Power*Architect User Guide

COLLABORATORS			
	TITLE : Power*Architect User Guide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		Last Modified	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Introduction

1.1 About Power*Architect

Power*Architect from SQL Power Group is a visual data modeling tool designed for data architects, DBAs, analysts, designers, and other professionals. Quickly design every aspect of your data model using diagrams and a hierarchical view of your model structure. Your data model remains platform-independent, allowing you to maintain a single database schema that works well with multiple database platforms.

Power*Architect is also well-suited to data warehouse and data mart design. You can open multiple source databases concurrently, then drag and drop objects (such as schemas, tables, and columns) into Power*Architect's data modeling playpen to create a new model. After fine-tuning the data model in the playpen, you can forward engineer the data model into new database on platforms such as Oracle, SQL Server, DB2, PostgreSQL, or MySQL. Power*Architect also creates ETL (Extract, Transform, Load) procedures you can use with Pentaho's popular open source Kettle ETL tool to populate the new database.

Power*Architect provides you with a variety of tools to view and compare data structures and mappings. For example, you can compare the structure of any two databases, highlighting the differences and similarities and generating the required DDL statements to synchronize them. You can also create a visual mapping report listing the source tables used in your data model, or create an easy-to-read profile summarizing the data contained in a database.

Whether you're building or maintaining a data model, Power*Architect provides the tools to help you design your model in a fraction of the time.

1.2 About This Guide

The Power*Architect User Guide provides step-by-step instructions for using Power*Architect and covers all of Power*Architect's features and capabilities.

The guide assumes you are familiar with basic database operations and terminology (please refer to Chapter 13 for a list of some common database terms). If you plan to use Kettle jobs, the guide assumes you have some knowledge about ETL (Extract, Transform, Load) procedures. If you are looking for more information about ETL, two books you may want to try are *Building the Data Warehouse* by W. H. Inmon and *The Data Warehouse Toolkit: The Complete Dimensional Modeling* by Ralph Kimball and Margy Ross.

1.3 Power*Architect Licensing and Distribution

Power*Architect is free and open source software, meaning that the source code is readily available. Everyone is free to inspect, comment on, and modify Power*Architect's source code. Anyone who modifies the program code is invited (but not required) to contribute their changes back to the project. All contributions are subject to review and acceptance by the Power*Architect team. We always welcome suggestions from Power*Architect users, in the spirit of making the application easier to use and providing the features that matter the most to you.

Power*Architect is distributed to the public under the GNU Public License Version 3. A copy of the license is available in Section of this User Guide.

Chapter 2

Getting Started

To get started using Power*Architect, begin by reading Section 2.6 . This section gives you a quick introduction to the main Power*Architect areas, the playpen and the database tree. You may then want to work through the hands-on Section 2.7 . This example shows you how to create a simple data model, set up a database connection, and forward engineer your model to any database you choose.

Power*Architect contains many features, and you may choose to use some or all of these features depending on what you are trying to accomplish. Please see the following sections for an overview of typical activities you would perform with Power*Architect.

- Section 2.1

- Section 2.2

- Section 2.3

- Section 2.4

- Section 2.5

2.1 About Data Models

As a general guideline, you would typically follow these steps to create and use a data model:

1. Create a data model using the playpen. You can do this by creating a data model from scratch, reverse engineering an existing database, or by using a combination of these two methods.

   For more information, see:

   - Chapter 3
   - Chapter 7

2. Forward engineer your data model to create the data structure in a new database. To use forward engineering, you must first set up a database type and connection for the target database.

   For more information, see:

   - Chapter 8
   - Chapter 5

3. Use a Kettle job to copy data into your new database.

   For more information, see:

   - Section 10.2

2.2 About Data Structure Analysis

You can use Power*Architect's many data structure analysis features to view information about a data model or database. You can:

- Compare two data models to view the differences and similarities. Generate and run a SQL script to update an older database to match a newer data model.

  For more information, see:

  - Section 9.1

- View a profile of the data in a database table.

  For more information, see:

  - Section 9.4

- Create a report listing the source tables used for the tables in your Power*Architect data model.

  For more information, see:

  - Section 9.5

- Export the source-to-target column mappings between a source database and your Power*Architect data model.

  For more information, see:

  - Section 9.6

2.3 Copying and Transforming Data

Power*Architect provides two methods (one basic, one complex) for copying data between databases. You can:

- Copy data across database platforms to create a verbatim copy of an existing database.

  For more information, see:

  - Section 10.1

- Create multiple transformations based on a data model.

  For more information, see:

  - Section 10.2

2.4 About Advanced Features

Power*Architect contains a general-purpose SQL query tool that that allows you to work at the raw SQL command level. This feature is meant as a convenience for advanced users. It is not necessary to use this tool during routine data modeling, profiling, or database comparison activity.

For more information, see:

- Chapter 11

2.5 About System Preferences

You can set project and user preferences for Power*Architect.

For more information, see:

- Chapter 6

2.6 Understanding the Power*Architect User Interface

Each data model you create in Power*Architect is saved as a separate project. When you open a project, the data model information is shown in Power*Architect's two main areas: the database tree and the playpen.
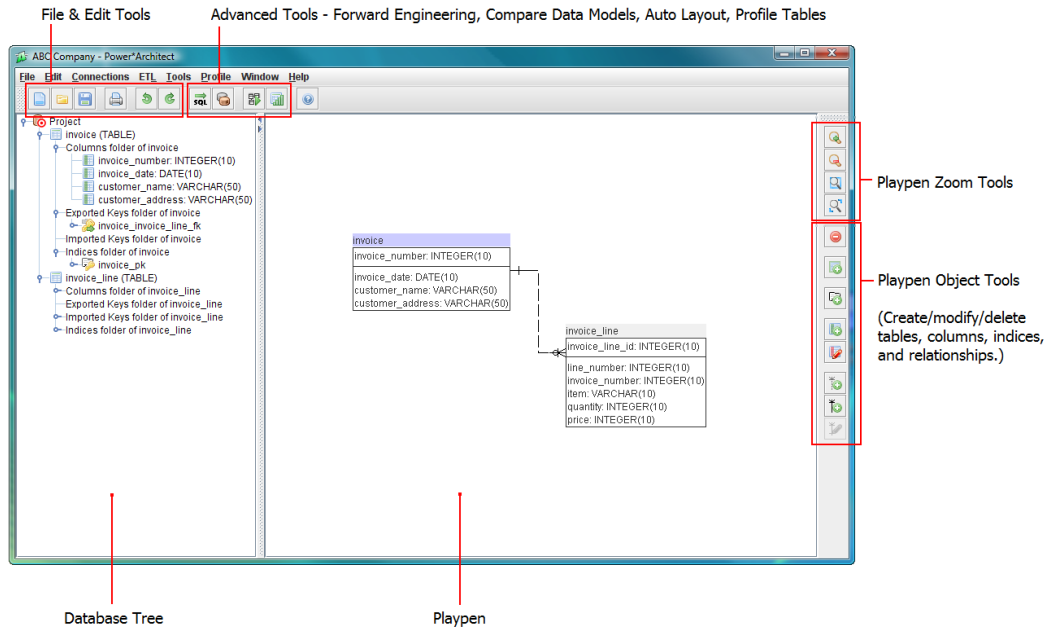
2.6.1 About the Database Tree

The database tree contains a hierarchical view of your project. The hierarchy includes:

- The objects in your data model (tables, columns, keys, indices, etc.).

- The database connections you've added to the project.

- Any objects you've obtained through reverse engineering an existing database. You can drag these objects into the playpen to add them to the data model you're building in Power*Architect. (Large objects may take some time to load in the playpen.)

You can expand the branches in the tree to view objects and can often right-click an object to perform actions. The following icons are used in the database tree to identify the object type.

 Database

 Catalogue

 Schema

 Owner

 Table

 Column

 Primary Key

 Exported Key

 Imported Key

Index

Unique Index

2.6.2 About the Playpen

The playpen is your main work area in Power*Architect, where you create and modify your data model. You can use the playpen to experiment and manipulate tables and relationships. Your changes are not saved until you decide to save them.

Your data model can includes tables, columns, indices, and relationships. You can create these objects in Power*Architect or obtain them by reverse engineering an existing database. For more information on working in the playpen, see Chapter 3 .

2.6.3 Using Power*Architect on Different Operating Systems

Power*Architect supports multiple operating systems, such as Windows, Macintosh and Linux. Power*Architect works the same on all operating systems, with a few minor exceptions:

- On Windows and Linux, CTRL is used as the accelerator key. On Macintosh, CMD is the accelerator key.

- On Windows and Linux, the Power*Architect menu bar is shown below the Power*Architect title bar. On Macintosh, the menu bar is shown at the top of the Power*Architect window.

2.7 Example - Creating a Data Model

This section will show you how to set up a simple database "from scratch", just to get you started using the tools, without modifying any live data. If you follow the example literally, you will create a trivial "customer and orders database".

Important: You must create the target database needed in this example. You can use standard vendor-specific database tools to create the database.

2.7.1 Setting Up Databases

1. Setup Driver. Select File->User Preferences if you are using Windows and Architect->Preferences if you are using Macintosh. Then select the JDBC Drivers tab. Select the database connection type you wish to use from the list on the left. If there is already a driver for the connection type you wish to use, click OK and go on to the next step. Otherwise, click the Add JAR button, navigate to where you have the driver Jar file installed, and click OK.

2. Create a Connection. In the Database Tree section of the main window, right click and choose Add Source Connection->New Connection. For this example you can use a name like SampleDB for the Connection Name. Then select a data type you wish to use, which should be the driver you set before. Then fill the hostname, port, database, file if asked. The JDBC URL will automatically generate when you fill those blanks, so you don't have to type it an extra time. Fill in the Username and password (which is set on the server machine).

2.7.2 Designing a Database

You are now ready to design some tables. For this example, we will create the Customer and Orders table shown here.

1. Click the New Table icon at the right side. The cursor will change to a crosshair. Move the cursor near the left of the Playpen area, and click. A "New Table" will appear.

2. Also, the Table Properties Dialog will appear. Rename this table to Customers.

3. Click the Insert Column icon, and a column property window will appear for the new column. Rename the column to customer_id and make it part of the primary key.

4. Insert additional columns for Firstname, Lastname, Address, City, Province, Country Code [1] and Postal Code. The table should look something like the following:
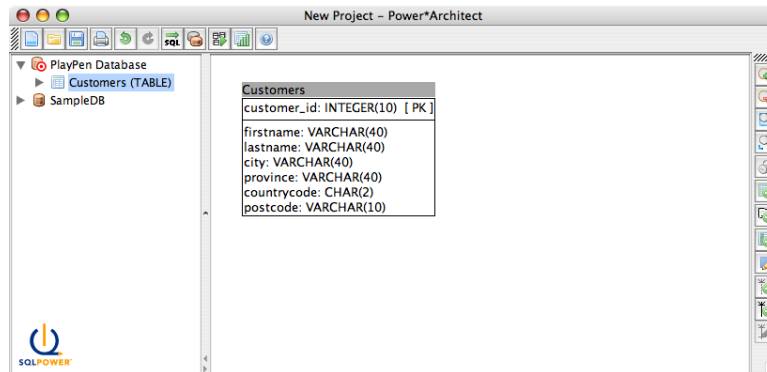
5. Create a second table, and name it Orders.

6. Create columns named order_id (in the primary key), Quantity, Total Amount, and customer_id. Your project should now look something like the following:
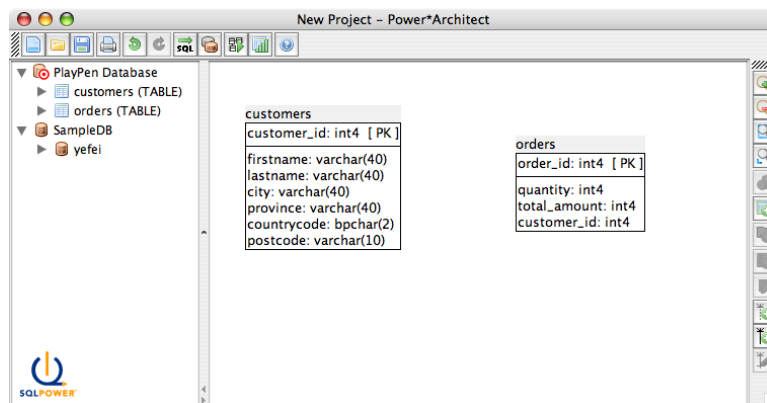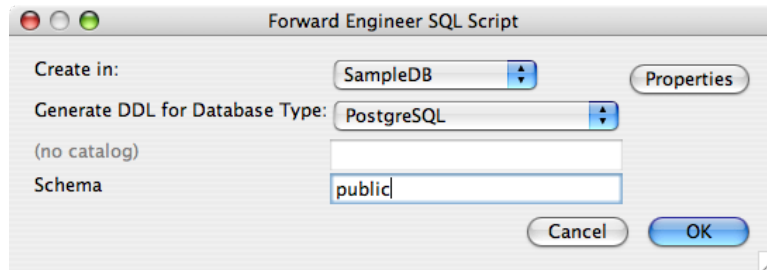
7. We need a relationship between these tables. An order should have a foreign key that refers to the customer. Click the "New Non-Identifying Relationship" icon. Select the Customers table, then the Orders table, and a link will be drawn as shown. Click this link and the keys that take part in the relationship will be highlighted in red.
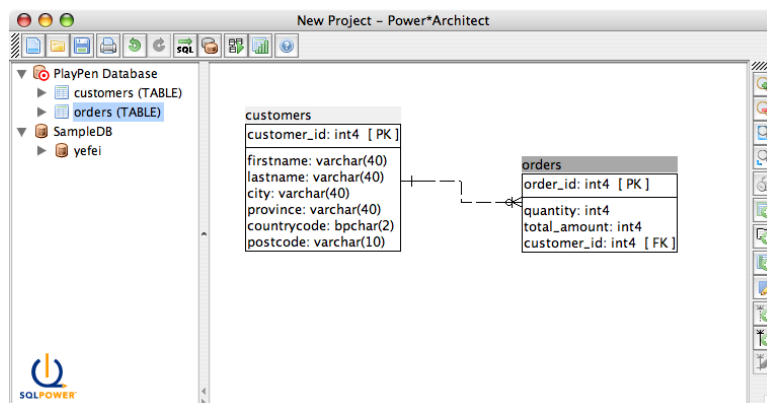
2.7.3 Forward Engineer

1. If you're happy with the database layout (you can always change it later), it's time to create the database. Click on the Forward Engineer button. You should see a window similar to the following:

---

[1] There is an ISO standard, ISO-3166, which specifies a two-letter code (and a rarely-used three letter code) for every country of the world: us for the United States, ca for Canada, and so on. Some developers like to use these in the Country field of a database, as we are doing here.

2. Set the "Create in" database to be the source connection we defined earlier. Set the database type to be the type that was set in the user preferences. Fill in the remaining fields based on the database type that was selected and press ok. You should see a window similar to:
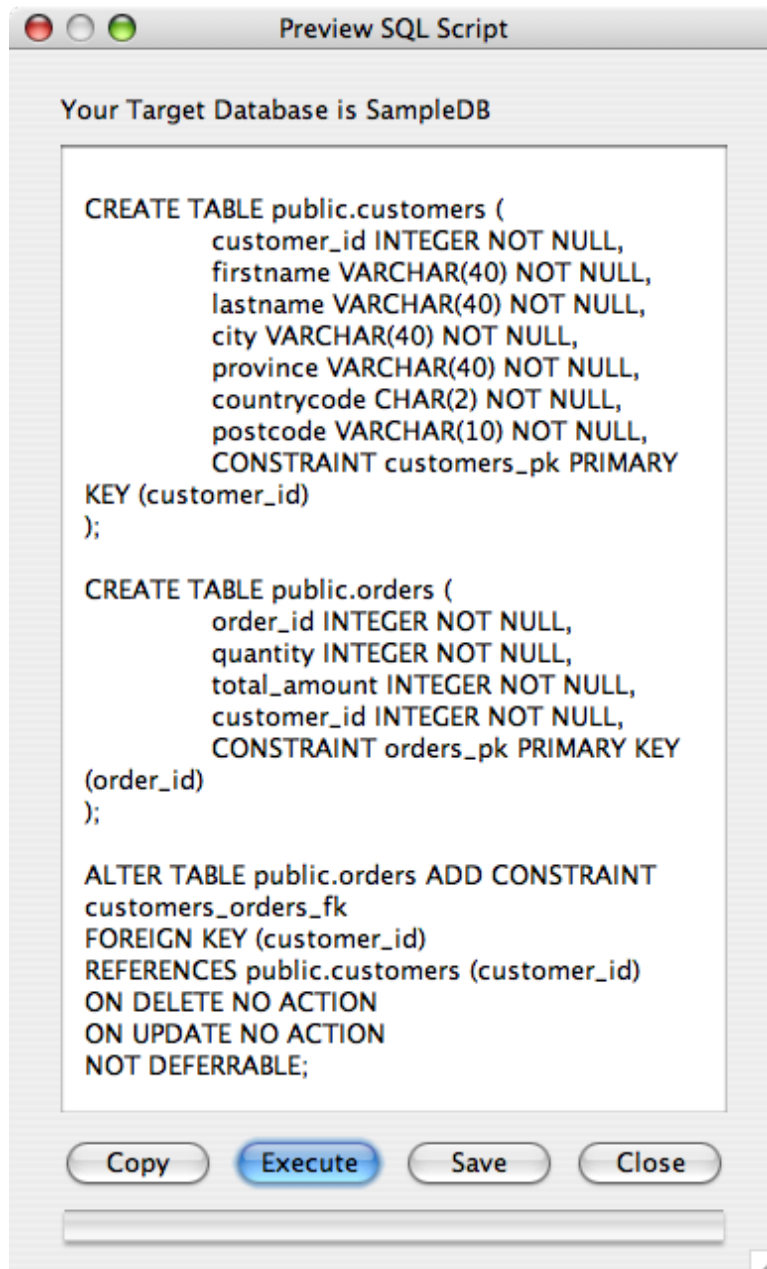
3. If this looks plausible, click Execute, and the tables and their relationship will be created. Congratulations! You have now created a simple database using the visual tools in Power*Architect.

2.7.4 Comparing Data Models

Suppose that after using this database, you realize that there should be a "shipping amount" field in the Order table (we never promised this would be completely realistic example).

1. Select the Order table by clicking on its title.

2. Click the Insert Column field and, as before, rename the New Column, this time to Shipping_Amount. Change its type to Decimal with precision 10 and scale 2.

3. Now we need to compare two different Data Models, the original database and the current project. Click the Compare DM icon. Set the "Older" to Physical Database SampleDB (you may need to change the Schema to Public). Set the "Newer" to "Current Project" (since it is now newer than the database you created in Step 6). Set the output format to SQL.

   PS: swap button can help you easily swap the newer the the older.
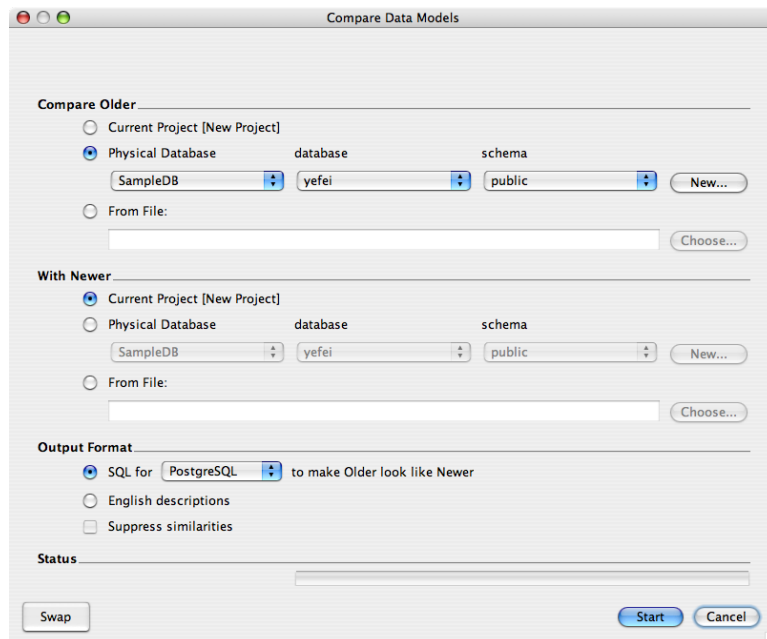
4. Click Start. You should see the SQL Preview window again, but this time with just an ADD for the column you just added:
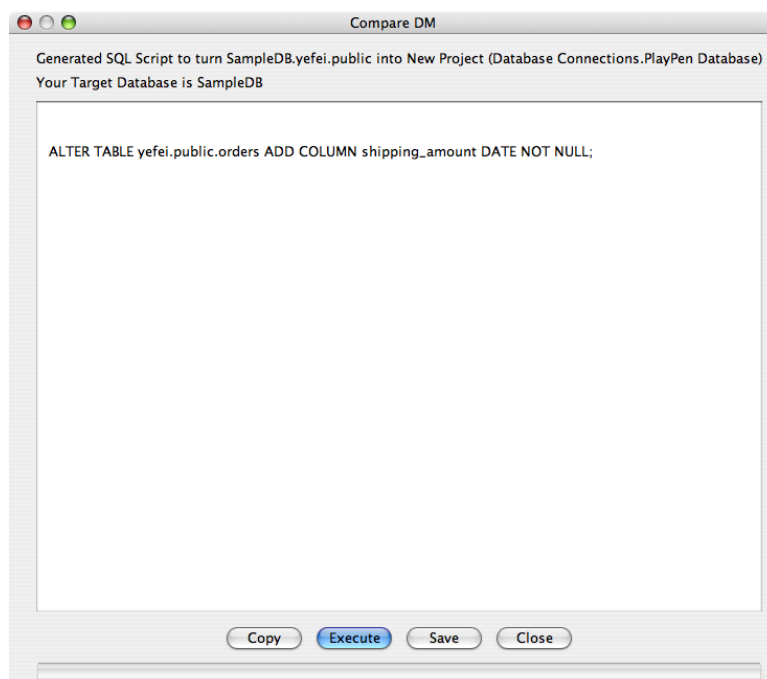
5. Click Execute, and the new column will be added to your database table.

When you exit the program, it will ask to save your project. Since you might want to alter this in future, to experiment with some of the other tools without damaging any live data, you may wish to save the Project file.

The remainder of this document provides a more comprehensive explanation of the various functions that Power*Architect offers.

Chapter 3

Creating a relational Data Model

Use the Power*Architect playpen to create a data model diagram that includes tables, columns, indices, and relationships. Before you begin, be sure to read Chapter 2 , which explains how to use the playpen and the database tree.

When you create a data model in Power*Architect, the model is saved in its own project. The project contains the data model diagram in the playpen and the database tree. You can have multiple projects (and therefore multiple data models) open in Power*Architect at once. Each project opens in a separate window.

3.1 Working with Tables

3.1.1 Creating New Tables

To create a new table:
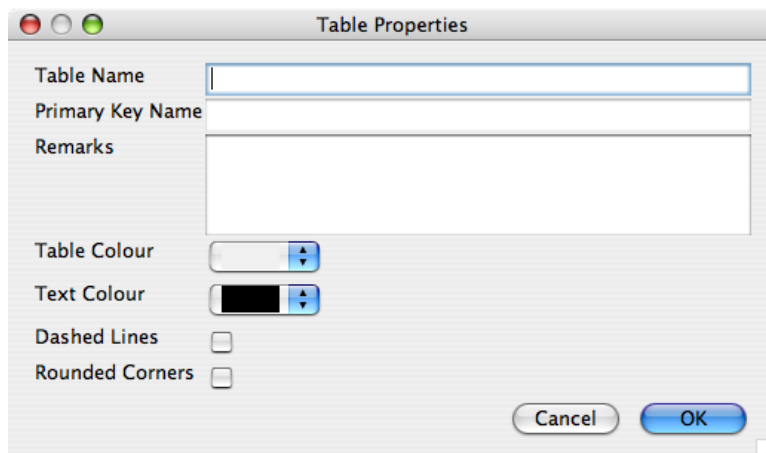
1. Click ![icon] in the side toolbar. The cursor changes to a +.

   Note: To cancel creating a new table, press ESC.

   Alternate methods:

   - Right-click in the playpen, then click New Table.
   - Place the cursor over the playpen, then press T.

2. Click in the playpen where you want to create the table. The Table Properties dialog box appears.

3. Enter the following information:

In this field ...	Do this ...
Table Name	Enter a table name.
Primary Key Name	You cannot enter a primary key name until you have added columns to the table and defined the primary key. The primary key name is used when you forward engineer the data model. For more information, see Section 3.2.1 . Note: Primary key names are not used when forward engineering to a MySQL database (MySQL does not support custom primary key names).
Remarks	Enter a description of the table. When you forward engineer the data model, the remarks will be included as comments in the database.

4. Click OK.

3.1.2   Modifying Tables

To modify a table:

- Click a table in the playpen, then click in the side toolbar.

  The Table Properties dialog box appears.
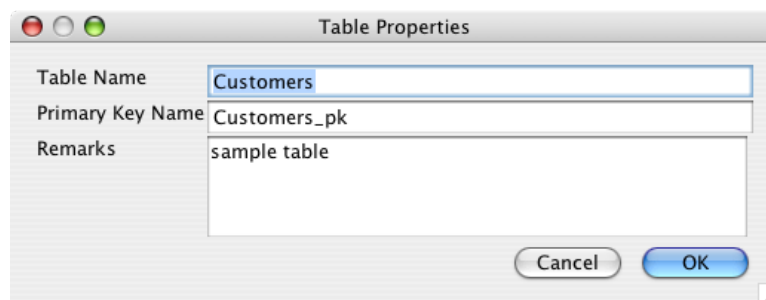
Alternate methods:

- Right-click a table in the playpen, then click Table Properties.

- Click a table in the playpen, then press Enter.

- Modify the table properties as required. For a description of the properties, see Section 3.1.1 .

- Click OK.

3.1.3   Copying Tables from external documents

To copy tables from external documents (for example spreadsheets) into Power*Architect you can select & copy the table names from your external documents and paste (CTRL + V) them into the Playpen.
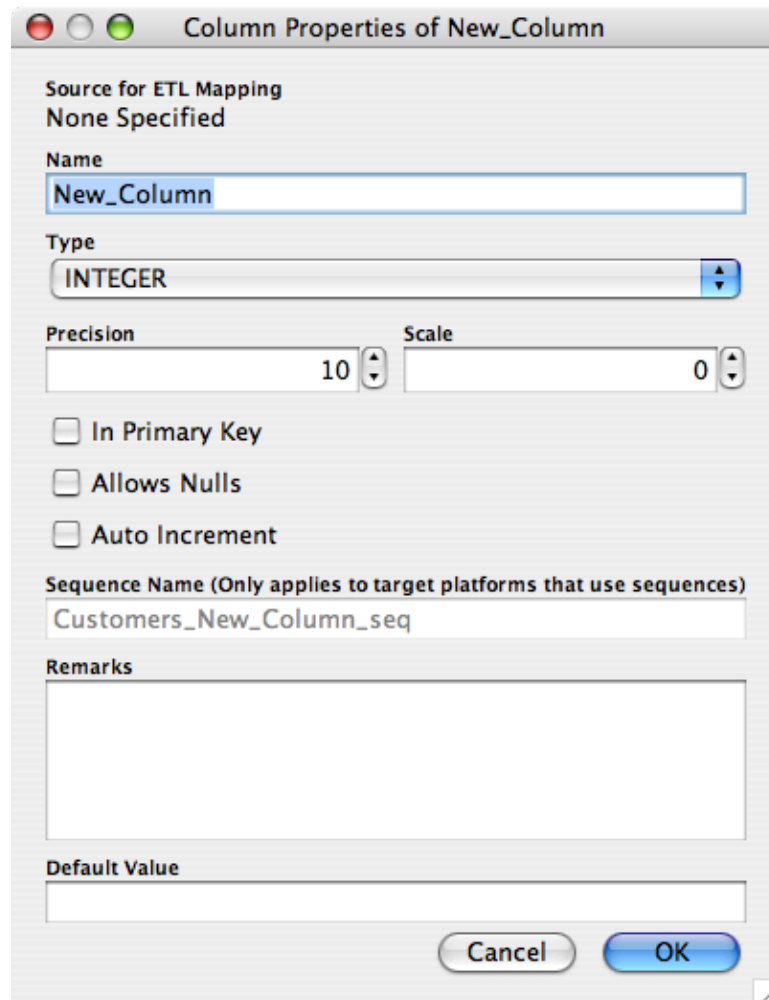
3.2   Working with Columns

3.2.1   Creating New Columns

When you create a column, you can choose where the new column is inserted in the table.

To add a column to a table:

1. Click a table in the playpen. The location you click determines where the column will be inserted in the table.

   - If you click the table name or if the table does not contain any columns, the new column is added to the end of the column list.
   - If you click an existing column, the new column is added above the selected column.

2. Click ![icon] in the side toolbar. The Column Properties dialog box appears.

   Alternate methods:

   - Right-click a table, then click New Column.
   - Click a table, then press C.

3. You can enter the following information:

In this field ...	Do this ...
Name	Enter the column name.
Type	Select the type of data the column holds.
Precision	Set the data precision.
Scale	Set the scale.
In Primary Key	Select the check box if the column is in the primary key.
Allows Nulls	Select the check box if the column handles null information.
Auto Increment	Select the check box if auto increment is allowed.

In this field ...	Do this ...
Sequence Name	When Power*Architect creates a table in a database platform that uses sequences (such as Oracle or PostgreSQL), Power*Architect creates a sequence for each auto-increment column in the table. Enter the name to use for the sequence. Note: This option is only available if you have selected the Auto Increment option for the column.
Remarks	Enter comments about the column. When you forward engineer the data model, the remarks will be included as comments in the database.
Default Value	Enter a default value for the column. Note: Power*Architect does not validate the default value, so ensure you use a valid format. The following examples show valid formats for different data types: • 'word' for a String • {d '2007-12-10'} for a Date • {t '5:38:00'} for a Time • {ts '2007-12-10 5:38:00'} for a Timestamp

4. Click OK.

3.2.2 Modifying Columns
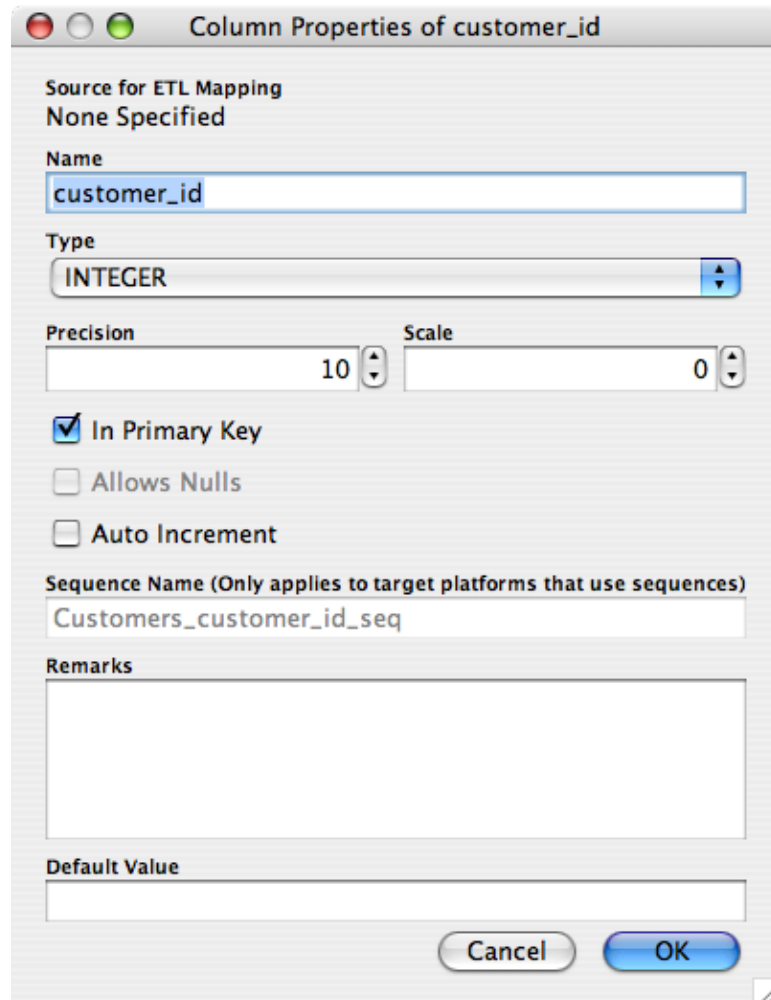
To modify a column:

1. Click a column, then click in the side toolbar.

   Alternate methods:

   • Right-click a column, then click Column Properties.

   • Click a column, then press ENTER.

   The Column Properties dialog box appears.

If you added this column to your data model using reverse engineering, the source database and table from which the column originated are shown at the top of the Column Properties dialog box.

2. For modifying multiple columns at a same time: select multiple columns, then open column properties as above. The window looked like this:

You can modify the properties of every column you selected by checking the checkboxes in front of them.

3. Modify the column properties as required. For a description of the properties, see Section 3.2.1 .

4. Click OK.

3.2.3 Moving Columns

You can move a column from one table to another or rearrange columns within a table.

• To move a column, click the column and drag it to a new location.

• To move multiple columns, use CTRL+click to select the columns, then drag them to a new location.

Note: You can also add or remove columns from the primary key. For more information, see Section 3.3 .

You can also use cut and paste to move to a column from one table to another. The column keeps the source for ETL mapping from the original table.

3.2.4 Copy Columns from external documents

To copy column from external documents (for example spreadsheets) into Power*Architect:

1. Select & copy the columns from your external document

2. Select the table in the Playpen you want to add the column to

3. Press CTRL + V to copy the columns into the table

3.3   Working with Primary Keys

After adding one or more columns to a table, you can define the column(s) used for the primary key.

To add a primary key:

1. Select one or more columns.

2. Drag the column(s) to the primary key area in the table.

To remove a primary key:

1. Select the column(s) in the primary key area.

2. Drag the column(s) from the primary key area to the table's column list.

Note: You can change the primary key name for the table. For more information, see Section 3.1.2 .

3.4   Working with Relationships

3.4.1   About Identifying and Non-Identifying Relationships

You can create relationships between tables. For example, a typical one-to-many relationship might describe how invoices and invoice line items relate to each other. The relationship might indicate that the invoice_line table is a child of the invoice table, and every row in the invoice_line table relates to exactly one row in the invoice table.

You can create identifying and non-identifying relationships:

- In an identifying relationship, the child table cannot be uniquely identified without the parent.

- In a non-identifying relationship, the child can be identified independently of the parent.

You could choose to create the invoice and invoice line relationship from the previous example as either an identifying or non-identifying relationship.

- If you create an identifying relationship, an invoice line cannot be uniquely identified without also knowing the invoice number it belongs to. For example, assume that invoice line numbers always start at 0 or 1 within each invoice. The same line numbers will appear in different invoices - each invoice will have a line 0, line 1, line 2, etc.
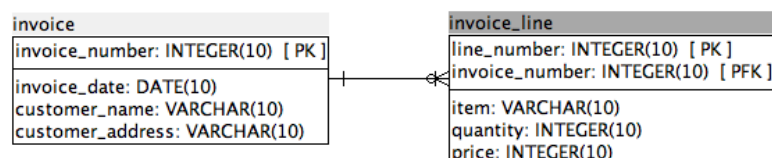
- If you create a non-identifying relationship, an invoice can be uniquely identified without knowing the invoice number it belongs to. For example, assume each invoice line has its own unique identifier (invoice_line_id). In this example, invoice_line_id is referred to as a "surrogate key," because it's just a made-up number which has no special meaning in terms of the invoice line.
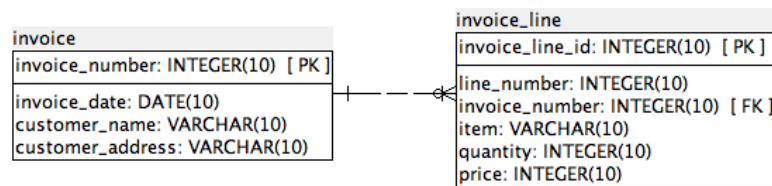
For this relationship, you would also want to create a unique index on the combination of (invoice_number, line_number) to guarantee there are no two line items with the same line number on the same invoice. In the identifying relationship example, the primary key enforces this rule.

3.4.2 Creating Relationships

To create a new relationship:

1. Do one of the following:

    - To define an identifying relationship, click in the side toolbar, or press R. The cursor changes to a +.

    - To define a non-identifying relationship, click in the side toolbar, or press SHIFT+R. The cursor changes to a +.

    Note: To cancel creating a relationship, press ESC or click a blank area in the playpen.

2. Click the parent table, then click the child table. A relationship is created between the two tables and is shown as a line.
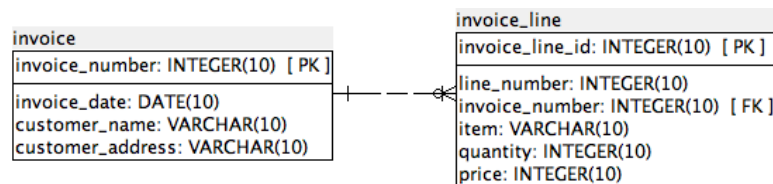
    The mapping between the tables is based on the parent table's primary key. For each column in the primary key of the parent table:

    - If the child table contains a column with the same name and this is the first relationship between the two tables, the relationship is mapped to the existing column in the child table.

    - If the child table does not contain a column with the same name, or the child table contains a column that has the same name but the column has a different data type, or a relationship already exists between the tables, a new column is created in the child table. The relationship is mapped to the new column.
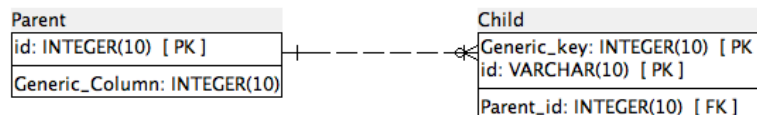
3. To view the columns that are mapped by the relationship, click the relationship link. The mapped columns are shown in red.
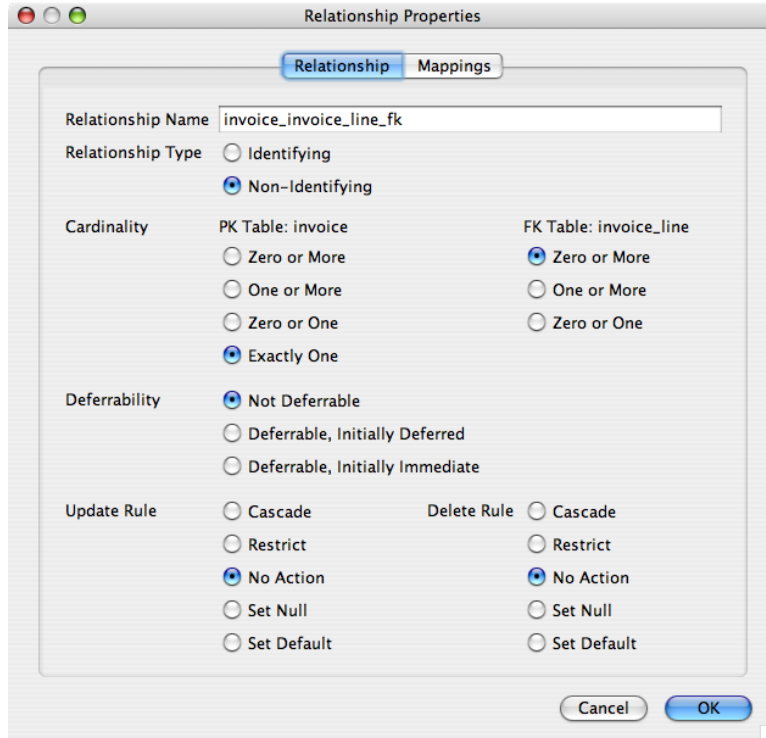
You can now define the relationship properties, view the individual column mappings or change the mapping of the child table to the parent table. For more information, see Section 3.4.3 .

Note: You can automatically straighten the relationship lines between tables. For more information, see Section 3.6.6 .

3.4.3 Modifying a Relationship

To modify a relationship:

1. Click a relationship link in the playpen, then click [icon] in the side toolbar. The Relationship Properties dialog box appears.

Alternate method:

- Right-click the relationship link, then click Relationship Properties.

2. You can enter the following information on the Relationship tab:

In this field ...	Do this ...
Relationship Name	Enter a name for the relationship. When you forward engineer the data model, the relationships are created as foreign key constraints in the target database. These constraints are named based on the relationship name. You can also view a relationship's name in the playpen when you hover over the relationship line.
Relationship Type	Select the type of relationship (identifying or non-identifying).
Cardinality	Select the end cardinality for the primary and foreign keys.

In this field ...	Do this ...
Deferrability	Select the deferrability options. • Not Deferrable - Foreign key constraints are checked immediately at the time an INSERT, UPDATE, or DELETE statement is issued. • Deferrable, Initially Deferred - If the database transaction doesn't specify whether to defer constraint checks, the foreign key constraints will be deferred, meaning that they are not checked until the INSERT, UPDATE, or DELETE transaction is committed. • Deferrable, Initially Immediate - If the database transaction doesn't specify whether to defer constraint checks, foreign key constraints are checked immediately at the time an INSERT, UPDATE, or DELETE statement is issued. Important: Before selecting an option, read the following description to ensure you fully understand the effect of each option. When manipulating data in a database (using INSERT, UPDATE, and DELETE statements), the foreign key constraints created by Power*Architect are used to ensure data integrity between the two tables. The deferrability options control when these constraints are enforced. Within the context of a transaction, deferred constraints are not checked until the transaction is committed, while immediate constraints are checked at the time the INSERT, UPDATE, or DELETE statement is issued (in the middle of the transaction). This means that if you are using immediate constraints, you must be careful about the order in which data is changed. With deferred constraint checking, you can make changes in any order as long as all constraints have been satisfied by the time you commit. For databases that support deferred and immediate constraint checking, each transaction can specify whether to defer constraint checks or carry them out immediately. If a transaction does not specify this option, each deferrable foreign key constraint is evaluated according to its "initially immediate" or "initially deferred" option. On the other hand, constraints marked as "not deferrable" will always be checked immediately regardless of the transaction's setting. Important Notes: • For data manipulation done outside the context of a database transaction, there is no difference between immediate constraint checking and deferred constraint checking. • Not all database platforms support this option. Some only support deferred constraint checking, while others only support immediate. When Power*Architect forward engineers to these types of systems, the DDL script includes comments warning about this lack of support.

In this field ...	Do this ...
Delete Rule	• Restrict - Prevents deletion of a referenced row. • No Action - If any referencing rows still exist when the constraint is checked, an error raised; this is the default behavior if you do not specify anything. (The essential difference between two choices is that No Action allows the check to be deferred until later in the transaction, whereas Restrict does not.) • Casecade - When a referenced row is deleted, row(s) referencing it should be automatically deleted as well. • Set Null - Cause the referencing columns to be set to nulls when the referenced row is deleted. • Set Default - Cause the referencing columns to be set to default values when the referenced row is deleted. NOTE: Set Default and Set Nulls do not excuse you from observing any constraints. For example, if an action specifies Set Default but the default value would not satisfy the foreign key, the operation will fail.
Update Rule	Analogous to Delete Rule there is also Update Rule which is invoked when a referenced column is changed (updated). The possible actions are the same.
Platform Supports	• Oracle - Supports only Restrict, No Action in the Update Rules and everything except Set Default in the Delete Rules. • PostgreSQL - Supports every rule. • MySQL - Supports everything except Set Default in both the Update and Delete Rules. • IBM DB2 - Supports only Restrict, No Action in Update Rules and everything except Set Default in Delete Rules. • HSQLDB - Supports everything except Restrict in both Update and Delete Rules. • SQL Server 2000 - Supports only Casecade and No Action in both Update Rule and Delete Rule. • SQL Server 2005 - Supports everything except Restrict in both Update and Delete Rule.

3. On the Mappings tab, you can change the mapping to the child table. Click and drag the relationship link to the column in the child table that is mapped to the parent table.

   Note: If a column in the child table has [FK] beside it, this means the column is a foreign key in another parent table. This alerts you that the column is already "in use", since you wouldn't normally use the same column as a foreign key in multiple tables.
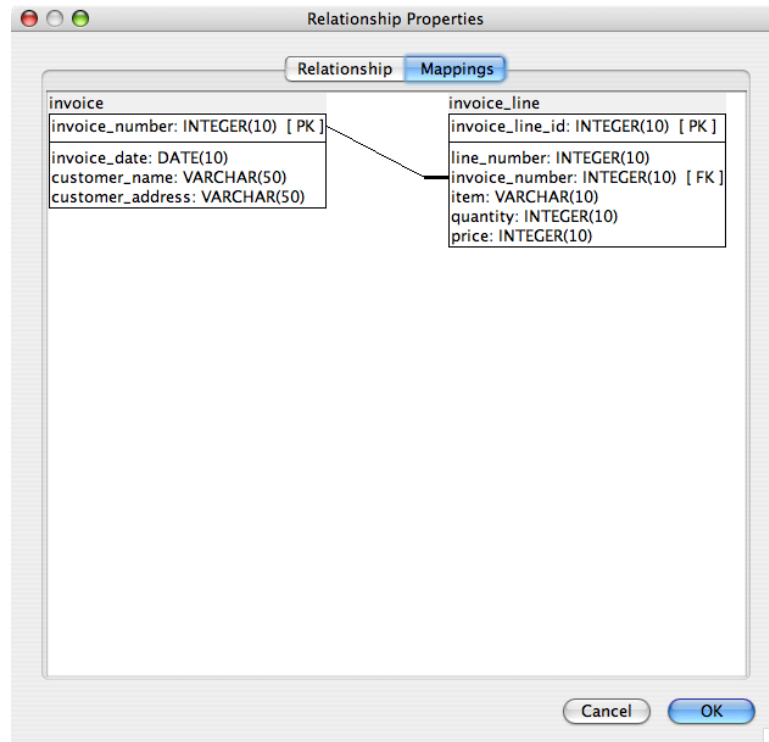
4. Click OK.

3.5 Working with Indices

3.5.1 Creating an Index

You can create multiple indices for a table.

To create an index:

1. Select a table in the playpen, then click in the side toolbar. The Index Properties dialog box appears.
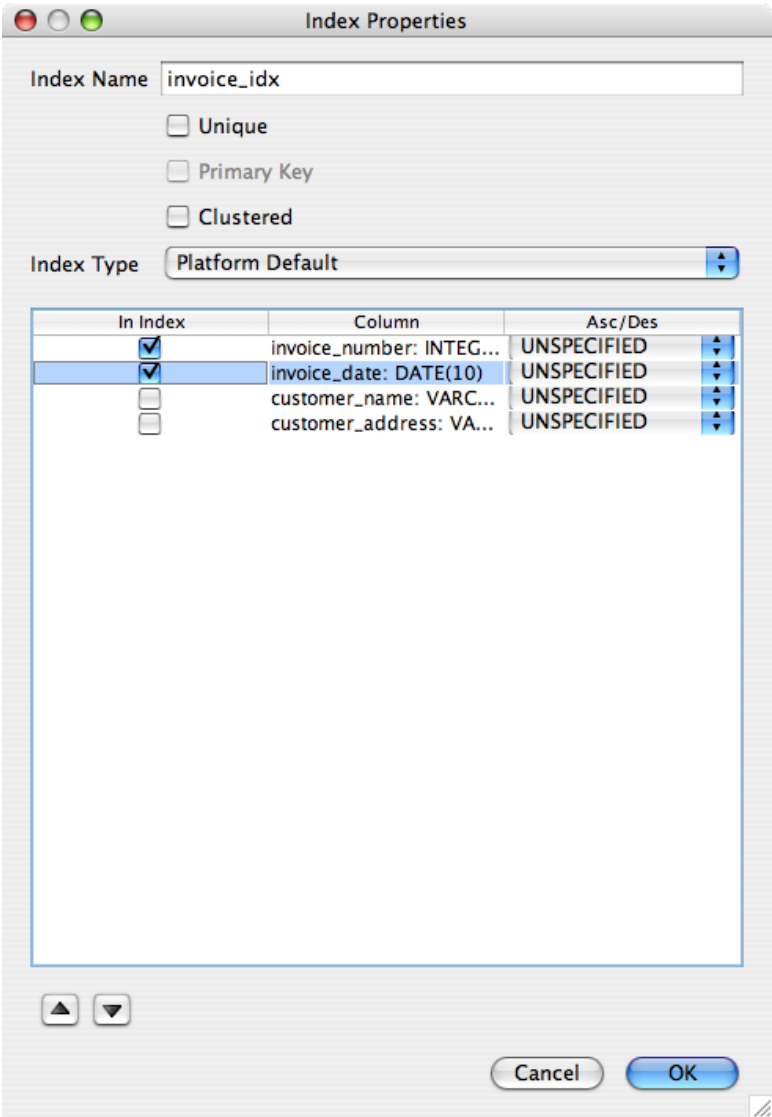
Alternate methods:

- Left-click a table in the playpen, then press I.
- Right-click a table in the playpen, then click New Index.
- Right-click a table in the database tree, then click New Index.

2. You can enter the following information:

In this field ...	Do this ...
Index Name	Enter a name for the index.
Unique	Select the check box if the index will act as a constraint which guarantees the values in this index's columns are unique across all rows in the table. This is similar to the primary key constraint, with two exceptions: A unique index may contain nullable columns, and a table can have any number of unique indices.

In this field ...	Do this ...
Primary Key	Select the check box to set this index as the table's primary key. The primary key is a special type of index which enforces uniqueness: The values in the primary key's columns are unique across all rows in the table. A table can only have one primary key, and none of the columns in the primary key may be nullable. It is considered good practice to have a primary key on every table in the data model.
Clustered	Select the check box to create a clustered index. Many databases support the notion of a clustered index. The exact meaning varies by platform, but marking an index as clustered often affects the physical ordering of the rows within the table (which may increase or decrease performance based on the types of SQL queries being run). Most database platforms allow only one clustered index per table.
Index Type	Select the index type. The list includes all known index types for all database types configured in your user preferences. If you are building a cross-platform data model, it's best to leave this setting at "platform default." However, if you are tuning your data model for a specific target database, you may choose the desired index type for your platform.
List of columns	Select the In Index check box beside each column you want to include in the index. For each column, select the sort order (Ascending, Descending, or Unspecified). Use the arrows at the bottom of the dialog box to set the order of the columns within the index. Columns higher in the list will come first in the index's column list. Notes: • If the table contains columns in the primary key, a separate index will always be created for the primary key column(s), even if you don't select any columns. • On some database platforms, the column order in the index and the column order in the SQL WHERE clause must match in order for the query optimizer to use the index. • On most database platforms, a WHERE clause that references a subset of a multi-column index can usually be used when those columns in the WHERE clause are the leading columns in the index. Example: Table A has columns B, C, D, E, F. Table A has an index on (F, E, D). SELECT * FROM a WHERE f='x'; - index can be used on most platforms SELECT * FROM a WHERE e='x'; - index can not be used on most platforms SELECT * FROM a WHERE f='x' AND e='x' AND d='x'; - index can be used SELECT * FROM a WHERE d='x' AND e='x' AND f='x' ; - index can be used on some platforms, but index order and WHERE clause order are different so some platforms will not use the index

3. Click OK.

3.5.2 Modifying an Index

To modify an index:

1. Right-click a table in the playpen, then click Index Properties. If there are multiple indices for the table, select the index you want to modify.

   Alternate method:

   - Right-click the index in the database tree, then click Index Properties.

   The Index Properties dialog box appears.

2. Modify the index properties as required. For a description of the properties, see Section 3.5.1 .

3. Click OK.

3.5.3 Deleting an Index

Right-click the index in the database tree, then click Delete Selected.

3.6 Working with Diagram Objects in the Playpen

3.6.1 Using Undo and Redo

Power*Architect keeps track of your actions and allows you to undo them at a later time. The 100 most recent actions you have performed are remembered and can be undone in sequence.

If you undo an action accidentally, you can choose to redo the action. However, be careful: If you make a new change after undoing one or more actions, your redo history is lost.

To undo an action, click   in the top toolbar. You can also select Edit » Undo or press CTRL+Z.

To redo an action, click   in the top toolbar. You can also select Edit » Redo or press CTRL+Y.

3.6.2 Selecting Multiple Objects in the Playpen

To select multiple objects (tables, columns, or relationships) in the playpen, do any of the following:

- Press CTRL or SHIFT and click the objects.

- Click a blank area in the playpen, then drag to form a grey box around the objects.

- Press CTRL+A to select all the objects in the playpen.

To cancel the selection, click a blank area in the playpen.

3.6.3 Deleting Diagram Objects in the Playpen

To delete a diagram object (table, column, or relationship) in the playpen, select one or more objects in the playpen, then click   in the side toolbar.

Alternate methods:

- Right-click an object, then click Delete Selected.

- Select one or more objects, then press DELETE.

3.6.4 Rearranging Diagram Objects in the Playpen

You can change the layout of your data model diagram by rearranging the tables in the playpen. You can also change where relationship links visually connect to a table in the diagram. (To change the columns mapped by a relationship link, you must modify the relationship. For more information, see Section 3.4.3 .)

Notes:

- You can rearrange columns within a table or move columns from one table to another. For more information, see Section 3.2.3 .

- You can automatically arrange the tables in the playpen. For more information, see Section 3.6.5 .

- You can automatically straighten the relationship lines between tables. For more information, see Section 3.6.6 .

To move a table, select one or more tables, then drag the table(s) to a new location in the playpen.

To move the placement of a relationship link, select a relationship link, then drag either end of the link to a new location on the parent or child table.

3.6.5 Automatically Arranging Tables in the Playpen

You can automatically arrange tables in the playpen. Automatic layout works best when you have a large or medium-sized collection of tables, and may not work as well with a small number of tables.

To automatically arrange tables, select several tables in the playpen, then click in the top toolbar.

Note: If you don't select any tables or select only one table, all of the tables will be arranged.

3.6.6 Straightening Diagram Lines in the Playpen

You can automatically create straight lines for the relationship links in your data model diagram. All relationship links will be changed to horizontal or vertical straight lines, as long as the tables connected by the link are aligned horizontally or vertically. If the tables are not aligned, the relationship link will not be changed.

To straighten the relationship lines, right-click a blank area in the playpen, then click Straighten Lines.

3.6.7 Using the Playpen Zoom Options

You can use the zoom options on the side toolbar to control the magnification level in the playpen. The four zoom buttons, in order from top to bottom, are:

- Zoom in

- Zoom out

- Reset the zoom to the default level

- Zoom to fit

To use the zoom options on specific objects in the playpen, select the objects before clicking a zoom button. If you don't select any objects in the playpen, the zoom options affect the entire diagram.

3.6.8   Finding and Replacing Playpen Objects

You can search for objects in the playpen. You can then quickly rename the items or select them in the playpen.

1. Select Edit » Find/Replace, or press CTRL+F. The Find dialog box appears.
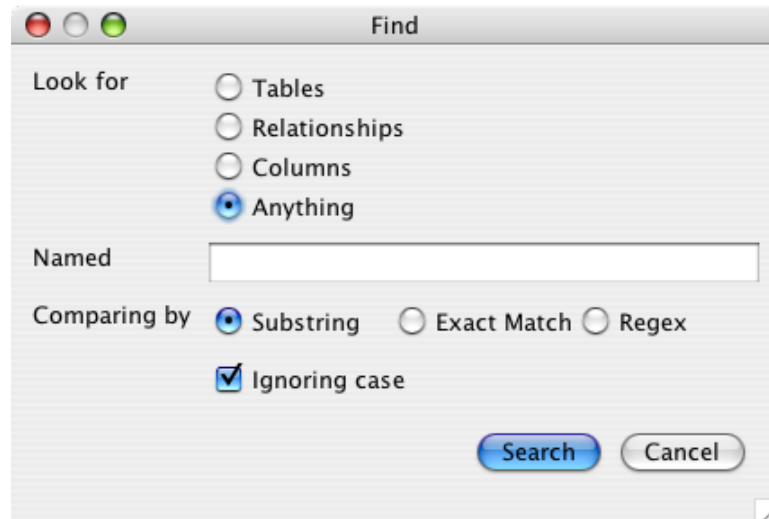
2. Enter your search criteria, then click Search. The Search Results dialog box appears with your results.
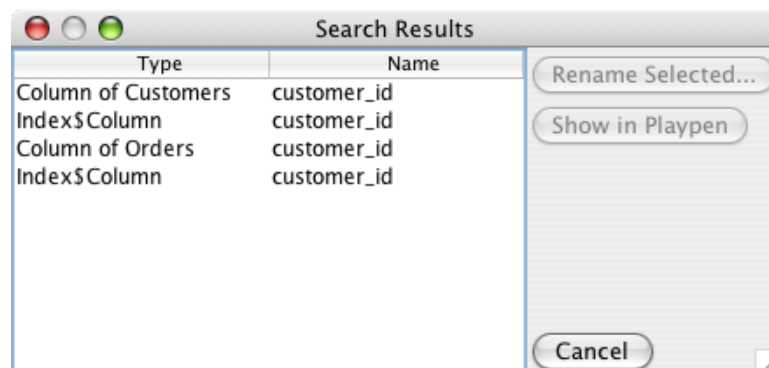
3. To rename an object, select the object and click Rename Selected. You can also select multiple objects if you want to rename all the objects using the same name.

4. To select an object in the playpen, select the object and click Show in Playpen.

You can also find tables in a model by selecting the table in the Playpen Database in the Database Tree and clicking on 'Zoom to fit' in the Playpen Zoom tools on the right side.

3.6.9   Printing or Exporting a Data Model Diagram

To print the data model diagram currently in the playpen, select File » Print.

To export the data model diagram currently in the playpen:

1. Select File » Export Playpen to PDF. The Save dialog box appears.

2. Select the location and filename for the PDF, then click Save.

3. To hide the Creating PDF dialog box, click Run in Background.

Chapter 4

Creating a multidimensional (OLAP) Data Model

Using the Power*Architect OLAP Schema Editor you can also create a multidimensional data model that includes dimensions, cubes and measures.
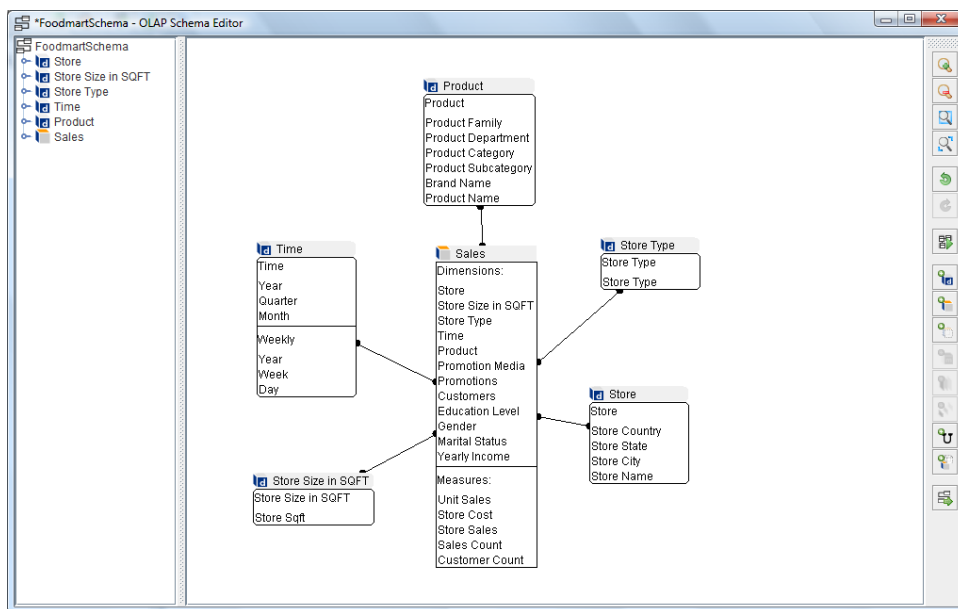
The development of the OLAP Schema Editor is still in an early stage so not all it's functionality is implemented yet. We welcome you to try it out and give us your feedback. This part of the manual covers only a small part of the functionality and is intended for users with experiences in building Mondrian OLAP schemas.

4.1 Working with multidimensional Schemas

4.1.1 Creating a multidimensional Schema

To create an OLAP schema select OLAP->Edit Schema->New Schema. Select the database your schema will be based on and enter a name for the new schema. The list of database connections corresponds to the databases available in the DB Tree in the main Architect frame. If you'd like to base your OLAP schema on an existing database, simply connect to that database in the DB Tree before creating your OLAP schema. Now you can easily add cubes, measures and dimensions to your schema by clicking on the according icon on the right side.

For more details on multidimensional schemas, please read the Mondrian schema specifications.

4.1.2 Importing a multidimensional Data Model

To import an OLAP schema select OLAP->Import Schema. After selecting the schema file you can enter the name of the schema and select the related database.

4.1.3 Exporting a multidimensional Data Model

You can export OLAP schemes to PDF and XML with right-clicking on the Playpen in the OLAP Schema Editor. To export it to XML select 'Export Schema...', to export it to PDF select 'Export Playpen to PDF'.

4.1.4 OLAP Schema Manager

The OLAP Schema Manager (OLAP -> OLAP Schema Manager) give the user access to all schemas that are available in the project.

4.2 Working with Cubes

4.2.1 Adding a Cube

To add a cube:

1. Click on to create a new cube. You can also press "c" instead.

2. The cursor will change to a crosshair. Move the cursor near the left of the Playpen area, and click. The Cube Properties box will appear.

3. Enter the following information:

In this field ...	Do this ...
Name	Enter a measure name.
Caption	Enter the measure caption.
Default measure (optional)	Enter the default measure. Leave this field empty if you create a new cube.
Fact table	Either select an existing table or create a new view using a SQL statement.

4. Click on OK.

5. You can now add measures to the cube.

4.2.2 Adding a Virtual Cube

Virtual cubes are not supported yet.

4.2.3 Adding a Dimension to a Cube

To add a dimension to a cube please readSection 4.4.2.

4.3   Working with Measures

Every cube can contain several measures. A measure can either be derived from a column of the fact table or from a MDX formula, further referenced as calculated member.

4.3.1   Adding a Measure

To add a measure:

1. Select the cube you want to add the measure to.

2. Click on to add a measure to the cube.

3. Enter the following information:

In this field ...	Do this ...
Name	Enter a measure name.
Caption	Enter the measure caption.
Aggregator	Select the aggregate function (sum, count, min, max, avg, distinct-count).
Value	Either select the column the measure is based on or write an expression.

4.3.2   Adding a Calculated Member

To add a calculated member:

1. Select the cube you want to add the measure to.

2. Click on to add a calculated member to the cube.

3. Enter the following information:

In this field ...	Do this ...
Name	Enter a measure name.
Caption	Enter the measure caption.
Dimension	Enter the dimension name. Measures should be default value.
Visible	Select if calculated member should be visible or not.
Formula	Enter the formula for the calculation.
Format	Enter the format.

4.4   Working with Dimensions, Hierarchies, Levels

In Power*Architect all dimensions are modeled as shared dimensions. Degenerate dimensions are not supported.

4.4.1   Adding a Dimension

To add a shared dimension:

1. Click on to create a new dimension.

2. The cursor will change to a crosshair. Move the cursor near the left of the Playpen area, and click.

3. The dimension properties dialog box appears.

4. Enter the following information:

In this field ...	Do this ...
Name	Enter the dimension name.
Caption	Enter the dimension caption.
Type	Select if your dimension is a standard dimension or a time related dimension.

4.4.2 Adding a Dimension usage

To add a dimension usage:

1. Click on to create a new dimension usage

2. The cursor will change to a crosshair. Do a left click on the cube you want to use the dimension in.

3. Do a left click on the dimension you want to use.

4. The dimension usage property box appears.

5. Enter the following information:

In this field ...	Do this ...
Caption	Enter the dimension usage caption.
Foreign Key	Select the related foreign key.

4.4.3 Adding a Hierarchy

To add a hierarchy:

1. Right click on the dimension you want to add the hierarchy to.

2. Select "New hierarchy".

3. The hierarchy properties dialog box appears.

4. Enter the following information:

In this field ...	Do this ...
Name	Enter the hierarchy name.
Caption	Enter the hierarchy caption.
Has All	Select if Hierarchy has an all level element.
All Level Name	Enter the name of the all level element.
Table	Select the table the hierarchy belongs to.
Primary Key	Select the related primary key.

4.4.4 Adding a Level

To add a level

1. Right click on the hierarchy you want to add a level to.

2. The level properties dialog box appears.

3. Enter the following information:

In this field ...	Do this ...
Name	Enter the level name.
Caption	Enter the level caption.
Column	Select the column your level is based on
Unique members	Select if the level has unique members.
Level Type	Enter the type of the level. This selection is only available if you set the dimension type to a time dimension.
Properties	Add properties to the level.

4.5 Working with Roles

Roles are not supported yet.

Chapter 5

Setting up Database Support

There are many features within Power*Architect that involve connecting to a database, such as reverse and forward engineering. Power*Architect allows you to use any JDBC- or ODBC-accessible source database. For more information on supported databases, see Section 5.1 .

Connecting to a database with Power*Architect involves the following steps:

1. Define general settings and drivers for the database platform you plan to connect to (such as SQL Server or Oracle). For more information, see Section 5.2 .

2. Create a connection to a specific database server. This connection uses the general settings and drivers you have configured for the database platform. For more information, see Section 5.3 .

5.1 Supported Database

Power*Architect provides full or partial support for the following database platforms.

Database	Support Notes
Oracle	Fully supported.
SQL Server	Fully supported.
PostgreSQL	Fully supported.
MySQL	Fully supported.
IBM DB2	Partial support; needs more testing.
HSQLDB	Works; used in samples.
Derby	Preliminary support exists. Reverse engineering databases in Derby 10.3.2 or later is possible. Derby-specific forward engineering is not yet available; however, you can try using the forward engineering support for another platform such as MySQL or HSQLDB. Please post to our web support forum if you are interested in forward engineering your data models to Derby.

5.2 Setting up Database Types

You must define general settings for the database platforms you plan to work with (such as SQL Server, MySQL, Oracle, DB2, etc.). These settings will be used by Power*Architect when you set up a connection to a specific database server.

Note: Remember, at this point you are configuring general settings only and are not connecting to a specific database. For more information on connecting to a database, see Section 5.3 .

- General settings for several database platforms are already pre-configured in Power*Architect. If you plan to work with one of these database platforms, all you need to do is define the location of the JDBC driver. For more information, see Section 5.2.2
.

- You can also define additional database platforms in Power*Architect. For more information, see Section 5.2.1 .

5.2.1 Adding a New Database Type

To add a new database type:

1. Select File » User Preferences.

   Alternate method:

   - Select Connections » Database Connection Manager or Window » Database Connection Manager. On the Database Connection Manager dialog box, click JDBC Drivers.

   The User Preferences dialog box appears, with the JDBC Drivers tab open. Existing database types, including the predefined database types included with Power*Architect, are listed on the left.
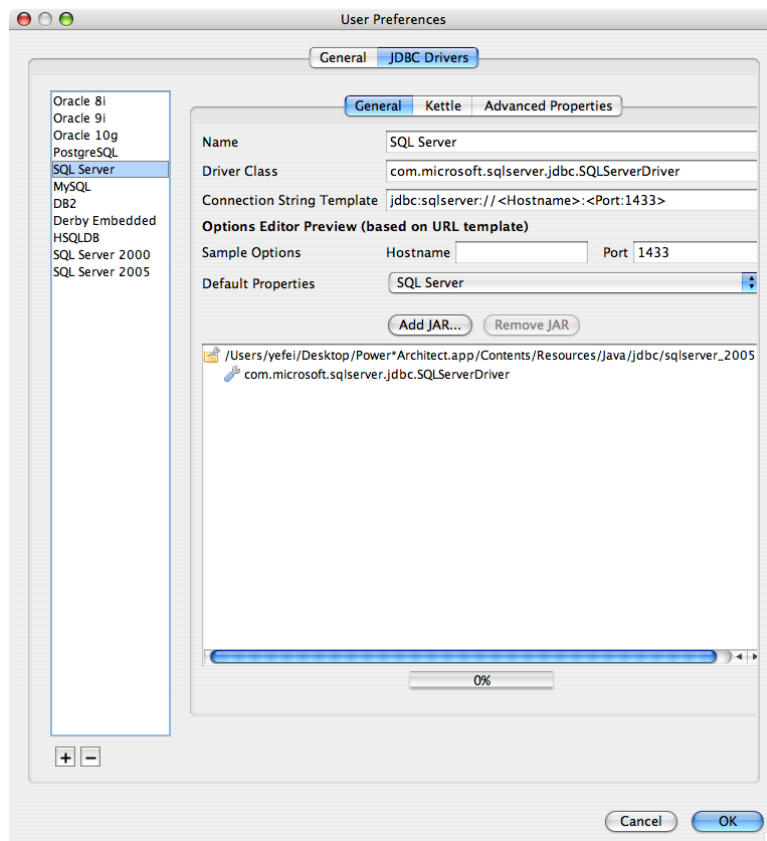
   Note: You can modify an existing database type by clicking on it in the list.

2. Click + below the list of database types.

3. Enter the following information on the General Tab:

In this field ...	Enter the following information ...
Name	Name for the database type (for example, PostgreSQL or SQL Server).
Driver Class	Java class name of the driver. This is the driver class within the JDBC driver JAR file that will be used for database connections.

In this field ...	Enter the following information ...
Connection String Template	General format of the JDBC URL for the database platform. Important: You are not creating a connection for a specific database - you are entering a generic connection string that applies to the database platform. Later on, when you set up a connection to a specific database, Power*Architect will use this template to create the URL to connect to the database. The connection string template must conform to a specific pattern that includes literals and variables. • Literals are entered like normal text but may not contain angle brackets (< or >), which are reserved for defining variables. As the name implies, literals appear in the URL in the same position and way they appear in the template. • Variables are used to for values that change often, such as the schema or database name you wish to connect to. To define a variable in the template, use the format <variable_name:default_value> (to include a default value) or <variable_name> (if you don't want to include a default value). If you use a default value. it is entered automatically when you create a database connection. You can modify the value if the database you are connecting to is configured to use a different value. Each variable you define is shown below the Connection String Template field. This provides you with a preview of the values you will be able to modify when creating a database connection. For example, the connection string template to connect to a Microsoft SQL Server database might look like this: jdbc:sqlserver://<Hostname>:<Port:1433> When you create a connection to a specific SQL Server database, Power*Architect will use this template to create the connection URL. In this example, the template will create the URL "jdbc:sqlserver://:1433", where 1433 is the default port value. Since SQL Server databases listen to port 1433 by default, it makes sense to include this value in the template. When you're creating the actual database connection, you can change the port value if the database you're connecting to is configured differently.

4. The settings on the Kettle tab are only used when you create a Kettle job. For more information on these settings, see Section 10.2 .
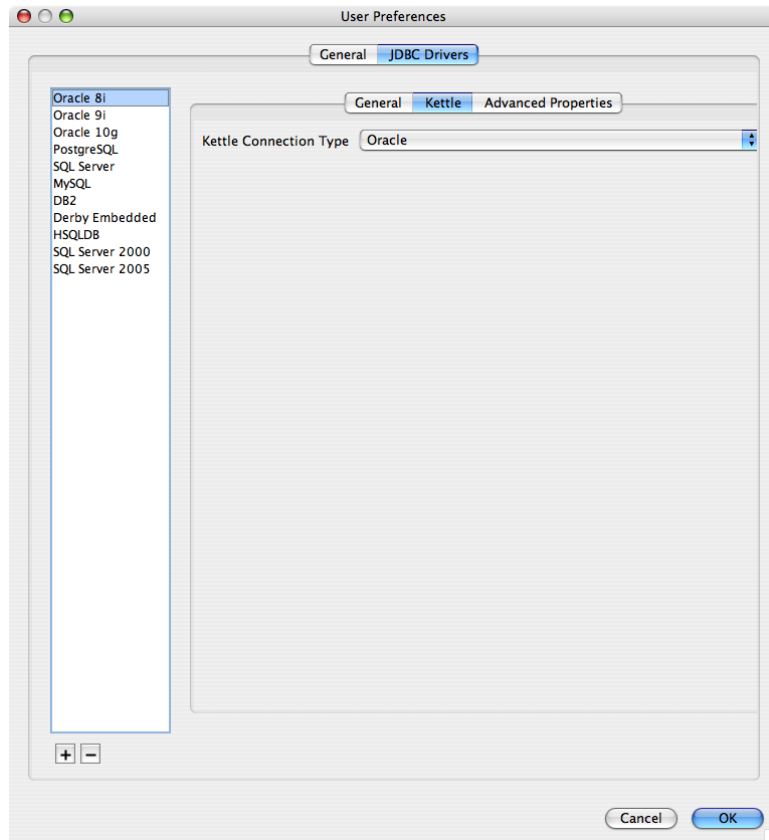
5. Click OK.

Next, you must define the location of the JDBC driver for the database type. For more information, see Section 5.2.2 .

5.2.2   Defining the JDBC Driver

Whether you are adding a new database platform to Power*Architect or want to use one of the pre-configured platforms, the last step in setting up a database type is to locate the JAR file (or files) that contain the JDBC drivers for the database platform.

Note: Remember, at this point you are just telling Power*Architect where the drivers are. You must set up a database connection in order to connect to a specific database server (for more information, see Section 5.3).

Unlike most applications, which need a distinct driver program to communicate with each type of database, Power*Architect uses Java-based drivers. These drivers normally come from the database vendor in the form of JAR (Java Archive) files. JAR files are an extension to the file format used by PKZip/WinZip archives.

Most database platforms provide drivers that are fully backward compatible. This means that it is best to use the newest driver available, regardless of the software version on the specific database server you intend to connect to. One exception to this is the Oracle database. It is important to match the major version number of your JDBC driver with the major version number of the Oracle database server you connect to. For example, if you are connecting to an Oracle 10g database, use the latest Oracle 10g driver. If you are connecting to an Oracle 9i database, use the Oracle 9i driver.

To define the JDBC driver for a database type:

1. If you do not have the JDBC driver for a specific database platform, you can usually obtain one from the database vendor. If that fails, you can find a directory of databases drivers on Sun's web site . There is also a permanent thread in the Power*Architect user support forum , where you can share information with other Power*Architect users about finding and configuring drivers for a particular database platform.

2. Decide on a permanent location to store your JDBC drivers. A good strategy is to create a JDBC folder under your Documents folder and collect all of you JDBC driver files there.

3. Save the JDBC driver (it will usually be one or more JAR files) in the location you've chosen.

4. If the User Preferences dialog box is not already open, select File » User Preferences.

5. On the JDBC Drivers tab, select a database type.

6. Click Add JAR.

7. Locate the JAR file and click Open. If there is a valid driver class in the JAR file, a file tree will appear showing the JDBC driver classes within the JAR file.

8. Select the driver you want to use.

9. Click OK.

5.3   Setting up Database Connections

You must set up a connection to allow Power*Architect to connect to a specific database. When you create a connection, it is automatically added to the current Power*Architect project. You can also use the connection in all your projects.

Before creating a connection, you must define the general settings for the database platform. For more information, see Section 5.2 .

5.3.1   Creating a New Database Connection

To create a new database connection:

1. Select Connections » Add Source Connection » New Connection.

   Alternate methods:

   - Select Connections » Database Connection Manager (or Window » Database Connection Manager), then click New.
   - Right-click a blank space in the database tree, then click Add Source Connection » New Connection.

   The Database Connection dialog box appears.
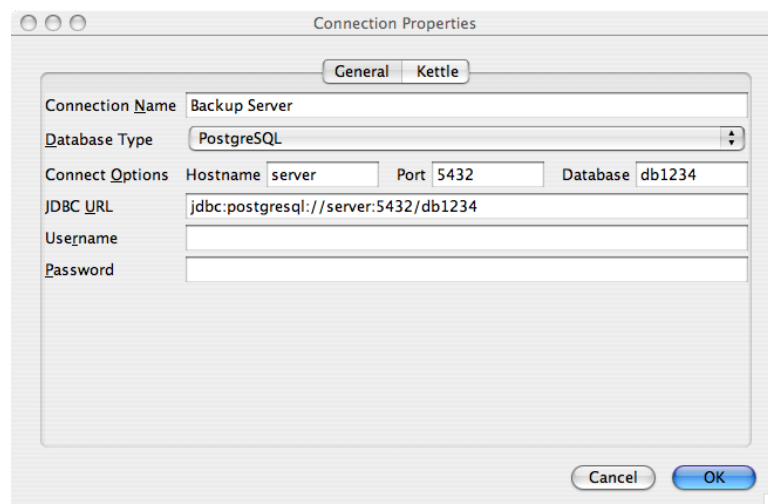
2. On the General tab, enter the following information:

In this field ...	Do this ...
Connection Name	Enter a name for the database connection.

In this field ...	Do this ...
Database Type	Select the database platform you want to connect to. Note: This list contains the database types you defined in your user preferences. For more information, see Section 5.2 .
Connect Options and JDBC URL	Enter the connection options for the database driver. (Theses options are based on the database type you select.) If you are using one of the fully-supported drivers, the connection option parameters are added into the JDBC URL field in the order that the Java driver expects to see them (this string is sometimes called a "db URL" in Java terminology). In the following example: • The default port number from the database type has been entered automatically in the Connect Options. Note: You would not usually change a default value unless the database server you're connecting to has been configured to use a different value. • The hostname and database name have been entered manually in the Connect Options. • The PostgreSQL driver is being used.
Username and Password	If necessary, enter the username and password to connect to the database.

3. The settings on the Kettle tab are only used when you create a Kettle job. For more information on these settings, see Section 10.2 .

4. Click OK. The new connection is added to the current project (you can view the connection in the database tree) and is also added to the Database Connection Manager.

5.3.2 Adding or Removing Database Connections for a Project

You can add a previously created database connection to a project. (When you create a new connection, it is automatically added to the current project. For more information, see Section 5.3.1 .) You can also remove a connection from a project. You cannot remove a connection if it is being used as a source connection in the playpen.

Note: You can permanently delete connections. For more information, see Section 5.3.3 .

To add a database connection to a project, do one of the following:

• Select Connections » Add Source Connection, then select a database connection.

• Right-click a blank space in the database tree, click Add Source Connection, then select a database connection.

The database connection is added to the database tree.

To remove a database connection from a project, do one of the following:

• Right-click a database connection in the database tree, then click Remove Connection.

• Click a database connection in the database tree, then select Connections » Remove Connection.

5.3.3 Modifying or Deleting Database Connections

You can modify a database connection's properties or permanently delete it. You cannot delete a connection if it is being used as a source connection in the playpen.

Note: You can also remove a connection from a project without permanently deleting the connection. For more information, see Section 5.3.2 .

To modify a database connection:

1. Select Connections » Database Connection Manager (or Window » Database Connection Manager).

2. Select a database connection, then click Edit.

   Alternate methods:

   - Right-click a database connection in the database tree, then click Connection Properties.
   - Select a database connection in the database tree, then select Connections » Connection Properties.

   The Database Connection dialog box appears.

3. Modify the connection settings. For information on the settings, see Section 5.3.1 .

4. Click OK.

To permanently delete a database connection:

1. Select Connections » Database Connection Manager (or Window » Database Connection Manager).

2. Select a database connection, then click Remove.

Chapter 6

Setting Preferences

6.1 Defining Project Settings

You can define several settings that apply to all Power*Architect projects.

1. Select File » Project Settings. The Project Settings dialog box appears.

2. Enter the following information:

In this field ...	Do this ...
Snapshot Entire Source Database in Project File?	When you open a source database in Power*Architect (for example, to use for reverse engineering), the database's data structure (catalogues, schemas, tables, etc.) is shown in the database tree. Select this check box if you always want to save the entire data structure in your Power*Architect project. This allows you to view the objects at any time without having to reconnect to the source system. Important: If you use this option, the first time you save your project will be very time-consuming and involve a lot of database activity.

In this field ...	Do this ...
Number of Common Values in Profiles	When profiling a database using graph view, you can view the most common values that occur in a column. Use this option to set the number of common values to include in the profile. For example, enter 10 if you want to include the ten most common values. For more information about profiling, see Section 9.4 .
Profile Creator Mode	Select the mode used to create a profile. • Remote Database - This mode sends a query to the database and the database calculates all of the statistics. This works well over a large network because very little data is transferred. • Local Reservoir - This mode transfers all of the data to the local computer where it is sampled and processed. This works well over a fast network. This option is still experimental and is known to cause an out of memory error when profiling large tables. For more information about profiling, see Section 9.4 .
Draw Relationships With	Select the method used to draw relationship lines in the playpen. Note: Changing this option affects new and existing relationship lines. • Rectilinear Lines - Use horizontal and vertical line segments to connect tables. One to three line segments will be used (at right angles to each other) depending on the position of the tables at each end of the relationship line. • Direct Lines - Use a single line segment (usually diagonal) to connect the tables.
Show Options	Select how to display your project in the playpen. Note: Changing these options affects new and existing columns and tags. • Show Primary Key Columns - display Primary Key Columns if checked, else otherwise. • Show Foreign Key Columns - display Foreign Key Columns if checked, else otherwise. • Show Indexed Key Columns - display Indexed Key Columns if checked, else otherwise. • Show Unique Key Columns - display Unique Key Columns if checked, else otherwise. • Show Remaining Key Columns - display rest Columns if checked, else otherwise. • Show PK tags - display [PK] tags for Primary Key columns if checked, else otherwise. • Show FK tags - display [FK] tags for Foreign Key columns if checked, else otherwise. • Show AK tags - display [AK] tags for the columns which belong to the unique indexes.

3. Click OK.

6.2 Setting User Preferences

You can set preferences that apply to all Power*Architect projects.

Note: This section describes general user preferences and default column settings only. For information on JDBC drivers preferences, see Chapter 5 . You can also set project settings (see Section 6.1).

1. Select File » User Preferences. For Macintosh, select Architect » Preferences.

   The User Preferences dialog box appears.

2. On the General tab, enter the following information:

In this field ...	Do this ...
PL.INI File	Enter the location for the pl.ini file. This file stores the settings for the database connections you create. If you leave this location blank, you will be prompted to use a default location when you start Power*Architect. If you have a pl.ini file from another SQL Power application, you can use the same file for Power*Architect so that you don't have to re-enter all of your database connection information.
ETL Log File	Enter the location for the etl.log file. This log file is written to when you use the ETL features in Power*Architect.
Forward Engineering Log File	Enter the location of the ddl.log file. This log file is written to when you forward engineer a data model.
Antialiased Rendering in PlayPen	Turn on this option to improve the display of the data model diagrams in the playpen, especially when zoomed out. This option may cause slower performance on some systems. Using this option is recommended unless you experience poor performance.

In this field ...	Do this ...
Error Reporting	Turn on this option to send automatic error reports to SQL Power. Error reports never include any information that could be used to identify you or the contents or subject matter of your project. They simply include a Java stack trace that tells developers where in the program code Power*Architect encountered a failure, as well as generic information such as the version of your Java Runtime Environment and the amount of RAM Power*Architect is currently using. These error reports help the Power*Architect development team prioritize bug fixes based on the estimated number of times a particular problem has been encountered.
Show Welcome Screen	Turn this option on to view the welcome screen when you start Power*Architect.

3. Click OK.

4. On the Default Column Setting, enter the following information:

In this field ...	Do this ...
Column name by Default	Enter the default column name.
Column type by Default	Select the default column type.
Column Precision by Default	Enter the default column precision.
Column Scale by Default	Enter the default column scale.
Column in Primary Key by Default	Select if column should be in primary key by default.
Column Allows Nulls by Default	Select if column should allow Nulls by default.
Column Auto Increment by Default	Select if column should Auto Increment by default.
Column Remarks by Default	Enter Column default remarks.
Value by Default	Enter default value.

Chapter 7

Reverse Engineering a Data Model

You can use reverse engineering to obtain a data model from an existing database, then work with the data model in Power*Architect. You can also use Power*Architect to create an upgrade script for the original database (for more information, see Section 9.1).

You can also use reverse engineering for data warehouse design, where your objective is to unify several data models and then import the data from the multiple source systems. To do this, you would typically reverse engineer one table at a time from several different source systems, then make modifications in Power*Architect, using the playpen. You can then forward engineer the new data warehouse data model to a new, separate database (for more information, see Chapter 8 , then use an ETL tool to transfer the data from the source systems to the data warehouse.

For more information, on ETL tools in Power*Architect, see the following sections:

- Section 10.2

- Section 9.5

- Section 9.6

To reverse engineer a data model:

1. To create a new Power*Architect project, select File » New Project.

2. If necessary, create a connection for the database you want to reverse engineer. For more information, see Chapter 5 .

3. Add the database connection to your project. For more information, see Section 5.3 .

   A database node is added to the database tree. Expand this node to view the hierarchy of objects in the database (such as catalogues and schemas, tables, columns, indices, and relationships). The hierarchy is presented the same way a native database tool for the source database platform would present the hierarchy.

   As you click objects in the database tree, the object changes from grey to black to indicate you've viewed it. All viewed items are saved with the project so you can view them later without having to reconnect to the source system.

   Note: If you want to save the entire hierarchy in the project, enable the snapshot option in project settings. For more information, see Section 6.1 .

4. You can now create a new data model using the objects from the database tree. Simply drag objects from the tree into the playpen.

   If you drag higher-level containers (such as a schema, catalogue, or the entire database), individual tables, or multiple tables, all items within the container will be added to the playpen. For example, if you drag a table into the playpen, all of the columns within the table will be added as well. You can also drag individual or multiple columns from the database tree into tables in the playpen. Just drag the columns to the position within the table where you want to insert them.

In addition to using objects from the database tree, you can create new objects (tables, columns, etc.) in the playpen. For more information on working with the playpen, see Chapter 3 .

You can also do the following:

- Create a report listing the source tables used for the tables in the playpen. For more information, see Section 9.5 .

- Compare your current data model to the original database. For more information, see Section 9.1 .

- Forward engineer the schema. For more information, see Chapter 8 .

- Use a Kettle job to move data from the original database to your new database. For more information, see Section 10.2 .

Chapter 8

Forward Engineering a Data Model

A key design principle of Power*Architect is that the data models you create always remain generic. This allows you to use the same data model with a variety of database platforms. You can then use forward engineering to transform a data model for a specific database platform.

When you forward engineer a data model, Power*Architect creates a physical model that represents the idealized generic model as closely as possible, given the constraints of the target system. Power*Architect then creates a SQL Script that you can run to place the components of the data model into a database.

It is important to note that Power*Architect creates the structure of the target database only and does not create the actual database. Before using forward engineering, you must create the target database. You would typically do this using the administrative tools provided for the database platform.

Note: You can view or change the location of the forward engineering log file in user preferences. For more information, see Section 6.1 .

To forward engineer a data model:

1. Open the Power*Architect project containing the data model you want to use. Ensure that all of the data model elements you want to forward engineer are included in the diagram in the playpen. Make any required changes, such as:

   • Creating new tables.

   • Renaming or deleting existing tables.

   • Creating new columns.

   • Renaming or deleting existing columns.

   • Moving columns between tables.

   • Modifying column data types.

   • Merging two or more tables together. (If the tables you merge have a parent-child relationship, this is called denormalizing the data model.)

   • Splitting a table into several related tables (this is often called normalization).

2. Create the target database. You would typically do this using the administrative tools provided for the database platform.

3. If necessary, create a connection for the target database. For more information, see Chapter 5 .

4. Add the database connection to your project. For more information, see Section 5.3.2 .

5. Click SQL on the top toolbar, or select Tools » Forward Engineering. The Forward Engineer SQL Script dialog box appears.

6. Enter the following information:

In this field ...	Do this ...
Create in	Select the database connection for the target database.
Generate DDL for Database Type	Select the database platform. This is the same database type you specified when you created the connection for the target database.
All remaining fields (Catalog, Database, Schema, Owner, ...)	Exactly which fields appear here depends on which target database platform you choose in the "Generate DDL for Database Type" field. Different database platforms organize tables in different hierarchies. You can experiment with choosing different target platform types to see which combinations exist. The names you enter here determine the qualifiers that will come before the names of the tables the script creates and alters. In any case, you can choose to leave these fields blank if you wish your create script to refer to tables (and other objects) by unqualified names.

7. Click OK. Power*Architect generates a SQL script to create the data structure currently in the playpen.

   Note: As Power*Architect is generating the script, warnings or error reports may appear.

8. The Preview SQL Script dialog box appears. For example:

```
CREATE TABLE public.customers (
        customer_id INTEGER NOT NULL,
        firstname VARCHAR(40) NOT NULL,
        lastname VARCHAR(40) NOT NULL,
        city VARCHAR(40) NOT NULL,
        province VARCHAR(40) NOT NULL,
        countrycode CHAR(2) NOT NULL,
        postcode VARCHAR(10) NOT NULL,
        CONSTRAINT customers_pk PRIMARY
KEY (customer_id)
);

CREATE TABLE public.orders (
        order_id INTEGER NOT NULL,
        quantity INTEGER NOT NULL,
        total_amount INTEGER NOT NULL,
        customer_id INTEGER NOT NULL,
        CONSTRAINT orders_pk PRIMARY KEY
(order_id)
);

ALTER TABLE public.orders ADD CONSTRAINT
customers_orders_fk
FOREIGN KEY (customer_id)
REFERENCES public.customers (customer_id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

9. To run the script, click Execute. The database objects are created in the target database.

Chapter 9

Analyzing Data Structures

9.1 Comparing Data Models

You can compare two data models to view the differences and similarities. You can compare a database to a Power*Architect project or to another database.

The data model comparison provides you with a description of the two data models, highlighting their differences and similarities, which you can copy into a document or save to a text file. You can also use the data model comparison to generate and run a SQL script that will update the older database to match the newer data model.

To compare two data models:

1. Click on the top toolbar, or select Tools » Compare DM. The Compare Data Models dialog box appears.
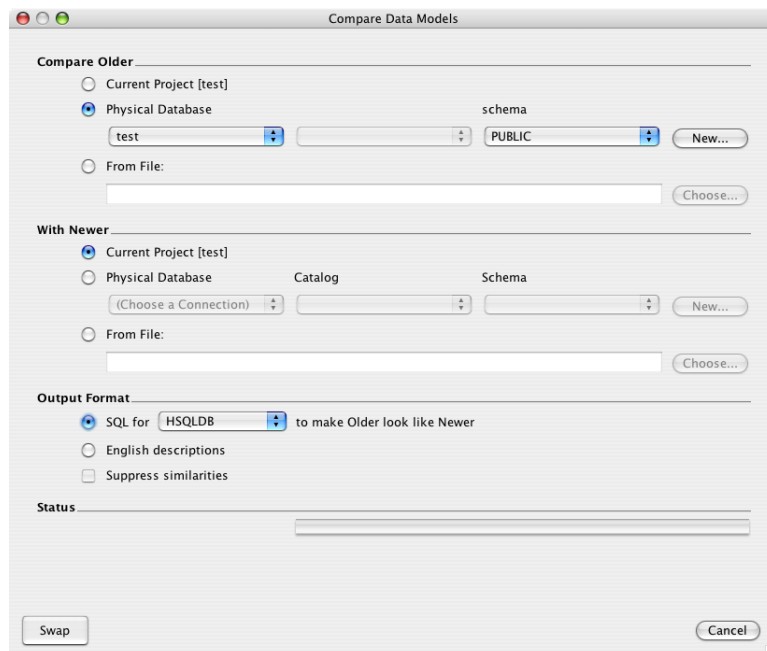
2. In the Compare Older and With Newer sections, select the data models you want to compare.

   - Select Current Project - Include an open Power*Architect project in the comparison. The data model currently in the playpen will be used.

   - Physical Database - Include an existing database in the comparison. You must also select the connection Power*Architect will use to connect to the database. For more information, see Section 5.3 .

- From File - Include an existing Power*Architect project in the comparison. Click Choose and select the project.

   Note: If you want to switch the items you've selected in the Compare Older and With New sections, click Swap.

3. In the Output Format area, select whether you want to create a SQL script or an English comparison.

4. Select the Suppress similarities check box if you want to include only the differences in the output.

5. Click Start. The data model comparison is created.

   Note: The Start button is only available if both data models in the comparison are valid.

See the following sections for details on the information shown in the data comparison.

9.2   Data Model Comparison with English Descriptions

If you chose English descriptions as the output format, the older and newer data models are shown side-by-side. You can copy the results to the clipboard or save them to a text file.

The comparison includes descriptions to make the older data model the same as the newer data model. The components are also colour coded to indicate similarities and differences.

The following table summarizes the meaning of the colour codes used in the data model comparison:

Colour	Description
Black	The component exists in both data models.
Green	The component exists in this data model only.
Red	The component does not exist in this data model but does exist in the other data model.
Blue	The component is a column and is on different keys in the two data models.

9.3   Data Model Comparison in SQL Script

If you chose SQL script as the output format, a script is created to make the older data model the same as the newer data model. You can copy the script to the clipboard or save it to a text file.

To run the script and apply the changes to the older database, click Execute.

Note: The Execute button is only available if the older database has a valid database connection. For more information, see Section 5.3 .

9.4   Profiling Data

Profiling allows you to view a summary of the data in a database. You can use profiles to quickly learn the characteristics of data in an unfamiliar database. You can also use profiles for activities such as database optimization and data migration. When you create a profile, the results are saved as part of the Power*Architect project.

Note: Power*Architect contains two different menu items related to profiling. Use Profile » Profile only when you want to create a new profile. If you want to view existing profiles, use Window » Profile Manager. (The profile manager window is similar to the download manager window in a web browser.)

9.4.1   Setting the Profile Mode

You can select the mode used to create a profile.

1. Select File » Project Settings.

2. In the Profile Creator Mode list, select one of the following options:

   - Remote Database - This mode sends a query to the database and the database calculates all of the statistics. This system works well over a large network because very little data is transferred.
     Warning: Profiling moderate-to-large tables (for example, with over 250,000 rows) remotely will put a significant demand on the database server's resources and may impact the database performance for other users.
   - Local Reservoir - This mode transfers all of the data to the local computer and then samples and processes the data there. This works well over a fast network. This option is still experimental and may cause an out of memory error when profiling large tables.

3. Click OK.

9.4.2 Creating a Profile

To create a profile:

1. Connect to the database you want to profile. For more information, see Section 5.3 .

2. In the database tree, select the tables you want to profile. (You can also select a column. If you do, a profile will be created for the entire table.)

3. Click ![icon] in the top toolbar.
   Alternate methods:

   - Select the tables you want to profile, then select Profile » Profile.
   - Right-click a table in the database tree, then click Profile.

   The Table Profiles window opens. The new profile is listed in the window, along with previous profiles you've created for the project.
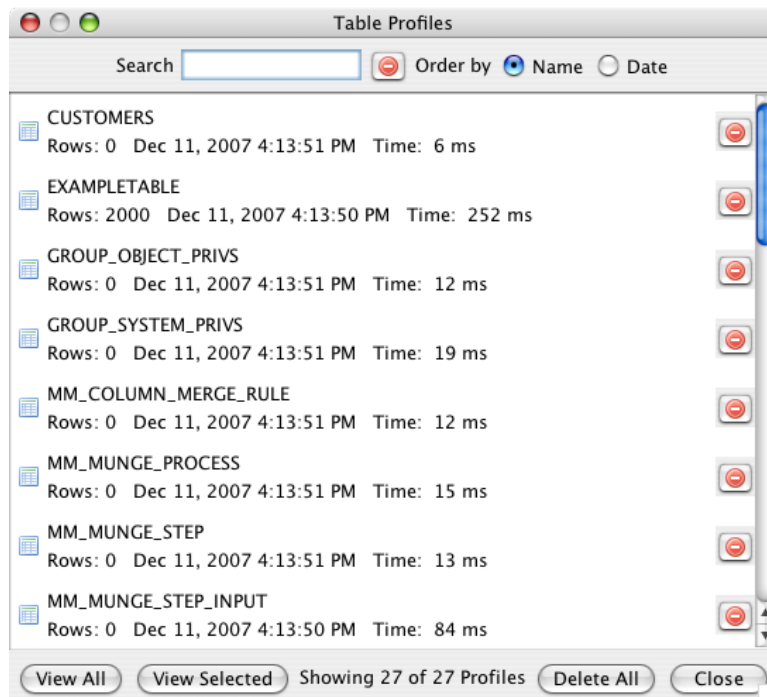
4. You can view details about each profile in the Table Profiles window. For more information, see Section 9.4.3 .

Note: To create a new profile of the same table, select the table in the Table Profiles window and click ![icon] . The previous profile will be retained as well. (Power*Architect will connect to the source database to create the new profile, regardless of the profile mode you're using.)

9.4.3 Viewing Profile Details

To view profile details:

1. If the Table Profiles window is not already open, select Window » Profile Manager.

2. You can use the Search box and Order by options to find a profile.

3. To view details for all profiles, click View All.

4. To view details for some profiles only, select one or more profiles in the window, then click View Selected.

You can view the profile details as a graph or table. For more information, see Section 9.4.4 and Section 9.4.5 .

9.4.4 Using Profile Graph View

To view the profile results in a graph:

• Click the Graph View tab.
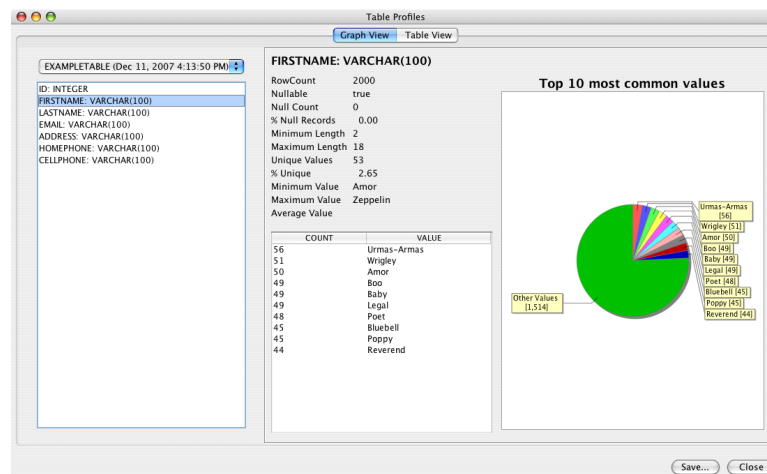
• On the left side of the window, select a column.

The column statistics are shown in the centre of the window. The most common values and their frequency within the table are also shown.

The pie chart on the right side of the window shows the frequency of the most common values in the column.

Note: You can set the number of common values to include in the comparison. For more information, see Section 6.1 .

• To save the profile results in CSV, PDF or HTML format, click Save.

9.4.5 Using Profile Table View

To view the profile results in a table format:

1. Click the Table View tab.

2. To narrow the results, use the Search box in the top-right corner.

3. To sort a column in ascending or descesending order, click the column header.

4. In the Most Frequent column, hover over a cell to view the value and frequency of the most common items in the column.

5. To save the profile results in CSV, PDF or HTML format, click Save.

9.4.6 Deleting Profiles

To delete a profile:

1. If the Table Profiles window is not already open, select Window » Profile Manager.

2. To delete a profile, click beside the profile or press delete.

3. To delete multiple profiles, use CTRL to select all the profiles you want to delete, then press delete.

4. To delete all the profiles, click Delete All.

9.4.7 Saving Your Profile Results in a PDF

You can easily create a PDF document that presents your profile results in an attractive format.

1. Create one or more profiles (see Section 9.4.2).

2. Select Window » Profile Manager.

3. In the Table Profiles window, select the profiles you want to include in the PDF, then click View Selected (see Section 9.4.3). Or click View All to include all of the profiles in the PDF.

4. Click Save.

5. Select PDF as the file type and enter a filename, then click Save.

9.5 Creating a Visual Mapping Report

When you create a data model using reverse engineering, you can create a report listing the source tables used for the tables in the data model. You can export this report to a CSV (comma-separate values) file.

1. Select ETL » Visual Mapping Report.

2. To save the report to a CSV file, click Export to CSV.

9.6 Exporting Column Mappings

When you create a data model using reverse engineering, you can export a CSV (comma-separated values) file describing the source-to-target column mappings between the original database and the data model you created in Power*Architect.

1. Open the project containing the data model you want to use.

2. Select ETL » Export CSV. The Save dialog box appears.

3. Select the location and filename for the CSV file, then click Save.

Chapter 10

Copying and Transforming Data

10.1 Copying Data Across Database Platforms

You can use Power*Architect to quickly copy data from one database platform (such as Oracle) and create a verbatim copy on another database platform (such as PostgreSQL). Power*Architect automatically checks for foreign key constraints in the target database and orders the inserts and deletes accordingly.

You can also use Power*Architect to copy data if the source and target databases are on the same database platform. However, in this case, it's usually faster and more reliable to use the database vendor's own tools to do a "dump-and-restore".

If you want to do something more complex than a verbatim copy, use an ETL tool such as Kettle. ETL tools offer great flexibility in extracting, transforming, and loading data between databases. For more information, see Section 10.2 .

To copy data:

1. Select Tools » Copy Table Data. The Copy table data dialog box appears.
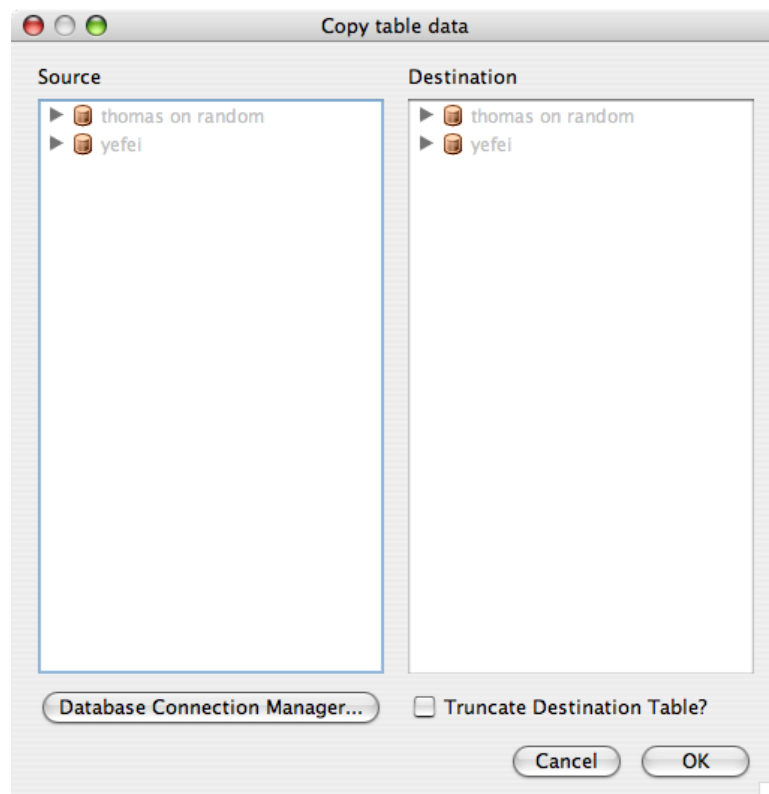
2. Select the Source and Destination databases. If necessary, click Database Connection Manager to set up a new database connection.

3. Select the Truncate Destination Table check box to delete all existing data in the destination tables before copying the data from the source tables.

   Warning: Only use this option if you are sure you want to delete the existing data in the destination tables.

4. Click OK.

10.2 Using Kettle Jobs

You can use Power*Architect to create a Kettle job, which you can then use to create multiple transformations based on a data model you've created in Power*Architect. You would typically create a Kettle job to copy data to a new database you've created through reverse engineering.

Note: The Kettle ETL tool is provided by Pentaho as free and open source software. SQL Power does not maintain or distribute Kettle. To obtain a copy, visit kettle.pentaho.org .

10.2.1 Before Creating a Kettle Job

Before you create a Kettle job, you must use reverse and forward engineering to create a new data model and database.

1. Create a new data model in Power*Architect using reverse engineering (see Chapter 7).

2. Forward engineer the data model into a new database (see Chapter 8). This creates the tables and relationships in the target database.

10.2.2 Creating a Kettle Job

Before creating a Kettle job, ensure you've completed the prerequisites (see Section 10.2.1).

Note: You can view or change the location of the Kettle (ETL) log file in user preferences. For more information, see Section 6.2 .

1. Open the project containing the data model you want to use for the Kettle job.

2. Select ETL » Create Kettle Job. The Create a Kettle Job dialog box appears.

3. Enter the following information:

In this field ...	Do this ...
Job Name	Enter a name for the job.
Target Database	Select the database connection for the target database. Click Properties to view the connection and modify it if necessary. Ensure the connection contains the following information: • General tab - Enter all the required connection properties for the database platform. (See Section 5.3 .) • Kettle tab - Enter the hostname, port, and database for the target database, if applicable. If a field does not apply to the database platform, it will be disabled. You do not have to enter a login name and password. Note: The hostname, port, and database information may be entered automatically based on the information on the General tab.
Schema Name	Enter the name of the schema in the target database that contains the target tables. If the target database doesn't contain any schemas, or the target tables are in the default schema, you can leave this field blank.
Default Join Type	Select the join type to use in all merge-joins. Merge-joins are used to create tables with multiple sources. Note: Merge-joins that are created in transformations from Power*Architect will usually have to be updated manually, since Power*Architect cannot tell which fields to compare during the join.
Save Job to File	Select this option to save the Kettle job settings and transformations to a file. Click Browse and select the location and filename.
Save Job to Repository	Select this option to save the Kettle job settings and transformations in a repository. In the Repository list, select the database connection for the repository. You can use a connection you have set up previously (if the database contains a repository) or you can set up a new connection to a repository. (See Section 5.3 .) Click Properties to view the connection and modify it if necessary. Ensure the connection contains the following information: • General tab - Enter all the required connection properties for the database platform. (See Section 5.3 .) • Kettle tab - Enter the hostname, port, and database for the repository, if applicable. If a field does not apply to the database platform, it will be disabled. Enter the repository login name and password. Note: The hostname, port, and database information may be entered automatically based on the information on the General tab.

4. Click OK to create the Kettle job and transformation files.

If you are using a repository, you are prompted to select the directory location in the repository where the files will be saved.

Once the job has been created, a window appears with the steps you need to complete before running the Kettle job.

Note: The transformation files are stored in the same location as the Kettle job. You must use Kettle to run the job. "doc/images/sqlquery_html_output.png"

Chapter 11

Universal SQL Access

Universal SQL Access is a "fall-back" tool that lets you work at the raw SQL command level. This is an advanced topic and should only be used by those familiar with the intricacies of SQL commands and the details of your database.

Universal SQL Access is started from the menu entry under the Tools menu, and begins with the GUI window shown below. The first thing you should do is select which database connection you wish to use. The list of Connections is the same as the main program uses, as set up in the JDBC Connections window.

The basic steps to using Universal SQL Access are to type a command in the top (SQL Command) section of the window and click the Execute button; the results are displayed in the bottom (SQL Results) section. To save you some typing, you can just drag the databases, tables and columns shown in the DB tree at the left of the Universal SQL Access window and drop them on the top (SQL Command) section.
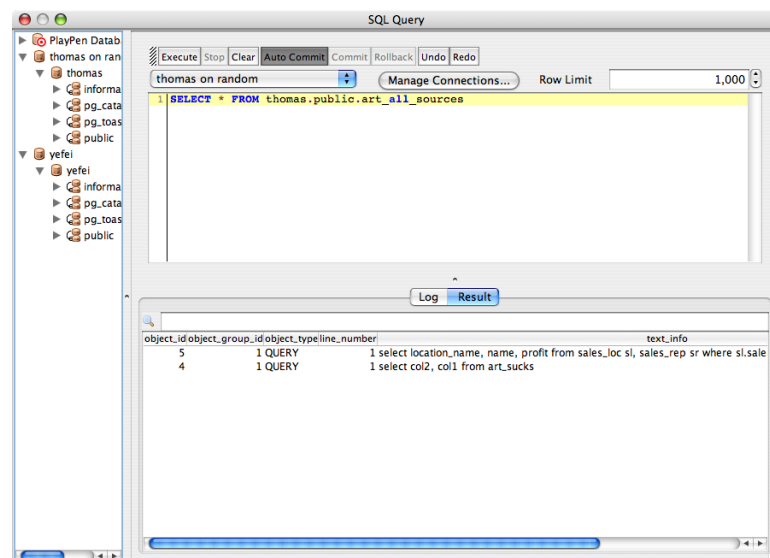
You can enter more than one SQL statement in the SQL Command section. If you do this, be sure to terminate each one with a semicolon. If you enter only one SQL statement, the terminating semicolon is unnecessary. The statements are not interpreted by Universal SQL Access itself, so anything that the given database and driver accepts can be used. For example, with Oracle, you can use PL/SQL statements. With most drivers you should be able to use stored procedures.

To execute your commands, use the mouse to press the execute button, or hit Ctrl-Enter (Command-Enter on Mac). Normally, all statements in the SQL Command text area will be executed in sequence. However, if there is a selection (in other words, some of the text is highlighted) when you execute, only the selected text will be executed. So if you only want to execute a subset of the commands, simply select the part you wish to execute.

By default, auto-commit is enabled, which means each SQL statement is executed in its own transaction context. Be careful, because this means the effects of your update and delete statements are not undoable! If you turn the Auto Commit off, and your database supports transactions (most do), you can manually Commit or Rollback later.

11.1   Output (Results) Window

Command Output appears in the SQL Output window. A scrollbar will appear if the information cannot all be seen at once.

Universal SQL Access keeps a log of statements executed and errors encountered. If you run an undate-type statement, or your statement is rejected by the database, the bottom half of the window will automatically switch to the "Log" tab so you can see the row count or database error message, respectively. Conversely, when you execute a select-type statement, the bottom half of the window switches to the tabular result set view so you can see the data that was selected. Note that in this case, there will still be an entry in the log tab with the date and time the query was executed, how long it took to execute, and how many rows were selected. You can see this information by manually switching to the "Log" tab.

11.2   Output Formats

The output format in the result window is always Table mode, which provides a friendlier interface which ensures all of the columns are lined up properly. If you right-click on a table or multiple tables, you can save the results to a file in HTML or CSV format. The HTML format consists of a table containing results of the select statement. The CSV format, produces a comma-separated values file with a header row, which is suitable for importing to any spreadsheet program, such as Microsoft Excel or OpenOffice Calc.

CSV files are often used as a simple way to transfer a large volume of spreadsheet or database information between programs, without worrying about special file types. For example, transferring a home-made address book from Power*Architect into a database program such as Filemaker Pro could be done by exporting the file as a CSV from Excel, then importing that CSV into Filemaker.

For example, here are the results of executing the select statement, "select * from art_logs":
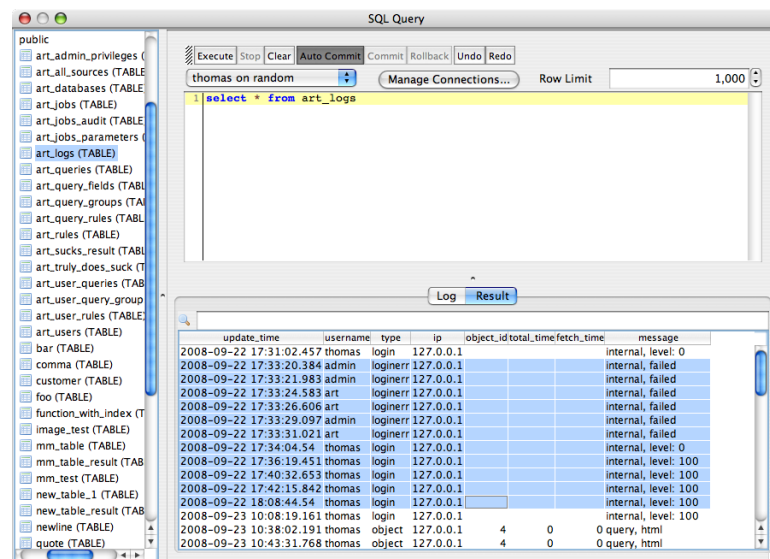
You can view this same data in HTML by right-clicking the selected table, then choosing "Exported selected to HTML..". Here is what the generated HTML looks like in a text editor:
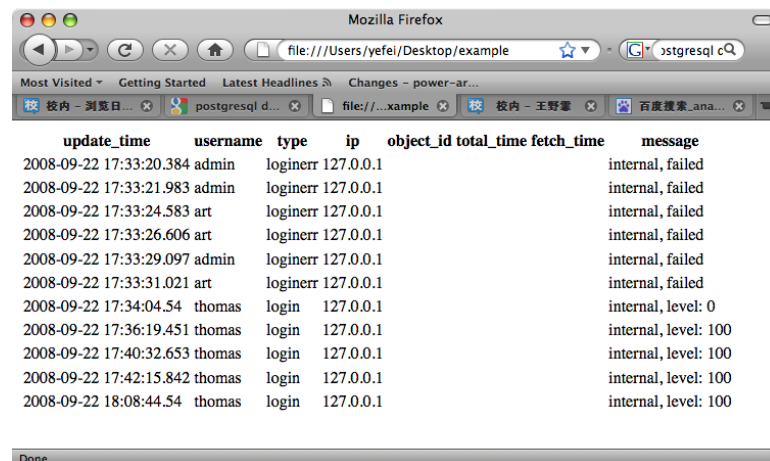
and formatted in a web browser:

With the addition of a custom CSS style sheet, this HTML page could fill a basic reporting need. For more advanced report layout, be sure to try Wabit, the open source reporting tool from SQL Power.

Chapter 12

Troubleshooting

Although we have done our best to ensure you don't experience any problems when using Power*Architect, there may be times when combinations of different database products, database configurations, and so on, cause issues. We apologize in advance for any inconvenience this may cause.

If you are having trouble with Power*Architect, we ask that, in order to help us to diagnose the problem, you take some or all of the following actions:

- Prepare a description of what you were doing.

- Prepare a copy of any errors you encountered.

- Post your problem to the Power*Architect help forum.

Chapter 13

Glossary

This section lists some database-related terms and their meanings.

Some of these terms are from FolDoc, "The Free On-line Dictionary of Computing", http://www.foldoc.org/, Editor Denis Howe.

Column	The set of all instances of a given field from all records in a table .
Database	One or more large structured sets of persistent data, usually associated with software to update and query the data. A simple database might be a single file containing many records , each of which contains the same set of fields where each field is a certain fixed width.
Data Modelling	The product of the database design process which aims to identify and organize the required data logically and physically.
Data Warehousing	A database, often remote, containing recent snapshots of corporate data. Planners and researchers can use this database freely without worrying about slowing down day-to-day operations of the production database.
ETL	Extraction, Transforming and Loading - the process of maintaining and transforming data into and out of a relational database.
Foreign key	A column in a database table containing values that are also found in some primary key column (of a different table). By extension, any reference to entities of a different type. Some RDBMSs allow a column to be explicitly labelled as a foreign key and only allow values to be inserted if they already exist in the relevant primary key column.
Identifying Relationship	Where the key of the parent table is a subset of the key of the child table.
JDBC	Java DataBase Connectivity, an unofficial acronym for the "java.sql" package of functionality used to access relational databases from programs written in the Java programming language.

Key	A value used to identify a record in a database, derived by applying some fixed function to the record. The key is often simply one of the fields (a column if the database is considered as a table with records being rows, see "key field"). Alternatively the key may be obtained by applying some function, e.g. a hash function, to one or more of the fields. The set of keys for all records forms an index. Multiple indices may be built for one database depending on how it is to be searched.
Primary key	The candidate key selected as being most important for identifying a body of information (an entity, object or record).
Record (row)	One or more structured sets of persistent data, usually associated with software to update and query the data. A simple database might be a single file containing many records, each of which contains the same set of fields where each field is a certain fixed width.
SQL	Originally SEQUEL and still pronounced that way by many practitioners, SQL is the Standard Query Language; a unified language for creating queries that is accepted (with some variations) by all modern relational databases.
Table	A collection of records in a relational database.

Chapter 14

Appendices

14.1   Appendix A: GNU GPL Version 3

The Power*Architect is distributed under the terms of the GNU General Public License, version 3 or later. Here is the text of that license:

GNU General Public License version 3

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most

unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL , as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

    a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

    b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

    c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

    d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that

part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

    a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

    b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

    c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

    d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

    e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

    f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously

your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-POSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR COR-RECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.    Copyright (C) year name o-
f author
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/-licenses/ .

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

program Copyright (C) *year name of author* This program comes with ABSOLUTELY NO WARRANTY; for details type ' show w '. This is free software, and you are welcome to redistribute it under certain conditions; type ' show c ' for details.

The hypothetical commands ' show w ' and ' show c ' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL , see http://www.gnu.org/-licenses/ .

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/-philosophy/why-not-lgpl.html .

14.2 Appendix B: Third Party Licenses

14.2.1 The FAMFAMFAM Silk Icon Set

The Power*Architect development team is grateful to Mark James for the beautiful Silk icon set which is used extensively throughout Power*Architect's user interface.

The Silk icons are freely available under the Creative Commons Attribution 2.5 License . If you want to use these icons in your own work (and why wouldn't you?), you can obtain the full set from the FAMFAMFAM web site .

14.2.2 The Apache Software Foundation

The Power*Architect development team is grateful to the Apache Software Foundation and their contributors; their high-quality reusable Java libraries have been invaluable in the development of Power*Architect. The text of the Apache License follows, because we are redistributing several Apache libraries upon which Power*Architect depends.

The following license applies to these library jar files, which are distributed as part of the Power*Architect download:

- commons-beanutils.jar

- commons-digester.jar

- commons-logging.jar

- commons-beanutils-bean-collections.jar

- commons-beanutils-core.jar

- jakarta-regexp-1.2.jar

- commons-collections-3.1.jar

- commons-dbcp-1.2.1.jar

- commons-pool-1.3.jar

- commons-vfs-1.0.jar

- log4j.jar

- opencsv-1.8.jar

Apache License Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions .

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License . Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License . Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution . You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and

- You must cause any modified files to carry prominent notices stating that You changed the files; and

- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions . Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks . This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty . Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability . In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability . While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

14.2.2.1 APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner] Licensed
        under the Apache License, Version 2.0 (the "License"); you may not
        use this file except in compliance with the License. You may obtain
        a copy of the License at http://www.apache.org/licenses/LICENSE-2.0
        Unless required by applicable law or agreed to in writing, software
        distributed under the License is distributed on an "AS IS" BASIS,
        WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
        implied. See the License for the specific language governing
        permissions and limitations under the License.
```

14.2.3   JGoodies Karsten Lentzsch

The Power*Architect team is also grateful to JGoodies for their excellent forms layout manager for Swing. JGoodies forms is released under the BSD license, reproduced below.

The following license applies to these library jar files, which are distributed as part of the Power*Architect download:

- forms-1.2.1.jar

The BSD License for the JGoodies Forms

Copyright (c) 2002-2006 JGoodies Karsten Lentzsch. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of JGoodies Karsten Lentzsch nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EX-PRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER-CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUB-STITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (IN-CLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

14.2.4   PostgreSQL JDBC Driver

The Power*Architect team would like to thank the PostgreSQL JDBC Driver team for their JDBC driver.

The following license applies to these library jar files, which are distributed as part of the Power*Architect download:

- postgresql_8.jar

Copyright (c) 1997-2005, PostgreSQL Global Development Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the PostgreSQL Global Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

14.2.5 iText

The Power*Architect team is also grateful to Bruno Lowagie and Paulo Soares for their excellent Java PDF library. iText is released under the MPL license, reproduced below.

The following license applies to these library jar files, which are distributed as part of the Power*Architect download:

• itext-2.0.8.jar

MOZILLA PUBLIC LICENSE Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is: A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims: (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via

Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters (a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations. Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions. Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions. Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works. If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUD-ING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIM-ITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____. Portions created by _____ are Copyright (C) _____ _____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

14.2.6   GNU LGPL version 3

The Power*Architect team is also grateful to the JFree team for their top-notch charting library, which has a nice API as well as nice-looking output.

The following license applies to these library jar files, which are distributed as part of the Power*Architect download:

• jcommon-1.0.0.jar

• jfreechart-1.0.1.jar

• rsyntaxtextarea.jar

GNU Lesser General Public License version 3

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the " GNU GPL " refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL .

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL .

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

   a. under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

   b. under the GNU GPL , with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a. Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b. Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a. Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b. Accompany the Combined Work with a copy of the GNU GPL and this license document.

c. For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d. Do one of the following:

   1. Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

   2. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e. Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL , and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b. Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

14.2.7   JUnit

The Power*Architect team would also like to extend our sincere thanks to the JUnit.org team. JUnit forms an invaluable part of our development process, but it is not redistributed as part of the Power*Architect download so its license is not reproduced here.

If you develop software, you should become test infected too! Learn about JUnit at http://www.junit.org/ .

14.2.8   Pentaho Data Integration

The Power*Architect provides ETL integration features with Pentaho Data Integration (the tool formerly known as Kettle), and we redistribute a portion of the Kettle library along with the Architect in order to support those features.

We gratefully acknowlegde the work of Matt Casters and the Pentaho corporation for their support and hard work on this product.

We redistribute kettle (kettle.jar), and edtftpj (edtftpj-1.5.4.jar), an FTP library upon which it depends, under the terms of the GNU LGPL, which is reproduced in full elsewhere in this section.

14.2.9   The Eclipse Foundation

The Power*Architect was primarily developed and tested using the Eclipse Java Development Tools, one of the more productive Java environments around.

14.2.10   Sun Microsystems

Last but not least, many thanks to Sun Microsystems and their various Java development teams for creating, extending, bugfixing, documenting, and supporting the Java platform over the past *N* years!

We redistribute the JavaHelp runtime library with Power*Architect. Although the JavaHelp website says that the system will be redistributable royalty-free, it does not actually link to the specific license terms. If someone can point us to the license text for JavaHelp redistributions, we would be grateful!

The portion of JavaHelp that we redistribute is in the following JAR file:

- jhall.jar

We also redistribute the JavaMail library and the JavaBeans Activation Framework, on which it depends (Kettle depends on JavaMail).

The following JAR files are covered by the CDDL, reproduced below:

- mail.jar

- activation.jar

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTIC-ULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PAR-TIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAM-AGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAM-AGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITA-TION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSE-QUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.