detail the order in which a series of data transformation tasks is executed (see [Building Transformation Control Flow Diagrams](#) on page 57).

You can use the completed model to guide you in writing a transformation for your ETL engine.

Creating a Data Transformation with the Transformation Wizard

The Transformation Wizard helps you set up a basic transformation process with input and output sources, and automatically creates one or more data transformation diagrams.

1. Launch the Transformation Wizard in any of the following ways, and then click Next to go to the next step:

- Select **Tools > Transformation Wizard**.
- Right-click a transformation process, and select Transformation Wizard.
- Right-click an input or output source, and select Transformation Wizard.
- Right-click a task in the Browser, and select Transformation Wizard.



2. The Transformation Process page [not available if launched from a transformation process] lets you select or create the transformation process to use for the data to be transformed. You can:

- Select an existing transformation process by clicking the Browse button to the right of the list. If your model has an extended model definition attached, the Type list is automatically filled with the type of the selected process.
- Create a transformation process by entering a name. If your model has an extended model definition attached, you can specify a type for the transformation process.

Make your selection, and then click Next.

3. The Transformation Tasks page [not available if launched from a data transformation diagram] lets you create or select the task(s) that will contain the details of your transformation. You can choose between the following options:

- Create or select an existing task to gather all source objects in the same data transformation diagram.
- Create a task and a data transformation diagram for each source object.

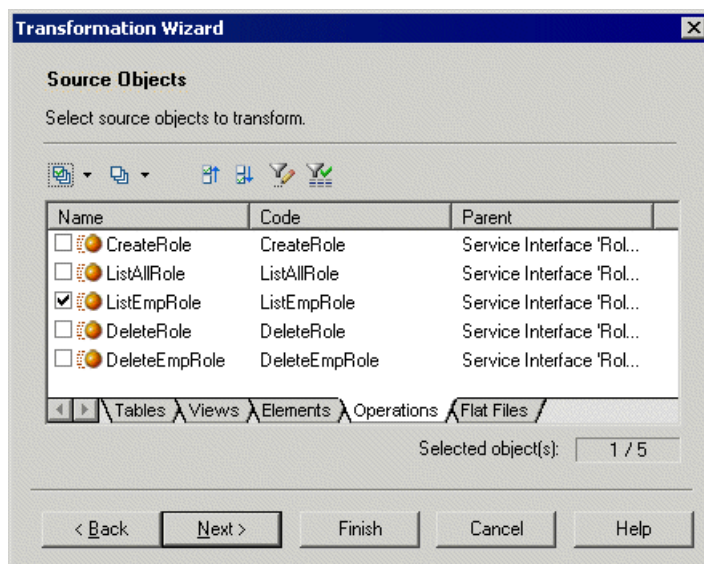
Make your selection, and then click Next.

4. The Source Models page lets you select the source models from which to extract the data to be transformed, and which will become input sources in the information liquidity diagram. Select one or more models open in the workspace. You can also:

- Click the Open Model tool to browse for other model files.
- Click the Reverse Engineer tool to create a new model by reverse engineering a database, XML schema or process language.

Make your selection, and then click Next.

5. The Source Objects page lets you select the source tables, views, elements, operations, and flat files to transform, and which will become data inputs in the data transformation diagram.



Make your selection, and then click Next.

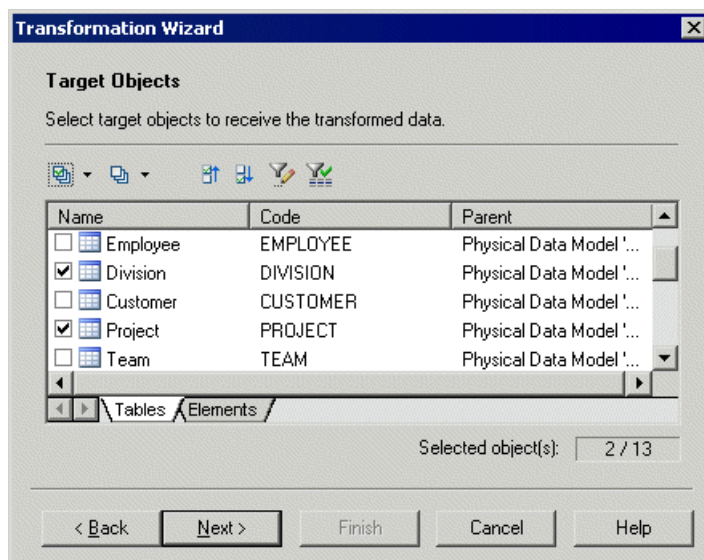
6. The Target Mode page lets you specify the location where you want to load your transformed data. You can choose between the following options:
 - Select existing models – in the following pages, you will select target models and objects, in which to load the data.
 - Create a new model [default if creating a task for each input source]. Enter a model name, type, and language, and click the Share or Copy radio button. Click Next to go to the completion page.
 - No target model [only available if creating a task for each input source]. Click Next to go to the completion page.

Make your selection, and then click Next.

7. The Target Models page [only available if creating a single task for all source objects] lets you select the target models to which to load the transformed data, and which will become output sources in the information liquidity diagram. Select one or more models open in the workspace. You can also:
 - Click the Open Model tool to browse for other model files.
 - Click the Reverse Engineer tool to create a new model by reverse engineering a database or XML schema.

Make your selection, and then click Next.

8. The Target Objects page [not available for new models] lets you select the target tables, views, elements, and flat files that will contain the transformed data, and which will become data outputs in the data transformation diagram.

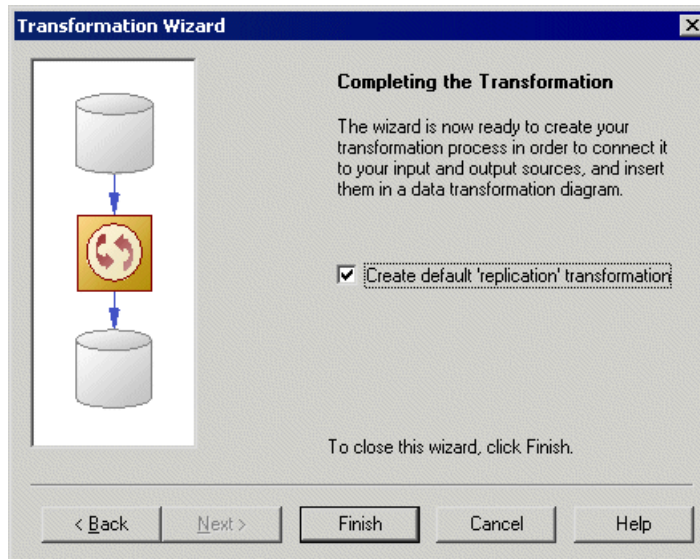


Make your selection, and then click Next.

9. The last page of the wizard summarizes the objects that will be created. You can choose to create a default transformation to connect input to output sources, and create a default replication, if their names match. This option is not available if you chose the No target model option.

When you click Finish the wizard creates:

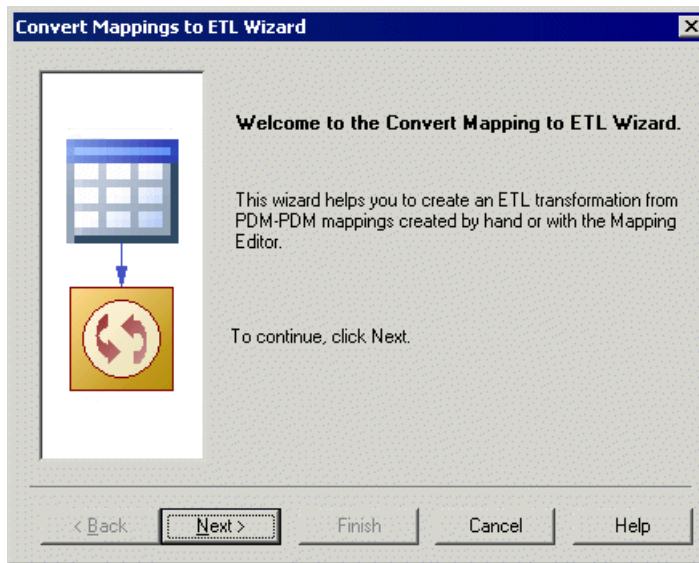
- An information liquidity diagram containing a transformation process connected to its input and output sources.
- One or more data transformation diagrams containing data inputs and, if required data outputs. Press ctrl and double-click the transformation process to open diagrams.



Creating a Data Transformation with the Convert Mappings to ETL Wizard

You can create a data transformation from an existing PDM-PDM mapping with the Convert Mappings to ETL Wizard. It helps you set up your transformation process with PDMs connected to it as input and output sources, along with basic data transformation diagrams with the appropriate data inputs and outputs, and appropriate actions.

1. Launch the Convert Mappings to ETL Wizard in any of the following ways, and click Next to go to the next step:
 - Select **Tools > Convert Mappings to ETL Wizard**.
 - Right-click a target database attached to one or more PDM(s) containing mappings, and select Convert Mappings to ETL Wizard.



2. The Database Selection page [not available if launched from a target database attached to a PDM containing mappings] lets you define the target database containing mappings. You can:

- Select an existing data connection.
- Type an existing database name or select an existing one using the Select a Database tool next to the list. This assumes that no data connection to your transformation process exists, and creates one.
- Type a new database name. This assumes that neither the database nor the data connection exists, and creates them.

Make your selection, and then click Next.

3. The Target Models page lets you select target models from which to extract mapping information, and which will serve as targets for the transformation. Any models attached to the previously selected database are selected by default. You can click the Open Model tool to browse for other model files.

Make your selection, and then click Next.

4. The Transformation Process page lets you select or create the transformation process to use for the data to be transformed. You can:

- Select an existing transformation process by clicking the Browse button to the right of the list. If your model has an extended model definition attached, the Type list is automatically filled with the type of the selected process.
- Create a transformation process by entering a name. If your model has an extended model definition attached, you can specify a type for the transformation process.

Make your selection, and then click Next.

5. The Transformation Tasks page lets you create or select the task(s) that will contain the details of your transformation. You can choose between the following options:

- Create a task to gather all source objects in the same data transformation diagram.
- Create a task and a data transformation diagram for each source object.

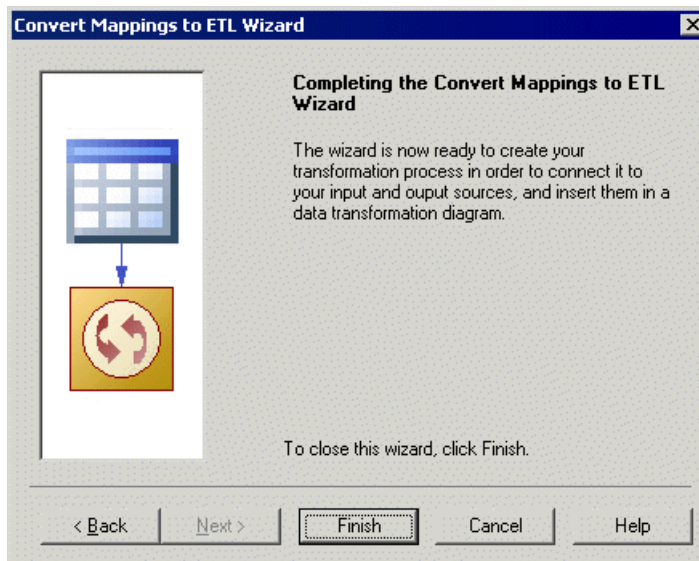
Make your selection, and then click Next.

6. The Target Objects page lets you select the target tables that will contain the transformed data.

Make your selection, and then click Next.

7. The last page of the wizard summarizes the objects that will be created:

- An information liquidity diagram containing a transformation process connected to its input and output sources.
- One or more data transformation diagrams containing data inputs and outputs, and any appropriate actions retrieved from the mapping conversion. Press ctrl and double-click the transformation process to open diagrams.



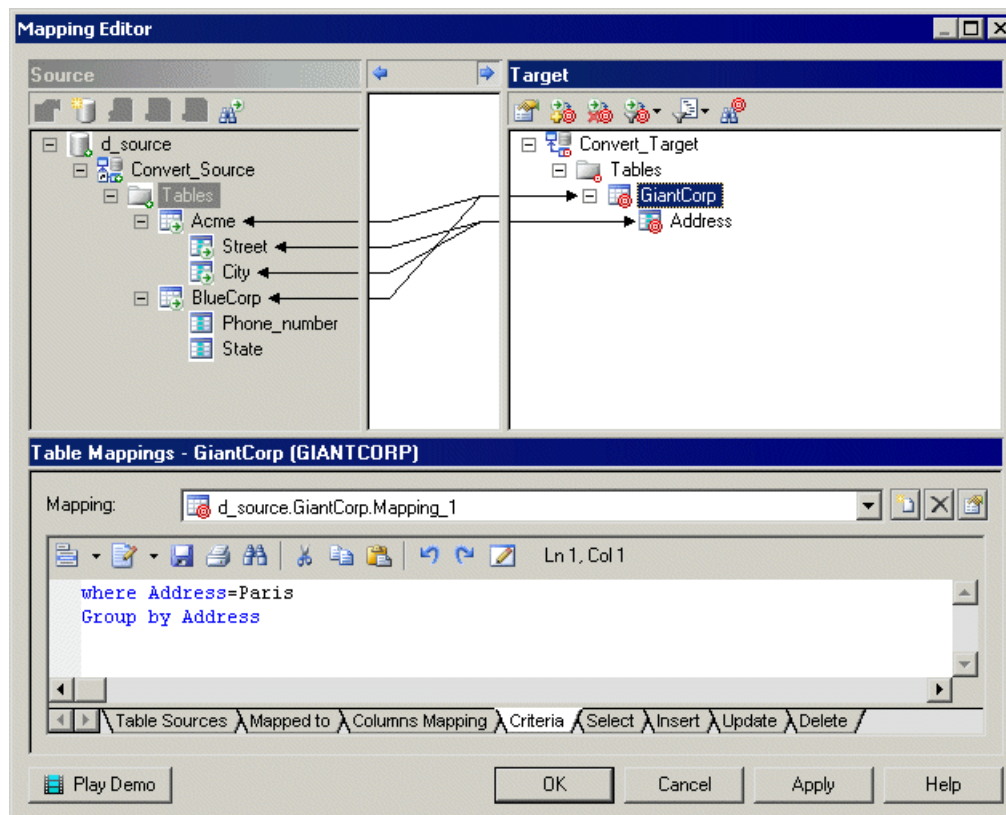
Mapping Conversion

A PDM-PDM mapping is specified in a query or a textual expression defined in a target table, and from which its source(s) and associated model(s) can be retrieved.

When you select a target mapped table, the wizard creates the corresponding output and retrieves the input sources from the mapped table. Additional objects (actions) can be created, if at least one of the following situations occurs:

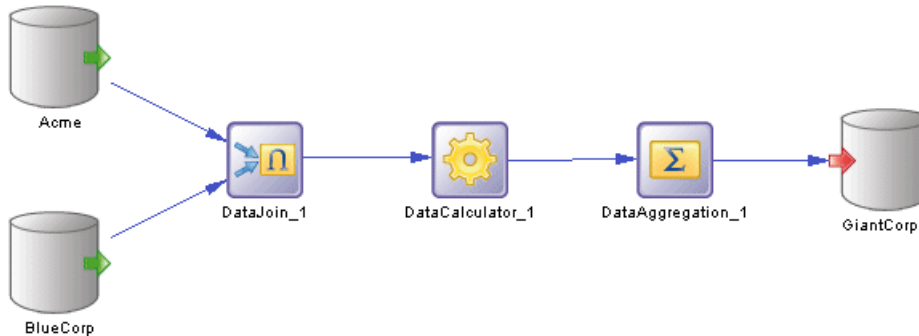
- The mapped table owns criteria (Where, Group by).
- The mapped table has more than one source.
- The columns of the mapped table have more than one source.

In the following example, the GiantCorp target table is mapped to the Acme and BlueCorp source tables, has Where and Group by criteria, and has an Address column mapped to the Street and City source columns:



The Convert Mappings to ETL Wizard will create the following objects in a data transformation diagram:

- The Acme and BlueCorp data inputs and a data join for the source tables.
- A data calculator for the two column sources and the Where criterion.
- A data aggregation for the Group by criterion.
- The GiantCorp data output for the mapped target table.

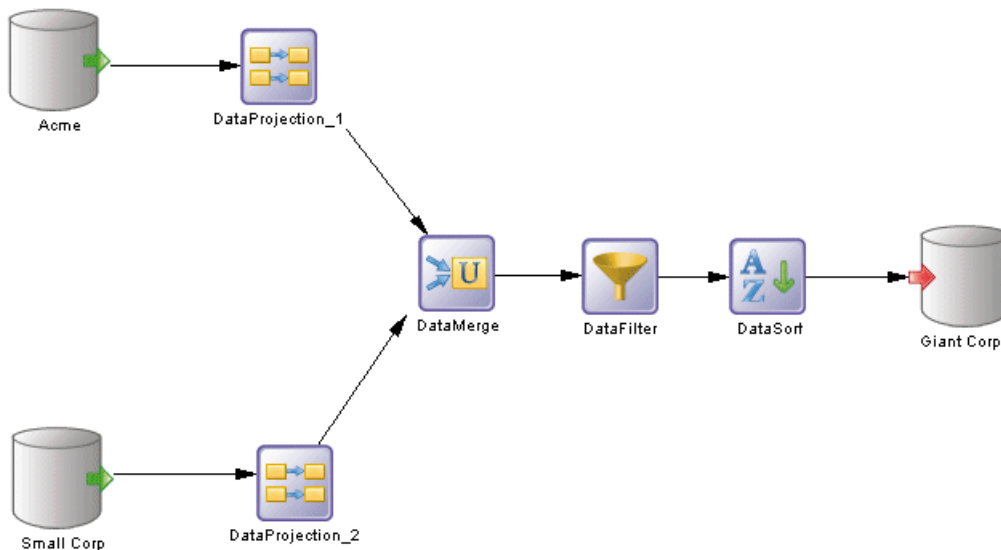


Data Transformation Diagram Basics

A *data transformation diagram* provides a graphical view of the inputs, outputs, and steps involved in a data transformation task.

Data comes from *data inputs*, are transformed by *actions*, and loaded to *data outputs*. These steps are linked together by *data flows*. The data that is to be transformed is represented by *data structure columns*, which are contained in each of these steps.

In the following example, data extracted from the Acme and Small Corp database inputs are merged into DataMerge, filtered by DataFilter, sorted by DataSort, and are then loaded into the Giant Corp database output:

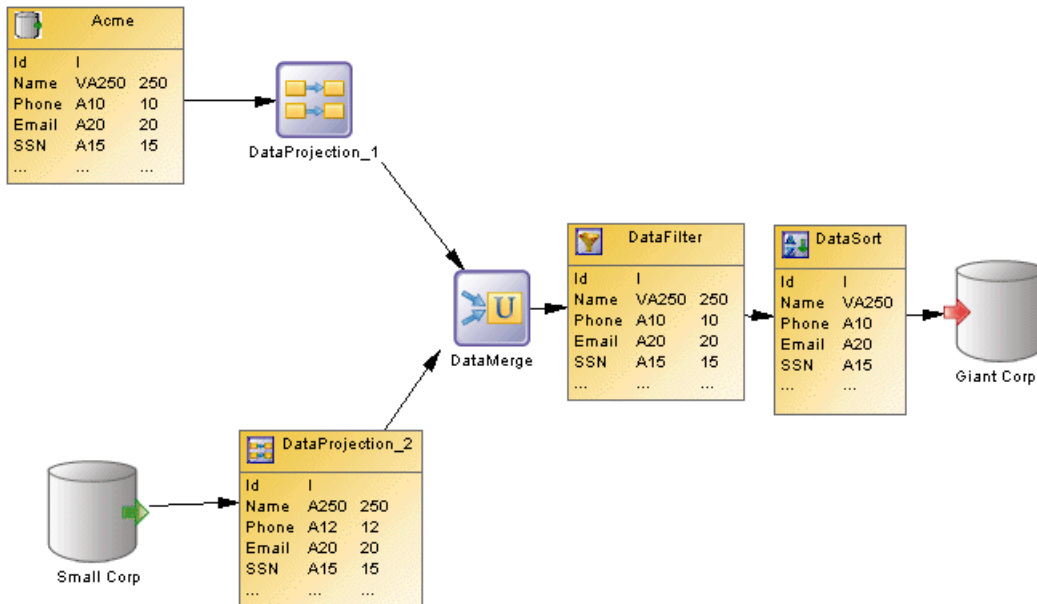


You can display a step's data structure columns directly in its symbol. To toggle between the basic and the detail formats:

- Right-click a symbol, and select Show Detail.
- *or*
- Select one or more symbols, and press *ctrl*+ *Q*.

The number of displayed columns is specified in the Object View page of the Display Preferences dialog box. See [Setting ILM Display Preferences](#) on page 90.

In the following example, Acme, Data Filter, DataSort, and DataProjection_2 display their data structure columns in their symbol:



Data Transformation Diagram Objects

You can create the following kinds of objects in a data transformation diagram:

- Data inputs — represent the sources from where data is extracted, such as a database or an XML document.
- Actions — specify how the data is transformed using, for example, data query executions or data aggregations.
- Data outputs — represent the targets to where data is loaded, such as a database or an XML document.
- Data flows — conveys data structure columns from one object to another.

Creating a Data Transformation Diagram

You create a transformation diagram in any of the following ways:

- Double-click a transformation process symbol that has no sub-diagram. A task and a data transformation diagram are created.
- Open the property sheet of a transformation process, click the Data Transformation Tasks tab, click the Add a Row tool to create a new transformation task, then click the Open Data Transformation Diagram tool to navigate to the new diagram.

Data Transformation Task Properties

You can modify an object's properties from its property sheet. To open a data transformation task property sheet, double-click its Browser entry in the Data Transformation Tasks folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for data transformation tasks.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.

Property	Description
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Data Transformation Task Property Sheet Inputs Tab

The *Inputs* tab displays a list of data inputs, in which you can add, edit, and delete inputs.

For more information about inputs, see [Data Inputs \(ILM\)](#) on page 61.

Data Transformation Task Property Sheet Actions Tab

The *Actions* tab displays a list of actions, and allows you to add, edit and delete actions.

For more information about actions, see [Actions \(ILM\)](#) on page 63.

Data Transformation Task Property Sheet Outputs Tab

The *Outputs* tab displays a list of data outputs, and allows you to add, edit and delete outputs.

For more information about outputs, see [Data Outputs \(ILM\)](#) on page 73.

Data Transformation Task Property Sheet Parameters Tab

The *Parameters* tab displays a list of parameters, and allows you to create, edit, or delete parameters.

For more information about parameters, see [Transformation Parameters \(ILM\)](#) on page 78.

Data Structure Mapping Editor

The Data Structure Mapping Editor allows you to visualize or define data structure columns in the Data Transformation Task Diagram. It is available from:

- The contextual menu of all data transformation steps (data inputs, data outputs, and actions)
- The data flow symbol
- The Data Structure Columns tab of all data transformation steps (data inputs, data outputs, and actions) using the Open Mapping Editor tool

You can use the Data Structure Mapping Editor to represent the mapping between the data structure columns of the source and target objects of a data flow.

The output data structure of a step becomes the input data structure of the next step and a mapping is defined between the output of the previous step and the input of the current step.

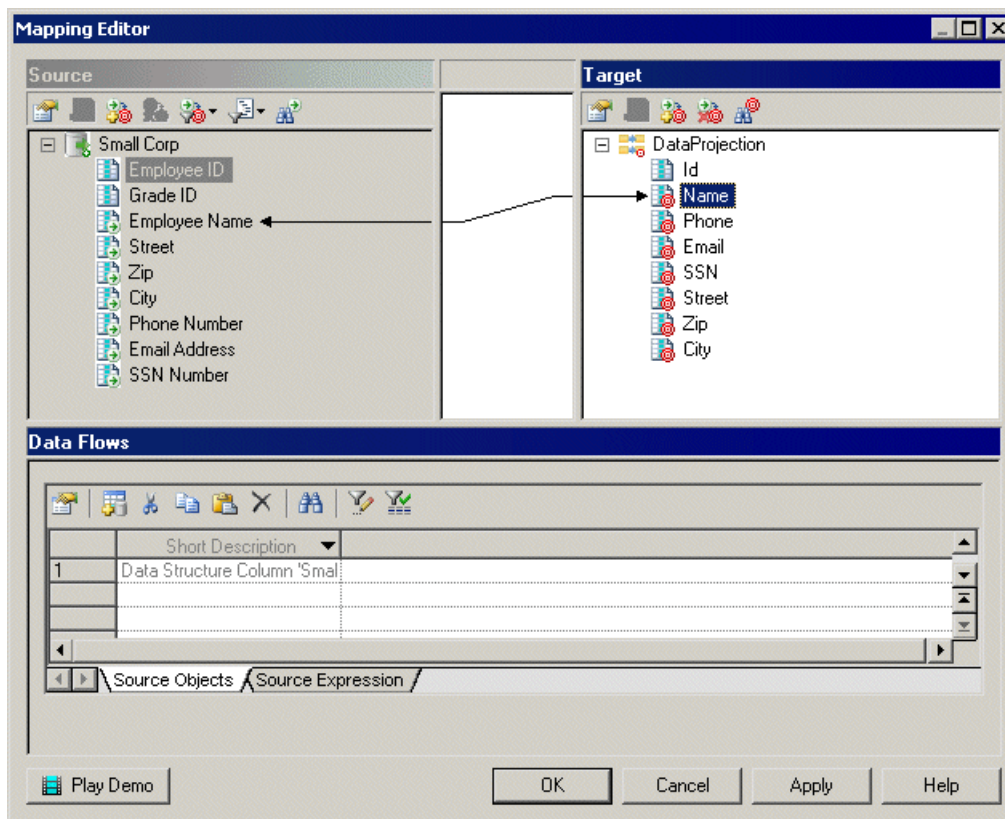
Types of Mapping

The object symbol from which you open the Data Structure Mapping Editor determines the type of mapping you can perform:

Open from a....	Description
Data input	Allows the mapping of a source PDM, XSM, BPM or flat file data structure columns to the current data input data structure columns. The target pane is active.

Open from a....	Description
Action	Allows the mapping of the data structure columns of a previous step to the current action data structure columns. The target pane is active.
Data output	Allows the mapping of a target PDM, XSM, or flat file data structure columns to the current data output data structure columns. The source pane is active.
Data flow	Allows the mapping of the data structure columns of the source and target objects of the flow. The target pane is generally active, except when the data flow links an action and an output. In this case the Source pane is active.

In the following example, the Mapping Editor shows the mapping between the Employee Name and the Name data structure columns. The Target pane is active, and the Data Flows pane lets you add source objects for the current column, and edit its source expression:



Creating a Mapping from the Data Structure Mapping Editor

You can create a mapping in the Mapping Editor in any of the following ways:

- Drag an object from one pane and drop it on an object in the other.
- Select an object in each of the target and source panes, and then click the Create Mapping between Source and Target Objects tool.
- Select an object in each of the target and source panes, right-click one, and select Create Mapping.

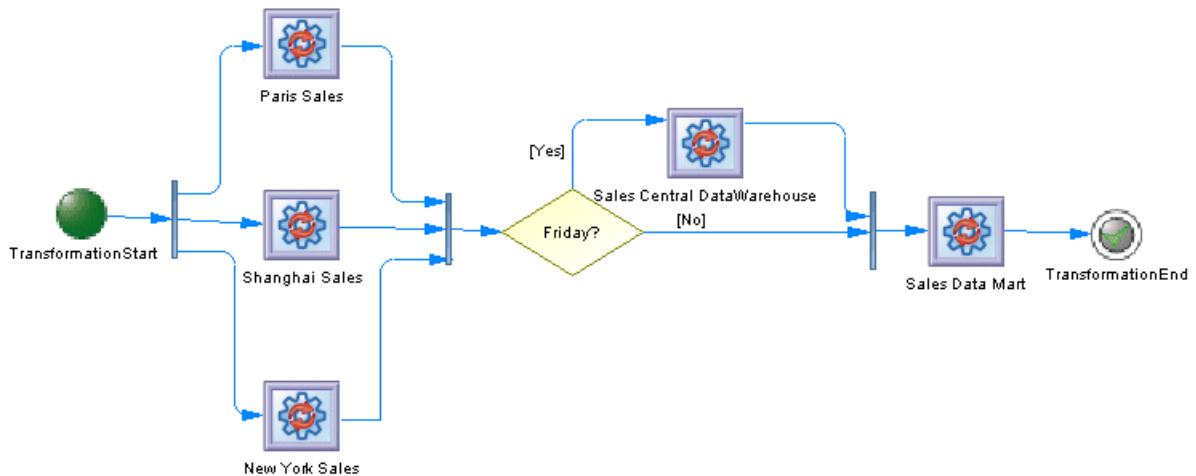
For more information, see the "Creating Mappings" chapter in the *Core Features Guide*.

Transformation Control Flow Diagram Basics

A *transformation control flow diagram* provides a graphical view of the order in which a series of data transformation tasks is linked together in a control flow.













The transformation control flow can include *starts*, *task executions*, *decisions*, *synchronizations*, and *ends*, which are linked together by *control flows*

In the following example, the Paris Sales, Shanghai Sales, and New York Sales tasks are performed in parallel. If it is Friday, the Sales Central DataWarehouse task is executed. Whether or not it is Friday, the Sales Data Mart transformation task execution is performed:



Transformation Control Flow Diagram Objects

You can create the following objects in a transformation control flow diagram:

Object	Tool	Symbol	Description
Transformation start			Starting point, which initiates the execution of a transformation control flow (see Transformation Starts (ILM) on page 80).
Transformation task execution			Instance of one or more data transformation tasks (see Transformation Task Executions (ILM) on page 81).
Transformation synchronization			Synchronization of the execution of several tasks (see Transformation Synchronizations (ILM) on page 83).
Transformation decision			Choice to make when different paths are possible (see Transformation Decisions (ILM) on page 84).
Transformation end			Ending point, which terminates the execution of a transformation control flow (see Transformation Ends (ILM) on page 85).
Control flow			Oriented link between objects (see Control Flows (ILM) on page 86).

Creating a Transformation Control Flow Diagram

You create a transformation control flow diagram in the following way:

- Open the property sheet of a transformation process, click the Transformation Control Flows tab, click the Add a Row tool to create a new transformation control flow, then click the Open Transformation Control Flow Diagram tool to navigate to the new diagram.

Transformation Control Flow Properties

You can modify an object's properties from its property sheet. To open a transformation control flow property sheet, double-click its Browser entry in the Transformation Control Flows folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Transformation Control Flow Property Sheet Task Executions Tab

The Task Executions tab displays a list of transformation task executions, and allows you to create, edit and delete task executions.

. A *task execution* is an instance of one or more data transformation tasks (see [Transformation Task Executions \(ILM\)](#) on page 81).

Transformation Processes (ILM)

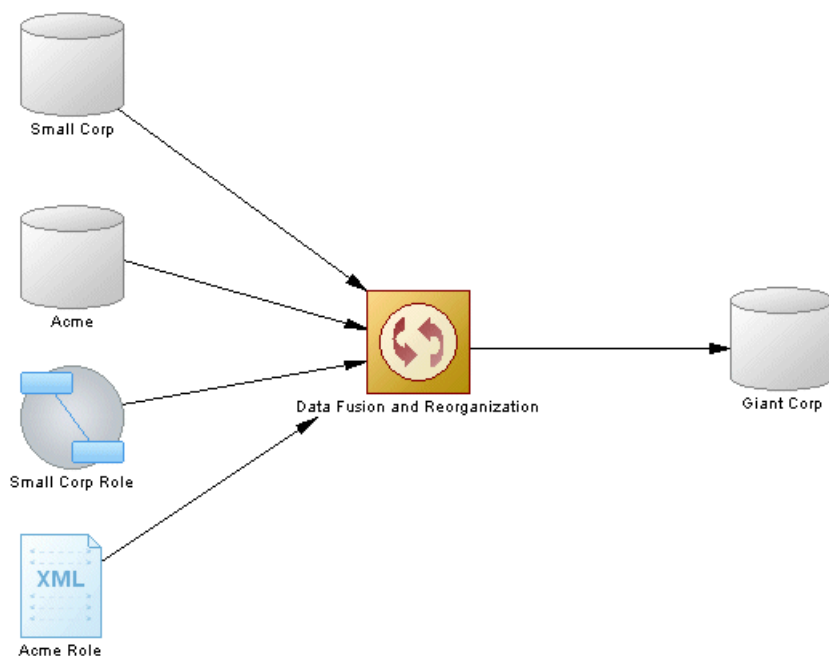
A *transformation process* is a high-level process, which models the extraction of data from input sources, its transformation, and loading to output sources. Input and output sources are linked to databases, flat files, XML documents, or business processes.

The transformation process can contain:

- Data Transformation Diagrams – identify data transformation tasks, which model the extraction, transformation, and loading of data. For detailed information, see [Building Data Transformation Diagrams](#) on page 54.
- Transformation Control Flow Diagrams – detail the order in which data transformation tasks are executed. For detailed information, see [Building Transformation Control Flow Diagrams](#) on page 57.

Press ctrl and double-click a transformation process symbol to open its diagram. When several diagrams are available, a selection dialog box opens to let you select the diagram you want.

In the following example, the Small Corp, Acme, Small Corp Role, and Acme Role input sources are transformed by the Data Fusion and Reorganization transformation process, and then loaded to the Giant Corp output sources:



Creating a Transformation Process

You can create a transformation process in any of the following ways:

- Use the Transformation Wizard.
- Use the Convert Mappings to ETL Wizard.
- Use the Transformation Process tool in the diagram Palette.
- Select **Model > Process Transformations** to access the List of Process Transformations, and click the Add a Row tool.
- Right-click the model or package in the Browser, and select **New > Transformation Process**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Process Properties

You can modify an object's properties from its property sheet. To open a transformation process property sheet, double-click its diagram symbol or its Browser entry in the Transformation Processes folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Server	Name of the server on which the transformation process is deployed. 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.

Property	Description
Type	Type of the engine you are using specified in an extended model definition (XEM). It influences the generation and the display of additional information within specific tabs. If you change the type, the data types used by the data structure columns that exist beneath in the different transformation steps will be converted to their equivalents in the new transformation engine. This conversion also applies to SQL functions used to build the source expression of data structure columns. For more information about data types, see Script/Data Type Category in the DBMS Resource File Reference chapter of the <i>Customizing and Extending PowerDesigner</i> manual. For more information about SQL functions, see "Keyword category" in the same chapter.

Transformation Process Property Sheet Data Transformation Tasks Tab

The *Data Transformation Tasks* tab displays a list of data transformation tasks for the transformation process, in which you can create, edit, and delete tasks.

For more information, see [Building Transformation Control Flow Diagrams](#) on page 57.

Transformation Process Property Sheet Transformation Control Flows Tab

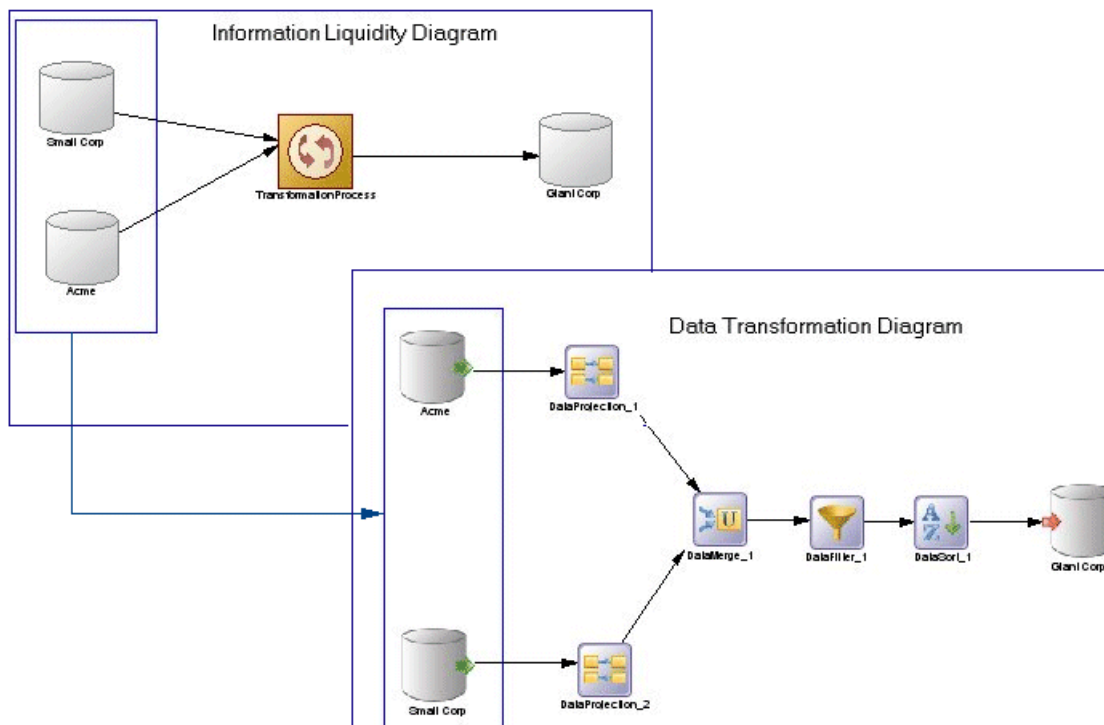
The *Transformation Control Flows* tab displays a list of transformation control flows for the transformation process, in which you can create, edit, and delete control flows.

For more information, see [Building Transformation Control Flow Diagrams](#) on page 57.









Data Inputs (ILM)

A *data input* represents a source of data in a data transformation diagram, and is linked to a database, an XML document, a web service or a flat file.

In the following example, the Small Corp and Acme databases in the Information Liquidity Diagram are represented by the Small Corp and Acme database inputs in the Data Transformation Diagram:



You can create the following data inputs in a data transformation diagram:

Object	Tool	Symbol	Description
Database input			Database from where data is extracted.
XML input			XML document from where data is extracted.
Web service input			Business process from where data is extracted.
Flat file input			Flat file from where data is extracted.

Creating a Data Input

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

- Drag a source data store (database, XML document, business process, or flat file) from the browser or from an information liquidity diagram, and drop it onto the data transformation diagram.
- Drag a PDM table or view, an XSM element, or a BPM operation from the browser, and drop it onto the data transformation diagram.
- Use the appropriate *Data Input* tool in the diagram Palette.
- Right-click a data transformation task in the Browser, and select **New > Data Input**.
- Open a transformation task property sheet, click the Inputs tab, and click the Add a Row tool.
- Select **Model > Data Inputs** to access the List of *Data Inputs*, and click the Add a Row tool.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Data Input Properties

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

The General tab contains the following properties:



Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Property	Description
Data connection	Specifies the data store represented by the input. You must select a data connection to access the list of available data stores. This field will be automatically completed if you drag the data store from the browser, and drop it onto the diagram.
Source object	[XML and Web service inputs only] Specifies the particular object from the source model to be used as input. You can use the tools to the right of the list to browse the available objects, or view the properties of the currently selected object.

Database Input Property Sheet SQL Query Tab

This tab is only available for database inputs. It allows you to edit the default SQL query to help you create your data structure columns.

The following tools are available on the *SQL Query* tab:

Tool	Description
	Retrieve Columns by Parsing Query — Parses the query you have specified in the textbox using the SQL Editor. The columns of the query are automatically created in the Data Structure Columns tab of the database input and their parent tables or views are displayed in the Data Structure Source Objects tab. You can also click this tool to update data structure columns and source tables when you have modified source expressions of data structure columns.
	Edit SQL Query — Opens the query in the SQL editor that helps you select PDM objects (tables, views, columns, procedures, and users) to build the SQL query script.

Data Structure Source Objects

This tab is only available for database inputs. It lists the source objects to which the database input is attached.

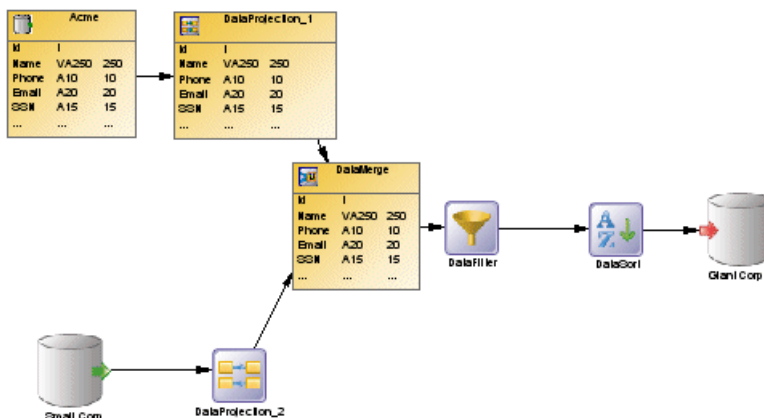
For more information, see [Data Structure Source Objects tab](#) on page 78.

Actions (ILM)




An *action* represents a transformation to execute on input flows in a data transformation diagram. Filtering, aggregating or duplicating data are examples of transformation you may need to perform in your activities.

Actions are linked to the previous step (data input or another action) using a data flow. Values of the input flow automatically appear in the Data Structure Columns tab of the action.

In the following example, the values of the Acme database input are propagated to the DataProjection_1 action, and are in turn propagated to the DataMerge action, and so on until they reach the GiantCorp database output:



You can create the following actions in a data transformation diagram:

Object	Tool	Symbol	Description
Script execution			Executes a script (see Inserting a script execution on page 65).
Data query execution			Executes an SQL Query in the database (see Inserting a data query execution on page 66).
Data aggregation			Reduces the number of rows from an input data flow in order to group the data (see Inserting a data aggregation on page 67).
Data sort			Sorts input rows from an input data flow (see Inserting a data sort on page 68).
Data filter			Filters rows from an input data flow (see Inserting a data filter on page 69).
Data split			Duplicates an input data flow into several output data flows (see Inserting a data split on page 70).
Data join			Joins data from several input data flows into one output data flow (see Inserting a data join on page 70).
Data lookup			Finds the corresponding value to a key column and store it in a new column of the output data flow (see Inserting a data lookup on page 71).
Data merge			Merges all the rows from several input data flows into one output data flow (see Inserting a data merge on page 70).
Data projection			Defines basic data transformations, such as removing columns or changing the order of columns (see Inserting a data projection on page 73).
Data calculator			Defines complex data transformations, such as filtering or aggregating data (see Inserting a data calculator on page 73).

Creating an Action

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

- Use the *Action* tool in the diagram Palette.
- Select **Model** > **Actions** to access the List of *Actions*, and click the Add a Row tool.
- Right-click a data transformation task in the Browser, and select **New** > **Action**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Action Properties

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

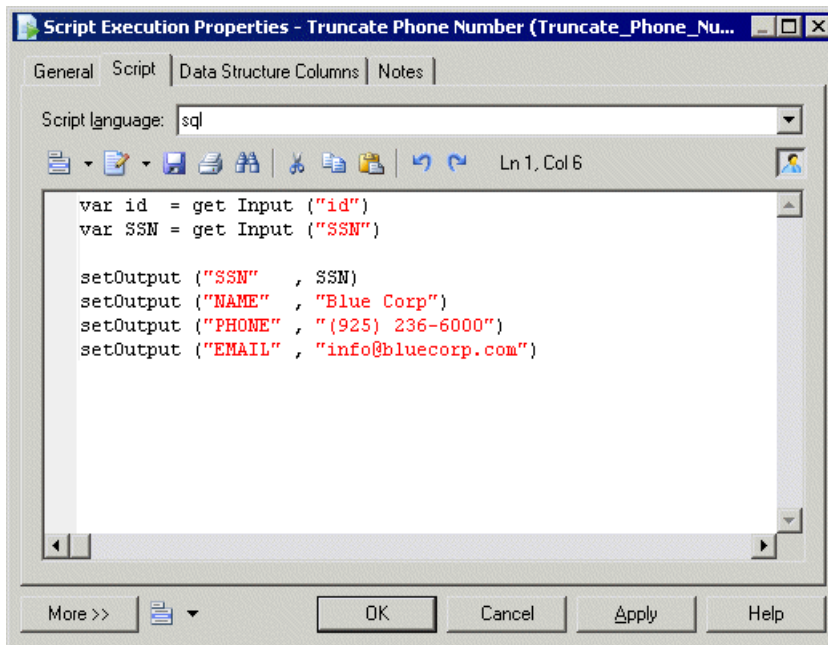
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Mode	[Data lookup only] Specifies the mean by which values are mapped. You can choose between the following options: <ul style="list-style-type: none"> • Database – [Default] The mapping is performed against a database table. This option triggers the display of the Script tab. • Predefined – The mapping is performed against a list of key value pairs. This option triggers the display of the Lookup Keys tab.
Source column	[Data lookup only] Specifies the source column key to replace.
Target column	[Data lookup only] Specifies the target column, which contains the resulting value.

Inserting a Script Execution

A *script execution* executes a script for each row of the input flow. For example, it can be used to create a log file, a mail or a text file related to the input flow.

1. Select the Script Execution tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the script execution to initialize it with the incoming data structure columns.
3. Double-click the script execution symbol to open its property sheet, click the Script tab, select or enter a script language, and enter a script in the textbox.

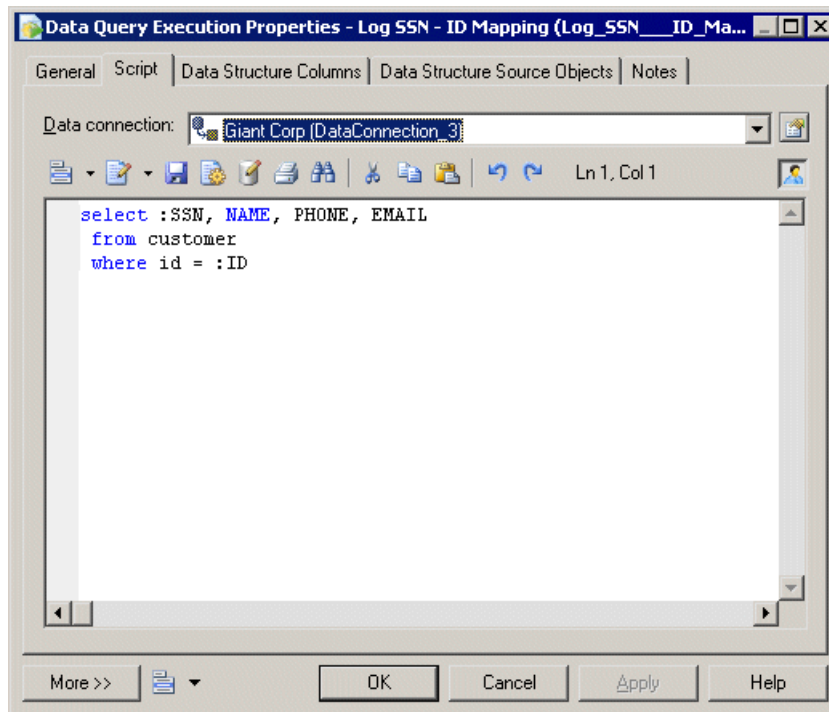


4. [optional] Click the Data Structure Columns tab, and add, edit, reorder or delete columns as appropriate.
5. Click OK to save your changes and return to the diagram.

Inserting a Data Query Execution

A *data query execution* executes an SQL Query against a database for each row of the input flow to transform it, and create a new data flow. Data from the input flow can be used as parameter.

1. Select the Data Query Execution tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data query execution.
3. Double-click the data query execution symbol to open its property sheet, click the Script tab, and select a data connection to access the database.
4. Enter an SQL query script in the textbox or click the Edit SQL Query tool to select PDM objects in the SQL Editor, and build the script.
5. Click the Retrieve Columns by Parsing Query tool that lets you parse the query you have specified in the textbox using the SQL Editor. The columns of the query are automatically created in the Data Structure Columns tab, and their parent tables or views are displayed in the Data Structure Source Objects tab.

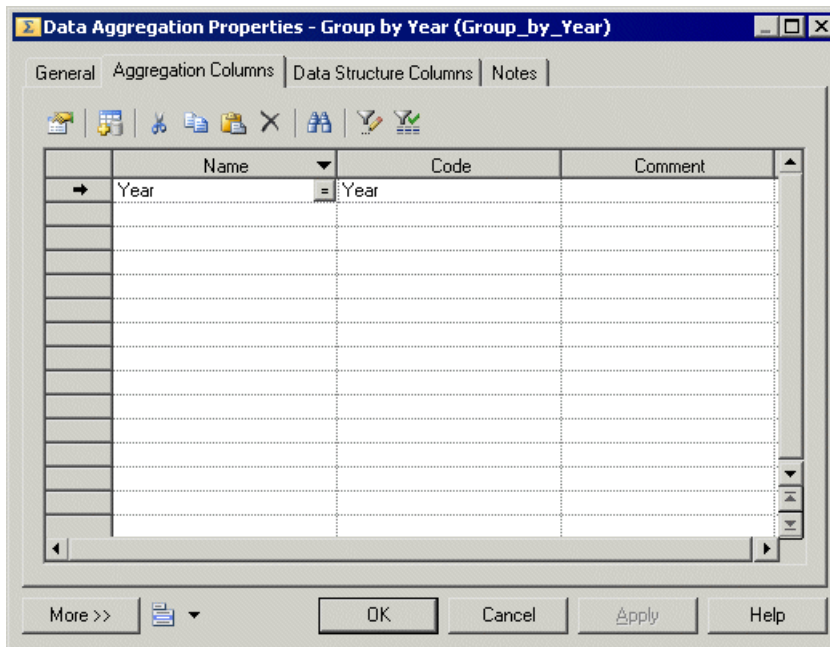


6. [optional] Click the Data Structure Columns tab, and add, edit, reorder or delete columns as appropriate.
7. Click OK to save your changes and return to the diagram.

Inserting a Data Aggregation

A *data aggregation* aggregates incoming data using functions such as Avg, Min, Max, Count, Sum etc.

1. Select the Data Aggregation tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data aggregation to initialize it with the incoming data structure columns.
3. Double-click the data aggregation symbol to open its property sheet, click the Aggregation Columns tab, and click the Select Aggregation Columns tool to open a selection dialog box, which allows you to select one or more columns to aggregate. Make your selection, click OK to add the columns and return to the tab, then click Apply.



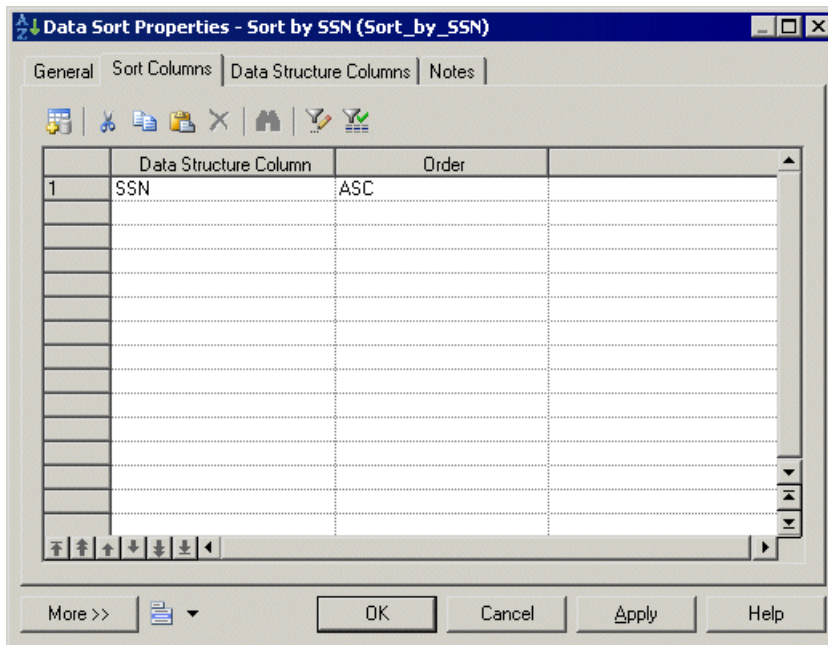
4. Click the Data Structure Column tab and for each column, enter an aggregation function in the Source Expression column. Delete columns that will not be aggregated.
5. Click OK to save your changes and return to the diagram.

Note: You can right-click a data aggregation symbol, and select Aggregated Columns to access the Aggregation Columns tab directly.

Inserting a Data Sort

A *data sort* sorts incoming rows by one or more data structure columns.

1. Select the Data Sort tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data sort to initialize it with the incoming data structure columns.
3. Double-click the data sort symbol to open its property sheet, click the Sort Columns tab and click the Select Sort Columns tool to open a selection dialog box, which allows you to select one or more columns to sort by. Make your selection, click OK to add the columns and return to the tab.
4. For each of the sort column, click in the Order column, and specify whether it should be sorted in ascending (default) or descending order.



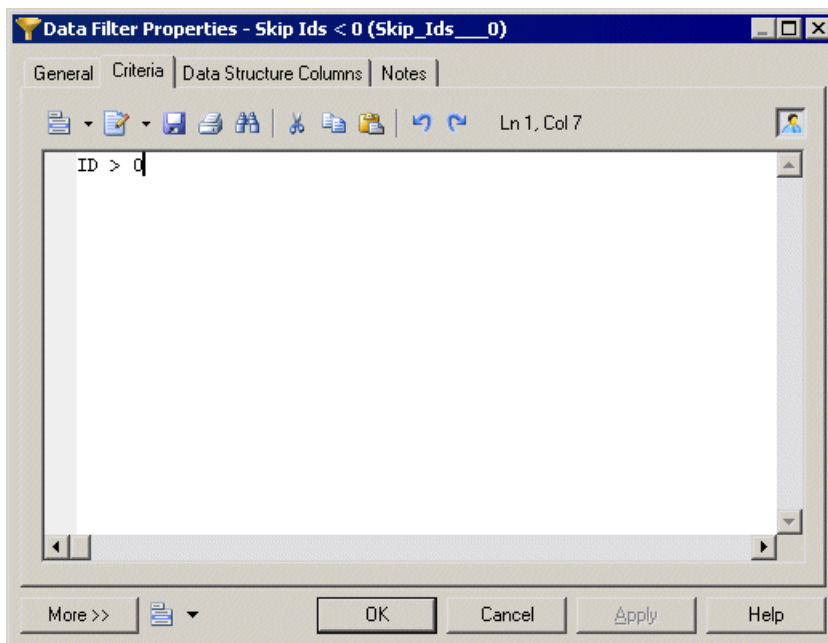
- 5. Click OK to save your changes and return to the diagram.**

Note: You can right-click a data sort symbol, and select Sorted Columns to access the Sort Columns tab directly.

Inserting a Data Filter

A *data filter* filters incoming rows using SQL criteria.

1. Select the Data Filter tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data filter to initialize it with the incoming data structure columns.
3. Double-click the data filter symbol to open its property sheet, click the Criteria tab, and enter a criterion expression to filter by.



4. Click OK to save your changes and return to the diagram.

Note: You can right-click a data filter symbol, and select Criteria to access the Criteria tab directly.

Inserting a Data Split

A *data split* duplicates a simple input data flow into two or more identical output data flows.

1. Select the Data Split tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data split to initialize it with the incoming data structure columns.
3. Click OK to save your changes and return to the diagram.

Note: When a data input or an action has more than one output flow, you can right-click the data input or action, and select Insert Split. This automatically creates a data split after the data input or action. Conversely, you can select Remove Split to display each output flow instead of the data split.

Inserting a Data Merge

A *data merge* combines two or more identical input flows into a single output flow.

To combine different input flows, see [Inserting a data join](#) on page 70.

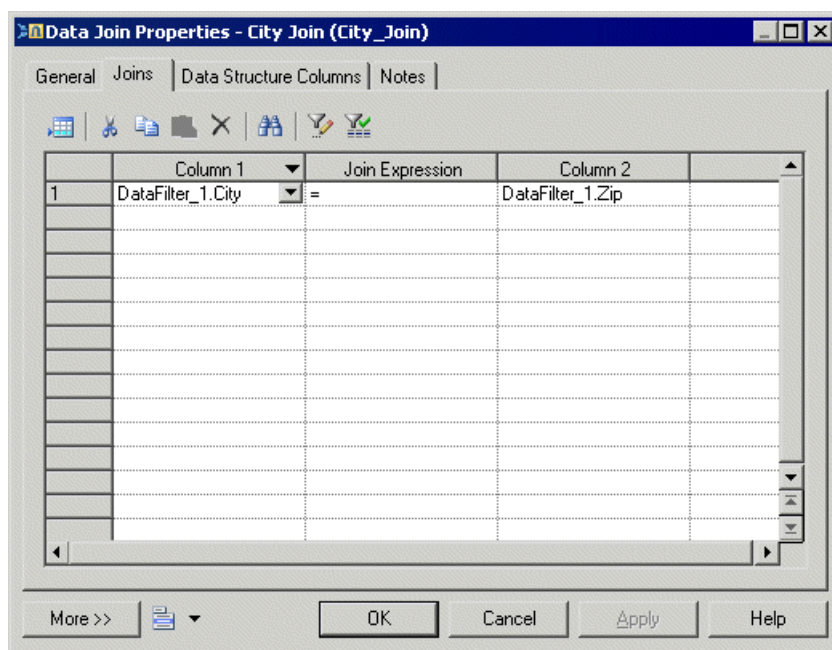
1. Select the Data Merge tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data merge to initialize it with the incoming data structure columns.
3. Click OK to save your changes and return to the diagram.

Note: When a data output or an action has more than two input flows, you can right-click the data output or action and select Insert Merge. It automatically creates a data merge before the data output or action. Conversely, you can select Remove Merge to display each input flow instead of the data merge.

Inserting a Data Join

A *data join* performs a join on two or more input flows, and combines them.

1. Select the Data Join tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data join to initialize it with the incoming data structure columns.
3. Double-click the data join symbol to open its property sheet, click the Join Columns tab and click the Add a Row tool to create a join.
4. Click Column 1 and select a column to join on. Click Column 2, and select a second column to join on.
5. Click the Join Expression column to select a join expression, and click Apply.



6. [optional] Click the Data Structure Columns tab and delete any unnecessary columns. For example, you may want to delete one of the two columns you selected for the join.
7. Click OK to save your changes and return to the diagram.

Note: You can right-click a data join symbol, and select Joins to access the Joins tab directly.

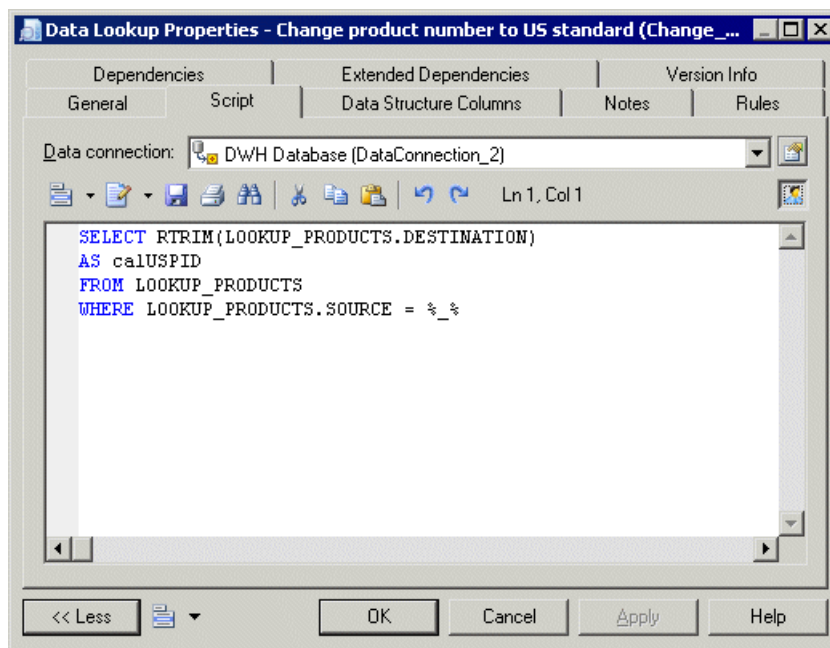
Inserting a Data Lookup

A *data lookup* lets you find the corresponding value to a key column, and replace it by creating a new column for the output data flow.

To Insert a Data Lookup in Database Mode:

The mapping between the key column and its value can come from database tables.

1. Select the Data Lookup tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data lookup to initialize it with the incoming data structure columns.
3. Double-click the data lookup symbol to open its property sheet, and select the Database mode in the Mode list. This triggers the display of the Script tab.
4. Select the source column from which you want to draw the values to be replaced. Create the target column, which will contain the values returned by the lookup. The target column will automatically replace the source column in the Data Structure Columns tab.
5. Click the Script tab, select a data connection, and specify a SQL query in the textbox. The query will be executed against the database tables and will return two columns (a key column to search for a corresponding value and a value column to store the corresponding value).

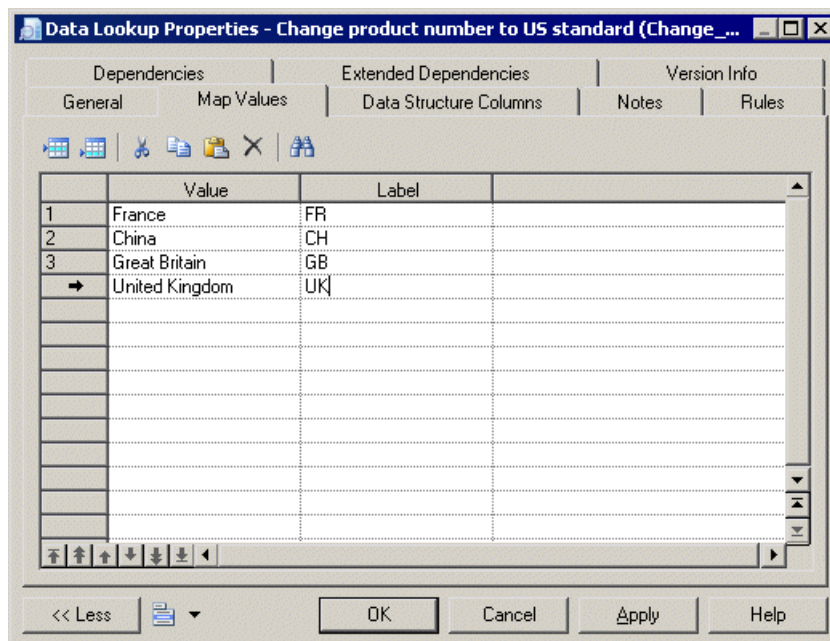


- 6.** Click OK to save your changes and return to the diagram.

To Insert a Data Lookup in Predefined Mode:

The mapping between the key column and its value can come from a predefined list of key value pairs.

1. Select the Data Lookup tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data lookup to initialize it with the incoming data structure columns.
3. Double-click the data lookup symbol to open its property sheet, and select the Predefined mode in the Mode list. This triggers the display of the Lookup Keys tab.
4. Select the source column from which you want to draw the values to be replaced. Create the target column, which will contain the values returned by the lookup. The target column will automatically replace the source column in the Data Structure Columns tab.
5. Click the Lookup Keys tab, and enter key value pairs in the list as appropriate.



6. Click OK to save your changes and return to the diagram.

Inserting a Data Projection

A *data projection* performs basic data transformations, such as removing columns or changing the order of columns.

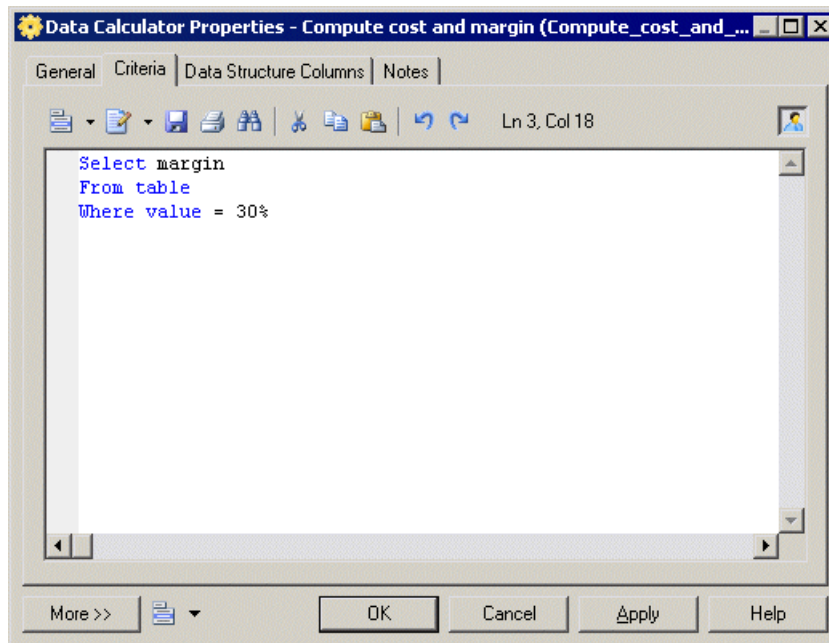
For more advanced transformations, see [Inserting a data calculator](#) on page 73.

1. Select the Data Projection tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data projection to initialize it with the incoming data structure columns.
3. Double-click the data projection symbol to open its property sheet, and click the Data Structure Columns tab, which is automatically completed with the values from the input flow. Reorder or delete columns as appropriate.
4. Click OK to save your changes and return to the diagram.

Inserting a Data Calculator

A *data calculator* allows you to perform any kind of data transformations, by specifying an SQL query.

1. Select the Data Calculator tool in the palette, and create the action in the diagram.
2. Select the Data Flow tool, and draw a flow from the preceding step (a data input or action) to the data calculator to initialize it with the incoming data structure columns.
3. Double-click the data calculator symbol to open its property sheet, click the Criteria tab, and enter the appropriate SQL script to perform the desired data transformation.



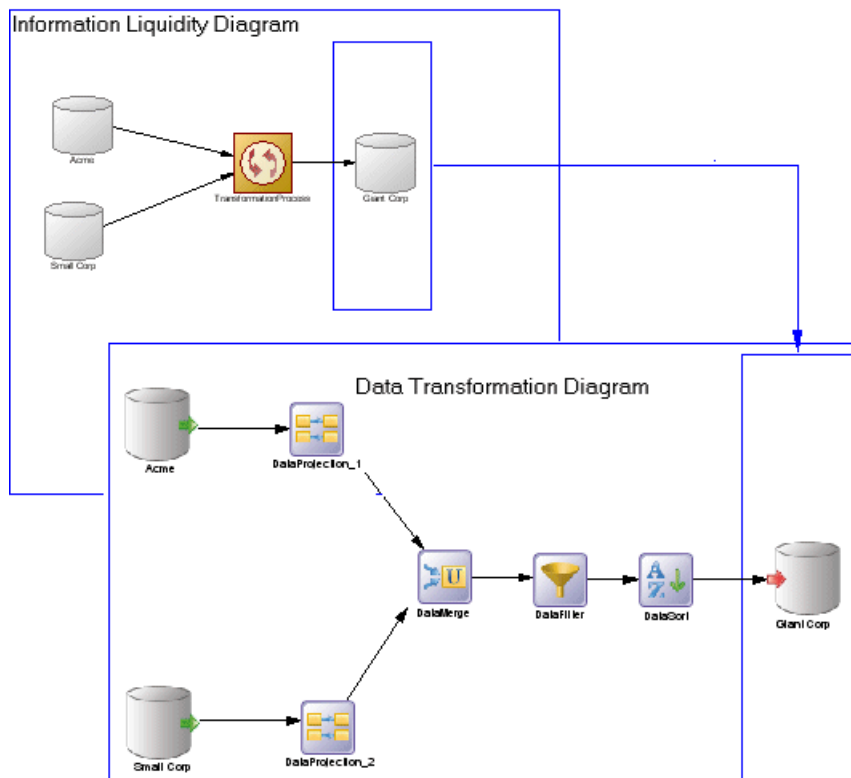
4. [optional] Click the Data Structure Columns tab, and add, edit, reorder or delete columns as appropriate.
5. Click OK to save your changes and return to the diagram.

Note: You can right-click a data calculator symbol, and select Criteria to access the Criteria tab directly.







Data Outputs (ILM)

A *data output* represents a target destination to load data in a data transformation diagram, and is linked to a database, an XML document, or a flat file.

In the following example, the Giant Corp database in the Information Liquidity Diagram is represented by the Giant Corp database output in the Data Transformation Diagram:



You can create the following data outputs in a data transformation diagram:

Object	Tool	Symbol	Description
Database output			Database to where data is loaded.
XML output			XML document to where data is loaded.
Flat file output			Flat file to where data is loaded.

Creating a Data Output

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

- Drag a target data store (database, XML document, or flat file) from the browser or from an information liquidity diagram, and drop it onto the data transformation diagram.
- Drag a PDM table or view, or an XSM element from a model attached to a target data store in the browser, and drop it onto the data transformation diagram.
- Use the appropriate *Data Output* tool in the diagram Palette.
- Right-click a data transformation task in the Browser, and select **New > Data Output**.
- Open a transformation task property sheet, click the Outputs tab, and click the Add a Row tool.

- Select **Model > Data Outputs** to access the List of *Data Outputs*, and click the Add a Row tool.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Data Output Properties

You can modify an object's properties from its property sheet. To open a data output property sheet, double-click its diagram symbol or its Browser entry in the *Data Outputs* folder. The following sections detail the property sheet tabs that contain the properties most commonly entered for data outputs.

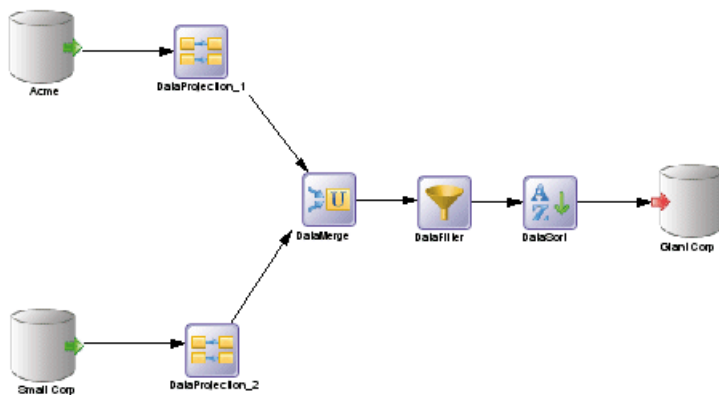
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Data connection	Specifies the data store represented by the output. You must select a data connection to access the list of available data stores. This field will be automatically completed if you drag the data store from the browser, and drop it onto the diagram.
Target object	[Database and XML document outputs only] Specifies the particular object from the target model to be used as output. You can use the tools to the right of the list to browse the complete tree of available objects or view the properties of the currently selected object.
Mode	<p>[Database output only] Specifies the type of action the database output performs on the target object by analyzing its input flows. You can choose one of the following options:</p> <ul style="list-style-type: none"> • Clear – Deletes rows in the target object. • Delete & insert – Deletes rows in the target object, and inserts a new row. • Insert – Inserts a row in a target object. • Insert or update – Looks for a row in a target object, and inserts it if it does not exist or updates it. • Update – Looks for an existing row in a target object, and updates it

Data Flows (ILM)

A *data flow* conveys data between steps in a data transformation diagram.

In the following example, data flows convey data from the Acme database input through several actions, and to the Giant Corp database output:



When you link two steps with a data flow, the data structure of the destination object is initialized with the data structure of the source object.

Any name, code or data type changes you perform on the data structure columns of a source object are automatically applied to the data structure columns of the target object, when they match.

Creating a Data Flow

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

- Use the Data Flow tool in the diagram Palette.
- Select **Model > Data Flows** to access the List of Data Flows, and click the Add a Row tool.

Data flows can be created in a list of data flows only if you have previously defined the required source and destination objects.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Data Flow Properties

You can modify an object's properties from its property sheet. To open a data flow property sheet, double-click its diagram symbol or its Browser entry in the Data Flows folder.

The General tab contains the following properties :

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Source	Specifies the object from which the data flow originates. 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.
Destination	Specifies the object to which the data flow leads. 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.

Data Structure Columns (ILM)

A *data structure column* represents a database table column, a flat file column, an XML element or attribute, or an output parameter of a web service operation at a particular point in the transformation.

For example, you may have a column called Name in your source database, which is extracted and processed by a number of transformation actions before being loaded into your target database. Each of these steps in the transformation task will contain a data structure column, which represents the column at that point in the transformation. The column may be renamed filtered, reordered, and/or have its data type, length, default value etc, changed, and you can trace each of these changes by referring to the data structure column at the relevant point in the transformation.

You can use the Data Structure Mapping Editor to show how source and target objects data structure columns are mapped. For more information, see [Data Structure Mapping Editor](#) on page 56.

Creating a Data Structure Column

You create a data structure column from the Data Structure Columns tab of a data input, an action or a data output.

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

- Click the Add Columns tool and select columns from a column selection of the input flow.
- Click the Add a Row tool to create a new data structure column.

Note that for actions that do not modify the structure of the data format, such as the data lookup object or the data split object for example, the Add a Row tool is not available.

Data Structure Column Properties

You can modify an object's properties from its property sheet. To open a data structure column property sheet, double-click its row in the Data structure Columns tab of a data input, an action or a data output property sheet or its Browser entry in the Data Structure Column folder under its parent object.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Data type	Specifies the type of the column, such as numeric, alphanumeric, boolean, etc. The types available are either those of the transformation engine or, if you have not specified an engine, those of PowerDesigner. If you change the type of the transformation process, the data type used by the data structure column will be converted to its equivalent in the new transformation engine. For more information about data types, see Script/Data Type Category in the DBMS Resource File Reference chapter of the <i>Customizing and Extending PowerDesigner</i> manual.
Length	Specifies the maximum length of the data type.
Precision	Specifies the maximum number of places after the decimal point.

Property	Description
Mandatory	Specifies a column that must be assigned a not null value.
Default value	Specifies a default value for the data structure column.
Identifier	Specifies the data structure column as an identifier. This is useful when you update the target tables used to create a join.
Target object	[data structure column owned by a data output only] Specifies the target object in which the data structure column is loaded. You can use the tools to the right of the list to browse the complete tree of available objects or view the properties of the currently selected object.

Note: If you change the type of the transformation process, the SQL functions used to build the source expression of a data structure column will be converted to its equivalent in the new transformation engine. For more information about SQL functions, see Keyword category in the DBMS Resource File Reference chapter of the *Customizing and Extending PowerDesigner* manual.

Data Structure Source Objects Tab

This tab is available in the property sheet of data structure columns, inputs and data query executions.

The *Data Structure Source Objects* tab lists the source objects to which the object is attached. In general, this tab is automatically propagated and read-only, but you can add or delete source objects for data structure columns belonging to the following steps:

- Inputs
- Script executions
- Data query executions
- Data calculators
- Data aggregations
- Data projections
- Outputs

In this tab, you can also click the Edit Source Expression tool, in the lower part of the dialog box, to open the Source Expression Editor, and specify a more complex source expression using functions.

You can also add or delete source objects for the following steps:

- Inputs (see [Data Inputs \(ILM\)](#) on page 61) – source objects correspond to objects of source data stores linked to a database, an XML document, a web service or a flat file.
- Data query executions (see [Inserting a data query execution](#) on page 66) – source objects originate from the previous steps to which data query executions are linked.

Transformation Parameters (ILM)

A *parameter* is an input or output variable global to a transformation task that you can use to customize your data transformations. A parameter is available to all diagrams within a given task. For example, if you manipulate sales figures, you might require a parameter specifying the sales region you are interested in.

Parameters are used in the source expression of data structure columns (see [Data Structure Columns \(ILM\)](#) on page 77).

Creating a Transformation Parameter

You can create a transformation parameter in any of the following ways:

- Open a transformation task property sheet, click the Parameter tab, and click the Add a Row tool.
- Right-click a data transformation task in the Browser, and select **New > Transformation Parameter**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Parameter Properties

You can modify an object's properties from its property sheet. To open a transformation parameter property sheet, double-click a transformation parameter on the Parameters tab of a data transformation task.

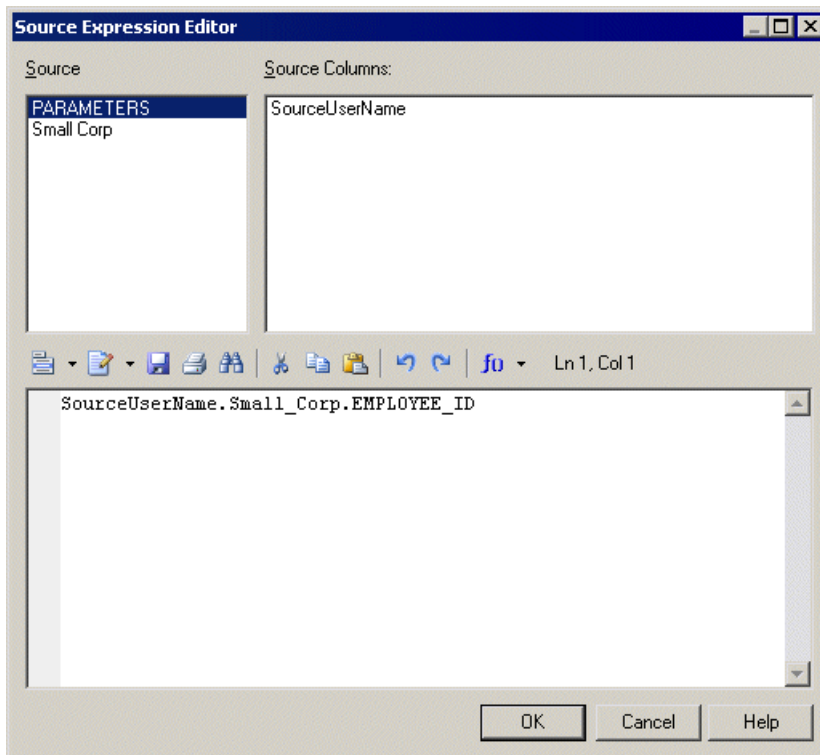
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Data type	Specifies the type of the parameter. If you change the type of the transformation process, the data type used by the parameter will be converted to its equivalent in the new transformation engine. For more information about data types, see Script/Data Type Category in the DBMS Resource File Reference chapter of the <i>Customizing and Extending PowerDesigner</i> manual.
Type	Specifies the type of the parameter. You can choose one of the following values: <ul style="list-style-type: none"> • In – specifies an input parameter. • Out – specifies an output parameter.
Default value	Specifies a default value for the parameter.

Example: Assigning a Parameter to a Data Structure Column

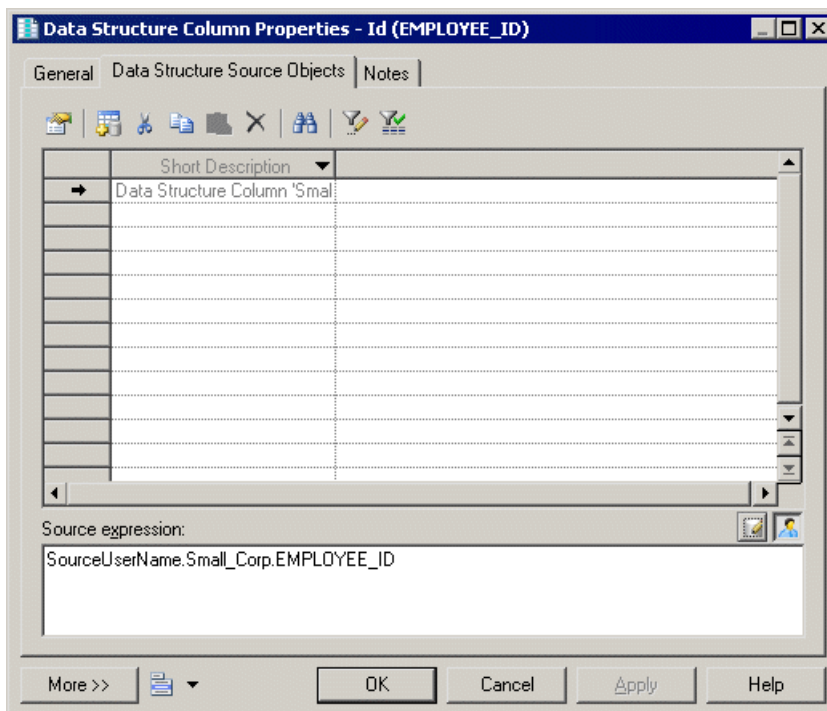
You can assign a parameter to a data structure column whose source expression can be modified.

1. Create a parameter and call it SourceUserName.
2. Open a step's property sheet whose data structure columns can be modified.
3. Click the Data Structure Columns tab, and double-click a data structure column to open its property sheet.
4. Click the Data Structure Source Objects tab, and in the source expression box, click the Edit Source Expression tool to open the corresponding editor.
5. Click Parameters in the Source field to display the available parameters in the Source Columns field, position the cursor in the script textbox where you want to add the parameter, and then double-click the SourceUserName parameter to add it to the query script.
6. Complete the script as appropriate.



7. Click OK to close the dialog box.

The source expression of the data structure column is updated.

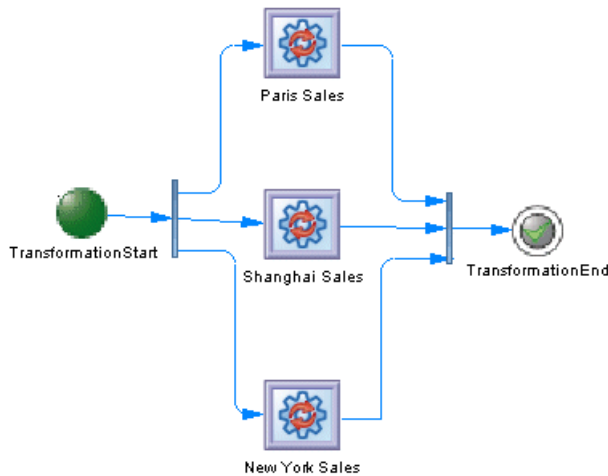


8. Click OK to close the dialog box.

Transformation Starts (ILM)

A *transformation start* initiates the sequence of execution of a series of data transformation tasks in a transformation control flow diagram.

In the following example, TransformationStart initiates the sequence of the Paris Sales, Shanghai Sales and New York Sales tasks:



Creating a Transformation Start

You can create a transformation start in any of the following ways:

- Use the Transformation Start tool in the diagram Palette.
- Select **Model > Starts** to access the List of Transformation Starts, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Transformation Start**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Start Properties

You can modify an object's properties from its property sheet. To open a transformation start property sheet, double-click its diagram symbol or its Browser entry in the Transformation Starts folder.

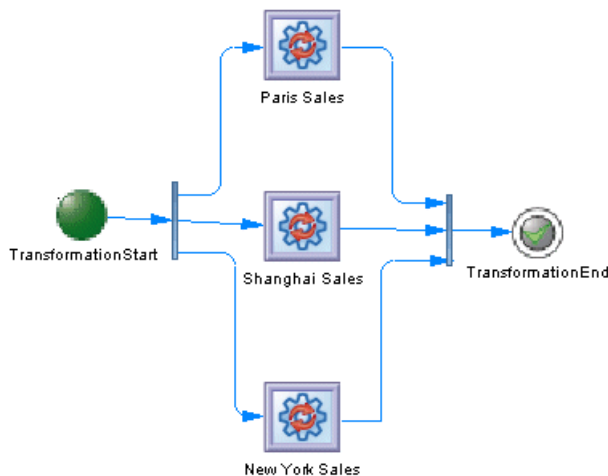
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Transformation Task Executions (ILM)

A *transformation task execution* is an instance of one or more data transformation tasks in a transformation control flow diagram. Tasks can be executed serially or in parallel.

In the following example, the Paris Sales, Shanghai Sales and New York Sales tasks are executed in parallel:



Creating a Transformation Task Execution

You can create a transformation task execution in any of the following ways:

- Drag a data transformation task from the browser and drop it onto a transformation control flow diagram.
- Use the Transformation Task Execution tool in the diagram Palette.
- Select **Model > Transformation Task Executions** to access the List of Transformation Task Executions, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Transformation Task Execution**.

Note: You can also create and automatically link multiple task executions (see [Creating multiple transformation task executions](#) on page 83).

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Task Execution Properties

You can modify an object's properties from its property sheet. To open a transformation task execution property sheet, double-click its diagram symbol or its Browser entry in the Transformation Task Executions folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Transformation Task Execution Property Sheet Tasks Tab

The *Tasks tab* displays a list of data transformation tasks, and allows you to add, edit or delete tasks..

Each transformation task execution must be associated with a data transformation task (see [Building Data Transformation Diagrams](#) on page 54)

Creating Multiple Transformation Task Executions

You can create multiple transformation task executions in your diagram and manage the mode (parallel or serial) and order of execution.

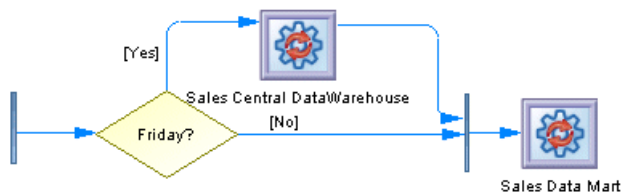
1. Click the diagram background, and select Create Task Executions (or Right-click a transformation control flow in the Browser, and select Create Task Executions) to open the Create Task Executions dialog box.
2. Select one or more data transformation tasks to add to your diagram.
3. [optional] Reorder the selected task executions according to your needs, using the arrows at the bottom of the dialog box.
4. Specify the execution mode. You can choose between:
 - Parallel
 - Serial

A task execution is created for each selected data transformation task. You can open the diagram of the data transformation task associated with the task execution, by pressing ctrl and double-clicking the symbol. If several tasks are associated, a selection dialog box opens to let you select the diagram you want.

Transformation Synchronizations (ILM)

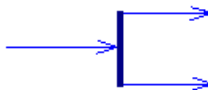
A *transformation synchronization* enables the synchronization of control flows between two or more concurrent actions.

In the following example, the output flows of the Friday decision and of the Sales Central DataWarehouse task are synchronized into one output flow, which goes to the Sales Data Mart task execution:



A transformation synchronization can be either a:

- Fork – Splits a single input flow into several independent output flows executed in parallel:



- Join – Merges multiple input flows into a single output flow. All input flows must reach the join before the single output flow continues:



Creating a Transformation Synchronization

You can create a transformation synchronization in any of the following ways:

- Use the Transformation Synchronization tool in the diagram Palette.

- Select **Model > Transformation Synchronizations** to access the List of Transformation Synchronizations, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Transformation Synchronization**.

By default, the transformation synchronization symbol is created horizontally. To toggle between horizontal and vertical display, right-click the symbol and select Change to Vertical or Change to Horizontal in the contextual menu.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Synchronization Properties

You can modify an object's properties from its property sheet. To open a transformation synchronization property sheet, double-click its diagram symbol or its Browser entry in the Transformation Synchronizations folder.

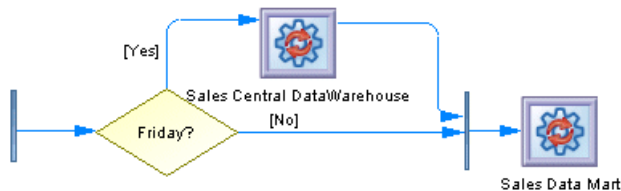
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.

Transformation Decisions (ILM)

A *transformation decision* lets you choose between different paths by evaluating guard conditions, which must be satisfied for an associated flow to execute some action.

In the following example, the Sales Central DataWarehouse task will only be executed on Friday:



A transformation decision allows you to create complex flows, such as:

- if ... then ... else ...
- switch ... case ...
- do ... while ...
- loop
- for ... next ...

Note: It is not possible to attach two flows of opposite directions to the same corner on a transformation decision symbol.

Creating a Transformation Decision

You can create a transformation decision in any of the following ways:

- Use the Transformation Decision tool in the diagram Palette.

- Select **Model > Transformation Decisions** to access the List of Transformation Decisions, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Transformation Decision**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation Decision Properties

You can modify an object's properties from its property sheet. To open a transformation decision property sheet, double-click its diagram symbol or its Browser entry in the Transformation Decisions folder.

The General tab contains the following properties

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Type	Dynamically specifies the type of the transformation decision: conditional branch, merge, or incomplete.

Transformation Decision Property Sheet Condition Tab

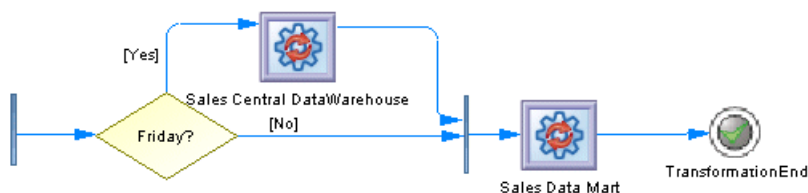
The Condition tab contains the following properties:

Properties	Description
Alias	Specifies a short name for the condition, to be displayed in the transformation decision symbol.
Condition (text box)	Specifies a condition to be evaluated to determine how the transformation decision should be traversed. You can enter any appropriate information in this box, as well as open, insert and save text files.

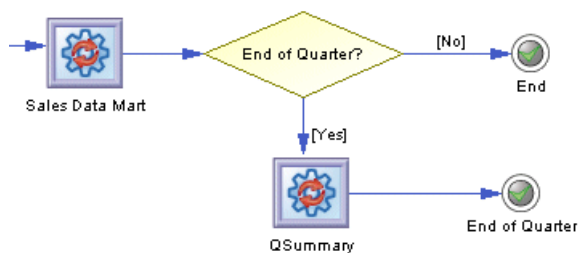
Transformation Ends (ILM)

A *transformation end* terminates the sequence of execution of a series of tasks in a transformation control flow diagram, and specifies the result for the execution, which can be either Success or Error.

In the following example, TransformationEnd terminates the sequence of execution of Sales Central DataWarehouse and Sales Data Mart:



You can create several ends within the same diagram if you want to show different end cases. In the following example, two different ends are possible depending on whether or not it is the end of the quarter.



Creating a Transformation End

You can create a transformation end in any of the following ways:

- Use the Transformation End tool in the diagram Palette.
- Select **Model > Transformation Ends** to access the List of Transformation Ends, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Transformation End**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Transformation End Properties

You can modify an object's properties from its property sheet. To open a transformation end property sheet, double-click its diagram symbol or its Browser entry in the Transformation Ends folder.

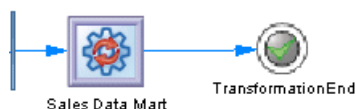
The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Type	Specifies whether the control flow execution has succeeded (Success) or has failed (Error).

Control Flows (ILM)

A *control flow* connects transformation starts, task executions, decisions, synchronizations and ends.

In the following example, a synchronization is connected to the Sales Data Mart task execution, which is in turn connected to TransformationEnd:



Creating a Control Flow

You can create a control flow in any of the following ways:

- Use the Control Flow tool in the diagram Palette.
- Select **Model > Control Flows** to access the List of Control Flows, and click the Add a Row tool.
- Right-click a transformation control flow in the Browser, and select **New > Control Flow**.

For general information about creating objects, see the Objects chapter in the *Core Features Guide*.

Control Flow Properties

You can modify an object's properties from its property sheet. To open a control flow property sheet, double-click its diagram symbol or its Browser entry in the Control Flows folder.

The General tab contains the following properties:

Property	Description
Name	Specifies the name of the item, which should be clear and meaningful, and should convey the item's purpose to non-technical users.
Code	Specifies the technical name of the item, which is used for generating code or scripts.
Comment	Specifies a descriptive comment for the object.
Stereotype	Extends the semantics of an object derived from existing objects but specific to your needs. You can type stereotypes directly in this field, or add stereotypes to the list by specifying them in your model's resource file or in an extended model definition.
Source	Specifies the object from which the control flow originates. 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.
Target	Specifies the object to which the control flow leads. 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.

Working with Information Liquidity Models

Information liquidity models benefit from the powerful model tools available in the PowerDesigner enterprise modeling environment.

Customizing the ILM Environment

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

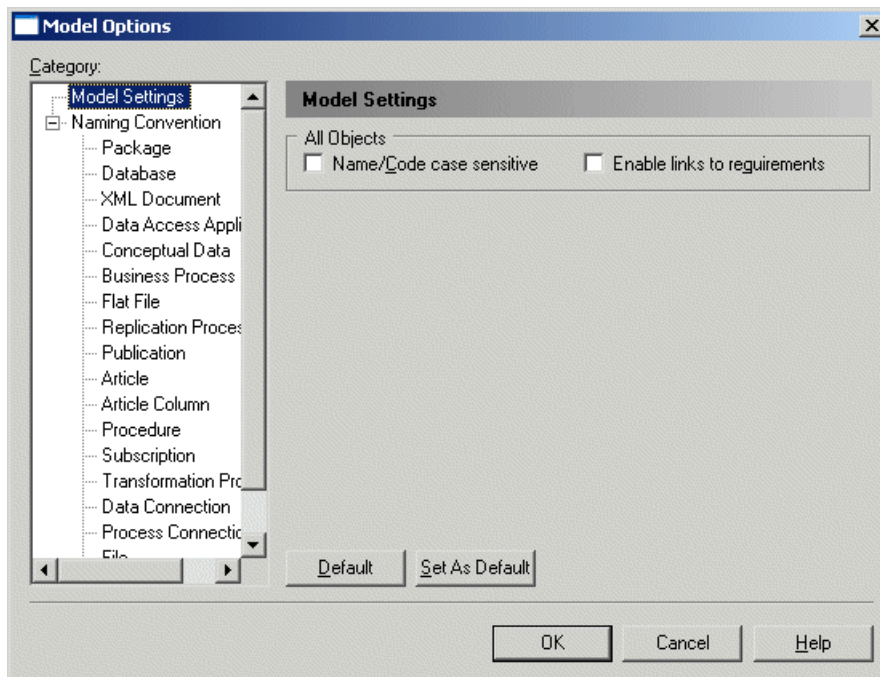
- At model creation
- After creating a model with default options and parameters
- When creating a model template

Setting ILM Model Options

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

For information about controlling the naming conventions of your models, see "Naming Conventions" section in the Models chapter of the *Core Features Guide*.

To set ILM model options, select **Tools > Model Options** or right-click the diagram background and select Model Options from the contextual menu.



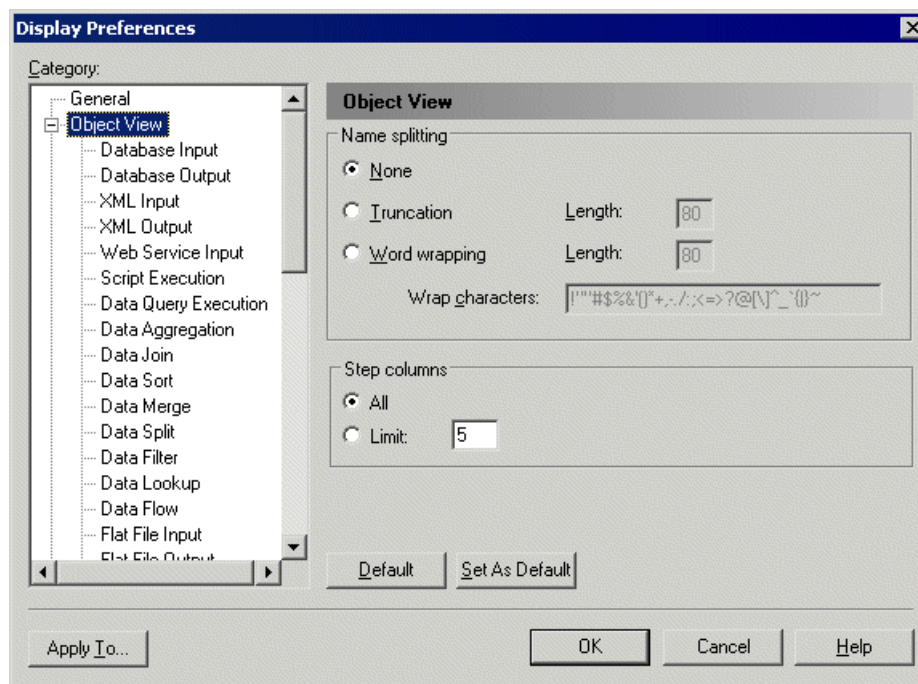
You can set the following options on this page. They apply to all objects:

Option	Description
Name/Code case sensitive	<p>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.</p> <p>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.</p>
Enable links to requirements	<p>Requirements are descriptions of customer needs that must be satisfied during development processes.</p> <p>You can enable links to requirements for all objects in the current model. When this check box is selected, it implies that the <i>Requirements</i> 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.</p> <p>For more information on requirements, see the <i>Requirements Modeling</i> guide.</p>

Setting ILM Display Preferences

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

To set ILM display preferences, select **Tools > Display Preferences** or right-click the diagram background, and select Display Preferences from the contextual menu.



Note: For information about changing the format of symbols, see "Format display preferences" in the Customizing your Modeling Environment chapter of the *Core Features Guide*.

The Object View category allows you to control what information is displayed on the symbol. By clicking on the category itself, you can set the following ILM preference. For other general preferences in this category, see "Content display preferences" in the Customizing your Modeling Environment chapter of the *Core Features Guide*.

Preference	Description
Step columns	Specifies the number of columns to display in a transformation step. You can choose between: <ul style="list-style-type: none"> • All – displays all columns. • Limit – displays columns up to the limit specified in the adjacent box.

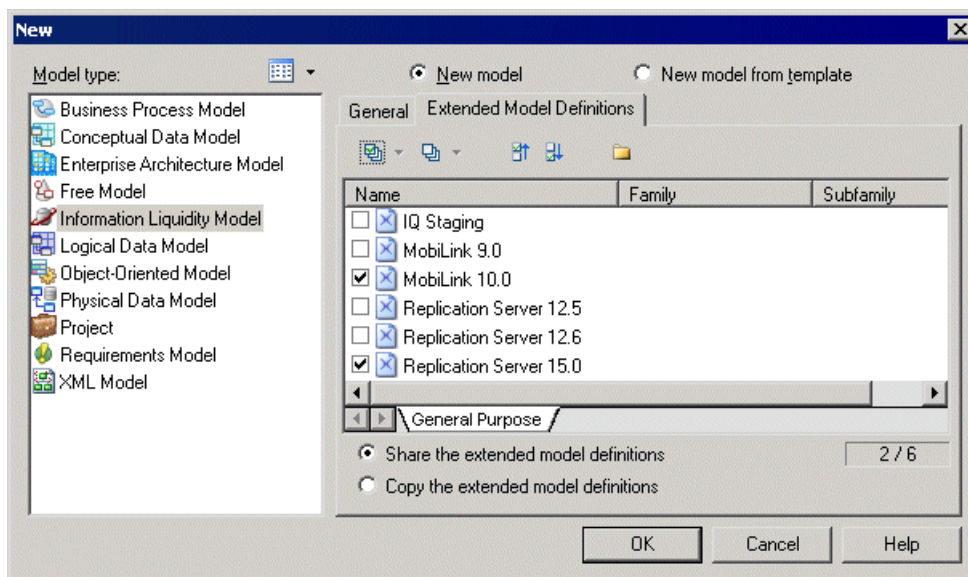
To set display preferences for a particular object, select it under the Object View category. The following table provides an alphabetical list of the display preferences available. Note that each object does not necessarily support all these preferences, and that the objects available to be customized in the Display Preferences window depend upon the current diagram type.

Preference	Description
Comment	Displays the comment of the object.
Stereotype	Displays the stereotype of the object.
Name	Displays the name of the object.
Criteria	Displays the criteria expression.
Type	Displays the type of the object.
Aggregated Columns	Displays the list of columns used to aggregate data.
Sorted Columns	Displays the list of columns used to sort data.
Joins	Displays the list of columns used to join data.

Working with ILM Extended Model Definitions

Extended model definitions (.XEM files) provide means for customizing and extending PowerDesigner metaclasses, parameters and generation. Extended model definitions are typed like models in PowerDesigner. You create an extended model definition for a specific type of model and you cannot share these files between heterogeneous models.

When you create a new ILM, you can select one or several extended model definitions and attach them to the model from the Extended Model Definitions page of the New dialog box:



When you attach an extended model definition to an ILM, you can choose between one of the following options:

Option	Definition
Share the extended model definitions	Current extended model definition constantly refers to the extended model definition stored in the Resource Files\Extended Model Definitions directory. Any changes made to the extended model definition are shared by all linked XEM.
Copy the extended model definitions	Current extended model definition is a unique copy of the extended model definition stored in the Resource Files\Extended Model Definitions directory. The current extended model definition is independent of the original one, so modifications made to the extended model definition in the Resource Files\Extended Model Definitions directory are not available to the copied XEM. This one is saved with the ILM and cannot be used without it

For more information on extended model definitions, see "Extended Model Definitions" in the Resource Files and the Public Metamodel chapter of the *Customizing and Extending PowerDesigner* manual.

Working with ILM Extended Dependencies

Extended dependencies are links between ILM 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 the semantics of extended dependencies between objects in the ILM.

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 "Using extended dependencies", in the Objects chapter of the *Core Features Guide*.

Generating Other Models from an ILM

You can generate another ILM from your ILM. 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.

1. Select **Tools > Generate Information Liquidity Model** to open the Information Liquidity Model Generation Options Window:
2. On the General tab, select a radio button to generate a new or update an existing model, and complete the appropriate options.
3. [optional] Click the Detail tab and set any appropriate options. We recommend that you select the Check model checkbox to check the model for errors and warnings before generation.
4. [optional] Click the Target Models tab and specify the target models for any generated shortcuts.
5. [optional] Click the Selection tab and select or deselect objects to generate.
6. Click OK to begin generation.

Note: For detailed information about the options available on the various tabs of the Generation window, see the Linking and Synchronizing Models chapter of the *Core Features Guide*.

Checking an ILM

The Information Liquidity Model is a very flexible tool, which allows you quickly to develop your model without constraints. You can, however, check the validity of your ILM at any time.

Note: We recommend that you check your PDM before generating other models from it. If the check encounters errors, generation will be stopped. The Check model option is enabled by default in the Generation dialog box.

You can check an ILM in any of the following ways:

- Press F4, or
- Select **Tools > Check Model** , or
- 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 Models chapter of the *Core Features Guide*.

The following sections document the ILM-specific checks available by default. For information about checks made on generic objects available in all model types, see "Checking a Model" in the Models chapter of the *Core Features Guide*.

Package Checks

The following ILM model checks are made on packages:

Check	Description and Correction
Package name and code uniqueness	Package 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

Database Checks

The following ILM model checks are made on databases:

Check	Description and Correction
Database name and code uniqueness	Database 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
Existence of data connection or data access link	A database must either be linked to at least one replication process or transformation process using a data connection, or to at least one data store [database, data access application or XML document] using a data access link Manual correction: Add any missing data connection links between the database and the replication process or the transformation process or add any missing data access links between the database and the data store Automatic correction: None
Database code maximum length	The database code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options. Manual correction: Modify the code length to meet this requirement Automatic correction: Truncates the code length to the maximum length specified in the XEM definition
Existence of model	At least one model must be attached to the database. Manual correction: Add any missing models in the Physical Data Models tab of the database property sheet Automatic correction: None

Replication Process Checks

The following ILM model checks are made on replication processes:

Check	Description and Correction
Replication process name and code uniqueness	Replication process 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
Existence of data or process connection	A replication process must be linked to at least one process using a process connection or to at least one database or XML document using a data connection. Manual correction: Add any missing connections to the replication process Automatic correction: None

Publication Checks

The following ILM model checks are made on publications:

Check	Description and Correction
Publication name and code uniqueness	Publication 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
Publication code maximum length	The publication code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options. Manual correction: Modify the code length to meet this requirement Automatic correction: Truncates the code length to the maximum length specified in the XEM definition
Existence of data connection	A publication must be linked to a data connection. Manual correction: Add any missing data connection links to publication from the replication property sheet Automatic correction: None
Existence of subscription	A subscription establishes a link between a publication and a database connection to define where data published via the publication must be replicated. A publication should be linked to at least one subscription. Manual correction: Add any missing subscription links to publication from the replication property sheet Automatic correction: None

Subscription Checks

The following ILM model checks are made on subscriptions:

Check	Description and Correction
Subscription name and code uniqueness	Subscription 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 data connection	<p>A subscription must be linked to a data connection.</p> <p>Manual correction: Add any missing data connection links to subscription from the replication property sheet</p> <p>Automatic correction: None</p>
Existence of publication	<p>A subscription establishes a link between a publication and a database connection to define where data published via the publication must be replicated.</p> <p>A subscription must be linked to at least one publication.</p> <p>Manual correction: Add any missing publications to subscription from the replication property sheet</p> <p>Automatic correction: None</p>

User Checks

The following ILM model checks are made on users:

Check	Description and Correction
User name and code uniqueness	<p>User names and codes must be unique in the model.</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>

Article Checks

The following ILM model checks are made on articles:

Check	Description and Correction
Article name and code uniqueness	<p>Article names and codes must be unique in the namespace.</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>
Article code maximum length	<p>The article code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options.</p> <p>Manual correction: Modify the code length to meet this requirement</p> <p>Automatic correction: Truncates the code length to the maximum length specified in the XEM definition</p>
Undefined source	<p>An article corresponds to a table or a view to replicate. An article should have a source table or view defined.</p> <p>Manual correction: Select articles from source database from the Articles tab of the Publication property sheet</p> <p>Automatic correction: None</p>

Article Column Checks

The following ILM model checks are made on article columns:

Check	Description and Correction
Article column name and code uniqueness	Article column 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
Undefined source	An article column corresponds to a table or view column to replicate. An article column should have a source table or view column defined. Manual correction: Select article columns from source database from the Columns tab of the Article property sheet Automatic correction: None

Procedure Checks

The following ILM model checks are made on procedures:

Check	Description and Correction
Procedure name and code uniqueness	Procedure names and codes must be unique in the namespace. The list of parameters must be taken into account. Manual correction: Modify the duplicate name/code Automatic correction: Appends a number to the duplicate name/code
Procedure code maximum length	The procedure code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options. Manual correction: Modify the code length to meet this requirement Automatic correction: Truncates the code length to the maximum length specified in the XEM definition
Undefined source	A procedure corresponds to a stored procedure linked to a table to replicate. A procedure should have a source stored procedure defined. Manual correction: Select procedures from source database from the Procedures tab of the Publication property sheet Automatic correction: None

Article Event Script Checks

The following ILM model checks are made on article event scripts:

Check	Description and Correction
Event script code maximum length	The event script code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options. Manual correction: Modify the code length to meet this requirement Automatic correction: Truncates the code length to the maximum length specified in the XEM definition
Event script event uniqueness	Event script events must be unique in the namespace. Manual correction: Modify the duplicate event script event Automatic correction: Deletes the duplicate event script event

Check	Description and Correction
Undefined event	<p>An event script allows you to define how events on an article are implemented. An event script must have its event defined.</p> <p>Manual correction: Select an event from the Event Selection dialog box accessible from the Event Scripts tab of an article property sheet</p> <p>Automatic correction: None</p>
Undefined script	<p>An event script allows you to define how events on the article are implemented. An event script must have its implementation script defined.</p> <p>Manual correction: Type a script for the event in the Script column accessible from the Event Scripts tab of an article property sheet</p> <p>Automatic correction: None</p>

Replication Process Event Script Checks

The following ILM model checks are made on replication process event scripts:

Check	Description and Correction
Event script code maximum length	<p>The event script code length is limited by the maximum length specified in the XEM definition [CodeMaxLen entry, in the Objects > Settings category] and in the naming conventions of the model options.</p> <p>Manual correction: Modify the code length to meet this requirement</p> <p>Automatic correction: Truncates the code length to the maximum length specified in the XEM definition</p>
Event script event uniqueness	<p>Replication process event script events must be unique in the namespace.</p> <p>Manual correction: Modify the duplicate event script event</p> <p>Automatic correction: Deletes the duplicate event script event</p>
Undefined event	<p>An event script allows you to define how events on a replication process are implemented. An event script must have its event defined.</p> <p>Manual correction: Select an event from the Event Selection dialog box accessible from the Event Scripts tab of a replication process property sheet</p> <p>Automatic correction: None</p>
Undefined script	<p>An event script allows you to define how events on the replication process are implemented. An event script must have its implementation script defined.</p> <p>Manual correction: Type a script for the event in the Script column accessible from the Event Scripts tab of a replication process property sheet</p> <p>Automatic correction: None</p>

XML Document Checks

The following ILM model checks are made on XML documents:

Check	Description and Correction
XML document name and code uniqueness	<p>XML document names and codes must be unique in the namespace.</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>

Check	Description and Correction
Existence of data connection or data access link	<p>An XML document must be either linked to at least one transformation process using a data connection, or to at least one data store [database, data access application or XML document], using a data access link</p> <p>Manual correction: Add any missing data connection links between the XML document and the transformation process, or add any missing data access links between the XML document and the data store</p> <p>Automatic correction: None</p>
Existence of model	<p>At least one model must be attached to the XML document.</p> <p>Manual correction: Add any missing models in the XSM Models tab of the XML document property sheet</p> <p>Automatic correction: None</p>

Business Process Checks

The following ILM model checks are made on business processes:

Check	Description and Correction
Business process name and code uniqueness	<p>Business process names and codes must be unique in the namespace</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>
Existence of connection	<p>At least one business process must be linked to another business process using an extended dependency</p> <p>Manual correction: Add any missing extended dependency links between business processes</p> <p>Automatic correction: None</p>
Existence of model	<p>At least one model must be attached to the business process</p> <p>Manual correction: Add any missing models in the BPM Models tab of the business process property sheet</p> <p>Automatic correction: None</p>
Non existence of data connection	<p>A business process must not be linked to a replication process using a data connection</p> <p>Manual correction: Delete any data connection between the business process and the replication process</p> <p>Automatic correction: None</p>

Flat File Checks

The following ILM model checks are made on flat files:

Check	Description and Correction
Flat file name and code uniqueness	<p>Flat file names and codes must be unique in the namespace.</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>

Check	Description and Correction
Existence of data structure column	At least one data structure column must be defined in the flat file. Manual correction: Add any missing data structure columns in the Data Structure Columns tab of the flat file. Automatic correction: None

Data Connection Checks

The following ILM model checks are made on data connections. Extremities of the data connection must be taken into account.

Check	Description and Correction
Data connection name and code uniqueness	Data connection 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

Process Connection Checks

The following ILM model checks are made on process connections. Extremities of the process connection must be taken into account.

Check	Description and Correction
Process connection name and code uniqueness	Data connection 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

Connection Group Checks

The following ILM model checks are made on connection groups:

Check	Description and Correction
Connection group name and code uniqueness	Connection 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

Server Checks

The following ILM model checks are made on servers:

Check	Description and Correction
Server name and code uniqueness	Server 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

Transformation Process Checks

The following ILM model checks are made on transformation processes:

Check	Description and Correction
Transformation process name and code uniqueness	Transformation process 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
Existence of data or process connection	A transformation process must be linked to at least one process using a process connection or to at least one database, business process, XML document or flat file using a data connection. Manual correction: Add any missing connections to the transformation process Automatic correction: None

Data Transformation Task Checks

The following ILM model checks are made on data transformation tasks:

Check	Description and Correction
Data transformation task name and code uniqueness	Data transformation task 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
Existence of data transformation action	At least one data transformation action should be associated with the data transformation task. Manual correction: Add any missing data transformation actions in the Actions tab of the data transformation action property sheet Automatic correction: None
Existence of data input	At least one data input should be associated with the data transformation task. Manual correction: Add any missing data inputs in the Inputs tab of the data transformation action property sheet Automatic correction: None
Existence of data output	At least one data output should be associated with the data transformation task. Manual correction: Add any missing data outputs in the Outputs tab of the data transformation action property sheet Automatic correction: None
Existence of data flow	At data transformation task should contain at least one data flow between each transformation step in the data transformation diagram. Manual correction: Add any missing data flows in the data transformation diagram Automatic correction: None

Data Transformation Diagram Checks

The following ILM model checks are made on data transformation diagrams:

Check	Description and Correction
Data transformation diagram name and code uniqueness	Data transformation diagram 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

Data Input Checks

The following ILM model checks are made on data inputs (database input, XML input, Web Service input, flat file input, generic data input):

Check	Description and Correction
Data input name and code uniqueness	Data input 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
Existence of data connection	A data input must be linked to a data connection. Manual correction: Select a data connection in the Data Connection list of the property sheet Automatic correction: None
Existence of source object	A data input must have at least one data structure source object. Manual correction: Add any missing source objects in the Data Structure Source Objects tab of the data input. Automatic correction: None
Existence of data structure column	At least one data structure column must be defined in the data input. Manual correction: Add any missing data structure columns in the Data Structure Columns tab of the data input. Automatic correction: None
Data structure mismatch	The data type of the data structure column and its source objects must match. Manual correction: Set the same data type for the data structure column and its source column. Automatic correction: None

Data Output Checks

The following ILM model checks are made on data outputs (database output, XML output, flat file output, generic data output):

Check	Description and Correction
Data output name and code uniqueness	Data output 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
Existence of data connection	A data output must be linked to a data connection. Manual correction: Select a data connection in the Data Connection list of the property sheet Automatic correction: None

Check	Description and Correction
Existence of source object	<p>A data output must have at least one data structure source object.</p> <p>Manual correction: Add any missing source objects in the Data Structure Source Objects tab of the data output.</p> <p>Automatic correction: None</p>
Existence of data structure column	<p>At least one data structure column must be defined in the data output.</p> <p>Manual correction: Add any missing data structure columns in the Data Structure Columns tab of the data output.</p> <p>Automatic correction: None</p>
Existence of target object	<p>A data output must have at least one data structure target object.</p> <p>Manual correction: Add any missing target objects in the Data Structure Target Objects tab of the data output.</p> <p>Automatic correction: None</p>
Data structure mismatch	<p>The data type between the data structure column and its source objects must match</p> <p>Manual correction: Set the same data type for the data structure column and its source objects.</p> <p>Automatic correction: None</p>

Data Transformation Action Checks

The following ILM model checks are made on data transformation actions (data split, data sort, data merge, data projection, data calculator, generic data action, data join, script execution, data query execution, data aggregation, data filter, data lookup):

Check	Description and Correction
Data transformation action name and code uniqueness	<p>Data transformation action names and codes must be unique in the namespace [model or package].</p> <p>Manual correction: Modify the duplicate name/code</p> <p>Automatic correction: Appends a number to the duplicate name/code</p>
Existence of source object	<p>A data transformation action must have at least one data structure source object.</p> <p>Manual correction: Add any missing source objects in the Data Structure Source Objects tab of the data transformation action.</p> <p>Automatic correction: None</p>
Existence of data structure column	<p>At least one data structure column must be defined in the data transformation action.</p> <p>Manual correction: Add any missing data structure columns in the Data Structure Columns tab of the data transformation action.</p> <p>Automatic correction: None</p>
Existence of data structure sorted column [data sort only]	<p>A data sort must have at least one sort column defined to sort data.</p> <p>Manual correction: Add any missing sort columns in the Sort Columns tab of the data sort</p> <p>Automatic correction: None</p>

Check	Description and Correction
Undefined script [script execution and data lookup only]	<p>A script execution/data lookup must have a script defined.</p> <p>Manual correction: Define any missing script in the Script tab of the script execution/data lookup</p> <p>Automatic correction: None</p>
Existence of data connection [data query execution only]	<p>A data query execution must be linked to a data connection to insert or update data in the database.</p> <p>Manual correction: Select a data connection in the Data Connection list in the Script tab of the property sheet</p> <p>Automatic correction: None</p>
Existence of source expression for data structure columns [data aggregation only]	<p>A data aggregation must have at least one source expression defined to aggregate data.</p> <p>Manual correction: Add any missing source expressions in the Source expression box in the Data Structure Source Object tab of the data aggregation</p> <p>Automatic correction: None</p>
Existence of aggregated column [data aggregation only]	<p>A data aggregation must have at least one aggregation column defined to aggregate data.</p> <p>Manual correction: Add any missing aggregation columns in the Aggregation Columns tab of the data aggregation</p> <p>Automatic correction: None</p>
Undefined criterion [data filter only]	<p>A data filter must have a criterion defined to filter data</p> <p>Manual correction: Add any missing criteria in the Criteria tab of the data filter</p> <p>Automatic correction: None</p>
Existence of data structure join [data join only]	<p>A data join must have a data structure defined</p> <p>Manual correction: Add any missing data structure joins in the Join Columns tab of the data join</p> <p>Automatic correction: None</p>
Missing occurrences in join sources [data join only]	<p>A data structure join must have two sources defined.</p> <p>Manual correction: Add any missing sources for a data structure join in the Join Columns tab of the data join</p> <p>Automatic correction: None</p>
Match data structure column sources [data merge only]	<p>The two data structure column sources of a data merge must be equivalent [same number of data structure columns and same data type]</p> <p>Manual correction: Add any missing data structure column in the data merge sources and/or modify the source data type in the data structure column property sheet</p> <p>Automatic correction: None</p>

Data Flow Checks

The following ILM model checks are made on data flows:

Check	Description and Correction
Data flow name and code uniqueness	Data flow 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

Transformation Control Flow Checks

The following ILM model checks are made on transformation control flows:

Check	Description and Correction
Transformation control flow name and code uniqueness	Transformation control flow 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
Existence of transformation start	At least one transformation start should be associated with the data transformation control flow. Manual correction: Add any missing transformation starts in the Starts tab of the transformation control flow property sheet Automatic correction: None
Existence of transformation end	At least one transformation end should be associated with the transformation control flow. Manual correction: Add any missing transformation ends in the Ends tab of the transformation control flow property sheet Automatic correction: None
Existence of control flow	At transformation control flow should contain at least one control flow between each start, end, transformation task execution and synchronization in the transformation control flow diagram. Manual correction: Add any missing data flows in the transformation control flow diagram Automatic correction: None

Transformation Control Flow Diagram Checks

The following ILM model checks are made on transformation control flow diagrams:

Check	Description and Correction
Transformation control flow diagram name and code uniqueness	Transformation control flow diagram 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

Transformation Start/end Checks

The following ILM model checks are made on transformation starts and ends:

Check	Description and Correction
Transformation start/end name and code uniqueness	Transformation start and end 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

Transformation Synchronization Checks

The following ILM model checks are made on transformation synchronizations:

Check	Description and Correction
Transformation synchronization name and code uniqueness	Transformation synchronization 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

Transformation Decision Checks

The following ILM model checks are made on transformation decisions:

Check	Description and Correction
Transformation decision name and code uniqueness	Transformation decision 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

Transformation Task Execution Checks

The following ILM model checks are made on transformation task executions:

Check	Description and Correction
Transformation task execution name and code uniqueness	Transformation task execution 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
Existence of data transformation task	A transformation task execution must be linked to a transformation task. Manual correction: Select a transformation task in the Task list of the property sheet Automatic correction: None

Control Flow Checks

The following ILM model checks are made on control flows:

Check	Description and Correction
Control flow name and code uniqueness	Control flow 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

Comparing and Merging ILMs

You can compare and merge two Information Liquidity Models.

The comparison process allows you to highlight the differences between two ILMs.

Working with Information Liquidity Models

The merge process allows you to form a single model that combines design efforts performed independently by several team members.

Merge is performed from left to right, the model in the right pane is compared to the model in the left pane, differences are highlighted and merge actions are proposed in the model to be merged.

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

Generating and Reverse Engineering Replication Objects

PowerDesigner can generate and reverse engineer replication objects.

Generating Replication Objects

The generation process allows you to generate for an extended model definition (XEM) in an ILM.

There are several ways to generate a target in an ILM:

- To generate several targets at the same time, select **Tools > Extended Generation**
- To generate for a specific target only, use one of the following commands: **Tools > Replication Server 12.5 > Generate Scripts**, **Tools > Replication Server 12.6 > Generate Scripts**, **Tools > MobiLink 9.0 > Generate Scripts**

To Generate for Several Targets:

You can generate for several targets.

1. Select **Tools > Extended Generation**.

The Generation dialog box opens to the Targets page.

2. Select or browse to a directory that will contain the generated files (.SQL)
3. Select the extended model definition that you want to use for generation.
4. Click Apply if you want to generate at a later time.

or

Click OK to generate.

To Generate for a Specific Target:

You can generate for a specific target.

1. Select **Tools > Specific target > Generate Scripts**.

The Generation dialog box opens to the Selection page.

2. Select or browse to a directory that will contain the generated files (.SQL).
3. Click OK to generate

The Generation dialog box contains the following pages. Note that extended generations for a specific XEM, do not display the Targets page in the Generation dialog box:

Page	Description
Targets	Allows you to select a generation target (XEM), which can be MobiLink 9.0 or Replication Server 12.6 or 12.5
Selection	Allows you to select objects to generate
Options	Allows you to customize the generation
Tasks	Allows you to use specific commands defined for the target

Selecting Targets to Generate in an ILM

You can generate SQL script files for the MobiLink and Replication Server targets in an ILM.

User-defined Generation

You can use the Generation Template Language (GTL) mechanism to produce user-defined text for code generation. It allows you to access all elements of the model and to customize the generation process. It is available for all targets supported in an ILM.

For more information on the Generation Template Language mechanism, see the Customizing Generation with GTL chapter of the *Customizing and Extending PowerDesigner* manual.

Trace Mode in Preview Pages

You can display the generation templates that generate code from the Preview page in object property sheets, provided the *Enable Trace Mode* check box is selected in the target editor.

For more information on the trace mode property, see "Generation category" in the Resource Files and the Public Metamodel chapter of the *Customizing and Extending PowerDesigner* manual.

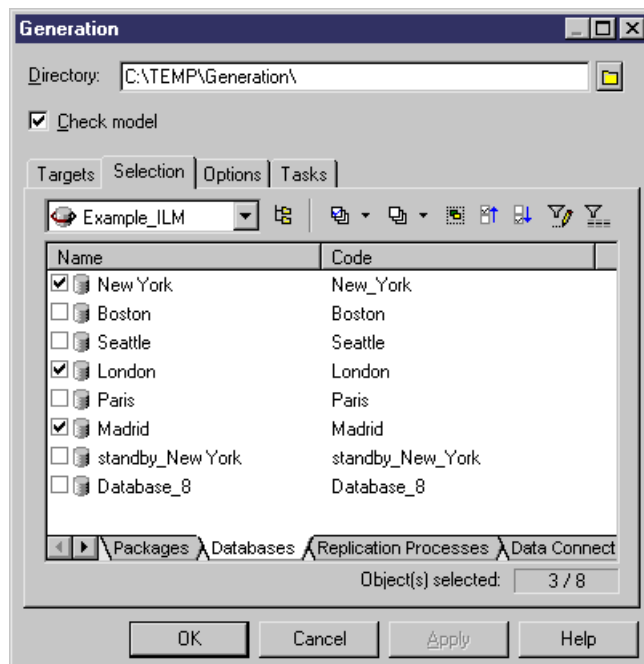
Check Model

You can check your model before generation if you select the Check Model check box. This check box is available for all targets. The generation stops if an error is found: you must correct this error before starting the generation again.

For more information on the Check Model feature, see [Checking an ILM](#) on page 92.

Selecting Objects to Generate in an ILM

You select objects for generation from the Selection page of the Generation dialog box. The Selection page contains tabbed pages for selecting specific objects of the model.



Listing Objects Contained in a Model or Package

You can display objects in the current model, or objects in individual packages contained in the model.

If you select the Include Sub-Packages tool, you can display either all objects in the current model, or all objects in a package.

The selection options are the following:

Parent object	Include Sub-Objects tool selected	Include Sub-Objects tool not selected
Model	All objects in the model including all objects contained in packages and sub-packages	All objects in the model except objects contained in packages and sub-packages
Package	All objects contained in the package including all objects contained in sub-packages	All objects in the package except objects contained in sub-packages

Defining Generation Options in an ILM

The Options page contains options that allow you to customize the generation process.

If you do not see the Options page in the Generation dialog box, it means that no option has been defined in the corresponding target (.XEM file).

You can however customize your target at any time, and add any options you may need during generation. You define generation options in the Generation/Options category of the target available from **Tools > Resources > Extended Model Definitions > Information Liquidity Models**.

For more information on generation options, see "Generation category" in the Resource Files and the Public Metamodel chapter of the *Customizing and Extending PowerDesigner* manual.

Defining Generation Tasks in an ILM

The Tasks page contains standard commands defined in the target. You can select any tasks to execute at generation time.

If you do not see the Tasks page in the Generation dialog box, it means that no command has been defined in the corresponding target (.XEM file).

You can however customize tasks for your current target so that, during generation, your files are automatically packaged in an archive for example. You define generation tasks and commands in the Generation/Tasks category and the Generation/Commands category of the target available from **Tools > Resources > Extended Model Definitions > Information Liquidity Models**.

For more information on how to add tasks, see "Generation category" in the Resource Files and the Public Metamodel chapter of the *Customizing and Extending PowerDesigner* manual.

Note: You can easily modify the order of commands execution using the Move up or Move down arrows. You can only select one value check box at a time.

Reverse Engineering Replication Objects

You can reverse engineer Replication Server or MobiLink objects from the embedded Replication Server database (RDSS) via a database connection to create or update them in your ILM.

You can reverse engineer into an ILM using two database connections:

- One for the consolidated database (or source database)
- One for the remote database

During the reverse engineering process, each time a database connection is required, the Select a Data Source dialog box opens to allow you to define one.

If you have not defined the consolidated database or remote database, PowerDesigner creates a default one for you during the reverse engineering process.

There are two ways for reverse engineering Replication Server or MobiLink objects:

- Reverse engineer a single replication process using the Reverse Engineering command from its contextual menu
- Reverse engineer several replication processes using the **Tools > Reverse Engineering Replication Server** command that allows you to select the replication processes to reverse engineer

Selecting a Data Source

The connection to a data source parameters are the following:

Parameter	Description
Data source	Specifies the connection profile that is used to connect to your database.
Login	Name of the user ID with which you log on to the database
Password	Password for the connection

For detailed information about creating, configuring, and using connection profiles, see "Connecting to a Database" in the Models chapter of the *Core Features Guide*.

Reverse Engineering a Single Replication Process

You reverse engineer a single replication process using the Reverse Engineering command from its contextual menu. It allows you to reverse engineer all its related objects.

Right-click a replication process symbol in the diagram and select Reverse Engineering from the contextual menu that is displayed.

Once the reverse engineering is performed, PowerDesigner displays the Merge Models window to show you the differences between the reverse engineered model and the current model. You can decide whether you want to accept or not the created or modified objects.

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

The reversed objects are added to your model. They are visible in the diagram and in the Browser. They are also listed in the Reverse page of the Output window, located in the lower part of the main window.

Reverse Engineering Several Replication Processes

You reverse engineer several replication processes using the **Tools > Reverse Engineering Replication Server** command that allows you to select the replication processes to reverse engineer.

1. Select **Tools > Reverse Engineering Replication Server** to open a selection dialog box.
2. Select the replication processes you want to reverse engineer.
3. Click OK to start the process.

Once the reverse engineering is performed, PowerDesigner displays the Merge Models window to show you the differences between the reverse engineered model and the current model. You can decide whether you want to accept or not the created or modified objects.

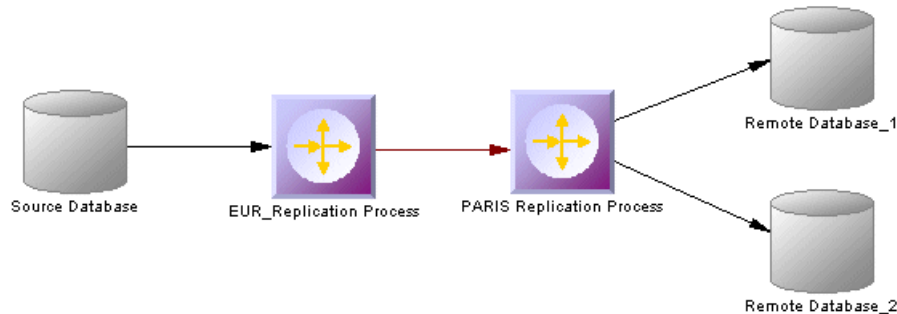
For more information on merging models, see the Comparing and Merging Models chapter in the *Core Features Guide*.

The reversed objects are added to your model. They are visible in the diagram and in the Browser. They are also listed in the Reverse page of the Output window, located in the lower part of the main window.

Working with Replication Server

Replication Server is a relational database replication engine which helps you to replicate data from a primary database to one or more replicate databases. PowerDesigner supports modeling for Replication Server version 12.5 and higher, including round-trip engineering.

The following example shows an information liquidity diagram representing a replication of data from a source database to two remote databases, each of which are modeled in Physical Data Models (PDMs):



Modeling for Replication Server

PowerDesigner supports the modeling of all the components required to deploy a Replication Server solution in your environment.

Network Components

The PowerDesigner ILM provides support for the following network components when modeling a Replication Server environment:

- *Replication Servers* – coordinate the data replication activities for the local data servers and exchange data with Replication servers at other sites. PowerDesigner models replication servers as replication processes (see [Replication Processes \(ILM\)](#) on page 26) with a Replication Server type and additional properties (see [Replication Server Properties](#) on page 118).
- *Primary and Replicate databases* – contain data that will be replicated and receive replicated data respectively. The structure of each database is modeled in an attached PDM. PowerDesigner models databases in a Replication Server environment as standard databases (see [Databases \(ILM\)](#) on page 18) with additional properties (see [Replication Server Database Properties](#) on page 118).
- *Servers* – provide a logical location for replication servers and databases. You should associate all of your network components to appropriate servers to ensure correct script generation and a model check is used to verify that each component is associated to a server (see [Servers \(ILM\)](#) on page 22).

Data Connections

Network components are connected via the following kinds of data connections:

- *Connections*– specify a message stream from a database to a Replication Server, or from a Replication Server to a database. PowerDesigner models connections as standard data connections (see [Data Connections \(ILM\)](#) on page 41) with additional properties (see [Replication Server Connection Properties](#) on page 119).

- *Routes*—specify one-way message streams that send requests from one Replication Server to another. PowerDesigner models routes as standard process connections (see [Process Connections \(ILM\)](#) on page 40) with additional properties (see [Replication Server Route Properties](#) on page 125).
- *Logical Connections*—consist of a pair of physical connections that are configured in a warm-standby environment (see [Creating a Warm Standby Database](#) on page 115) to link an active and a standby database. PowerDesigner models logical connections as data connection groups (see [Data Connection Groups \(ILM\)](#) on page 38) with additional properties (see [Replication Server Logical Connection Properties](#) on page 127).

Replication Definitions

Replication definitions describe the tables, views, databases, and stored procedures that you want to replicate:

- *Replication definitions*—describe a source table to be replicated, the columns you want to copy, and may also describe attributes of the destination table. Destination tables that match the specified characteristics can subscribe to the replication definition. PowerDesigner models Replication Server replication definitions as articles (see [Articles \(ILM\)](#) on page 31) with additional properties (see [Replication Server Replication Definition and Article Properties](#) on page 128).
- *Database replication definitions*—allow you to replicate an entire primary database to one or more replicate databases. PowerDesigner models database replication definitions as publications (see [Publications \(ILM\)](#) on page 29) with a Database type and additional properties (see [Replication Server Database Replication Definition Properties](#) on page 129).
- *Function replication definitions*—specify information about a stored procedure to replicate. PowerDesigner models function replication definitions as publication procedures (see [Procedures \(ILM\)](#) on page 35) with additional properties (see [Replication Server Function Replication Definition Properties](#) on page 129).
- *Articles*—specify a replication definition extension for tables or stored procedures that allow you to assign table or function replication definitions in a publication. PowerDesigner models Replication Server articles as standard articles (see [Articles \(ILM\)](#) on page 31) with additional properties (see [Replication Server Article Column Properties](#) on page 130).

Publications and Subscriptions

Replication definitions are grouped together into publications that replicate databases can subscribe to:

- *Publications*—collect replication definitions together, to simplify subscriptions. PowerDesigner models Replication Server publications as standard publications (see [Publications \(ILM\)](#) on page 29) with additional properties (see [Replication Server Publication Properties](#) on page 130).
- *Subscriptions*—instruct Replication Server to replicate the data specified in a replication definition or publication to a particular replicate database. PowerDesigner models Replication Server subscriptions as standard subscriptions (see [Subscriptions \(ILM\)](#) on page 36) with additional properties (see [Replication Server Subscription Properties](#) on page 131).

Other Objects

These sub-objects are created inside a Replication Server object:

- *Users*—specifies a user name and password to connect to a Replication Server. PowerDesigner models Replication Server users as standard users (see [Users \(ILM\)](#) on page 37) with additional properties (see [Replication Server User Properties](#) on page 131).
- *Function strings*—contain instructions for executing a function in a database. PowerDesigner models Replication Server function strings as event scripts (see [Event Scripts \(ILM\)](#) on page 39) with additional properties (see [Replication Server Function String Properties](#) on page 131).

Creating a Basic Replication Server Environment with the Replication Wizard

The replication wizard provides a quick way to configure a Replication Server process for replicating one database to another. You can replicate the entire database or choose specific tables to replicate. You can run the wizard as many times as necessary to create additional replications on a single or multiple Replication Servers.

Prerequisites

Although you can launch the Replication wizard without already having modeled your databases in PDMs, we recommend that you as a minimum create a PDM to represent the structure of your primary database. You can reverse-engineer an existing database by selecting **File > Reverse Engineer > Database**.

1. Select **File > New Model** to open the New Model window and select **Information Liquidity Model** in the Model Type list and Information Liquidity Diagram in the Diagram pane. Click the **Extensions** button to open the Extended Model Definitions dialog, click the General Purpose sub-tab, select the appropriate version of Replication Server, and then click OK to return to the New Model window.
2. Click **OK** to create the ILM, which opens with an empty diagram.
3. Click the Replication Server tool in the palette, and then click in the center of the diagram to create a replication process. Right-click the Replication server symbol, and select Replication Wizard to open a wizard that guides you through configuring Replication Server for replicating data between your source and target databases.

Note: For detailed information about using the Replication Wizard, see [Replicating Objects with the Replication Wizard](#) on page 16

When you click OK to close the wizard, PowerDesigner will create source and target database objects in your ILM, as well as all the necessary articles, publications, and subscriptions that Replication Server requires to manage the replication of data between them.

Completing your Replication Environment

Once you have created any appropriate replication processes, replication definitions, publications and subscriptions with the replication wizard, you should complete your environment with the following kinds of supporting objects.

Creating Servers

Though it is not compulsory to assign each of your databases and replication servers to a server, we strongly recommend that you do so, in order to enable the accurate generation of appropriate network addresses in your replication scripts and a model check is used to verify that each component is associated to a server.

For information about working with servers, see [Servers \(ILM\)](#) on page 22.

Creating Maintenance Users

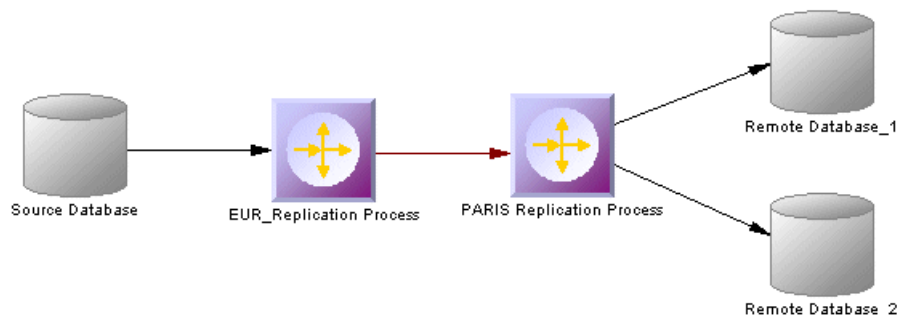
In order to access all the databases in the environment, Replication Server needs you to allocate Maintenance Users in each primary and replicate database. The maintenance user needs to have permission rights to access the source tables in the primary database and the target tables in the remote database. These are specified on the Connection tab of the property sheet of the data connection that links the database to the replication server (see [Replication Server Connection Properties](#) on page 119).

A SQL script that contains the definition of permission rights for the maintenance user is generated for the primary database and the remote database on the tables referenced in the replication definition.

Creating Additional Replication Servers

In many replication environments, individual replication servers are located at each physical site, and connected by routes, oriented links that transfer requests from one Replication Server to another.

Use the connection tool to draw a route between to replication processes.



The name of the administrator must be the same for both process servers. If the names differ, you must specify the user name and password for the target server in the Route Options tab of the route property sheet.

Controlling Subscription Materialization

Typically, creating a subscription causes Replication Server to immediately materialize the subscription by copying the initial requested data from the primary database to the replicate database. Once the subscription is created and materialized, Replication Server begins distributing primary data changes to the replicated data.

For large tables and non Sybase Adaptive Server® Enterprise databases, it can be more efficient to delay materializing data until after the creation of a subscription, displacing it to a time when the network is less busy.

Use the Materialize Subscription generation option to control when materialization is performed.

Populating Database Objects

Database symbols provide various shortcuts to assist you in defining their structures. You can:

- Reverse-engineer an existing database – by right-clicking the database and selecting Reverse Engineer Database, to create a new PDM.
- Create a primary or replicate database structure from article or subscription information - by right-clicking the database and selecting Update *Type* Database to deduce the database structure from the definition of articles in the Replication server, in which a subscription must be specified.
- Associate the same PDM with the source database and the remote database - if the remote database has the same structure as the consolidated database.

Note: You can connect to the Replication Server System Database (RSSD) at any time by right-clicking the replication process, and selecting the Connect and the Execute SQL commands.

Previewing scripts

When modeling, you can preview the script that will be generated for any object by clicking the Preview tab in its property sheet. Objects that belong to a replication server (such as replication definitions, publications, and subscriptions) have their own Preview tabs, which show the part of the replication server script that is dedicated to them.

Modeling for Heterogeneous Replications with Replication Agents and ECDA

Heterogeneous databases are databases other than Sybase Adaptive Server Enterprise or Adaptive Server Anywhere/SQL Anywhere®. Replication agents allow Replication Server to communicate with heterogeneous primary databases.

Modeling a Heterogeneous Primary Database

The replication agent captures the changes made in the primary database and sends the transaction log to the primary Replication Server. To model for a heterogeneous primary database using a replication agent, you need to:

- Specify server objects to contain the primary database and replication process, each with the appropriate host machine name and port number (see [Server Properties](#) on page 22).
- Specify the appropriate properties for the Replication Server connection and RSSD database on the **RepServer Connection** tab of the replication process property sheet (see [Replication Server Properties](#) on page 118)
- Specify the **RepAgent type**, the **RepAgent user** (to access the replication server), the **primary database user** (to access the database), and the other properties on the **RepAgent Options** tab of the primary database property sheet (see [Replication Server Database properties](#) on page 118).

Note: In order to generate and execute the replication agent SQL file using isql, you must select the "Execute generated scripts in Replication Agent" on the Tasks tab of the Generation window (see [Generating for Replication Server](#) on page 132).

Modeling a Heterogeneous Replicate Database

The ECDA communicates replicated data from a replication server to a heterogeneous replicate database. To model for a heterogeneous replicate database, you need to:

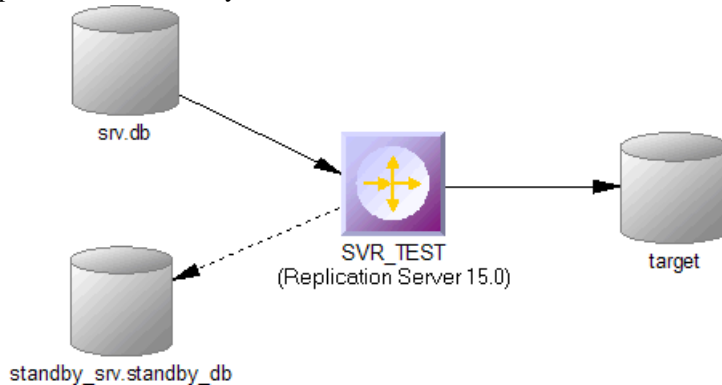
- Specify server objects to contain the replicate database and replication process, each with the appropriate host machine name and port number (see [Server Properties](#) on page 22).
- Specify the DirectConnect instance name in the Code of the replicate database in its property sheet (see [Replication Server Database properties](#) on page 118).

Modeling a Warm Standby Application

A warm standby application is a pair of Adaptive Server or SQL Server databases, one of which is a backup of the other. Client applications update the active database, and Replication Server maintains the standby database as a copy of the active database.

1. Create a database in your ILM and link it to a PDM that contains the structure of the active database.
2. Create a Replication Server replication process, and link the database to it with a connection.
3. Right-click the database and select **Create Standby Database**.

PowerDesigner converts the standard connection into a logical connection between the active database, the replication process, and a standby database, that it creates and links to the PDM used to describe the active database.



Modeling for Mirror Activator

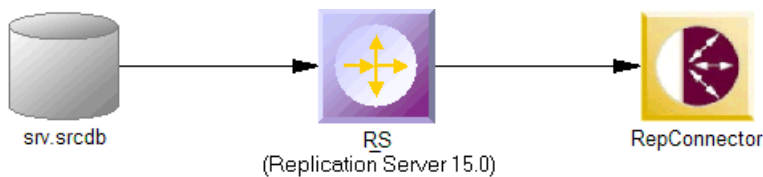
Mirror Activator is a combination of Mirror Replication Agent, Replication Server, and a third party disk replication system to add disk replication for transaction logs to the Replication Server transaction-based replication to provide an optimal disaster recovery solution.

The Mirror Replication replication agent reads the log file replicated by a disk replication system, and sends it to Replication Server. To model for a primary database to which the Mirror Replication replication agent is deployed, simply select Mirror Activator in the RepAgent type list on the RepAgent Options tab of its property sheet (see [Replication Server Database properties](#) on page 118).

Modeling for RepConnector

RepConnector allows you to use Sybase Real-Time Data Services to capture transactions in an ASE database and deliver them as events to external applications in real time. PowerDesigner supports modeling for replication environments in which RepConnector is deployed, but does not generate specific orders for RepConnector itself.

To specify that an ASE database has RepConnector enabled, simply select RepConnector in the Type list on the General tab of the database property sheet. The database symbol changes to reflect the use of RepConnector:



Note: If you are working with PowerDesigner in the Eclipse environment, you can invoke the RepConnector Manager directly from the remote database object contextual menu.

Modeling Replications to a Sybase IQ Data Warehouse

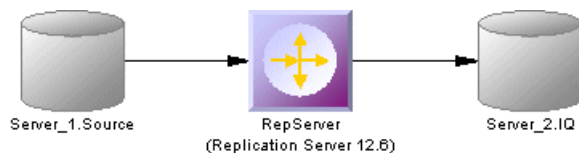
Sybase® IQ (IQ) is a high-performance decision support server designed specifically for data warehousing. Since IQ is not optimized for inserting, updating and deleting row by row, you should implement a *staging database* to replicate data from OLTP databases to an IQ data warehouse.

PowerDesigner can automate the creation of the staging database. You create a standard replication with Sybase IQ as the target database, and then, a single command allows you to create all the artifacts required to implement the staging database.

1. Create a PDM to represent the structure of your primary database. You can reverse-engineer an existing database by selecting **File > Reverse Engineer > Database**.
2. Select **File > New Model** to open the New Model window and select **Information Liquidity Model** in the Model Type list and Information Liquidity Diagram in the Diagram pane.
3. Click the **Extensions** button to open the Extended Model Definitions dialog, click the General Purpose sub-tab, select the appropriate version of Replication Server and the IQ Staging extended model definition, and then click OK to return to the New Model window.
4. Click **OK** to create the ILM, which opens with an empty diagram.
5. Click the **Replication Server** tool in the palette, and then click in the center of the diagram to create a replication process. Right-click the Replication server symbol, and select **Replication Wizard** to open a wizard that guides you through configuring Replication Server for replicating data between your source and target databases.

Note: The source database can be any supported database, the target database must be Sybase IQ. For detailed information about using the Replication Wizard, see [Replicating Objects with the Replication Wizard](#) on page 16

When you click OK to close the wizard, PowerDesigner will create source and target database objects in your ILM, as well as all the necessary articles, publications, and subscriptions that Replication Server requires to manage the replication of data between them:

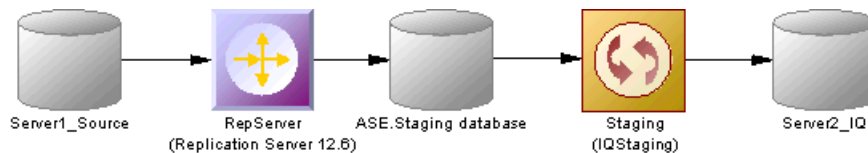


6. Open the property sheet of the IQ database, select the Staging Database tab, and enter the appropriate properties:

Staging database options	Description
Sybase ASE version	Version of the Sybase ASE staging database automatically created.
Database name	PDM name of the staging database.
Database code	PDM code of the staging database.
Server name	Server name of the staging database.
Server code	Serve code of the staging database.
Use insert table in Sybase IQ	Indicates that an insert staging table will be used in Sybase IQ to copy inserted rows from staging database in order to support transformation inside Sybase IQ.

Staging database options	Description
Support update in Sybase IQ	Indicates that an update statement will cause an update in Sybase IQ. If you do not select this option, update statements will be replaced by delete and insert statements.
Insert table code	Template for defining the code of an insert table.
Update table code	Template for defining the code of an update table.
Delete table code	Template for defining the code of a delete table.
Use stored procedure for function strings	Creates stored procedures in the staging database and uses them in RepServer function strings.
Insert procedure code	Template for defining the code of insert stored procedures.
Update procedure code	Template for defining the code of update stored procedures.
Delete procedure code	Template for defining the code of delete stored procedures.

- Click OK to return to the diagram and then select **Tools > Check Model** to verify that your model contains no errors and then save the model for reference.
- Select **Tools > Generate Information Liquidity Model** to open the Generate dialog.
- Click the **Enable transformations** button on the **Detail** tab, and then click the **Extended Model Definitions** tab and select **Sybase IQ Staging**.
- Click OK to launch the generation of a new ILM that recreates the original transformation but with an ASE staging database and an IQ Staging transformation process inserted between the replication server and the IQ database to represent the data transfer between the staging database and IQ:



The RepServer definition is modified, it is no longer directly connected to Sybase IQ but to the Sybase ASE staging database, and some function strings to replicate data into the Sybase ASE staging database have been added.

- Creating a Sybase ASE database with the same structure as Sybase IQ.
- Creating the stored procedures used by RepServer function strings in the staging database.
- Changing the RepServer connection to the staging database.
- Creating or modifying RepServer function strings to invoke the stored procedures.
- Creating staging tables in Sybase IQ to move data from the staging database into temporary tables in Sybase IQ before moving the data into Sybase IQ tables.
- Creating a stored procedure in Sybase IQ to load data from the staging database into Sybase IQ.
- Creating a stored procedure in the staging database to clean transferred data.

Note: If you need to change any aspect of your replication definitions, you must do so in the original ILM, and then regenerate to recreate the staging database. Any changes made to replication definitions in the generated ILM will not be accurately reflected in the staging database.

Generating Scripts for Replication to IQ

To generate the RepServer definition and Sybase ASE staging database, you need to generate the RepServer definition, the Sybase ASE staging database, and the Sybase IQ database.

- Right click the Replication Server process symbol and select **Generate Scripts**. Click the **Tasks** tab and select the **Execute generated scripts in Replication Server** task.

The Replication Server creation script is generated and executed using ISQL.

2. Click **OK** in the Generation dialog box.
3. Right-click the Sybase ASE staging database symbol and select **Generate Database**. Specify any appropriate database generation options and click **OK** to start generation.
4. Right-click the Sybase IQ database symbol and select **Generate Database** or **Modify Database**. Specify any appropriate database generation or modification options and click **OK** to start generation.

Transferring Staged Data to IQ

When the replication is set up, you can start RepServer to begin data replication. Data modifications made in the source database are replicated to the Sybase ASE staging database or the staging tables in Sybase IQ. At some point, you need to transfer the data from the staging database into Sybase IQ.

You automate this process using a script that performs the following tasks periodically.

1. Suspend replication to make sure data will not change during the transfer from staging database to Sybase IQ.
2. Run the IQ_LOAD_STAGING stored procedure in Sybase IQ to move data into Sybase IQ.
3. Run the IQ_CLEAN_STAGING stored procedure in Sybase ASE staging database to remove the already transferred data.
4. Resume replication.

Replication Server Object Properties

When modeling for a Replication Server environment, you use standard ILM objects with additional properties.

Replication Server Properties

Replication Server property sheets contain all the standard replication process tabs, along with the RepServer Connection tab, the properties of which are listed below.

Property	Description
Replication Server connection options	Specifies the connection information for Replication Server. You have to specify the following options: <ul style="list-style-type: none"> Port number – specifies the Replication Server port number (Scripting name: PortNumber) User name – specifies the name of the administration user (Scripting name: UserName) Password – specifies the password of the administration user (Scripting name: Password)
RSSD database options	Specifies the connection information for the RSSD. You have to specify the following options: <ul style="list-style-type: none"> Host name (Scripting name: HostName) Port number (Scripting name: PortNumber) Database name (Scripting name: DatabaseName) Database type Scripting name:DatabaseType()

Replication Server Database Properties

Primary databases connected to a Replication Server replication process have a property sheet that displays an additional, RepAgent Options, tab, the properties of which are listed below:

Property	Description
Use replication agent	Specifies if a replication agent should be used for primary database. This option is not necessary if the PDM of the primary database is defined and opened. Scripting name: UseReplicationAgent

Property	Description
RepAgent type	Specifies the replication agent type (Oracle, DB2, SQL Server, Informix, Mirror Activator). Scripting name: RepAgentType
RepAgent name	Specifies the replication agent instance name. It is used to generate replication agent script using isql. Scripting name: RepAgentName
RepAgent user name	Specifies the replication agent user login name. It is used to generate replication agent script using isql. Scripting name: RepAgentUserName
RepAgent password	Specifies the replication agent user login password. It is used to generate replication agent script using isql. Scripting name: RepAgentPassword
Primary database user name	Specifies the primary database server user login name for the replication agent instance. Scripting name: RepAgentPrimDBUserName
Primary database password	Specifies the primary database server user login password for the replication agent instance. Scripting name: RepAgentPrimDBPassword
RepServer user name	Specifies the Replication Server user login name for the replication agent instance. Scripting name: RepAgentRepServerUserName
RepServer password	Specifies the Replication Server user login password for the replication agent instance. Scripting name: RepAgentRepServerPassword
RSSD user name	Specifies the RSSD user login name for the replication agent instance. Scripting name: RepAgentRSSDUserName
RSSD password	Specifies the RSSD user login password for the replication agent instance. Scripting name: RepAgentRSSDPassword

Replication Server Connection Properties

Replication Server connection property sheets contain all the standard data connection tabs, along with the Connection tab, the Database Options tab, the Security tab, and the Transaction Options tab. The properties of these tabs are listed below.

Connection Tab

The following properties are available on the Connection tab:

Property	Description
Connection option	Option for a connection: dsi_suspended or log transfer on. Scripting name: ConnectionOption
Function string class	Name of the function class. Scripting name: FunctionClass

Property	Description
Error string class	Name of the error class. Scripting name: ErrorClass
User name	Maintenance user name. Scripting name: UserName
Password	Maintenance user password. Scripting name: Password
Dump marker	If this connection is in a connection group, then it can be flagged as dump marker. Scripting name: DumpMarker

Database Options tab

The following properties are available on the Database Options tab:

Property	Description
Number of commands in log	Specifies the number of commands to write into the exceptions log for a transaction. The value "-1" stands for all commands. Scripting name: DatabaseParameter_dsi_max_cmds_to_log
Number of bytes in log	Specifies the number of bytes to write into the exceptions log for each rs_writetext function in a failed transaction. Change this parameter to prevent transactions with large text, image or raw object columns from filling the RSSD or its log. The value "-1" means all text, image, or rawobject columns. Scripting name: DatabaseParameter_dsi_max_text_to_log
Number of transactions in group	Specifies the maximum number of transactions in a group. Larger numbers may improve data latency at the replicate database. Range of values: 1 – 100. Scripting name: DatabaseParameter_dsi_max_xact_in_group
Number of parallel threads	Specifies the number of parallel DSI threads to be reserved for use with large transactions. The maximum value is one less than the value of dsi_num_threads. Scripting name: DatabaseParameter_dsi_num_large_xact_threads
Cache size	Specifies the maximum SQT (Stable Queue Transaction interface) cache memory for the database connection, in bytes. The default, "0," means that the current setting of sqt_max_cache_size is used as the maximum cache size for the connection. To confirm the current value of sqt_max_cache_size, execute rs_configure. Scripting name: DatabaseParameter_dsi_sqt_max_cache_size
Group size	Specifies the maximum number of bytes, including stable queue overhead, to place into one grouped transaction. A grouped transaction is multiple transactions that the DSI applies as a single transaction. A value of -1 means no grouping. Scripting name: DatabaseParameter_dsi_xact_group_size
Number of commands per timeslice	Specifies the number of LTL commands an LTI or RepAgent Executor thread can possess before it must yield the CPU to other threads. Scripting name: DatabaseParameter_exec_cmds_per_timeslice

Property	Description
Save interval	Specifies the number of minutes that the Replication Server saves messages after they have been successfully passed to the destination data server. Scripting name: DatabaseParameter_save_interval
Partitioning rule	Specifies the partitioning rules (one or more) the DSI uses to partition transactions among available parallel DSI threads. Scripting name: DatabaseParameter_dsi_partitioning_rule
Use batch markers (Replication Server 15.0 and higher)	Controls the processing of function strings rs_batch_start and rs_batch_end. If use_batch_markers is set to on, the rs_batch_start function string is prepended to each batch of commands and the rs_batch_end function string is appended to each batch of commands. Set use_batch_markers to on only for replicate data servers that require additional SQL to be sent at the beginning or end of a batch of commands that is not contained in the rs_begin function string. Scripting name: DatabaseParameter_use_batch_markers
Dynamic sql	Specifies the mode (on, off, default) of the connection so that the replication definition allows the execution of dynamic SQL statements. Additional configuration parameters linked to dynamic SQL are only available when the mode is set to "on". Scripting name: DynamicSQL
Replication	Specifies whether or not transactions applied by the DSI are marked in the transaction log as being replicated. Scripting name: DatabaseParameter_dsi_replication
Serialization method	Specifies the method used to maintain serial consistency between parallel DSI threads when applying transactions to a replicate data server. Scripting name: DatabaseParameter_dsi_serialization_method
SQL data style	Formats datatypes (particularly date/time, binary, bit and money) to be compatible with: DB2 ("db2"), Lotus Notes ("notes"), SQL Anywhere, formerly Watcom SQL ("watcom") or SQL Remote ("sqlremote"). Scripting name: DatabaseParameter_dsi_sql_data_style
Text convert multiplier	Changes the length of text datatype columns at the replicate site. Use dsi_text_convert_multiplier when text datatype columns must expand or contract due to character set conversion. Replication Server multiplies the length of primary text data by the value of dsi_text_convert_multiplier to determine the length of text data at the replicate site. Its type is float. Scripting name: DatabaseParameter_dsi_text_convert_multiplier
Dump load	Enables coordinated dump when set to "on" at replicate sites only. Scripting name: DatabaseParameter_dump_load
Distributor write request limit	Specifies the amount of memory available to the Distributor for messages waiting to be written to the outbound queue. Scripting name: DatabaseParameter_md_sqm_write_request_limit
Subscription write request limit	Specifies the memory available to the subscription materialization or dematerialization thread for messages waiting to be written to the outbound queue. Scripting name: DatabaseParameter_sub_sqm_write_request_limit

Property	Description
LTI write request limit	Specifies the amount of memory available to the LTI or RepAgent Executor thread for messages waiting to be written to the inbound queue. Scripting name: DatabaseParameter_exec_sqm_write_request_limit
Parallel DSI	Provides a shorthand method for configuring parallel DSI threads. A setting of "on" configures these values: dsi_num_threads to 5 dsi_num_large_xact_threads to 2 dsi_serialization_method to "wait_for_commit" dsi_sqt_max_cache_size to 1 million bytes A setting of "off" configures these parallel DSI values to their defaults. Scripting name: DatabaseParameter_parallel_dsi
Replication DDL (Replication Server 15.0 and higher)	Specifies whether or not transactions are to be replicated back to the original database to support bidirectional replication. When set to "on", DSI sends set replication off to the replicate database, which instructs it to mark the succeeding DDL transactions available in the system log not to be replicated. Therefore, these DDL transactions are not replicated back to the original database, which enables DDL transaction replication in bidirectional MSA replication environment. Scripting name: DatabaseParameter_dsi_replication_ddl
Dynamic sql cache management (Replication Server 15.1 and higher)	Specifies the dynamic SQL cache for a connection. You can choose from one of the following values: <ul style="list-style-type: none"> • mru (default) – specifies that once dynamic_sql_cache_size is reached, the old dynamic SQL prepared statements are deallocated to give room for new statements. • fixed – specifies that once the dynamic_sql_cache_size is reached, allocation for new dynamic SQL statements stops. Scripting name: DatabaseParameter_dynamic_sql_cache_management
Dynamic SQL cache size (Replication Server 15.1 and higher)	Specifies an estimation of the number of database objects which can be used by SQL for a connection. This can be used to limit resource demand on a data server. Minimum value 1 is and Maximum value is 65,535.

Security Tab

The following properties are available on the Security tab:

Property	Description
Message confidentiality	Specifies whether Replication Server sends and receives encrypted data. If set to "required," outgoing data is encrypted. If set to "not required," Replication Server accepts incoming data that is encrypted or not. Scripting name: SecurityParameter_msg_confidentiality
Unified login	Specifies how Replication Server seeks to log in to remote data servers and accepts incoming logins. Scripting name: SecurityParameter_unified_login
Use security services	Specifies whether Replication Server can use security services. If use_security_services is "off," no security features take effect. This parameter can only be set by configuring Replication Server. Scripting name: SecurityParameter_use_security_services
Message integrity	Specifies whether data is checked for tampering. Scripting name: SecurityParameter_msg_integrity

Property	Description
Message origin check	Specifies whether the source of data should be verified. Scripting name: SecurityParameter_msg_origin_check
Message replay detection	Specifies whether data should be checked to make sure it has not been read or intercepted. Scripting name: SecurityParameter_msg_replay_detection
Message sequence check	Specifies whether data should be checked for interception. Scripting name: SecurityParameter_msg_sequence_check
Mutual authorization	Requires remote server to provide proof of identify before a connection is established. Scripting name: SecurityParameter_mutual_auth
Security mechanism	The name of the third-party security mechanism enabled for the pathway. Scripting name: SecurityParameter_security_mechanism

Transaction Tab

The following properties are available on the Transaction tab:

Property	Description
Disk affinity	Specifies an allocation hint for assigning the next partition. Enter the logical name of the partition to which the next segment should be allocated when the current partition is full. Scripting name: DatabaseParameter_disk_affinity
Packet size	Specifies the maximum size of a network packet. During database communication, the network packet value must be within the range accepted by the database. You may change this value if you have a System 10 or later SQL Server or Adaptive Server that has been reconfigured. Scripting name: DatabaseParameter_db_packet_size
Batch	Specifies how Replication Server sends commands to data servers. When batch is "on," Replication Server may send multiple commands to the data server as a single command batch. When batch is "off," Replication Server sends commands to the data server one at a time. Scripting name: DatabaseParameter_batch
Batch begin	Specifies whether a begin transaction can be sent in the same batch as other commands (such as insert, delete, and so on). Scripting name: DatabaseParameter_batch_begin
Command retry	Specifies the number of times to retry a failed transaction. The value must be greater than or equal to 0. Scripting name: DatabaseParameter_command_retry
Command batch size	Specifies the maximum number of bytes that Replication Server places into a command batch. Scripting name: DatabaseParameter_dsi_cmd_batch_size
Command separator	Specifies the character that separates commands in a command batch. Scripting name: DatabaseParameter_dsi_cmd_separator

Property	Description
Character convert	<p>The specification for handling character-set conversion on data and identifiers between the primary Replication Server and the replicate Replication Server. This parameter applies to all data and identifiers to be applied at the DSI in question.</p> <p>Scripting name: DatabaseParameter_dsi_charset_convert</p>
Check locks interval	<p>Specifies the number of milliseconds (ms) the DSI executor thread waits between executions of the rs_dsi_check_thread_lock function string. Used with parallel DSI function string. Used with parallel DSI.</p> <p>Scripting name: DatabaseParameter_dsi_commit_check_locks_intrvl</p>
Stop Unsupported Commands (Replication Server 15.0 and higher)	<p>When set to on, DIST suspends itself if a command is not supported by downstream Replication Server. When set to off, DIST ignores the unsupported command. Regardless of dist_stop_unsupported_cmd parameter's setting, Replication Server always logs an error message when it sees the first instance of a command that cannot be sent over to a lower-version Replication Server.</p> <p>Scripting name: DatabaseParameter_dist_stop_unsupported_cmd</p>
DSI bulk copy (Replication Server 15.1 and higher)	<p>Turns the bulk copy-in feature on or off for a connection. If dynamic_sql and dsi_bulk_copy are both on, DSI applies bulk copy-in. Dynamic SQL is used if bulk copy-in is not used.</p> <p>Scripting name: DatabaseParameter_dsi_bulk_copy</p>
Check locks times	<p>Specifies the number of times the DSI executor thread executes the rs_dsi_check_thread_lock function string before logging a warning message. Used with parallel DSI.</p> <p>Scripting name: DatabaseParameter_dsi_commit_check_locks_logs</p>
Max check locks times	<p>Specifies the maximum number of times a DSI executor thread checks whether it is blocking other transactions in the replicate database before rolling back its transaction and retrying it. Used with parallel DSI.</p> <p>Scripting name: DatabaseParameter_dsi_commit_check_locks_max</p>
Commit control	<p>Specifies whether commit control processing is handled internally by Replication Server using internal tables (on) or externally using the rs_threads system table (off).</p> <p>Scripting name: DatabaseParameter_dsi_commit_control</p>
Request stored procedure	<p>Turns on or off request stored procedures at the DSI of the primary Replication Server.</p> <p>Scripting name: DatabaseParameter_dsi_exec_request_sproc</p>
Fade out time	<p>Specifies the number of seconds of idle time before a DSI connection is closed. A value of "-1" specifies that a connection will not close.</p> <p>Scripting name: DatabaseParameter_dsi_fadeout_time</p>
Ignore underscore name	<p>When the transaction partitioning rule is set to "name," specifies whether or not Replication Server ignores transaction names that begin with an underscore.</p> <p>Scripting name: DatabaseParameter_dsi_ignore_underscore_name</p>
Keep triggers	<p>Specifies whether triggers should fire for replicated transactions in the database. Set off to cause Replication Server to set triggers off in the Adaptive Server database, so that triggers do not fire when transactions are executed on the connection. Set on for all databases except standby databases.</p> <p>Scripting name: DatabaseParameter_dsi_keep_triggers</p>

Property	Description
Number of transactions in log	Specifies the number of commands allowed in a transaction before the transaction is considered to be large. Scripting name: DatabaseParameter_dsi_large_xact_size
Number of threads	Specifies the number of parallel DSI threads to be used. The maximum value is 255. Scripting name: DatabaseParameter_dsi_num_threads
DSI isolated level (Replication Server 15.0 and higher)	Specifies the isolation level for transactions. The ANSI standard and Adaptive Server supported values are: 0 – ensures that data written by one transaction represents the actual data. 1 – prevents dirty reads and ensures that data written by one transaction represents the actual data. 2 – prevents nonrepeatable reads and dirty reads, and ensures that data written by one transaction represents the actual data. 3 – prevents phantom rows, nonrepeatable reads, and dirty reads, and ensures that data written by one transaction represents the actual data. NoteData servers supporting other isolation levels are supported as well through the use of the rs_set_isolation_level function string. Replication Server supports all values for replicate data servers. The default value is the current transaction isolation level for the target data server. Scripting name: DatabaseParameter_dsi_isolation_level
DSI bulk threshold (Replication Server 15.1 and higher)	Specifies the number of insert commands that, when reached, triggers Replication Server to use bulk copy-in. When Stable Queue Transaction (SQT) encounters a large batch of insert commands, it retains in memory the number of insert commands specified to decide whether to apply bulk copy-in. Because these commands are held in memory, Sybase suggests that you do not configure this value much higher than the configuration value for dsi_large_xact_size. Minimum: 1

Replication Server Route Properties

Replication Server route property sheets contain all the standard process connection tabs, along with the Route Options tab and the Security tab. The properties of these tabs are listed below.

Route Options Tab

The following properties are available on the Route Options tab:

Property	Description
User name	Specifies a name for the user. Scripting name: UserName
Password	Specifies a password for the user. Scripting name: Password
Next site	Specifies that the connection passes through an intermediate Replication Server site. Scripting name: NextSite
Disk affinity	Specifies an allocation hint for assigning the next partition. Enter the logical name of the partition to which the next segment should be allocated when the current partition is full. Scripting name: RouteParameter_disk_affinity
RSI batch size	Specifies a number of bytes sent to another Replication Server before a truncation point is requested. The range is 1024 to 262144. Scripting name: RouteParameter_rsi_batch_size

Property	Description
Save interval	Specifies the number of minutes that the Replication Server takes to save messages after they have been successfully passed to the destination Replication Server. Scripting name: RouteParameter_save_interval
Large message	Specifies route behavior if a large message is encountered. This parameter is applicable only to direct routes where the site version at the replicate site is 12.1 or earlier. Values are "skip" and "shutdown." Scripting name: RouteParameter_rsi_xact_with_large_msg
RSI synchronize interval	Specifies the number of seconds between RSI synchronization inquiry messages. The Replication Server uses these messages to synchronize the RSI outbound queue with destination Replication Servers. Values must be greater than 0. Scripting name: RouteParameter_rsi_sync_interval
RSI packet size	Specifies the packet size, in bytes, for communications with other Replication Servers. The range is 1024 to 8192. Scripting name: RouteParameter_rsi_packet_size
RSI fadeout time	Specifies the number of seconds of idle time before Replication Server closes a connection with a destination Replication Server. The value -1 specifies that Replication Server will not close the connection. Scripting name: RouteParameter_rsi_fadeout_time
Message confidentiality	Indicates whether Replication Server sends and receives encrypted data. If set to "required," outgoing data is encrypted. If set to "not required," Replication Server accepts incoming data that is encrypted or not encrypted. Scripting name: SecurityParameter_msg_confidentiality
Unified login	Specifies how Replication Server seeks to log in to remote data servers and accepts incoming logins. Scripting name: SecurityParameter_unified_login
Use security services	Specifies whether to use security services. If use_security_services is "off," no security features take effect. This parameter can only be set by configuring replication server. Scripting name: SecurityParameter_use_security_services
Message integrity	Specifies whether data is checked for tampering. Scripting name: SecurityParameter_msg_integrity
Message origin check	Specifies whether the source of data should be verified. Scripting name: SecurityParameter_msg_origin_check
Message replay detection	Specifies whether data should be checked to make sure it has not been read or intercepted. Scripting name: SecurityParameter_msg_replay_detection
Message sequence check	Specifies whether data should be checked for interception. Scripting name: SecurityParameter_msg_sequence_check
Mutual authorization	Specifies a remote server to provide proof of identify before a connection is established. Scripting name: SecurityParameter_mutual_auth

Property	Description
Security mechanism	Specifies the name of the third-party security mechanism enabled for the pathway. Scripting name: SecurityParameter_security_mechanism

Security Tab

The following properties are available on the Security tab:

Property	Description
Message confidentiality	Specifies whether Replication Server sends and receives encrypted data. If set to "required," outgoing data is encrypted. If set to "not required," Replication Server accepts incoming data that is encrypted or not encrypted. Scripting name: SecurityParameter_msg_confidentiality
Unified login	Specifies how Replication Server seeks to log in to remote data servers and accepts incoming logins. Scripting name: SecurityParameter_unified_login
Use security services	Specifies whether to use security services. If use_security_services is "off," no security features take effect. This parameter can only be set by configuring replication server. Scripting name: SecurityParameter_use_security_services
Message integrity	Specifies whether data is checked for tampering. Scripting name: SecurityParameter_msg_integrity
Message origin check	Specifies whether the source of data should be verified. Scripting name: SecurityParameter_msg_origin_check
Message replay detection	Specifies whether data should be checked to make sure it has not been read or intercepted. Scripting name: SecurityParameter_msg_replay_detection
Message sequence check	Specifies whether data should be checked for interception. Scripting name: SecurityParameter_msg_sequence_check
Mutual authorization	Specifies a remote server to provide proof of identify before a connection is established. Scripting name: SecurityParameter_mutual_auth
Security mechanism	Specifies the name of the third-party security mechanism enabled for the pathway. Scripting name: SecurityParameter_security_mechanism

Replication Server Logical Connection Properties

Replication Server logical connection property sheets contain all the standard data connection groups tabs along with the Connection Options tab, the properties of which are listed below:

Property	Description
Distribution	Specifies values for distribution: On or Off. Scripting name: Distribution
Logical name	Specifies a logical name for a logical connection. Scripting name: LogicalName
Materialization save interval	Specifies the materialization queue save interval. This parameter is only used for standby databases in a warm standby application. Scripting name: MaterializationSaveInterval
Replication minimum columns	Specifies whether Replication Server should send all replication definition columns for all transactions or only those needed to perform update or delete operations at the standby database. Replication Server uses this value in standby situations only when a replication definition does not contain a "send standby" option with any parameter. In the other case, Replication Server uses the "replicate minimal columns" or "replicate all columns" parameter in the replication definition. Scripting name: ReplicateMinimalColumns
Save interval	Specifies the number of minutes the Replication Server takes to save messages after they have been successfully passed to the destination data server. For more information, see the Replication Server Administration Guide. Scripting name: SaveInterval
Send standby columns	Specifies which columns Replication Server should send to the standby database for a logical connection and overrides the "send standby" option in the replication definition that tell Replication Server which table columns to send to the standby database. Scripting name: SendStandbyRepdefCols
Send truncate table	Specifies the truncated tables to send. Scripting name: SendTruncateTable

Replication Server Replication Definition and Article Properties

Replication Server replication definition and article property sheets contain all the standard article tabs, along with the RepServer Options tab, the details of which are listed below.

Property	Description
Enable replication	Enables replication for a table.
Primary table name	Specifies the name of the table in the primary database to be replicated. Scripting name: PrimaryTableName
Replication definition name	Specifies the name of the replication definition. Scripting name: ReplicationDefinitionName
Multiple owner	Specifies the mode of the table to replicate, so that both the table name and the owner name are considered for replication. Scripting name: MultipleOwner
Column replication type	Specifies the type of the column replication: "all columns", "minimal columns". Scripting name: ColumnReplication

Property	Description
Standby type	Specifies the type of standby: "all", "replication definition". Scripting name: StandBy
Dynamic SQL	(Replication Server 15.1 and higher) Specifies the mode (on, off, default) of the connection so that the replication definition allows the execution of dynamic SQL statements. Additional configuration parameters linked to dynamic SQL are only available when the mode is set to "on". Scripting name: DynamicSQL

Replication Server Database Replication Definition Properties

Replication Server replication definition property sheets contain all the standard publication tabs, along with the RepServer Options tab, the details of which are listed below:

Property	Description
Replicate DDL	Specifies whether the Data Definition Language (DDL) should be replicated. Scripting name: ReplicateDDL
Replicate table	Specifies the list of tables to replicate for the database. Scripting name: ReplicateTable
Replicate function	Specifies the list of stored procedures to replicate for the database. Scripting name: ReplicateFunction
Replicate transaction	Specifies the list of transactions to replicate for the database. Scripting name: ReplicateTransaction
Replicate system procedure	Specifies the list of stored procedures to replicate for the database. Scripting name: ReplicateSystemProcedure
Transaction set	Specifies the list of transactions to replicate for the database. Scripting name: TransactionSet

Replication Server Function Replication Definition Properties

Replication Server function replication definition property sheets contain all the standard procedure tabs, along with the RepServer Options tab, the details of which are listed below.

Property	Description
Standby type	Specifies the type of standby. You can choose from one of the following values: <ul style="list-style-type: none"> • All • Replication definition Scripting name: StandBy
Primary procedure name	Specify the name of the procedure. Scripting name: PrimaryProcedureName
Procedure option	Logs the execution of the stored procedure you are replicating either in the current database (log_current) or in the database where the stored procedure resides (log_sproc). Scripting name: ProcedureOption

Property	Description
Stored procedure option	<p>Specifies the options for the stored procedure. You can choose from one of the following values:</p> <ul style="list-style-type: none"> Function: Enables replication for a stored procedure associated with a function replication definition. Table: Enables replication for a stored procedure associated with a table replication definition. This option is equivalent to executing <code>sp_setreplicate</code> on the procedure. false : Disables replication for the stored procedure. <p>Scripting name: <code>StoredProcedureOption</code></p>
Function replication definition	<p>Specifies the name of the function replication definition.</p> <p>Scripting name: <code>FunctionReplicationDefinitionName</code></p>
Request (Replication Server 15.1 only)	<p>Specifies whether the function replication definition is a request one.</p> <p>Scripting name: <code>Request</code></p>

Replication Server Article Column Properties

Replication Server article column property sheets contain all the standard article column tabs, along with the RepServer Options tab, the details of which are listed below.

Property	Description
Column replication type	<p>Specifies an event for the replication of an article column. You can choose from one of the following values:</p> <ul style="list-style-type: none"> <code>always_replicate</code> <code>replicate_if_changed</code> <code>do_not_replicate</code> <p>Scripting name: <code>ColumnReplicationType</code></p>
Primary key	<p>Specifies whether the column is a primary key column.</p> <p>Scripting name: <code>IsPrimary</code></p>
Mandatory	<p>Specifies whether the column is mandatory in the replicated table.</p> <p>Scripting name: <code>IsMandatory</code></p>
Searchable	<p>Specifies whether the article column is searchable.</p> <p>Scripting name: <code>Searchable</code></p>
Data type	<p>Specifies the datatype of a column after a column-level datatype translation, but before any class-level translation and presentation to the replicated database.</p> <p>Scripting name: <code>PublishedDataType</code></p>
Identity (Replication Server 15.1 and higher)	<p>Specifies an article column computed from a table column when the source column attribute is specified. Only numerical typed column such as integer, numeric, or smallint can have this property.</p> <p>Scripting name: <code>Identity</code></p>

Replication Server Publication Properties

Replication Server publication property sheets include a specific Type property on the General tab. The name of the tabs for articles and procedures changes with the publication type.

Property	Description
Type	<p>Specifies the type of the publication. You can choose from one of the following properties:</p> <ul style="list-style-type: none"> • Undefined – no type is specified. Tabs for articles and procedures are renamed into the Articles & Replication Definitions tab and the Articles & Function Replication Definitions tab. • Database – specifies the creation of a database replication limited to the tables listed in articles. Tabs for articles and procedures are renamed into the Tables tab and the Procedures tab. • Publication – specifies the creation of an article and of a replication definition for each article. The validation of the publication is also generated. Tabs for articles and procedures are renamed into the Articles & Replication Definitions tab and the Articles & Function Replication Definitions tab. • Replication Definitions – specifies the creation of a replication definition for each article. Tabs for articles and procedures are renamed into the Replication Definitions tab and the Function Replication Definitions tab.

Replication Server Subscription Properties

Replication Server subscription property sheets contain all the standard subscription tabs, along with the RepServer Options tab, the details of which are listed below:

Property	Description
For new articles	Option for new articles
Subscribe for truncated tables	Specifies subscription for truncated tables.
Materialization	<p>You can choose from one of the following values:</p> <ul style="list-style-type: none"> • Incrementally • Without holdlock • Without materialization
Suspend replication	<p>Specifies the replication suspension. You can choose from one of the following values:</p> <ul style="list-style-type: none"> • Suspension – specifies the Data Server Interface (DSI) suspension for the replicate database after you change the subscription status. • Suspension at active replicate only – specifies the active database DSI suspension in a warm standby application.

Replication Server User Properties

Replication Server user property sheets contain all the standard user tabs, along with the RepServer Options tab, the details of which are listed below:

Property	Description
Permission	Specifies the permission granted to the user.
Password	Specifies the user password.

Replication Server Function String Properties

Replication Server function string property sheets contain all the standard event script tabs, along with the RepServer Options tab, the details of which are listed below:

Property	Description
Function string class name	Specifies the name of the function class. Scripting name: FunctionClass
Overwrite function class	Specifies whether or not you want to overwrite the function. Scripting name: FunctionClassOverwrite
Function string name	Specifies a name for the function. You can type one of the following values: rs_select, "rs_select_with_lock", "rs_get_textptr", "rs_textptr_init", "rs_writetext" events. Scripting name: FunctionString
Log type	Specifies the type of the log. You can choose from one of the following values: "use primary log", "with log" or "no log". Scripting name: Log
Scan template	Specifies the input template of a function string for the where clause in a Create Subscription command. Scripting name: ScanTemplate
Script output type	Specifies the type of the output script. You can choose from one of the following values: "language", "rpc", "writetext", or "none". Scripting name: ScriptOutput

Generating for Replication Server

You can generate Replication Server scripts (*.sql) for the replication process, and/or for the primary and target databases.

One SQL file is generated per server and contains all the orders for the server. The SQL file cannot be executed using an live database connection. You need to use the isql command to execute this SQL file. You can preview the script that will be generated for each object on the Preview tab of its property sheet.

Note: In order for you to connect properly to the replication process you must verify that:

- The code of the replication process object corresponds to the instance name of the Replication Server
- The User name and Password extended attributes correspond to the Replication Server login user name and password
- Each database has been generated using one of the Generate commands from its contextual menu

```
isql.exe -e -U USERNAME -P PWD -S SVR_TEST -i SVR_TEST.sql
```

1. Select **Tools > Replication Server version > Generate Scripts** to open the Generate dialog.

You can, alternately, right-click any database or replication process in the replication environment and select **Generate Scripts** to open the Generate dialog and generate a script for that element only.

2. Specify a directory in which to generate the scripts.
3. [optional] Select the **Check Model** option to verify the validity of your model before generation.
4. On the **Targets** tab, select the replication engine(s) that you want to generate for. This tab may not appear if you are generating for only a single replication process.
5. On the **Selection** tab, select the objects that you want to include in the generation. Use the sub-tabs to navigate between separate lists of object types. The selections you make here will affect the files that are available to select on the Generated Files tab.
6. On the **Options** tab, set generation options as appropriate. The following options are available:

Option	Description
Create <i>replication object</i>	Specifies to include create statements for this type of replication object in the generated script.
Drop <i>replication object</i> before creation if it already exists	Specifies to include drop statements for this type of replication object in the generated script before inserting the appropriate create statement.
Materialize subscriptions	Specifies how the data associated with subscriptions is to be materialized.

7. On the **Tasks** tab, select generation tasks as appropriate. The following tasks are available:

Task	Description
Execute generated scripts in Replication Server	Allows you to directly execute the generated scripts in Replication Server.
Execute generated scripts in RepAgent	Allows you to directly execute the generated scripts in RepAgent.

Note: To execute these tasks, you must have OpenClient isql installed on your machine. For information on how to install OpenClient isql for Replication Server, see the Replication Server documentation

8. Click OK to generate the scripts in the specified directory.

When the generation is complete, the Generated Files dialog opens listing the scripts, each of which you can open and review by selecting it and clicking Edit.

The Replication Server file is generated in the destination directory.

Reverse Engineering for Replication Server

You can reverse engineer an existing Replication Server definition into an ILM using two live connections:

- One for the consolidated database (or source database)
- One for the remote database

There are two ways for reverse engineering Replication Server objects:

- Reverse engineer a single replication process using the Reverse Engineering command from its contextual menu
- Reverse engineer several replication processes using the **Tools > Reverse Engineering Replication Server** command that allows you to select the replication processes to reverse engineer

The process of reverse engineering replication processes equals to retrieve Replication Server objects from the embedded Replication Server database (RDSS) via a live connection to create the corresponding ILM objects.

Reverse Engineering a Single Replication Process Object

You reverse engineer a single replication process object using the Reverse Engineering command from its contextual menu. It allows you to also reverse engineer all its related objects.

1. Open a replication process property sheet to define the data source, user name and password of the consolidated database in the Database Connection tab.

(If you do not define the data source, the Select a Data Source dialog box opens during the reverse engineering process).

2. If you have already defined the consolidated database and remote database, you can create a data connection from the consolidated database to the replication process and another data connection from the replication process to the remote database.

If you have not defined the consolidated database or remote database, PowerDesigner creates a default one for you during the reverse engineering process.

3. Right-click the replication process symbol and select Reverse Engineering from the contextual menu that is displayed.

If you have not defined the data source for the remote database in the Database Connection tab of the remote database property sheet, PowerDesigner asks you to select the data source of the remote database.

Once the reverse engineering is performed, PowerDesigner displays the Merge Models window to show you the differences between the reverse engineered model and the current model. You can decide whether you want to accept or not the created or modified objects.

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

The objects are added to your model. They are visible in the diagram and in the Browser. They are also listed in the Reverse tab of the Output window, located in the lower part of the main window.

Reverse Engineering Several Replication Processes

You reverse engineer several replication processes using the **Tools > Reverse Engineering Replication Server** command from the Menu bar.

You can customize your objects selection for reverse engineering.

1. For each replication process, define the data source, user name and password of the consolidated database in the Database Connection tab of the replication process property sheet.

(If you do not define the data source, the Select a Data Source dialog box opens during the reverse engineering process)

2. Select **Tools > Reverse Engineering Replication Server** .
3. Select the replication processes you want to reverse engineer.

For each remote database, PowerDesigner asks you to select the data source of the remote database.

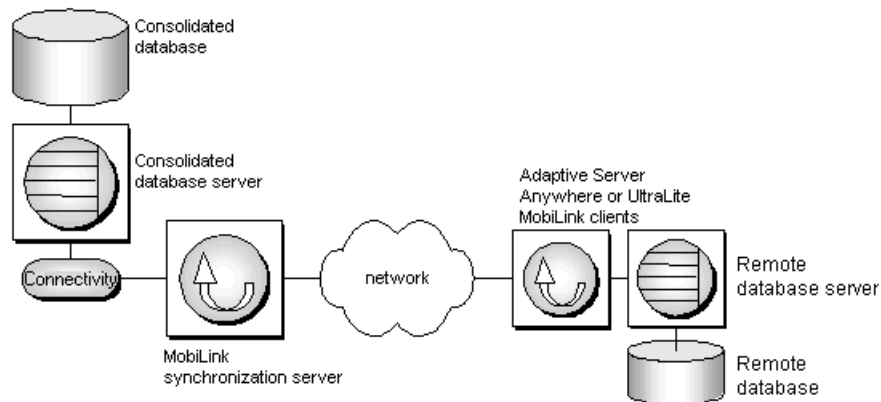
Once the reverse engineering is performed, PowerDesigner displays the Merge Models window to show you the differences between the reverse engineered model and the current model. You can decide whether you want to accept or not the created or modified objects.

Working with MobiLink

PowerDesigner provides full support for modeling all aspects of MobiLink 9.0 and 10.0 synchronization engines, including round-trip engineering. PowerDesigner's MobiLink objects are based on standard Information Liquidity Model (ILM) objects, but with additional properties.

MobiLink is a relational database synchronization system that enables the synchronization of a consolidated database with one or more remote databases.

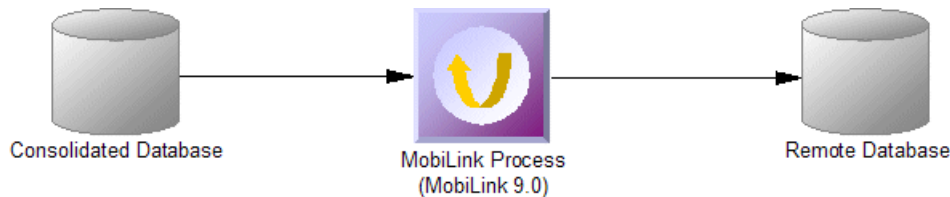
The following example illustrates a synchronization of data from the Consolidated database through the MobiLink synchronization server to the Remote database:



In the example:

- The Consolidated database – contains the central copy of all the data in the synchronization system, and can be any ODBC-compliant database, such as SQL Anywhere or Adaptive Server Enterprise.
- The MobiLink synchronization server – manages the synchronization and provides the interface between all MobiLink clients and the Consolidated database server. The synchronization can be one-way (download) or bi-directional (download and upload).
- The Remote database – is the database on the client side that contains a subset of the consolidated database. It can be a Sybase AS Anywhere or an UltraLite database, and a client can be a PC, a laptop, a pocket PC, or any other hand-held devices.

The following example illustrates how PowerDesigner represents a MobiLink synchronization in an Information Liquidity Model diagram:



The consolidated and remote databases are modeled in Physical Data Models (PDMs) attached to the database symbols, and the synchronization is represented as a MobiLink server replication process.

MobiLink Components

The main components of the MobiLink synchronization engine are modeled in PowerDesigner as follows:

MobiLink	PowerDesigner
Consolidated database – contains the central copy of all information in the synchronization system. It can be an SQL Anywhere, Adaptive Server Enterprise, Oracle DB2 or Microsoft SQL Server database.	Database (see MobiLink Consolidated and Remote Databases on page 138).
MobiLink server – manages the synchronization, and provides the interface between the MobiLink clients and the consolidated database server. The synchronization can be one-way or bi-directional.	Replication process with MobiLink type (see MobiLink Servers on page 137).
Remote database – runs on the client side, and can be installed on a PC, a laptop or a hand-held device, such as a Palm or a Pocket PC. Two database types are supported: UltraLite and SQL Anywhere.	Database (see MobiLink Consolidated and Remote Databases on page 138).
Publication – specifies the tables and columns to be synchronized.	Publication (see Publications (ILM) on page 29).
Article – specifies a table or a subset of the columns and rows in a table of the remote database.	Article (see MobiLink Articles on page 139).
Subscription – specifies the publication to use for synchronization.	Subscription (see MobiLink Subscriptions on page 140).
Connection script – specifies a synchronization script at connection level that matches a synchronization event, which occurred in the MobiLink server.	Event script (see MobiLink Connection Scripts/Table Scripts on page 141).
Table script – specifies a synchronization script at table level that matches a synchronization event, which occurred in the MobiLink server.	Event script (see MobiLink Connection Scripts/Table Scripts on page 141).
MobiLink user – specifies a user name and password to connect to a MobiLink server.	User (see MobiLink Servers on page 137).

Modeling a MobiLink Synchronization Definition

You can model a MobiLink synchronization by creating each synchronization object individually or you can use the Replication Wizard.

For detailed information about creating ILM objects, see [Building Information Liquidity Diagrams](#) on page 7.

The Replication Wizard guides you through the creation of a basic model containing a consolidated database that is synchronized with a remote database through a Mobilink server, and which manages at least one publication, a subscription and articles representing replicated data.

In order to model a MobiLink synchronization definition in PowerDesigner we recommend that you:

1. Identify your source databases to reverse engineer, or create a PDM to represent its structure.
2. Create an ILM and launch the Replication Wizard to create a basic synchronization of your databases through a MobiLink server (see [Creating a MobiLink synchronization with the Replication Wizard](#) on page 137).
3. Specify Mobilink options for the Mobilink server (see [MobiLink server properties](#) on page 138), remote database (see [MobiLink remote database properties](#) on page 138), subscription (see [MobiLink subscription properties](#) on page 140), articles (see [MobiLink article properties](#) on page 139), and connection and table scripts ([MobiLink connection script/table script properties](#) on page 141).
4. [optional] Specify any other necessary synchronization objects, such as additional remote databases, articles, and so on (see [Building Information Liquidity Diagrams](#) on page 7).

5. Generate your MobiLink synchronization definition (see [Generating for MobiLink](#) on page 142).

Creating a MobiLink Synchronization with the Replication Wizard

You can use the Replication Wizard to create a basic model of a Mobilink synchronization.

1. Select **Tools > Replication Wizard** to open the wizard, and click Next to open the Source Database Selection page.
2. Select or type the name of the database from which data are to be extracted, and click Next to open the Physical Data Model Selection page.
3. Select or create (via reverse engineering) a PDM to represent your source database, and then click Next to open the Replication Process Selection page.
4. Select or type the replication process to use, select a process type (MobiLink 9.0 or 10.0), and then click Next to open the Publication Selection page.
5. Create or select your publications, and click Next to open the Replicated Objects Selection page.
6. Select a data connection to access the remote database or create a new remote database, and then click Next to open the Remote Physical Data Model Selection page.
7. Select or create (via reverse engineering) a PDM to represent your remote database, and then click Next to open the Congratulations page.
8. Choose whether you want to update your remote database from the definition of articles, and then click Finish to close the wizard and create a basic synchronization corresponding to your selections.

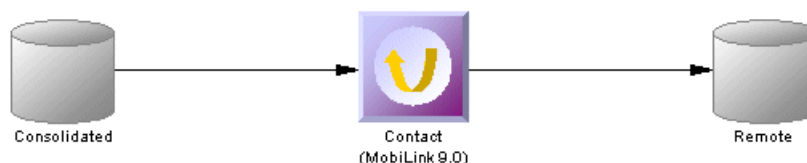
For detailed information about the Replication Wizard see "Replicating Objects Using the Replication Wizard" in the Building Information Liquidity Diagrams".

MobiLink Servers

PowerDesigner models MobiLink servers as standard replication processes with a MobiLink type and additional properties.

For information about creating and working with replication processes, see [Replication Processes \(ILM\)](#) on page 26.

In the following example, the Contact MobiLink server manages the synchronization, and provides the interface between the Consolidated database and the Remote database:



MobiLink Publications

A MobiLink publication belongs to the MobiLink server. It contains articles (see [MobiLink Articles](#) on page 139), and allows the identification of the data to synchronize in a remote database.

PowerDesigner models MobiLink publications as standard publications. For information about creating and working with publications, see [Publications \(ILM\)](#) on page 29.

Creating a MobiLink Server

You can create a MobiLink server in any of the following ways:

- Use the MobiLink tool in the diagram Palette. The type is automatically set to MobiLink.



- Select **Model > Replication Processes** to access the List of Replication Processes, and click the Add a Row tool. Open the replication property sheet, and select MobiLink from the Type list.
- Right-click the model or package in the Browser, and select **New > Replication Process** . Open the replication property sheet, and select MobiLink from the Type list.

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

MobiLink Server Properties

MobiLink server property sheets contain all the standard replication process tabs along with the MobiLink Options tab, the properties of which are listed below:

Property	Description
Download	Specifies the method for obtaining the data to synchronize from the consolidated database. You can choose from one of the following values: <ul style="list-style-type: none"> • Use timestamp column – compares values in a timestamp column. You must specify the name of the column that will contain this information. • Use shadow table – the shadow table tracks changes to records. You have to use triggers to manage the contents of the shadow table. • Use snapshot – retrieves all the records.
Use deleted column	Specifies the method for obtaining the list of deleted records. You can choose from one of the following values: <ul style="list-style-type: none"> • Use deleted column – you must specify the column name, data type, and the deleted value and non-deleted value designators. • Use shadow table – you have to use triggers to manage the contents of the shadow table. You must specify the timestamp column name in this table.
Row type column for conflict data tables	Specifies the row type column to be used for data conflicts.
User name column for data partitioning	Specifies the column name to store the synchronization user name for data partitioning.
Default script version	Specifies the default version name for synchronization scripts.
Always use stored procedure for scripts	Specifies the use of stored procedures for all synchronization scripts. This property should be used for databases (such as ASE 12.0) that do not support long strings for stored procedure parameters.
Enable error logging	Specifies the use of a script to log synchronization errors.

MobiLink Consolidated and Remote Databases

PowerDesigner models MobiLink consolidated and remote databases as standard databases. The consolidated database targets the Mobilink server, while remote databases (or MobiLink clients) are targeted by the Mobilink server, and have additional properties. The MobiLink server must have a subscription to a publication specified.

For information about creating and working with databases, see [Databases \(ILM\)](#) on page 18.

MobiLink Remote Database Properties

MobiLink remote database property sheets contain all the standard database tabs along with the MobiLink Options tab, and the Users tab, the properties of which are listed below:

MobiLink Options Tab

Property	Description
MobiLink synchronization users	<p>Specifies a list of users and passwords for the remote database. You can choose from one of the following values:</p> <ul style="list-style-type: none"> List of users – uses the list of user names and passwords that are specified in the Users tab. List of users from a file – Uses user names and passwords specified in a CSV file. You must specify the path to the file in the File name box or select a file by clicking the Select File tool to the right of the box. In the file, user name and password must be separated by a tab space, and there must be only one user name/password couple per line. Range of users – Specifies a range of users, without specifying user names and passwords. The user name and password will be generated from a template. You must specify the number of users, the first user number, the user name template and the password template. The default value <UserNo> represents the user number.
Global database ID	<p>You have to specify the following values:</p> <ul style="list-style-type: none"> First database ID – specifies a global database ID for the first remote database. Increment – specifies the increment to be used to generate other remote databases.
MobiLink server connection	<p>Specifies the connection information to the MobiLink server. You have to specify the following values:</p> <ul style="list-style-type: none"> Synchronization type – specifies the type of the synchronization for the user (Active Sync, http, https, tcpip). Host name – specifies the name of the machine where the MobiLink server runs for the user. Port number – specifies the port number of the MobiLink server for the user. Network parameters – specifies the network parameters for the user. For more information about the Create Synchronization User command, see the MobiLink documentation. Option clause – specifies extended options for subscriptions.

Users Tab

This tab is only available when:

- The remote database is targeted by a MobiLink server, in which you have specified a subscription to a publication.
- You have selected the List of users radio button in the MobiLink tab.

This tab lets you specify a list of user names and passwords to identify the people allowed to connect to the database.

MobiLink Articles

PowerDesigner models MobiLink articles as standard articles with additional properties.

For information about creating and working with articles, see [Articles \(ILM\)](#) on page 31.

MobiLink Article Properties

MobiLink article property sheets contain all the standard article tabs along with the Download and Upload tabs, the properties of which are listed below:

Download Tab

Property	Description
Enable download	<p>Specifies the method for obtaining the data to synchronize from the consolidated database. You can choose from one of the following values:</p> <ul style="list-style-type: none"> • Use timestamp column – compares values in a timestamp column. You must specify the name of the column that will contain this information. By default, the timestamp column name specified in the MobiLink Options tab of the replication process is used. • Use shadow table – the shadow table tracks changes to records. You have to use triggers to manage the contents of the shadow table. You can also specify the name of the shadow table. <ConsolidatedTable> represents the code of the consolidated table. • Use snapshot – retrieves all the records.
Enable download delete	<p>Specifies the method for obtaining the list of deleted records. You can choose from one of the following values:</p> <ul style="list-style-type: none"> • Use deleted column – you must specify the column name, data type, and the deleted value and non-deleted value designators. • Use shadow table – you have to use triggers to manage the contents of the shadow table. You must specify the name of the deleted shadow table (<ConsolidatedTable> represents the code of the consolidated table), along with the timestamp column name.
Enable data partitioning	Specifies the column name to store the synchronization user name for data partitioning.

Upload Tab

Property	Description
Upload	<p>Specifies the method for obtaining the data to synchronize to the remote databases. You have to select or specify the following values:</p> <ul style="list-style-type: none"> • Enable insert – Specifies the upload of inserted data in the remote database. • Enable update – Specifies the upload of updated data in the remote database. • Enable delete – Specifies the upload of deleted data in the remote database. • Enable conflict resolution – Specifies the resolution of conflicts during the upload. You must specify the name of the tables that will store new and old data rows and the resolution type, which can take one of the following values: <ul style="list-style-type: none"> • First win – first user who synchronized. • Last win – last user who synchronized. • User-defined – an administrator has to write a resolution script, as a conflict cannot automatically resolve.

MobiLink Subscriptions

PowerDesigner models MobiLink subscriptions as standard subscriptions with additional properties.

For information about creating and working with subscriptions, see [Subscriptions \(ILM\)](#) on page 36.

MobiLink Subscription Properties

MobiLink subscription property sheets contain all the standard subscription tabs along with the MobiLink tab, the properties of which are listed below:

Property	Description
MobiLink server connection	<p>Specifies the connection information to the MobiLink server. You have to specify the following values:</p> <ul style="list-style-type: none"> • Synchronization type – specifies the type of the synchronization for the subscription (Active Sync, http, https, tcpip). • Host name – specifies the name of the machine where the MobiLink server runs for the subscription. • Port number – specifies the port number of the MobiLink server for the subscription. • Network parameters – specifies the network parameters for the subscription. For more information about the Create Synchronization User command, see the MobiLink documentation. • Option clause – specifies extended options for subscriptions.

MobiLink Connection Scripts/Table Scripts

PowerDesigner models MobiLink connection scripts and table scripts as standard event scripts with additional properties.

For information about creating and working with event scripts, see [Event Scripts \(ILM\)](#) on page 39.

MobiLink Connection Script/table Script Properties

MobiLink connection script and table script property sheets contain most of the standard event script tabs along with the MobiLink tab, the properties of which are listed below:

General Tab Additional Property

Property	Description
Version	Specifies an event script version in order to identify the script to use among a set of scripts. When working with synchronization scripts, you can right-click a MobiLink server and use the commands to rename, duplicate or delete a script version.

MobiLink Tab

Property	Description
Language	<p>Specifies the language in which the script is written. You can choose from one of the following values:</p> <ul style="list-style-type: none"> • SQL – specifies an SQL script. • Java – specifies a Java script. You must type the fully qualified Java class name and the operation name. • .NET – specifies a .NET script. You must type the fully qualified .NET class name and the operation name.
Enable stored procedure	<p>Specifies the use of a stored procedure, along with a stored procedure name template having the following syntax: <code>pdm1_<Event>_<Process></code> or <code><Table>_% Version%</code></p> <p><Event> is the name of the event, <Process> is the code of the MobiLink server, <Table> is the code of the MobiLink article, and % Version% is the version of the script.</p>

Generating for MobiLink

PowerDesigner can generate a MobiLink synchronization definition. You have to perform the following sequence of actions:

- Generate the consolidated database (see [Generating the consolidated database](#) on page 142).
- Specify MobiLink generation options (see [Specifying MobiLink generation options](#) on page 143).
- Generate MobiLink synchronization scripts (see [Generating MobiLink synchronization scripts](#) on page 143).
- Register MobiLink users (see [Registering MobiLink users](#) on page 143).
- Generate the remote database (see [Generating a remote Sybase AS Anywhere database for a single MobiLink user](#) on page 143).

Generating the Consolidated Database

When you specify a MobiLink synchronization definition, you usually have to modify the consolidated database to perform actions, such as update or delete shadow tables, add stored procedures, add timestamp columns, etc.

In this case, before you can generate a consolidated PDM, you must first update it.

Updating the Consolidated Database

In order to generate or modify the consolidated database, you first need to update the PDM that represents the consolidated database:

- If the consolidated database structure does not exist yet, and you specify a MobiLink synchronization based on the remote database – right-click the consolidated database, and select Update Source Database to create the consolidated database structure from the definition of articles.

The Merge Models dialog box opens to let you verify the merge actions. The PDM attached to the consolidated database will be updated to reflect the structure of the consolidated database.

- If the consolidated database structure already exists – right-click the consolidated database, and select Update Consolidated Database PDM to update the consolidated database and add missing objects. This creates an archived PDM first then update the PDM.

If you want to cancel the changes, you can right-click the consolidated database, select Open Model, then select **Edit > Undo** from the menu bar.

Generating or Modifying the Consolidated Database

Depending on whether the consolidated database has already been generated or not, you can perform one of the following actions:

Note

You can also generate or modify the consolidated database from the Database menu of the consolidated database PDM.

To Generate the Consolidated Database for the First Time:

You generate the consolidated database for the first time in the following way:

Right-click the consolidated database, and select Generate Database, Generate Triggers and Procedures, and Generate Privileges.

To Modify an Existing Consolidated Database:

You modify an existing consolidated database in the following way:

Right-click the consolidated database, and select Modify Database.

Specifying MobiLink Generation Options

The following options are available on the Options tab of the Generation dialog box:

Option	Description
Drop publication before creation	Specifies the drop of a publication if it exists before recreating it.
Drop subscription before creation	Specifies the drop of a synchronization subscription if it exists before recreating it.
Delete old synchronization scripts	Specifies the deletion of all the previously specified connection scripts, table scripts and script versions from MobiLink system tables.

Generating MobiLink Synchronization Scripts

You can generate the connection scripts and table scripts contained in your MobiLink server.

1. Right-click the MobiLink server, and select Generate and Execute Scripts.
or
2. Right-click a MobiLink server, and select Generate Scripts to generate a SQL script, and then use another SQL interpreter to execute it.

Registering MobiLink Users

Once the MobiLink synchronization users are specified in the remote database property sheet, you have to register them in order to let them connect to a MobiLink server.

Right-click the MobiLink server, and select Register MobiLink Synchronization Users. A batch file will be generated and executed.

Generating a Remote Sybase AS Anywhere Database for a Single MobiLink User

In the ILM, articles reflect the structure of the remote database. When you specify articles and article columns in the MobiLink server, you have to make sure that the remote database tables and columns have the same name as those of the definition of articles.

If you specify a MobiLink synchronization from:

- The consolidated database – you have to create the remote database structure from the definition of articles.
- The remote database and you have modified articles (add or remove articles and/or columns) – you have to update the remote database structure, then you can generate or modify it.

You also have to generate or modify the publications and synchronization subscriptions.

To Update, Generate or Modify the Remote Database for a Single MobiLink User:

You update, generate or modify the remote database for a single MobiLink user in the following way:

1. Right-click the remote database.
2. Select Update Remote Database. The Merge Models dialog box opens to allow you to verify the merge actions.

or

Select Generate Database to open the Generation dialog box, and choose a generation type (see "Generating a database" in the Generating a Database from a PDM chapter in the *PDM User's Guide*).

or

Select Modify Database, to open the Modify Database dialog box, and choose a generation type (see "Modifying a database" in the Generating a Database from a PDM chapter in the *PDM User's Guide*).

To Generate the Publications and Subscriptions:

You generate the publications and subscriptions in the following way:

Right-click the remote database, and select Generate and Execute Scripts.

Note: If you do not have a remote Sybase AS Anywhere database yet, you can right-click the remote database, and select Create New ASA Database File. This also creates a data source for the database file.

Generating Remote Sybase AS Anywhere Databases for All MobiLink Users

If you need to deploy the remote Sybase AS Anywhere database to multiple devices, you have to create a Sybase AS Anywhere database for each MobiLink user. This database should contain the synchronization user, the synchronization subscription, and the global database ID.

You specify MobiLink users from the remote database property sheet (see [MobiLink remote database properties](#) on page 138).

1. Generate a remote Sybase AS Anywhere database as for a single MobiLink user (see [Generating a remote Sybase AS Anywhere database for a single MobiLink user](#) on page 143).
2. Right click the remote database, and select Create ASA Databases for all MobiLink Users. This creates a subdirectory and a specific Sybase AS Anywhere database for each MobiLink user.

The remote Sybase AS Anywhere database is used as a database template. If the remote Sybase AS Anywhere database is in a different location, you can use the RemoteDatabaseFile extended attribute to indicate the file path.

Generating a Remote UltraLite Database

If you need to generate a remote UltraLite database, you have to create a remote Sybase AS Anywhere database first, then convert it to an UltraLite database.

1. Generate a remote Sybase AS Anywhere database as for a single MobiLink user.
2. Right click the remote database, and select Convert ASA Database to UltraLite Database. This converts the remote Sybase AS Anywhere database to an UltraLite database .usm file. The UltraLite database will have the same structure and publications as the Sybase AS Anywhere database.

The remote Sybase AS Anywhere database is used as a database template. If the remote Sybase AS Anywhere database is in a different location, you can use the RemoteDatabaseFile extended attribute to indicate the file path.

Generating All MobiLink Script Files

You can generate all the MobiLink script files of your model at once.

This command does not perform any database generation or modification scripts. These types of actions can be performed from the consolidated or remote database contextual menu.

1. Select **Tools > MobiLink > Generate Scripts** to open the Generation dialog box.
2. Select an output directory, and click OK.

Reverse Engineering for MobiLink

You can reverse engineer one or more existing MobiLink synchronization definitions into an ILM using a live database connection for each of the consolidated and remote databases.

1. Open the Database Connection tab of each consolidated database property sheet linked to a MobiLink server, and specify the data source, login and password.
2. Right-click a MobiLink server, and select Reverse Engineering. If you have not previously specified the data source, login and password for the remote database on the Database Connection tab of its property sheet, PowerDesigner asks you to specify them.

or

Select **Tools > Reverse Engineering Replication Server**, and select the MobiLink servers you want to reverse engineer from the Selection dialog box. For each remote database, PowerDesigner asks you to specify the data source, login and password on the Database Connection tab of its property sheet.

Once the reverse engineering is over, PowerDesigner displays the Merge Models window to show the differences between the reverse engineered model and the current model. You can choose to accept or not the created or modified objects (for information about merging models, see the Comparing and Merging Models chapter in the *Core Features Guide*). The objects are added to your model, and are displayed in the diagram and in the Browser.

Index

A

- All Objects model option 89
- article 130
 - check model 95
 - Columns tab 32
 - create 31
 - define 31
 - Download tab in MobiLink 139
 - event script 33
 - modify from the Mapping Editor 16
 - object 7
 - properties 32
 - publication 31
 - where clause 33
- article column
 - check model 95
 - modify from the Mapping Editor 16
- ASE staging database generation 117
- auto completion list 33

B

- business process
 - check model 98
 - create 24
 - define 24
 - object 7
 - properties 24

C

- check model
 - article 95
 - article column 95
 - business process 98
 - connection group 99
 - control flow 105
 - data connection 99
 - data flow 103
 - data input 101
 - data output 101
 - data transformation action 102
 - data transformation diagram 100
 - data transformation task 100
 - database 93
 - end 104
 - event script (article) 96
 - event script (replication process) 97
 - flat file 98
 - ILM 92
 - package 93
 - procedure 96
 - process connection 99
 - publication 94
 - replication process 94
 - server 99
 - start 104

- subscription 94
- transformation control flow 104
- transformation control flow diagram 104
- transformation decision 105
- transformation process 99
- transformation synchronization 105
- transformation task execution 105
- user 95
- XML document 97
- column
 - create 33
 - define 33
 - object 7
 - properties 34
- compare ILM 105
- conceptual data
 - deprecated model container 43
- connect
 - database 21
- connection
 - Connection tab in data connection 119
 - Database Options tab in data connection 119
 - Security tab in data connection 119
 - Transaction Options tab in data connection 119
- connection group
 - add connection 39
 - check model 99
 - object 7
 - replication process 29
- consolidated database (MobiLink) 138
- control flow
 - check model 105
 - create 86
 - define 86
 - properties 87
- Convert Mappings to ETL Wizard 51
- Create Mapping between source and target objects tool 15
- ctrl+Q shortcut for detailed view 55

D

- data
 - replicate with Replication Server 112
 - synchronize with MobiLink 136
- data access application
 - migrated into project 43
- data access link
 - migrated into project 43
- data aggregation
 - aggregation column 67
 - create 64
 - Data Structure Columns tab 77
 - define 63, 67
 - properties 65
- data calculator
 - create 64
 - Data Structure Columns tab 77
 - define 63, 73

- properties 65
- data connection
 - add to connection group 39
 - check model 99
 - create 42
 - define 42
 - object 7
 - properties 42
 - read-only 42
 - read/write 42
 - write-only 42
- data connection group
 - create 38
 - define 38
 - properties 39
 - standby database 38
- data filter
 - create 64
 - Data Structure Columns tab 77
 - define 63, 69
 - properties 65
- data flow
 - check model 103
 - create 76
 - define 75
 - properties 76
- data input
 - check model 101
- data join
 - create 64
 - Data Structure Columns tab 77
 - define 63, 70
 - join column 70
 - properties 65
- data lookup
 - create 64
 - Data Structure Columns tab 77
 - database 71
 - define 63, 71
 - Lookup Keys 71
 - predefined 71
 - properties 65
 - script 71
- data merge
 - create 64
 - Data Structure Columns tab 77
 - define 63, 70
 - properties 65
- data output
 - check model 101
- data projection
 - create 64
 - criteria 73
 - Data Structure Columns tab 77
 - define 63, 73
 - properties 65
- data query execution
 - create 64
 - Data Structure Columns tab 77
 - define 63, 66
 - properties 65
 - script 66
- data replication 1
- data sort
 - create 64
 - criteria 69
 - Data Structure Columns tab 77
 - define 63, 68
 - properties 65
 - sort column 68
- data source 20, 110
- data source (administrator) 20
- data source administrator 110
- data split
 - create 64
 - Data Structure Columns tab 77
 - define 63, 70
 - properties 65
- data structure column
 - create 77
 - Data Structure Source Objects tab 78
 - define 77
 - properties 77
 - source expression 78
- Data Structure Mapping Editor 56, 77
- data transformation 1
 - model 48
- data transformation action (check model) 102
- data transformation diagram 1, 3
 - basics 54
 - check model 100
 - create 55
 - objects 55
- data transformation task
 - Actions tab 56
 - check model 100
 - define 54
 - Inputs tab 56
 - Outputs tab 56
 - Parameters tab 56
 - properties 55
- database
 - associate a PDM 20
 - check model 93
 - connect 21
 - create 19
 - Database Connection tab 20
 - define 18
 - display 21
 - object 7
 - Physical Data Models tab 20
 - properties 19
 - Replication Wizard from contextual menu 16
 - select objects to replicate 16
 - type 19
- database (remote)
 - MobiLink tab 138
 - Users tab 138
- database connection
 - access type 13
 - create in Mapping Editor 13
 - creation wizard 13
 - data source 110
 - Read-only 13

- Read/Write 13
- Write-only 13
- Database Connection tab
 - replication process property sheet 29
- database input
 - create 62
 - Data Structure Columns tab 77
 - Data Structure Source Objects tab 63, 78
 - define 61
 - properties 62
 - SQL Query tab 63
- database output
 - create 74
 - Data Structure Columns tab 77
 - define 73
 - properties 75
- database replication definition
 - properties in Replication Server 129
- Direct Connect 114
- display preference
 - ILM 90

E

- Enable links to requirements model option 89
- end
 - check model 104
- event script
 - article 33
 - create 39
 - define 39
 - object 7
 - properties 40
 - replication process 29
- event script (article)
 - check model 96
- event script (replication process)
 - check model 97
- execute query 21
- extended dependency 92
- extended generation
 - MobiLink 107
 - Replication Server 107
- extended model definition 107
 - MobiLink 91
 - Replication Server 91

F

- flat file
 - check model 98
 - Columns tab 26
 - create 25
 - define 25
 - object 7
 - properties 25
- flat file input
 - create 62
 - Data Structure Columns tab 77
 - define 61
 - properties 62
- flat file output

- create 74
- Data Structure Columns tab 77
- define 73
- properties 75
- fork 83
- function replication definition
 - properties in Replication Server 129
- function string
 - properties in Replication Server 131

G

- generate
 - ASE staging database 117
 - check model 107
 - for MobiLink 107
 - for Replication Server 107
 - ILM 108
 - ILM package 108
 - ILM to ILM 92
 - MobiLink options 143
 - MobiLink script files 144
 - remote database (MobiLink) 143
 - RepServer definition 117
 - select ILM object 108
 - Sybase AS Anywhere database 144
 - target 107, 109
 - template language 107
 - UltraLite database 144
- generation link
 - deprecated model container 43
- generation option 109
- generation task 109

I

- ILM 1
 - business process 24
 - compare 105
 - control flow 86
 - data flow 75
 - Data Structure Mapping Editor 56
 - data transformation task 54
 - diagram 1
 - environment 89
 - extended dependencies 92
 - merge 105
 - model 1
 - objects 55, 58
 - overview 1
 - properties 5
 - transformation decision 84
 - transformation end 85
 - transformation parameter 78
 - transformation process 59
 - transformation start 81
 - transformation task execution 81
 - XEM 5
- ILM mapping 9
- information liquidity diagram 1, 7
 - create 9
- Information Liquidity Model 1

- input flow
 - fork 83
 - join 83
- IQ_CLEAN_STAGING 118
- IQ_LOAD_STAGING 118

J

- join 83

L

- logical connection
 - Connection Options tab in Replication Server 127

M

- map
 - child object 12
 - parent object 12
- mapping
 - create with drag and drop 14
 - create with the Create Mapping between source and target objects tool 15
 - create with the Data Structure Mapping Editor 57
 - create with the Mapping Editor 14
- Mapping Editor
 - automatic mapping 14
 - child object 12
 - create database connection 13
 - parent object 12
 - properties pane 12
 - Source pane 9, 11
 - Target pane 9, 12
 - visualize replications 9
- Mapping Editor video 12
- merge ILM 105
- Mirror Activator 115
- MobiLink
 - article 135, 139
 - concepts 135
 - connection script 135, 141
 - consolidated database 135, 138
 - consolidated database server 135
 - create database structure 142
 - event script 141
 - extended model definition 91
 - generate 142
 - generate .SQL 107
 - generate consolidated database 142
 - generate script files 144
 - generate Sybase AS Anywhere 144
 - generate UltraLite database 144
 - generation options 143
 - model synchronization definition 136
 - modify consolidated database 142
 - object 7
 - publication 135
 - register user 143
 - remote database 135, 138
 - reverse engineering 144

- server 135, 137, 138
- subscription 135, 140
- synchronization scripts 143
- synchronization server 135
- synchronization user 135
- synchronize data 136
- table script 135, 141
- target 107
- update consolidated database 142
- user 138

model

- copy extended model definition 91
- create 4
- ILM 1
- options 89
- properties 5
- share extended model definition 91

models

- adding to project diagram 44

- move generation tasks up and down 109

N

- Name/Code case sensitive model option 89

O

- object
 - generate in ILM 108
 - ILM 7, 55, 58
- output flow
 - fork 83
 - join 83

P

- package
 - check model 93
- parameter for procedure 36
- PDM
 - associate with database 20
- procedure
 - check model 96
 - create 35
 - define 35
 - modify from the Mapping Editor 16
 - object 7
 - parameters 36
 - properties 35
 - publication 31
- process connection
 - check model 99
 - create 41
 - define 40
 - object 7
 - properties 41
 - replication process 40
 - transformation process 40
- project
 - creating 44
- project diagrams

- adding models 44
- rebuilding inter-model links 45
- publication
 - article 31
 - check model 94
 - create 29
 - define 29
 - modify from Mapping Editor 16
 - object 7
 - procedure 31
 - properties 30
 - properties in Replication Server 130
 - replication process 28
 - subscription 31

Q

- query
 - execute from database 21
 - execute from replication process 21
 - SQL 21

R

- rebuilding
 - inter-model links 45
- register user 143
- remote database
 - define 16
 - update 16
- remote database (MobiLink) 138
 - generate 143
 - modify 143
- RepAgent 114
 - configuration 114
 - generation 114
- RepConnector 115
- replicate data with Replication Server 112
- replication
 - Mapping Editor 9
- replication definition
 - properties in Replication Server 128
- replication process
 - check model 94
 - Connection Group tab 29
 - create 27
 - define 26
 - event script 29
 - object 7
 - properties 27
 - publication 28
 - Replication Wizard from contextual menu 16
 - subscription 28
 - type 27
 - user 28
- Replication Server
 - article 111, 130
 - components 111
 - connection 111, 119
 - consolidated database 111
 - consolidated database server 111
 - create the consolidated database 113

- create warm standby database 115
- data replication 112
- database 118
- database replication definition 111, 129
- Direct Connect 114
- extended model definition 91
- function replication definition 111, 129
- function string 111, 131
- generate .SQL 107
- generate script file 132
- generation target 107
- global process 111
- introduction 111
- logical connection 111, 127
- object 7
- publication 111, 130
- publication type 128, 130
- remote database 111
- RepAgent 114
- replication definition 111, 128
- reverse engineer 133, 134
- reverse engineering 133
- route 111, 125
- server 118
- subscription 111, 131
- synchronization server 111
- tool palette 111
- update the consolidated database 113
- user 111
- work with RepConnector 115
- Replication Wizard 16, 26
 - define database schema 16
 - define publication 16
 - define remote database 16
 - define source database 16
 - select a replication process 16
 - select objects to replicate 16
 - update remote database 16
- répliquer
 - tables de base de données 16
- reverse engineer 109
 - into existing ILM 109
 - replication process object 133
 - Replication Server 133, 134
 - select data source 110
 - several replication processes 110, 134
 - single replication process 110
- route
 - Route Options tab in Replication Server 125
 - Security tab in Replication Server 125

S

- script execution
 - create 64
 - Data Structure Columns tab 77
 - define 63, 65
 - insert script 65
 - properties 65
- server
 - check model 99
 - create 22

- define 22
- MobiLink Options tab 138
- object 7
- properties 22
- Server
 - RepServer Connection tab 118
- Show Detail 55
- source database
 - define 16
 - define schema 16
- Source pane toolbar 11
- SQL
 - query 21
- staging database 116
 - transfer data to Sybase IQ 118
- standby database 115
- start
 - check model 104
- subscription
 - check model 94
 - create 37
 - define 36
 - MobiLink tab 140
 - object 7
 - properties 37
 - properties in Replication Server 131
 - publication 31
 - replication process 28
 - Where Clause tab 37
- Sybase ASE staging database 116
- Sybase IQ 116
 - transfer data from the staging database 118
- synchronization script 143
- synchronize
 - data with MobiLink 136

T

- target
 - generate 107
 - MobiLink 107
 - Replication Server 107
- Target pane toolbar 12
- task execution
 - create 83
- trace mode 107
- transformation
 - create with the Convert Mappings to ETL Wizard 51
 - create with the Transformation Wizard 49
 - data format 77
 - Data Structure Mapping Editor 56
- transformation action
 - Data Structure Columns tab 77
 - properties 65
- transformation control flow
 - check model 104
 - define 58
 - properties 59
 - Task Executions tab 59
- transformation control flow diagram 1, 4
 - check model 104
 - create 58

- objects 58
- transformation decision
 - check model 105
 - create 84
 - define 84
 - properties 85
- transformation end
 - create 86
 - define 85
 - properties 86
- transformation parameter
 - create 78
 - define 78
 - properties 79
- transformation process
 - check model 99
 - create 60
 - Data Transformation Tasks tab 61
 - define 59
 - object 7
 - properties 60
 - Transformation Control Flows tab 61
- transformation start
 - create 81
 - define 81
 - properties 81
- transformation synchronization
 - check model 105
 - create 83
 - define 83
 - properties 84
- transformation task execution
 - check model 105
 - create 82
 - define 81
 - properties 82
 - Tasks tab 82
- transformation wizard 49
- type
 - database 19
 - replication process 27

U

- user
 - check model 95
 - create 37
 - define 37
 - object 7
 - properties 38
 - properties in Replication Server 131
 - replication process 28

V

- validate ILM 92

W

- Web service input
 - create 62

- Data Structure Columns tab 77
- Data Structure Source Objects tab 78
- define 61
- properties 62
- where clause 7
- Where Clause tab
 - article 33
 - subscription 37

X

- XEM
 - ILM 5
 - MobiLink 142
 - Replication Server 132
- XML document

- check model 97
- create 23
- define 23
- object 7
- properties 23
- XML input
 - create 62
 - Data Structure Columns tab 77
 - Data Structure Source Objects tab 78
 - define 61
 - properties 62
- XML output
 - create 74
 - Data Structure Columns tab 77
 - define 73
 - properties 75

