

Power*Architect User Guide

SQL Power Group Inc. [<http://www.sqlpower.ca/>]

Power*Architect User Guide

SQL Power Group Inc. [<http://www.sqlpower.ca/>]

Copyright © 2007 SQL Power Group Inc.

Table of Contents

1. Introduction	1
Basic Concepts	2
Who this book is for	3
2. Installation Guide	4
Power*Architect Prerequisites	4
How to Obtain the Power*Architect Software	4
How to Install Power*Architect	4
Microsoft Windows	4
Macintosh OS X	4
Other Platforms	4
How to Run Power*Architect	4
Microsoft Windows	4
Macintosh OS X	5
Other Platforms	5
3. How to Use the Power*Architect	6
Power*Architect User Interface	6
Setting the User Preferences	7
Setting up the Architect for different database types	8
Defining a new Database Type	9
Locating the JDBC Driver	10
A Basic Example	11
Setting Up Databases	11
Designing a Database	11
Forward Engineer	12
Comparing Data Models	13
Using Diagram Components	15
Creating New Tables	15
Editing Tables	15
Creating New Columns	15
Editing Columns	16
Dragging Columns	17
Creating Non-Identifying and Identifying Relationships	17
Editing a Relationship	17
Selecting Multiple Items in the Playpen	18
Relocating Objects in the Playpen	18
Deleting Diagram Components	18
Database Usage in Power*Architect	19
Adding a New Connection to the Power*Architect	19
Adding an Existing Connection	20
Editing Database Connection Properties	20
Removing a Database Connection	20
Navigating through the Database Tree	20
Moving Items from the Database Tree	21
Undo and Redo	21
Find/Replace Function	21
What It Does	21
How to Use Find / Replace Function:	21
Forward Engineering and Compare Data Model	22
Forward Engineering	22
Compare Data Model Function	23
Auto-layout	25

What It Does	25
How to Use Auto-layout	25
SQLRunner	25
What It Does	25
How to Use SQLRunner	25
Output (Results) Window	26
Output Formats	27
Profiling	28
Profiling Mode	29
Table View	29
Graph View	30
How to Create a Kettle Job	30
What it Does	30
Setup for Kettle	30
Creating a Kettle Job	31
4. Database Product Notes	34
5. Troubleshooting	35
6. Glossary	36
7. Acknowledgements	38
The FAMFAMFAM Silk Icon Set	38
The Apache Software Foundation	38
JGoodies Karsten Lentzsch	43
PostgreSQL JDBC Driver	44
iText	45
JFree	54
Darwin Systems	55
Pentaho Data Integration	56
JUnit	56
The Eclipse Foundation	56
Sun Microsystems	56

List of Tables

3.1. Database Tree Icons	20
3.2. Compare Database Model Colour Codes	24
3.3. SQLRunner Escape Characters	26

Chapter 1. Introduction

Data Architects, DBAs, Analysts and Designers rely on state-of-the-art Data Modeling Tools to facilitate and simplify their data Modeling efforts, while maximizing the use of their resources. The Power*Architect software allows these busy technical professionals to perform this most intricate part of their job in a fraction of the time.

SQL Power Group's Power*Architect is a visual data modeling tool which places an emphasis on speed and ease of use, and to the maximum extent possible, database platform independence. You are not required to declare up-front which platform you intend to eventually create your data model in, and in fact, the Architect makes it easy to maintain a database schema which works well in multiple database platforms simultaneously. This is a particular advantage for application developers who have to maintain a database schema for an application that works with several different backend database platforms.

Additionally, the Architect includes several features designed primarily for Data Warehouse and Data Mart design. It allows the designer to open multiple concurrent source Database connections, drag and drop source schemas, tables and columns into the data modeling playpen, and forward-engineer the resulting target database and its associated ETL template. The Architect keeps track of the original location each object was reverse-engineered from, and can produce a visual mapping report as well as a starter set of Extract, Transform, Load (ETL) procedures for use with the popular open source Kettle ETL tool from Pentaho.

The Power*Architect is a user-friendly DW data modeling tool created by data warehouse designers, and has many features geared specifically for the data warehouse architect, including:

- Access any JDBC- or ODBC-accessible source database;
- Design every aspect of the target database Data Model;
- Compare the database structure of any two databases, highlighting the structural differences and generating the required DDL to synchronize;
- Compare the Data Model data structures to an existing target database;
- Save a snapshot of all source systems' data structures in the project file, allowing data warehouse designers to evolve their target data model remotely;
- Forward engineer to Oracle, SQL Server, DB2, PostgreSQL, MySQL, and other databases;
- Produce ETL Templates containing source-to-target data mappings;

Whether you're building or maintaining your Data Warehouse data model, the Power*Architect will provide you a complete view of all required database structures and will expedite every aspect of your data warehouse design.

We firmly believe you can#Design your Data Warehouse in a fraction of the time with the Power*Architect.

In addition to all of the benefits outlined above, another key benefit of the Power*Architect is its licensing and distribution model. The Architect is free and open source software, meaning that the source code is readily available. Everyone is free to inspect, comment on, and modify the 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, but we welcome contributions from the user community.

The key advantage here is that software which is developed by the actual users of that same software tends to be much easier and more fun to use, and to include the features that matter most to users.

Specifically, the Power*Architect is distributed to the public under the New BSD License. There is a large body of software already available under this license, so its terms are already well understood. Here is the license for the Power*Architect in its entirety:

Copyright (c) 2007, SQL Power Group Inc.

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 SQL Power Group Inc. 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.

Basic Concepts

Project - a Power*Architect project consists of a view of multiple databases; you can load and save a Project to work on it at leisure.

Driver - Most programs need a distinct driver program to communicate with each different type of database. Power*Architect uses Java-based drivers, which normally come from the database vendor in the form of "JAR" ¹ files. You need to inform Power*Architect about each driver before you can use it; do this from the User Preferences panel, under JDBC Drivers (just click Add and browse to the Jar file for your driver). If you do not have the JDBC driver for a given database, 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 [<http://developers.sun.com/product/jdbc/drivers>]. There is also a permanent thread [<http://www.sqlpower.ca/forum/posts/list/401.page>] in the Architect's user support forum where users can share information about how to find and configure drivers for a particular database platform.

Playpen - This is the main area of the Power*Architect window, in which you manipulate tables and relationships. You can play here to your heart's content, knowing it will not be saved until you ask the program to save.

¹ Java Archive files; these are an extension to the file format used by PKZip/WinZip archives

Who this book is for

This book is a step-by-step guide on how to use the full capabilities of Power*Architect. It covers topics from how to install the Power*Architect through setting up database connections to engineering your data model.

We assume you are familiar with basic database terms. If you meet any terms that are unfamiliar, please refer to the Glossary at the end of this book.

This book also assumes you are familiar with basic computer operations.

We also assume you have SQL Power's Power*Architect software installed on your computer; if not, please refer to the Installation Guide below.

For the section on creating a Kettle job we assume that the reader has some basic knowledge about ETL. For more information about ETL look for *Building the Data Warehouse* by W. H. Inmon or *The Data Warehouse Toolkit: The Complete Dimensional Modeling* by Ralph Kimball and Margy Ross.

Chapter 2. Installation Guide

Power*Architect Prerequisites

To run the Power*Architect, you need a Java Runtime Environment (JRE). The Architect depends on Java 5, Standard Edition, or newer. A current JRE for common operating systems can be obtained from Sun Microsystems [<http://java.com/getjava/>]. To ensure that your JRE is up-to-date, if you are an Apple Macintosh user, you should use Software Update from the Apple Menu. If you are a Windows user, use the Java Updater (found in the Microsoft Windows Control Panel). If you use a different system such as Linux, Solaris, or BSD, you should use the normal procedures for obtaining and updating software packages on your platform (packages, ports, etc.).

How to Obtain the Power*Architect Software

The Power*Architect can be obtained from the download section [<http://download.sqlpower.ca/architect/current.html>] of the SQL Power [<http://www.sqlpower.ca/>] Website. You should only need to download one file, choosing the platform-appropriate distribution (Windows-Installer for Microsoft Windows, "DMG" for Apple Macintosh, and ".tar.gz" for other platforms). You should normally choose the download with the highest revision number available.

How to Install Power*Architect

Microsoft Windows

Double click on the Architect-setup-Windows-n.m.jar. This will launch the Microsoft Windows installer. Follow the on-screen instructions.

If you are using Windows Vista, you will need to right-click the jar file and run it with administrative privileges, since it needs to write to your Program Files directory and to modify the Windows registry (to create the project file associations).

Macintosh OS X

Double-click the architect-n.m.dmg file, then agree to the license. At this point, a new Finder window will appear with the Power*Architect icon hanging off of a rather fetching green mobile crane. Drag the Architect program icon off of the crane hook and into your Applications folder. Eject the disk image, then delete the dmg download file.

Other Platforms

Extract the Architect-generic-n.m.tar.gz package into the desired directory.

How to Run Power*Architect

Microsoft Windows

From the start menu, select All Programs. Then select the Power Architect program group. Finally, click on the Power Architect shortcut.

Macintosh OS X

From the Finder, select Applications, then select Power*Architect. To enable launching of the Architect directly from the Dock, either drag the program icon there or, while it is running, control-click on the Architect's dock icon and select Keep In Dock.

Other Platforms

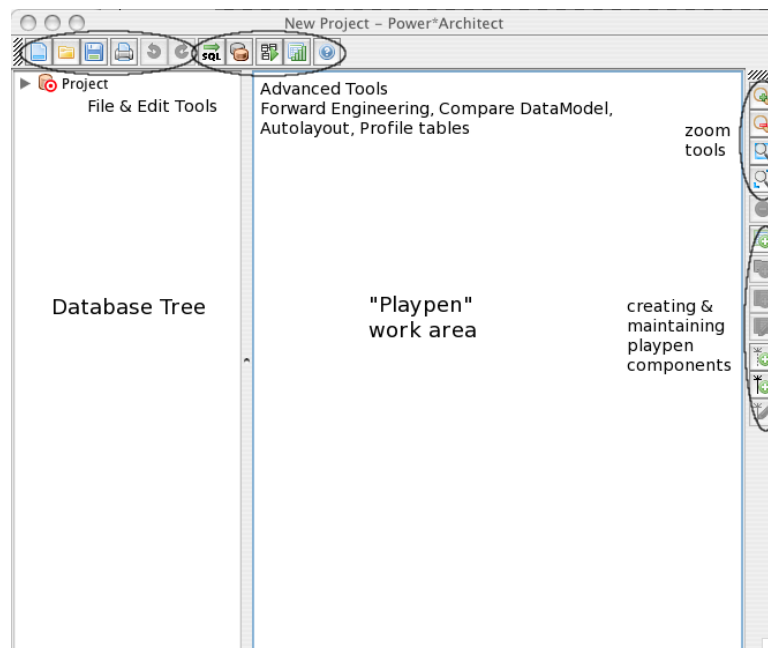
In the directory into which you extracted the Architect, run the command **java -jar architect.jar**. For larger projects, the default maximum amount of memory allowed by the Java runtime may not be enough. In that case, use a command such as **java -Xmx600m -jar architect.jar**, which increases the memory cap from the default 64MB to 600MB.

Alternatively, if your operating system has a visual filesystem browser, you may be able to launch the architect by clicking (or double-clicking) on the architect.jar file.

Chapter 3. How to Use the Power*Architect

Power*Architect User Interface

When you start the Power*Architect, you will see the Project window, shown below, which is the main view area and starting point for actions.



Database Tree - This is where you can add, maintain and explore imported connections. It uses a tree-node dropdown method. Therefore to explore inner components, you can expand components within this container as needed.

Playpen - This is the main area of the window, in which you manipulate tables and relationships. You can play here to your heart's content, knowing it will not be saved until you ask the program to save.

Playpen Components - These are the components that can go into the playpen. The playpen components are Tables and Relationships.

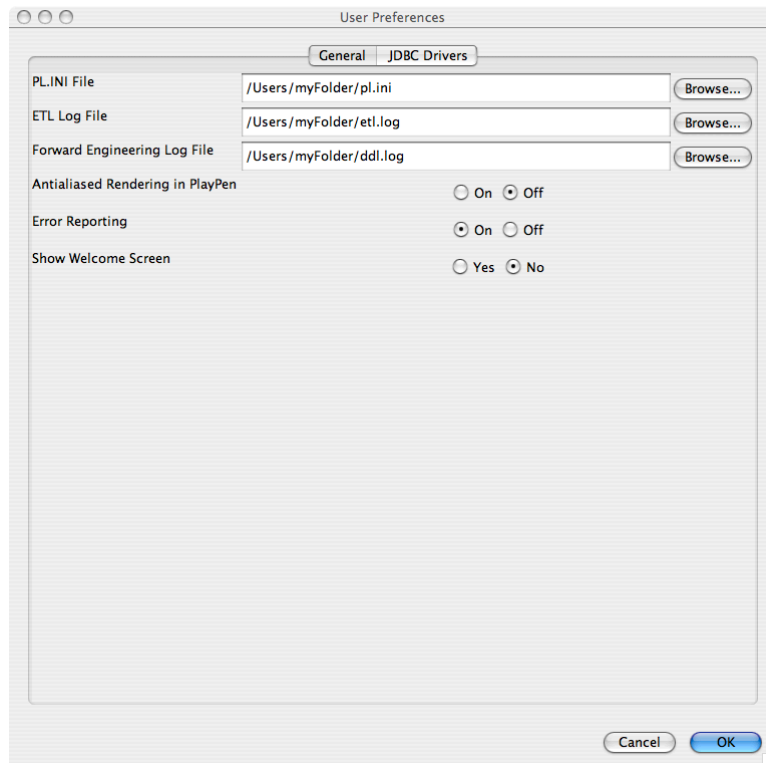
Zoom Tools - These tools allow you to control the magnification level of your view into the playpen. There are four zoom tools. In order from top to bottom, their functions are: Zoom in a bit; zoom out a bit; reset the zoom to its default level; and zoom to fit the whole diagram. If there are any selected components in the playpen, they affect the behaviour of the zoom in and zoom to fit tools: When zooming in, the Architect attempts to keep all selected components visible (by scrolling your view to them); when zooming to fit, the Architect will zoom to fit just the selected components, and not the whole data model.

Delete - Deletes the selected component or components.

Create/Maintain Playpen Components - This is discussed in [Using Components](#) below.

Setting the User Preferences

When getting started, you need to set up some files and drivers to use the full functionality of the Power*Architect. If you have not done so already, please go to "User Preferences" under the File Menu (on Mac OS X, this is found under the application menu) to configure the Power*Architect. The preferences dialog (shown below) will appear.



The pl.ini file stores the settings for database connections that you set up (see the JDBC Drivers section later in this chapter). If you leave this location blank, the Architect program will prompt you to use a default location when you start it up. If you have a pl.ini file from other SQL Power applications you should generally use the same file, because doing so will save you from having to re-enter all of your database connection information. The next two settings are for log files that will be written when you use the ETL and Forward Engineering functions of the application.

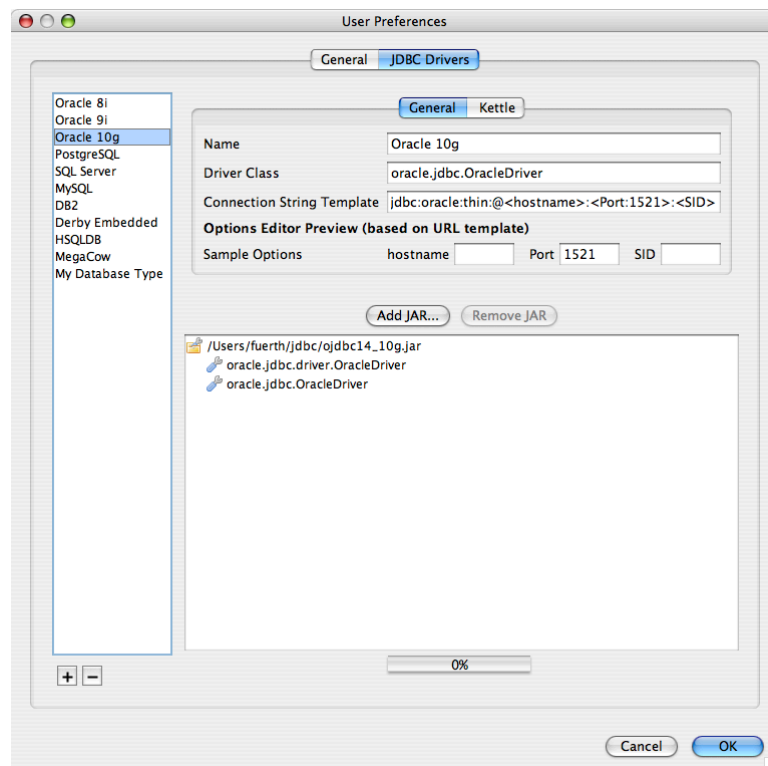
The next setting controls the operations of graphics in the PlayPen. Turning on "antialiased rendering" provides better display of the graphical database layouts shown in the PlayPen, especially when zoomed out, but may cause slower performance on some systems. We recommend that you turn it on unless you experience poor performance as a result.

The error reporting setting can be turned on to send SQL Power error reports when they occur. Error reports never include any information that could be used to identify you, nor do they include any information that could be used to deduce the contents or subject matter of your project. They simply include a Java stack trace that tells developers where in the program code the Architect encountered a failure, as well as generic information such as the version of your Java Runtime Environment, the amount of RAM the Architect is currently using, and so on. These automatic error reports help the Power*Architect development team prioritize bug fixes based on the estimated number of times a particular problem has been encountered.

The last setting, Show Welcome Screen, controls whether or not the welcome screen appears when the Power*Architect starts up.

Setting up the Architect for different database types

The Power*Architect comes pre-configured to work with several database platforms out-of-the-box. If you plan to use the Architect with Microsoft SQL Server 2000 or newer, any version of PostgreSQL, or any version of MySQL, you can safely skip this section. However, if you plan to use the Architect with another type of database, you will have to do a bit more work.



It is important to realize that this section describes the procedure for defining the general rules for how to connect to any database server of a particular type. At this point, you are not defining a connection to a specific database server.

The information about database types includes:

- The name you want to give the database type (for instance, PostgreSQL or SQL Server)
- Where to find the JDBC driver
- Which driver class within the JDBC driver JAR file to use
- The general format of a JDBC URL for this database type

If you need to configure a database type, there are two possibilities: First, if your database type is already present in the list of database types along the left-hand side of the dialog, you are in luck! All you need to do is tell the Architect the location of the JDBC driver. In that case, skip to the section Setting the JDBC Driver Location below. On the other hand, if your database platform is not listed on the left-hand side, you will need to define a new database type from scratch.

Defining a new Database Type

Start by clicking the + button near the lower left-hand corner of the dialog. This will create a new, blank entry in the list. From here, you simply need to fill in the fields as described below.

- The "Name" field is used to assign a name to the item in the list. These names will also be used to assign a type to a new connection that you are defining.
- The "Driver Class" field is the class name of the driver. This class will be contained within one of the JAR files you specified for this location.
- The "Connection String Template" field is used to define a URL template. The URL template is used by the Power*Architect to create a URL that is used to attempt a database connection. The pattern that the string must conform to is not trivial but is made up of two simpler parts, literals and variables.

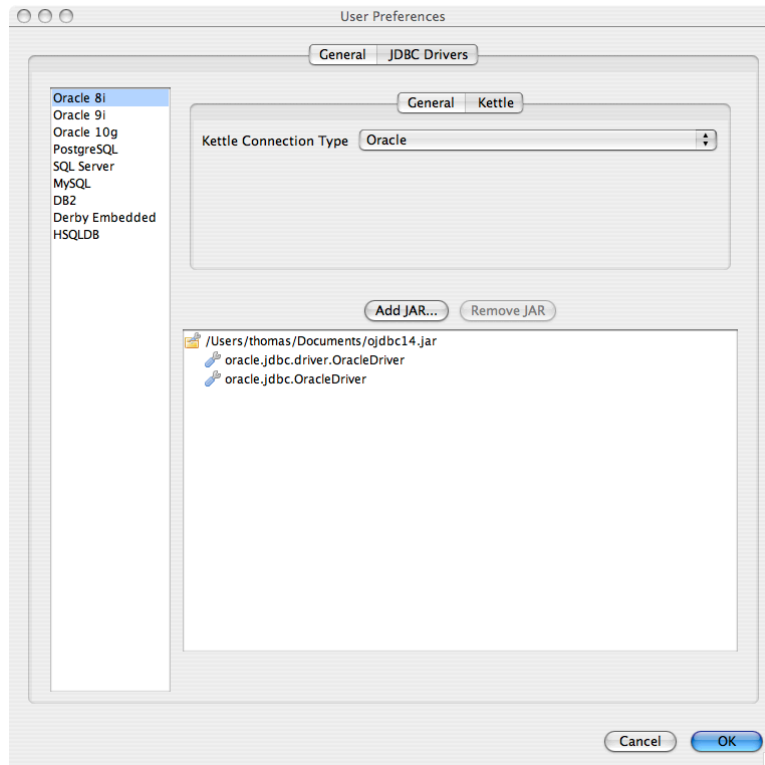
Literals are entered like normal text but may not contain angle brackets (< or >) because they are reserved for defining variables. As their name implies, literals appear in the URL in the same position and way they appear in the template.

Variables are used to make it easier for things that change often to be changed, such as the schema or database name you wish to connect to. They are defined in the template by <variable_name>:default_value>. Once a variable is defined in the template, a field labeled with the variable's name will appear below with the default value in it just below the template as a preview to what you will later be able to modify in "Connection Properties".

Note: Default values are often useful, but are optional. If you do not want a variable to have a default value, define it as <variable_name>. The default value for these variables is blank.

For example, if we wanted to connect to a Microsoft SQL Server database, the URL template might look like "jdbc:sqlserver://<Hostname>:<Port:1433>". The URL this template will output is "jdbc:sqlserver:///:1433" where the 1433 was a default value. If we were to enter "localhost" in the Hostname field, we would get the URL "jdbc:sqlserver://localhost:1433". By defining the "Hostname" variable, we make it easier and quicker to connect to a generic SQL Server instance. SQL Server databases listen to port 1433 by default but can be configured to listen to others so, by giving the variable this default, we can have the most common value inserted automatically while allowing us to modify what is in the field to change the port in specific cases.

- If you click on the "Kettle" tab, there will be a drop-down menu where you can specify the type of Kettle connection to associate with the selected item. If you do not use Kettle, do not worry about this setting because it is only necessary when you wish to create a Kettle ETL jobs.



You're almost done! All that is left to do is tell the Architect where to find the JDBC driver for your database platform.

Locating the JDBC Driver

Whether you have just configured the Architect for a new database platform, or you are using 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 means for connecting to your database platform.

Nearly all database platforms we have encountered provide drivers that are fully backward compatible, meaning that it is always best to use the newest driver available regardless of the software version on the specific database server you intend to connect to. The Oracle database is one exception. 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, you have to use the Oracle 9i driver. For this reason, the Architect comes pre-configured with three database types for Oracle, representing the three major versions still in regular use.

Download the JDBC driver for your database platform from the database vendor. Save this file (it will usually be one or more JAR files) in a safe and permanent location so that you will not be tempted to relocate it later on. For example, your Desktop is not an ideal location. A good strategy is to create a "JDBC" folder under your Documents folder and collect all of your JDBC driver JAR files there.

With the JDBC driver JAR file in a permanent location, click the **Add JAR...** button. Locate the JAR file in the file chooser and open it. If there is a valid driver class in it, a file tree will appear in the area below the buttons. This is a representation of the JAR file and the JDBC driver classes it contains. Select the driver you wish to use and click the "OK" button at the bottom.

A Basic Example

This section will show you how to set up ¹ 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".

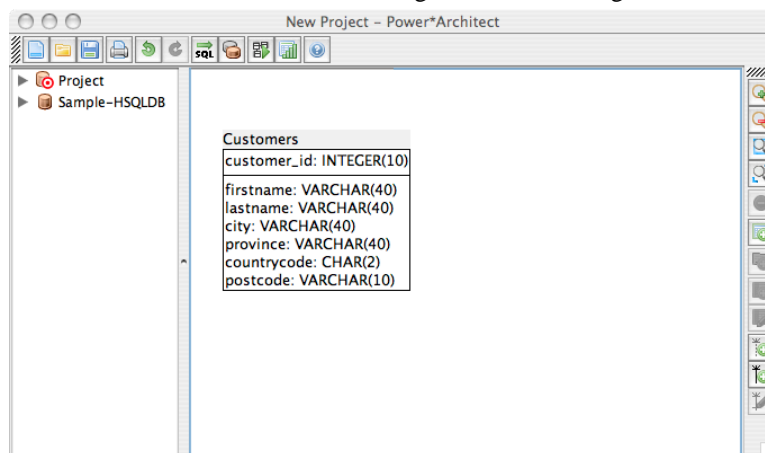
Setting Up Databases

1. Setup Driver. Select File->User Preferences and 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 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 both the Connection Name and the Database name (these names do not have to be the same, but we'll keep them the same for simplicity). If you select the JDBC Driver before you type the database name, then as you type the Database name, it will be added to the DB URL, so you don't have to type it an extra time. Fill in all the fields and click OK.

Designing a Database

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

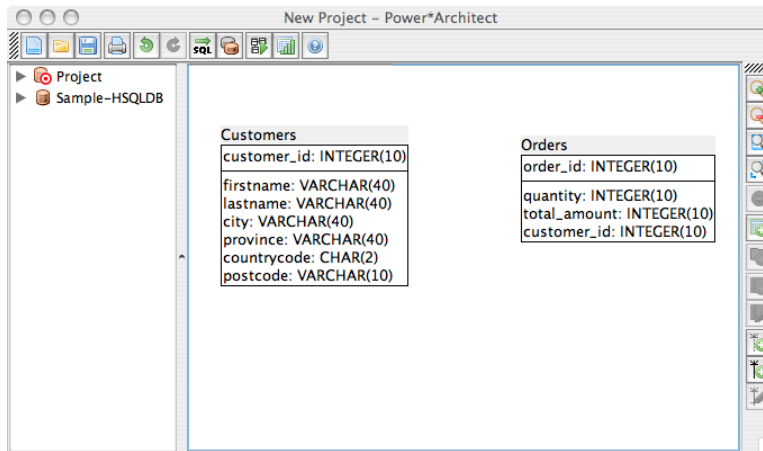
1. Click on the New Table icon at the right side. The cursor will change to a crosshair. Move the cursor near the left of the Playpen area, and click. A "New Table" will appear.
2. Double click on the title, and the Table Properties Dialog will appear. Rename this table to Customers.
3. Click on the Insert Column icon, and a "New Column" will appear. When the new column is created a property window will appear for it. 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 ² and Postal Code. The table should look something like the following:



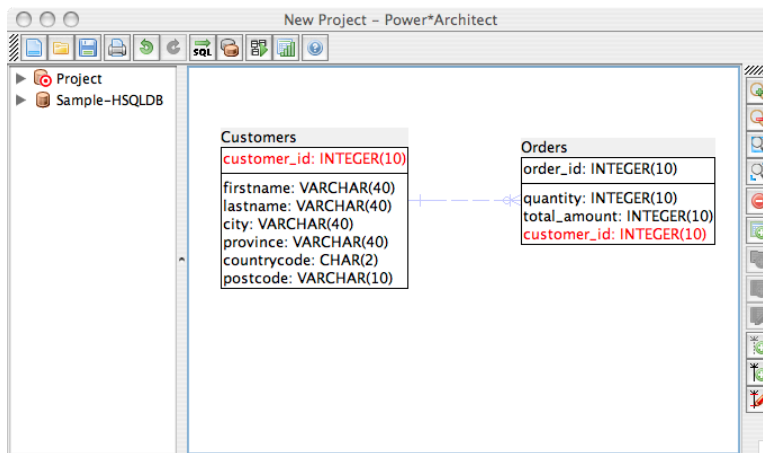
5. Create a second table, and name it Orders.

¹ Assumes you have used some vendor-specific external tool to create a new database.

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:

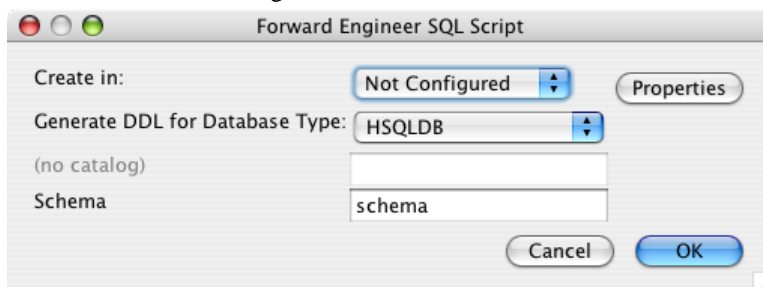


7. We need a relationship between these tables. An order should have a foreign key that refers to the customer. Click the "New Non-Identifying Relationship" icon. Select the Customers table, then the Orders table, and a link will be drawn as shown. Click on this link and the keys that take part in the relationship will be highlighted in red.

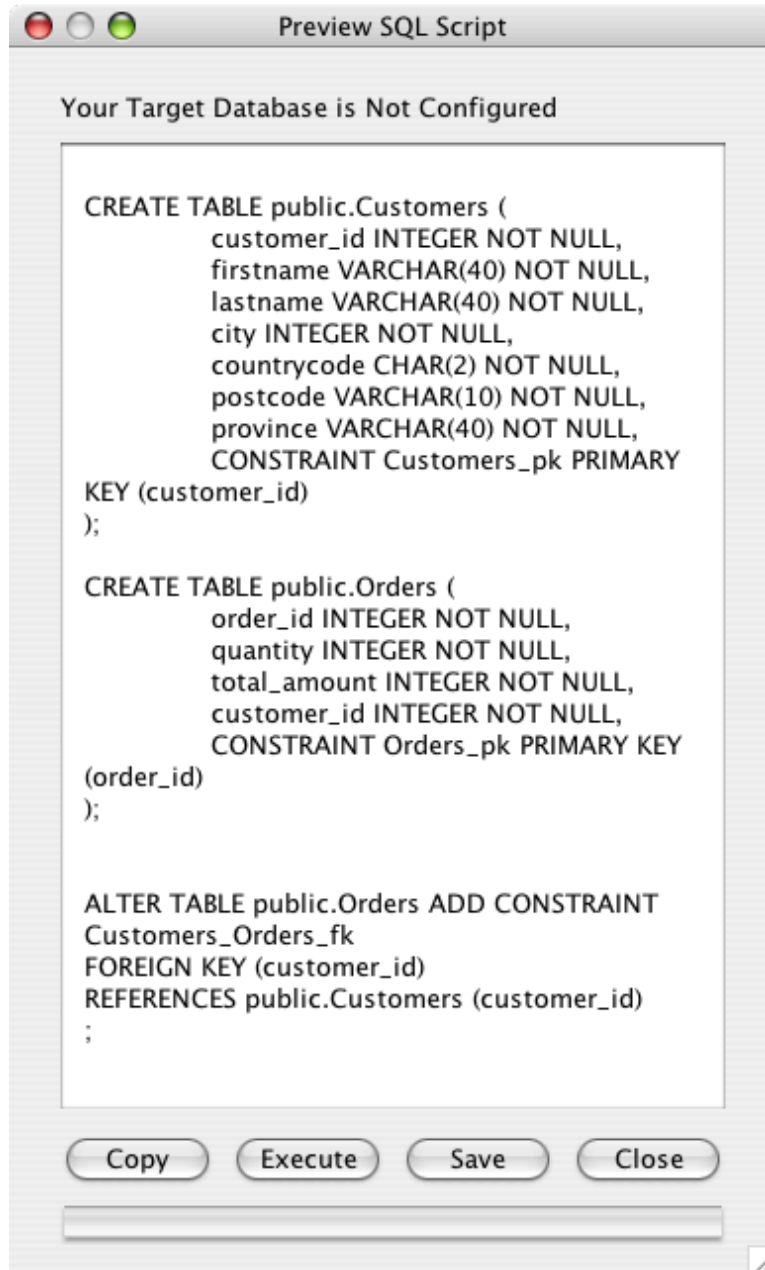


Forward Engineer

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:



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:



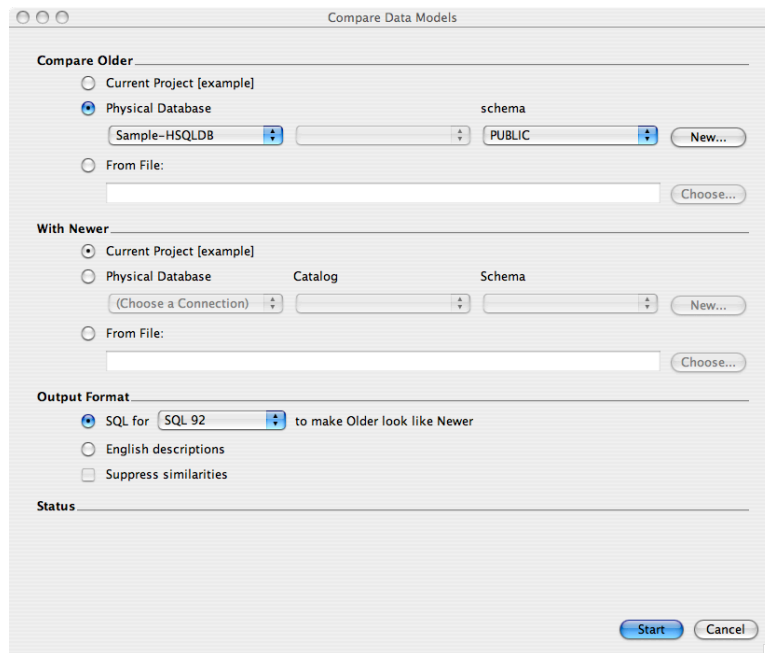
3. If this looks plausible, click Execute, and the tables and their relationship will be created. Congratulations! You have now created a simple database using the visual tools in Power*Architect.

Comparing Data Models

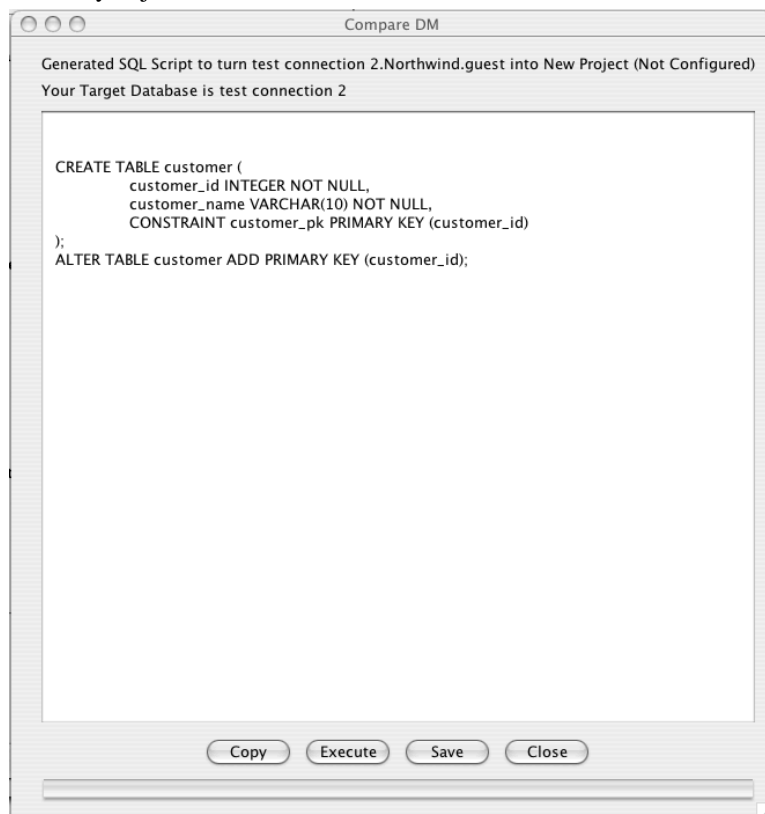
Suppose that after using this database, you realize that there should be a "shipping amount" field in the Order table (we never promised this would be completely realistic example).

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(10,2).

- 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.



- 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.

Using Diagram Components

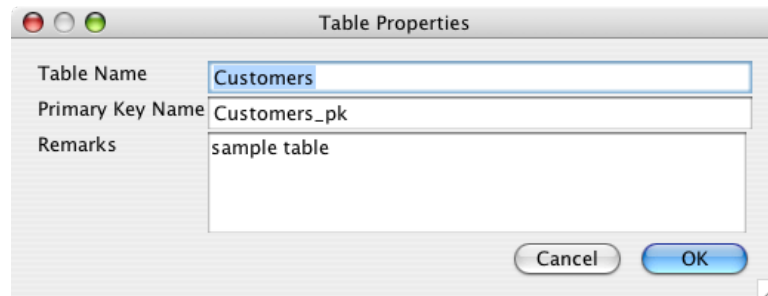
Creating New Tables



There are several ways to create new tables in Power*Architect. The first way is to click on the New Table Icon on the sidebar menu. The cursor turns into a "+" cursor indicating the mode change. Simply click on the playpen on the spot you desire to place the new table at. The second method to create a table is to right click on the playpen and select the "New Table" option. A newly created table will then be placed at the point of the right click. A third way is to type the letter T with the mouse over the playpen. To cancel creating a new table, press the esc button.

Editing Tables

To edit a table, right click on the table title and select "Table Properties". This pops up the Table Properties dialog.



In this dialog, you can:

- Change the name of the table
- Rename the primary key section of the table
- Add comments/notes about the table

Creating New Columns



First select the table the new column will be placed in. Click on the "Insert Column" button and a column is created in the selected table. Another way to add a new column is to right click on a table and select the "New Column" option. The new column will be added to the table below the selected column. It will be placed at the bottom of the table if no columns are selected. If a table has existing columns in the primary key and you wish to create new columns within the primary key, select a column that is already

in the desired primary key and then create a new column. The newly created column will be placed within the primary key as well.

Editing Columns



Select the desired column, right click and select "Column Properties...". The Edit Column Properties dialog pops up. Or you can select the column and click the "Edit Column" button on the Playpen toolbar.

Column Properties of Customers

Source Database: None Specified

Source Table.Column: None Specified

Name: customer

Type: INTEGER

Precision: 10

Scale: 0

In Primary Key: ☒

Allows Nulls: ☐

Auto Increment: ☐

Sequence Name: Customers_customer_seq

Only applies to target platforms that use sequences

Remarks:

Default Value:

Cancel OK

In this dialog, you can:

- Rename the column
- Change the type of data the column holds
- Set the precision of the data
- Set the scale
- Indicate if the column is in the primary key or not
- Indicate if the column should handle null information or not
- Indicate if auto increment is allowed or not
- Add additional comments about the column
- Set a default value for the column

The Power*Architect does not format the default value in anyway, you are responsible for entering a valid default value. The following examples are valid:

- '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

A special feature of the Power*Architect is that if a column originated from a database, the Power*Architect is able to remember the database and table it originated from.

Dragging Columns

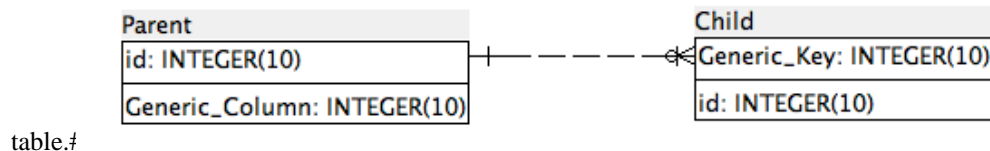
The Power*Architect allows you to drag columns within a table's key and from table to table freely. Simply select the desired columns and drag them into the desired destination. For now, only one column can be moved at a time.

Creating Non-Identifying and Identifying Relationships

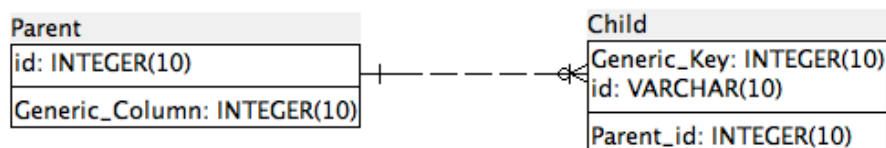
 To create a relationship, select the desired type of relationship on the Playpen ToolBar. The non-identifying relationship icon is  (keyboard shortcut is shift+R). The identifying relationship icon is  (shortcut key is R). The cursor changes to the "+" cursor to indicate the mode change. First click on the parent table and then click on the child table. Once this has been done, the relationship will be created, and will appear as a link between the two tables.

To cancel creating a new relationship, press the esc button or click anywhere in the background playpen.


If a column in the child table shares the name of the primary key of the parent table and it is the first relationship between the two tables, the relationship will map to the existing column in the child

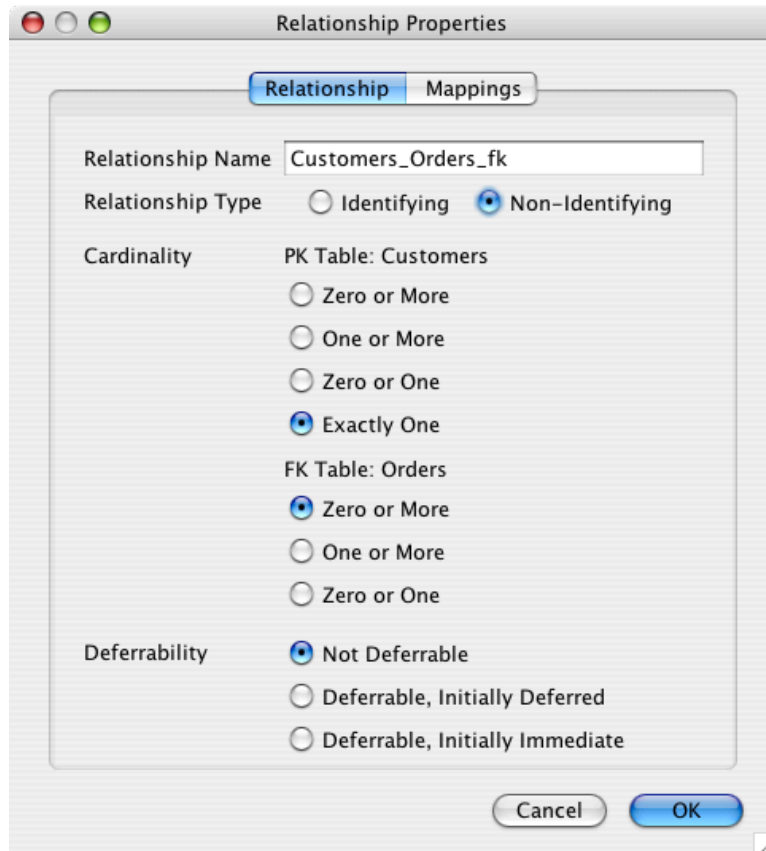


If a column exists in the child table that has the same name as the primary key in the parent table but a different data type or a relationship already exists, a new column will be created.



Editing a Relationship

 Right click on the relationship you wish to edit. Choose the "Relationship Properties" options. This can also be done by selecting the relationship and clicking on the relationship properties button. In both cases, the Relationship Properties dialog will appear.



In this dialog, you can:

- Rename the relationship
- Choose the relationship type
- Change the primary key end cardinality
- Change the foreign key end cardinality

Selecting Multiple Items in the Playpen

There are two ways to select multiple items in the playpen. One way is to hold down the shift key or the ctrl key as more items are being selected. The alternative method is to use the selection box.

Relocating Objects in the Playpen

The Power*Architect allows diagram objects to move around freely in the playpen. To do so, first select the items you want to move in the Playpen. Click and hold on one of the selected item and drag the items to a desired spot on the playpen.

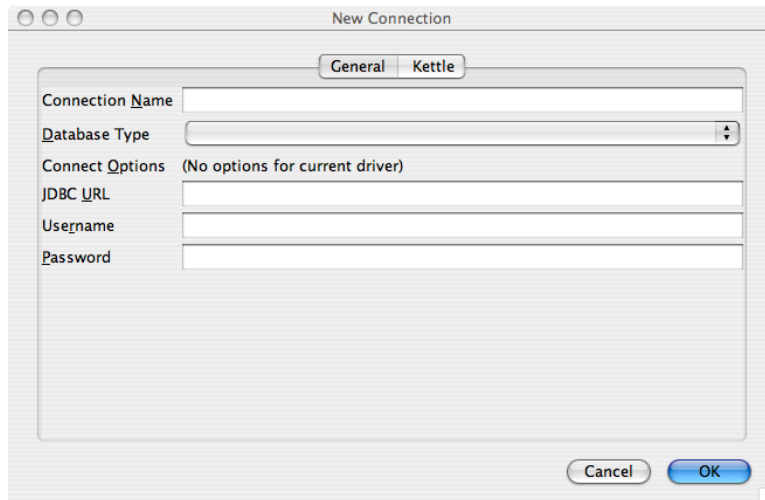
Deleting Diagram Components

Select the desired diagram components on the playpen and click on the delete items button on the Playpen Toolbar. It is also possible to delete the selected items by right clicking on one of the components and selecting "Delete Selected" or simply pressing the delete keys with the unwanted items selected.

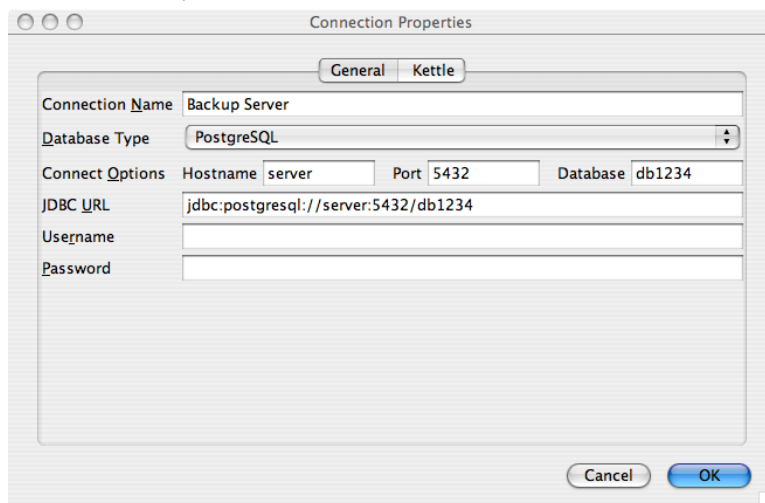
Database Usage in Power*Architect

Adding a New Connection to the Power*Architect

To add a new connection, go to "Add Connections" under "Connections" menu and select "New Connection..". The other method is to right click a white space on the Database Tree and select "New Connection" under "Add Connections". Both ways open up this dialog:

The image shows a "New Connection" dialog box with a title bar containing three window control buttons. It has two tabs: "General" (selected) and "Kettle". The "General" tab contains several input fields: "Connection Name" (text), "Database Type" (dropdown menu), "Connect Options" (text, with a note "(No options for current driver)"), "JDBC URL" (text), "Username" (text), and "Password" (text). At the bottom right are "Cancel" and "OK" buttons.

You must know which type of database you wish to use before you can connect to a database. When you have selected the database type, the Connect Options will change to allow you to enter the particular parameters that the given database driver needs. If you are using one of the fully-supported drivers, then as you enter these parameters, they will be 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 example below, we've selected the PostgreSQL driver and entered the hostname and database name (the "port number" was already filled in; do not change this unless the database server software has been reconfigured to use a different value).

The image shows a "Connection Properties" dialog box with a title bar containing three window control buttons. It has two tabs: "General" (selected) and "Kettle". The "General" tab contains several input fields: "Connection Name" (text, value "Backup Server"), "Database Type" (dropdown menu, value "PostgreSQL"), "Connect Options" (text, with sub-fields "Hostname" (value "server"), "Port" (value "5432"), and "Database" (value "db1234")), "JDBC URL" (text, value "jdbc:postgresql://server:5432/db1234"), "Username" (text), and "Password" (text). At the bottom right are "Cancel" and "OK" buttons.

When you are finished, press the OK button. Any new connection will automatically be added in the user-preference.

Adding an Existing Connection

Right click on empty space in the Database Tree and go to "Add Connection" to see a list of all database connections that were previously stored on the Architect.

Editing Database Connection Properties

Select the database connection you wish to change and go to "Connections" menu and select the option "Connection Properties.." which leads you to the Connection Setting dialog. An alternate solution is to right click on the database and select "Connection Properties..." option.












Removing a Database Connection

Select the database connection you wish to change and go to "Connections" and select the option "Remove Connection". Right clicking on the database connection and selecting "Remove Connection" will perform the same action. Connections can only be removed if they are not being used as sources in the play pen.

Navigating through the Database Tree

The Database Tree works like a tree-dropdown model. Clicking on any item will cause the component to expand display the items under that specific component. Each item will have a unique icon beside its name to identify the type of object it is. The table below shows what each icon means:

Table 3.1. Database Tree Icons

Icon	Representation
	Database
	Catalog
	Schema
	Owner
	Table
	Column
	Exported Key
	Imported Key
	Index
	Index Key
	Unique Index

Moving Items from the Database Tree

To copy items from the Database Tree to the playpen, simply select the desired items in the tree and drag them onto the playpen. Depending on the size of the items dragged onto the playpen, it will take some time to load.

Undo and Redo

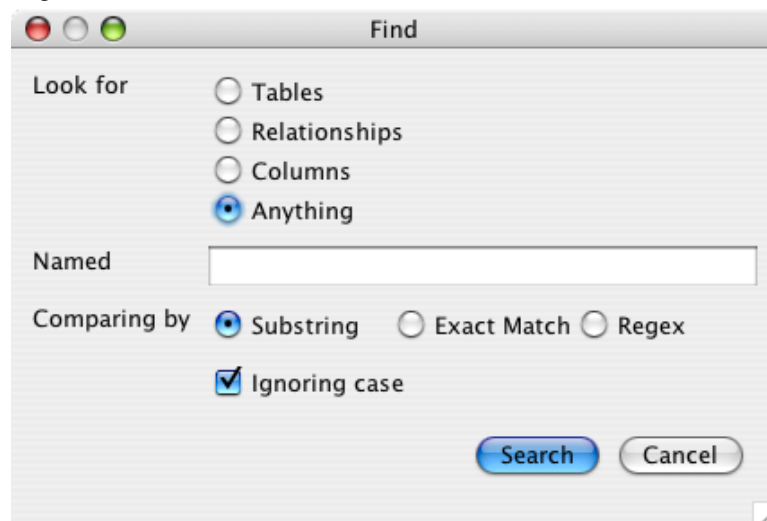
Like most modern application software, the Architect keeps track of your actions and allows you to undo them at a later time. Currently, the 100 most recent actions you have performed in the Architect are remembered and can be undone in sequence. If you undo too far back, you can redo any of the actions you have just undone. Be careful, however: After undoing one or more actions, once you make a new change, your redo history will be lost.

The undo and redo features can be accessed from the Edit menu, or by their keyboard shortcuts of Control-Z and Control-Y respectively (on Mac OS X, use the Command key instead of Control).

Find/Replace Function

What It Does

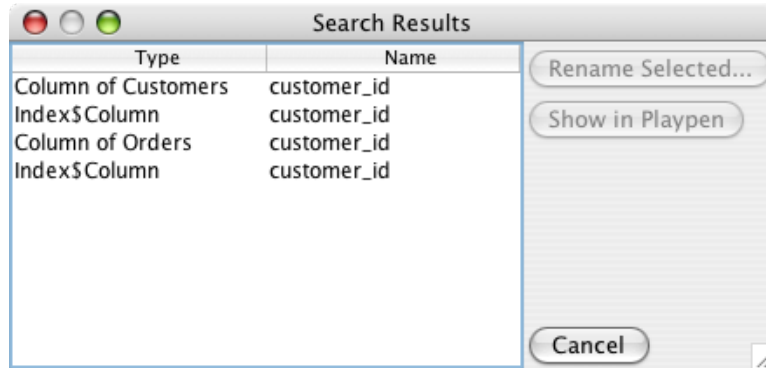
This function searches the playpen for any relevant matches within the search constraints and displays the resulting matches. If matches are found, you can have them selected in the playpen. This operation will also scroll the playpen to ensure the selected item is visible on the screen. This is particularly useful with large data models that do not fit on the screen all at once.



How to Use Find / Replace Function:

Go to "Edit" under the menu bar and click on Find/Replace option. This shows the Find/Replace function. In this window, enter the search constraints and press "OK" when you are finished. This will popup a new window with your search result in a chart format.

You can rename the component by selecting the column in the list and clicking on "Rename Component". You can also have Power*Architect focus on a certain component by selecting the component in the list and press the "Show in Playpen" button.



Forward Engineering and Compare Data Model

These two functions are similar; they both involve using the PlayPen (usually) and generating SQL. The Forward Engineer function always creates a SQL script to generate a database identical with the complete current project (current PlayPen contents). The Compare DM function can output either an English-language description or a SQL script describing the differences between two databases, or stored projects, or one of these and the PlayPen.

Forward Engineering

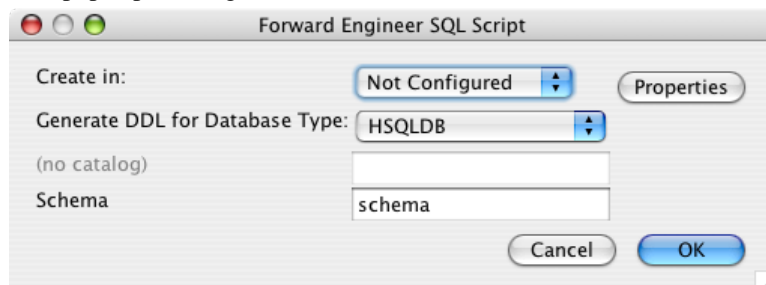
What It Does

One key design principle of the Power*Architect is that the data model remains generic for its entire lifetime. Forward#engineering is the process that makes the transformation to the specified database platform. During this process, the Architect attempts to create the physical model that most closely represents the idealized generic model, given the constraints of the target system. The result is a SQL Script that can be run to update or put the components of the current Playpen into a database.

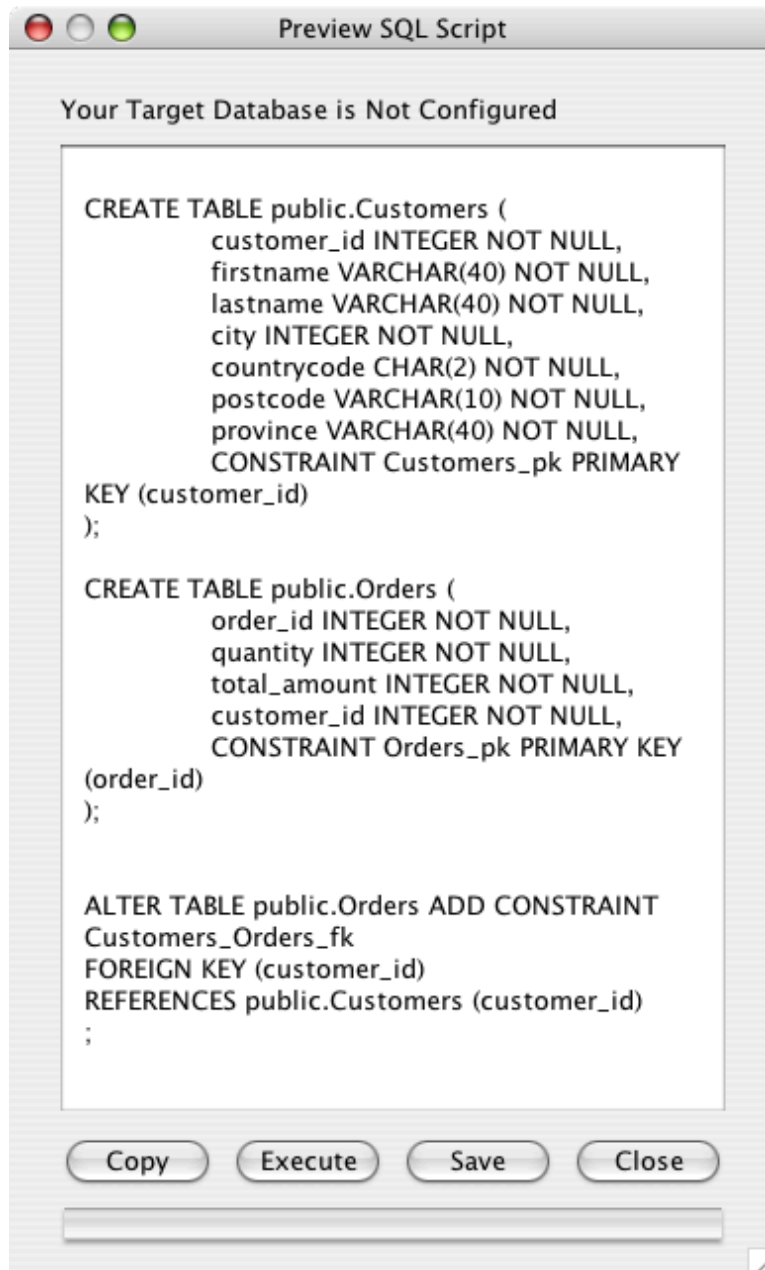
How To Use Forward Engineering



SQL First set the target connection to the database you want the changes to be made in. Then go to "Tools" and click "Forward Engineering". Another way is to press the "Forward Engineering" button at the top. This pops up a dialog that looks similar to the one below:



Fill in the fields as necessary and hit "OK" when you are done. Depending on the situation, a dialog warning you of possible side-effects of creating the script may pop up. Finally a script that would create data structure currently in the Playpen will be displayed. It is the same dialog used in CompareDataModel-Part 2- In SQL Script.




Compare Data Model Function

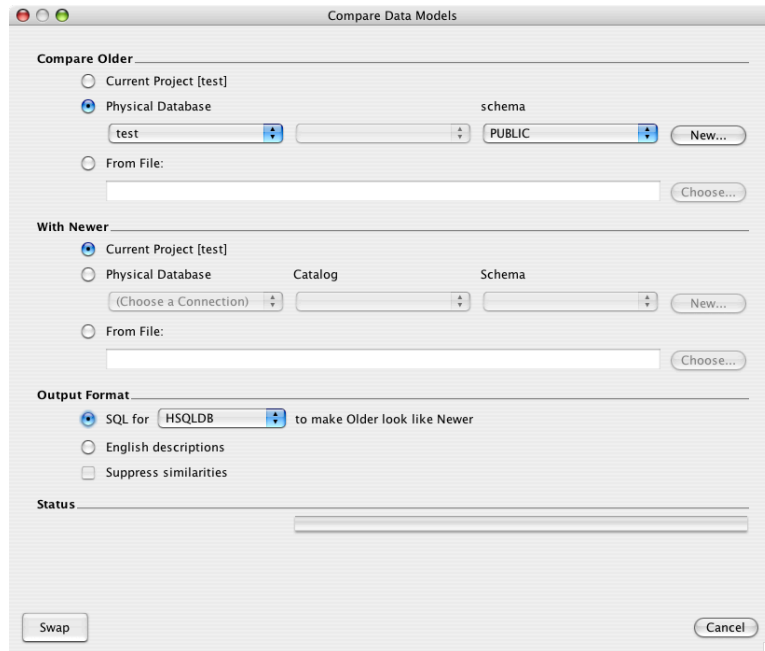
What It Does

The Compare DataModel Function takes two databases or Power*Architect projects (or one of each) or the current PlayPen, and compares and contrasts the structure of the two database/projects.

How to use Compare Data Model Function:

Part 1

 To start the Compare Data Model Function, either go to "Tools" and click "Compare Data Model" or just click the "Compare Data Model" icon on the project toolbar. This will bring up the main Compare Data Model window. Here select the source and target models you want to compare and contrast. You can choose either the current playpen, an existing database or select a saved project file function. If one of or both the source and the target has invalid models, the start button will not enable. The similarities and differences can be displayed either in SQL Script language or in English.



Part 2

English Descriptions

If this option is selected, a side-by-side document will be displayed stating the similarities and differences of the source and target in plain English. If the "Suppress Similarities" checkbox is selected then similarities will not be shown and only the differences will be displayed. The left text gives English descriptions to make the source database look like the target database. In addition to the text, they are also colour coded. You can copy the text to the clipboard by pressing the copy button, or save the results to a text file. The table below explains what each colour means.

Table 3.2. Compare Database Model Colour Codes

Colour	Explanation of the Colour Code
Black	This component exists in both databases
Green	This component only exists in this database but not the other
Red	This component does not exist on this database but exists on the other
Blue	This component is a column and is on different keys in the two databases

In SQL Script

If this option is chosen, this will produce a SQL Script in the SQL dialect chosen in Step 1 to make the source database look like the target database. You can either copy the results to the clipboard, or save the results in a text file. If the source has a valid connection database, the Execute button will enable and you can directly execute the changes. If the source does not have a valid connection, the execute button is disabled.

Auto-layout

What It Does

The auto-layout feature automatically rearranges the selected tables (all tables if none are selected) in an organized manner. It is also fun to watch.

How to Use Auto-layout



Select the tables on the playpen that you want to organize and hit the auto-layout button at the top. If one or zero tables are selected, the program will auto-layout every table in the playpen. Note that the layout algorithm may produce a few surprises when run with a small number of tables; it works best for a large or medium-sized collection of tables.

SQLRunner

What It Does

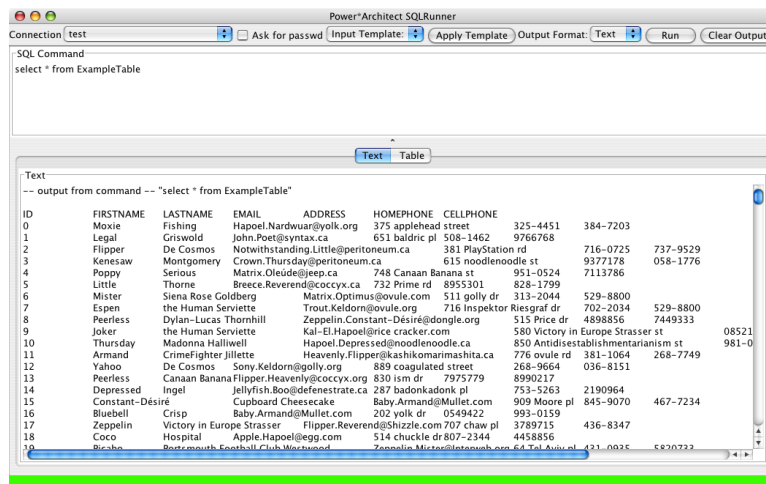
SQLRunner is a "fall-back" tool that lets you work at the raw SQL command level. This is an advanced topic and should only be used by (or made available to) those familiar with the intricacies of SQL commands and the details of your database; like a sharp knife, this tool is very useful in the hands of a skilled chef, but a slip of the fingers here can be quite messy...

SQLRunner was written by Ian Darwin, and is distributed under a liberal free-software, open-source license which permits its inclusion in programs such as Power*Architect.

How to Use SQLRunner

SQLRunner 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 SQLRunner are to type a command in the top (SQL Command) window and click the Run button; the results are displayed in the bottom (SQL Results) window. To save you some typing, there is a "Statement Template" mechanism that will insert a template for SELECT, INSERT or UPDATE SQL statements (just select the template you want and click "Apply Template" and the template will replace the current Input Statement.



The command can actually be one of two kinds: either one of a half-dozen escape commands listed below, or, anything that is valid input to your database's command interface (e.g., programs such as psql or Oracle SQL*Plus™).

Table 3.3. SQLRunner Escape Characters

Escape Sequence	Action
\dt	Describe list of all tables
\dtT	Describe column names of table named T
\dmX	Set the mode, where X is the first letter of the mode (t for text, s for SQL, h for HTML or x for XML; not needed in the embedded version because the GUI has a control for this)
\oF	Send output to the given file instead of the screen (though you can usually just view the output and copy-and-paste to save parts of it into a file; does not work in GUI versions).
\q	Exit the program (not supported in embedded versions).

SQL Statements are entered one at a time, can be more than one line long, and need not end with a semicolon. These statements are not interpreted by SQLRunner 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. Each SQL statement is executed in its own transaction context, that is, changes are committed immediately (so be careful!).

Output (Results) Window

Command Output in the chosen format (see below) appears in the SQL Output window. A scrollbar will appear if the information cannot all be seen at once.

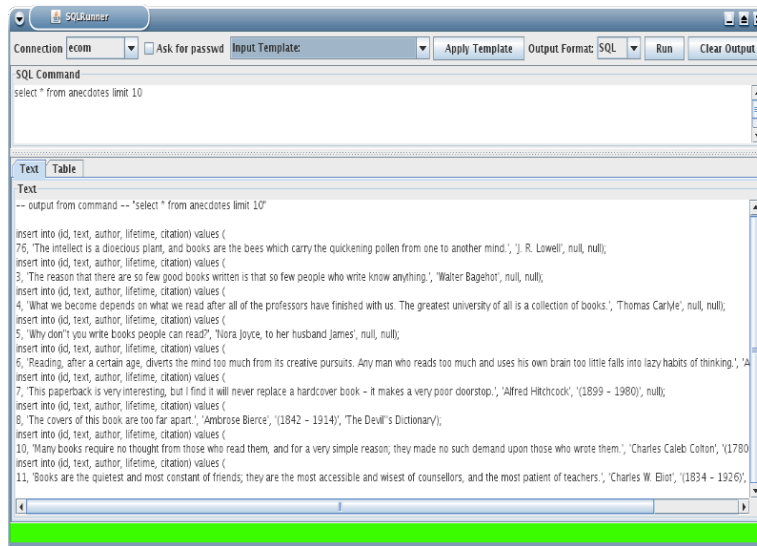
A visual indication of the success or failure of the command is displayed below the output: green for success, red for failure. As well, failures will be accompanied by a pop-up window containing details on the failure.

The Clear Output button clears the contents of the output window.

Output Formats

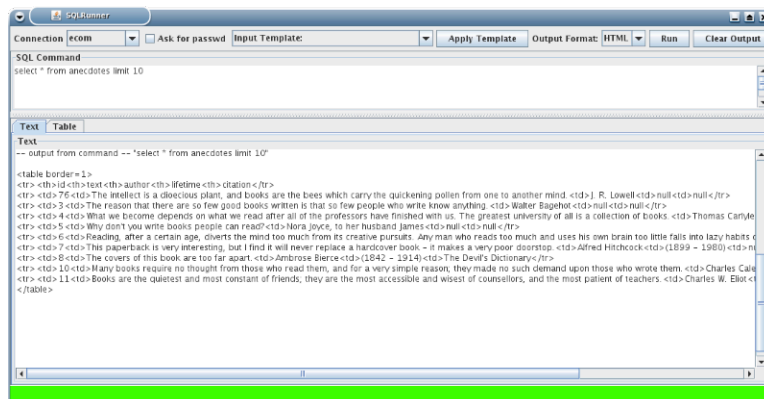
There are several output modes for the display of SQL "select" results: text, SQL, HTML, XML, and Table. Output from the escape commands are always displayed as plain text. Text mode is the default, and is primarily a raw display format. SQL output is most useful with the output of a SELECT statement; it will generate SQL that will attempt to re-create the data in another database. HTML mode generates an HTML table to display the results of a Select. XML format is similar but may be used for exporting data into other applications. Finally, table mode provides a friendlier interface which ensures all of the columns are lined up properly. In this mode, it is even possible to rearrange the columns by dragging them!

For example, with SQL mode selected, a "select * from anecdotes" (a table in a sample bookstore web site's database, used to display a casual quotation about books) looked like this:

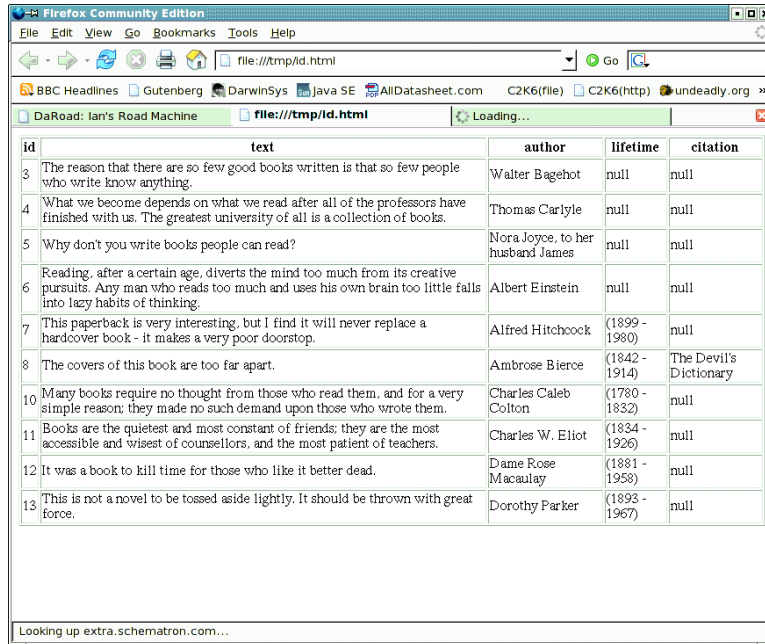


This could, as you can see, be used to create a SQL script to re-create the contents of the database. In fact, some developers use SQLRunner primarily for this purpose: to create stable test databases from "live" data that was created by their application.

You can view this same data in HTML just by changing the Format selection to HTML and clicking the Run button again:



When copied and pasted into an HTML file and viewed in a browser, the output looked like this:



id	text	author	lifetime	citation
3	The reason that there are so few good books written is that so few people who write know anything.	Walter Bagehot	null	null
4	What we become depends on what we read after all of the professors have finished with us. The greatest university of all is a collection of books.	Thomas Carlyle	null	null
5	Why don't you write books people can read?	Nora Joyce, to her husband James	null	null
6	Reading, after a certain age, diverts the mind too much from its creative pursuits. Any man who reads too much and uses his own brain too little falls into lazy habits of thinking.	Albert Einstein	null	null
7	This paperback is very interesting, but I find it will never replace a hardcover book - it makes a very poor doorstop.	Alfred Hitchcock	(1899 - 1980)	null
8	The covers of this book are too far apart.	Ambrose Bierce	(1842 - 1914)	The Devil's Dictionary
10	Many books require no thought from those who read them, and for a very simple reason; they made no such demand upon those who wrote them.	Charles Caleb Colton	(1780 - 1832)	null
11	Books are the quietest and most constant of friends; they are the most accessible and wisest of counsellors, and the most patient of teachers.	Charles W. Eliot	(1834 - 1926)	null
12	It was a book to kill time for those who like it better dead.	Dame Rose Macaulay	(1881 - 1958)	null
13	This is not a novel to be tossed aside lightly. It should be thrown with great force.	Dorothy Parker	(1893 - 1967)	null

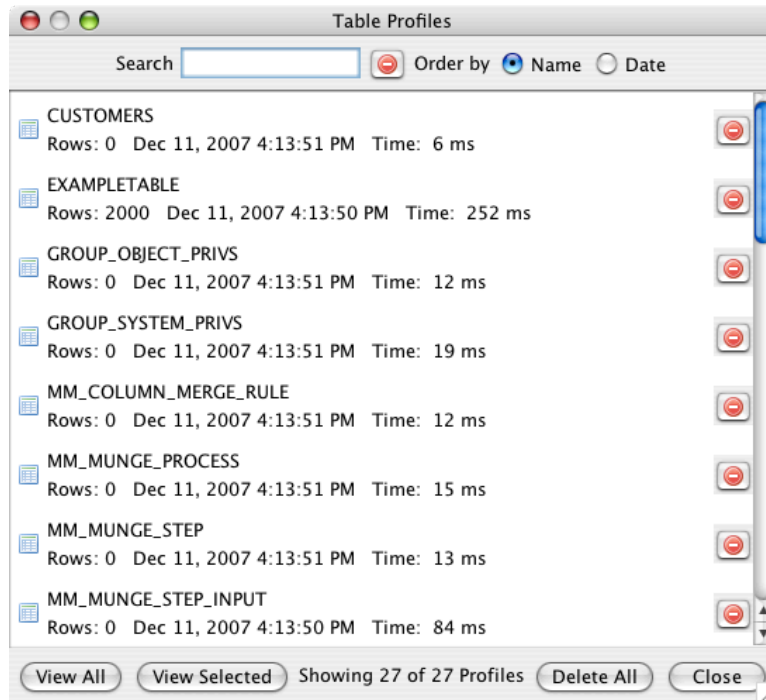
With a bit of formatting, or even a CSS style sheet, this HTML page could be made quite usable.

SQLRunner is not perfect, but it is adequate for many purposes involving direct use of SQL.

Profiling



Profiling displays a summary of the data found in a database. The summary can be used for such tasks as; database optimization and data migration. Select the columns and tables that you wish to profile from the database tree on the left hand side of the screen. Then activate the profile feature by either going to the "Profile" menu and selecting "Profile...", or by right clicking on a selected item and select "Profile.." from the context menu, or by selecting the Profile Icon in the Advanced Tool icon bar. If there is still an existing Profile window, the new profiling results will be added on to the existing window, otherwise, the resulting profile will be displayed in a new window.



Profiling Mode

The profiling mode can be changed under File -> Project Settings. There are two possible modes for profiling: First is the Remote Database option. This sends a query to the database and has it calculate all of the statistics. This system works well over a large network because very little data is transferred. The second option, Local Reservoir which transfers all of the data to the local computer and then samples and process the data there. 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.

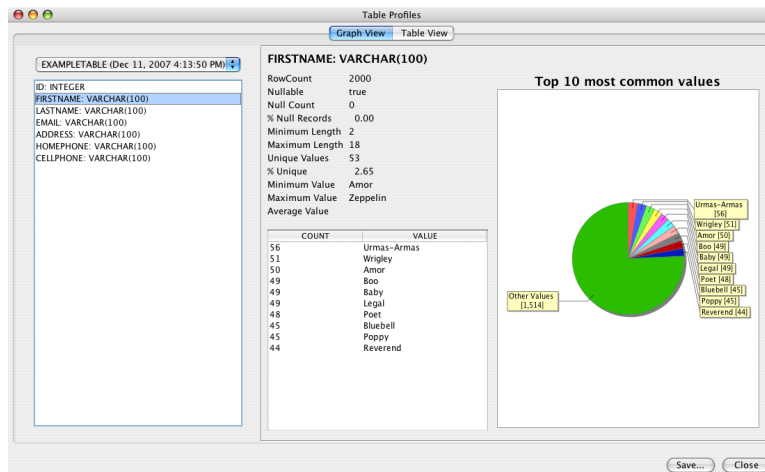
Table View

In the profiling window, the profiling information is sortable. Simply click on the column header and it will sort the data by ascending or descending order. In addition, if you place your mouse pointer over a most frequent cell, it will display the value and frequency of the most recurring items in the column. You can narrow down the results by using the search bar on the top right corner of the profile window. To delete columns from the profile result, simply select the desired columns and press the "Delete" button. As for refreshing the data within the profile table, select any one of the columns within the table and hit "Refresh". This will update the contents of the whole table. If you wish to save the profile results, you can highlight the desired columns to save, and click the "Save" button. If no column is selected, the Architect will save all the displayed results. You have the option to save it in CVS, PDF or HTML format.

Data...	Cata...	Sche...	Table Colu...	Run...	Rec...	Data...	# Null	% Null	# Un...	% Un...	Min...	Max...	Avg...	Min...	Max...	Avg...	Mos...
Ma...	null	PU...	EX...	AD...	20...	20...	VA...	0 0%	19...	97%	9	44 17	1 L...	99...	65...		
Ma...	null	PU...	EX...	CE...	20...	20...	VA...	0 0%	18...	93%	7	8 7	00...	99...	28...		
Ma...	null	PU...	EX...	EM...	20...	20...	VA...	0 0%	18...	92%	15	67 26	Am...	Ze...	Fli...		
Ma...	null	PU...	EX...	FIR...	20...	20...	VA...	0 0%	53 3%		2	18 6	Amor	Ze...	Ur...		
Ma...	null	PU...	EX...	HO...	20...	20...	VA...	0 0%	18...	92%	7	8 7	00...	99...	28...		
Ma...	null	PU...	EX...	ID	20...	20...	INT...	0 0%	20...	100%	1	4 3	0	1...	999	19...	
Ma...	null	PU...	EX...	LA...	20...	20...	VA...	0 0%	50 2%		3	33 11	Ali	the...	Cu...		

Graph View

Besides the tabular view, there is also the graphical view. Simply click on "Graph View" tab on the top to switch. On the left side of this window, you can select which column you want to profile. In the middle, it will show statistics about that particular column. It will also display the most frequent 'n' values and its frequency within the table. The pie chart on the right displays the portionality of the most frequent 'n' values in the column (the value of 'n' can be set in the "Project Settings" under the "File" menu).



How to Create a Kettle Job

What it Does

This feature allows a user to create a Kettle job and multiple transformations based on the information in the play pen. The Kettle job is used to take the data from the sources of the tables in the play pen and place them in the new tables of the target database.

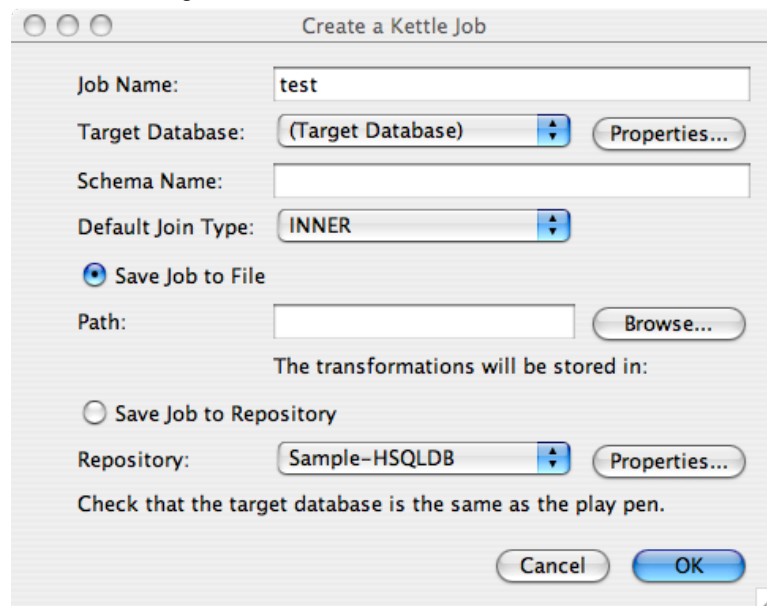
Setup for Kettle

Before you start creating jobs, some settings need to be configured. Go to the User Preferences and select the JDBC Drivers tab. For each driver that you use in a database connection, set the Kettle Connection Type on the Kettle sub-tab.

Creating a Kettle Job

Before creating a Kettle job we need to create the new database schema in the play pen. This includes creating new source connections, dragging tables into the play pen and modifying the play pen to have the desired layout. Once the play pen has the correct layout use the forward engineering tool to create the tables and relationships in the target database.

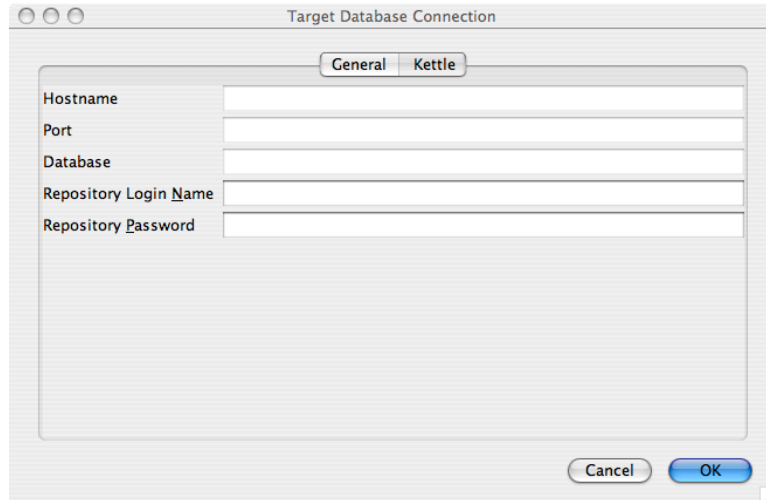
To start creating a Kettle job go to the ETL menu and select the "Create Kettle Job" menu item. You should see the following window.



Each Kettle job requires a name, a target database and either a file path or a repository to save to. To set the target database, click the "Properties" button. The default join type is used to define what join type will be used in all merge-joins. Merge-joins are used to create tables with multiple sources.

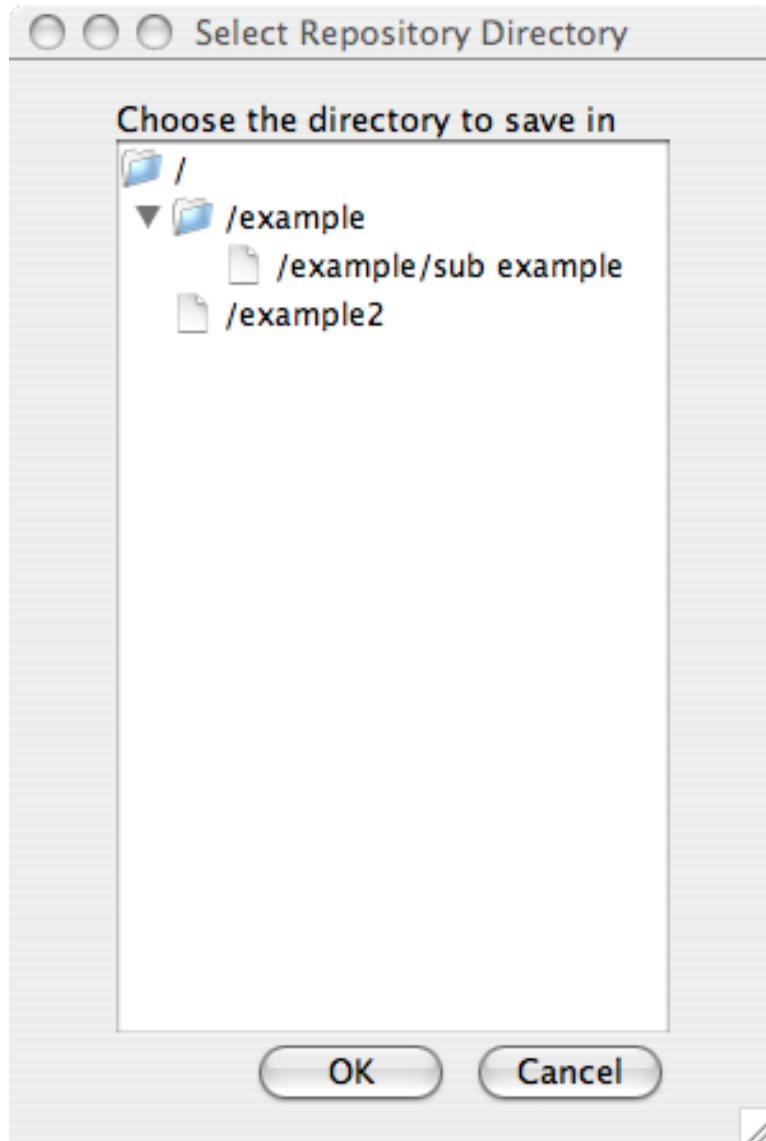
Note: Merge-joins that are created in transformations from Power*Architect will usually have to be updated manually. A manual update is required as Power*Architect cannot tell which fields to compare during the join.

When setting the target database, a hostname, port number and database name need to be specified. The URL template may contain properties for the database, host and/or port. If the template contains properties for Kettle, the values given in the URL will be used for the target connection in Kettle. If the template does not contain Kettle properties then they can be set on the Kettle tab.



To save the Kettle job and transformations to a repository you must first choose the repository option. Then, any of the source connections can be used as a repository provided they have a repository set up. If the desired repository is not one of the source databases the properties can be set manually. When setting up a database connection for use as a repository the login name and password can be set on the Kettle tab.

To start creating the Kettle job and transformation files, click the "OK" button. If a repository is being used a window will appear to choose the directory location in the repository to save to.



Once the job has been created, a window will display additional steps that need to be completed before running the job.

Note: The transformation files will be stored in the same location as the Kettle job. You must use Kettle to run the job.

Chapter 4. Database Product Notes

Database	Notes
Oracle	Is fully supported.
SQL Server	Is fully supported.
PostgreSQL	Is fully supported.
IBM DB2	Partial support; needs more testing.
HSQLDB	Works; used in samples.
Derby	Does not work; the current version (10.1.2) has what we consider some unwarranted chumminess with the JDBC Driver Manager that breaks because we use our own Java "ClassLoader"
MySQL	Is fully supported.

Chapter 5. Troubleshooting

We have worked hard to ensure that Power*Architect works correctly. However there are probably always going to be some combinations of different database products and database configurations, user actions, computer setups, and so on, that just don't work. We apologize in advance for any inconvenience this may cause...

If you are having trouble with Power*Architect, we may 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 [<http://www.sqlpower.ca/forum/forums/show/2.page>]

Chapter 6. Glossary

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

Column	The set of all instances of a given field from all records in a table [http://foldoc.org/foldoc/foldoc.cgi?table] .
Database	One or more large structured sets of persistent data, usually associated with software to update and query [http://foldoc.org/foldoc/foldoc.cgi?query] the data. A simple database might be a single file containing many records [http://foldoc.org/foldoc/foldoc.cgi?records] , each of which contains the same set of fields [http://foldoc.org/foldoc/foldoc.cgi?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	<p>A column [http://foldoc.org/foldoc/foldoc.cgi?column] in a database table [http://foldoc.org/foldoc/foldoc.cgi?table] containing values that are also found in some primary key [http://foldoc.org/foldoc/foldoc.cgi?primary+key] column (of a different table). By extension, any reference to entities of a different type.</p> <p>Some RDBMSs [http://foldoc.org/foldoc/foldoc.cgi?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.</p>
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 [http://foldoc.org/foldoc/foldoc.cgi?record] in a database, derived by

	<p>applying some fixed function to the record. The key is often simply one of the fields [http://foldoc.org/foldoc/foldoc.cgi?fields] (a column [http://foldoc.org/foldoc/foldoc.cgi?column] if the database is considered as a table with records being rows, see " key field [http://foldoc.org/foldoc/foldoc.cgi?key+field] "). Alternatively the key may be obtained by applying some function, e.g. a hash function [http://foldoc.org/foldoc/foldoc.cgi?hash+function] , to one or more of the fields. The set of keys for all records forms an index [http://foldoc.org/foldoc/foldoc.cgi?index] . Multiple indexes may be built for one database depending on how it is to be searched.</p>
Primary key	<p>The candidate key [http://foldoc.org/foldoc/foldoc.cgi?candidate+key] selected as being most important for identifying a body of information (an entity, object or record [http://foldoc.org/foldoc/foldoc.cgi?record]).</p>
Record (row)	<p>One or more structured sets of persistent data, usually associated with software to update and query [http://foldoc.org/foldoc/foldoc.cgi?query] the data. A simple database might be a single file containing many records [http://foldoc.org/foldoc/foldoc.cgi?records] , each of which contains the same set of fields [http://foldoc.org/foldoc/foldoc.cgi?fields] where each field is a certain fixed width.</p>
SQL	<p>Originally SEQUEL [http://en.wikipedia.org/wiki/SQL#History] 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.</p>
Table	<p>A collection of records [http://foldoc.org/foldoc/foldoc.cgi?records] in a relational database [http://foldoc.org/foldoc/foldoc.cgi?relational+database] .</p>

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

Chapter 7. Acknowledgements

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 the Architect's user interface.

The Silk icons are freely available under the Creative Commons Attribution 2.5 License [<http://creativecommons.org/licenses/by/2.5/>]. 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 [<http://www.famfamfam.com/lab/icons/silk/>].

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 the Architect. The text of the Apache License follows, because we are redistributing several Apache libraries upon which the Architect depends.

The following license applies to these library jar files, which are distributed as part of the 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

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:

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

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

(c) 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

(d) 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

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.

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 Architect download:

- forms-1.1.0.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:

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

o 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.

o 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 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.

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 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.

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 Architect download:

- itext-1.4.8.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-INFRINGEMENT. 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 declaratory 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 INCLUDING, 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 LIMITATION 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.]

JFree

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 Architect download:

- jcommon-1.0.0.jar
- jfreechart-1.0.1.jar

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 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, with the addition of the following:

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.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library used for any Combined Work is the version of the Library used by the Application.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding the source code for the Application.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, as well as the object code and/or source code for the Minimal Corresponding Source, as applicable.

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, you may:

- * a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the facility, the Combined Work can still run, and
- * 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 an object code under section 4 of this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the facility, the Combined Work can still run, and

- * 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 governed 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

- * a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and i
- * 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
- * d) Do one of the following:
 - o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application
 - o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses
- * e) Provide Installation Information, but only if you would otherwise be required to provide such information under sec

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 libr

- * a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other lib
- * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where

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 ti

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered ve

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Publi

Darwin Systems

Thanks to Ian Darwin of Darwin Systems for his many contributions to the Power*Architect. SQLRunner is part of his darwinsys Java library, which we redistribute with the Architect.

The following license applies to:

- darwinsys.jar

Copyright (c) Ian F. Darwin, <http://www.darwinsys.com/>, 1996-2006.

All rights reserved. Software written by Ian F. Darwin and others.

\$Id: PowerArchitectUserGuide.xml 2024 2007-12-20 19:18:06Z sean.oczkowski \$

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 author nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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 AUTHOR 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.

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

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

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 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/> [<http://www.junit.org/>] .

The Eclipse Foundation

The Power*Architect was primarily developed and tested using the Eclipse [<http://www.sqlpower.ca/>] Java Development Tools, one of the more productive Java environments around.

Sun Microsystems

Last but not least, many thanks to Sun Microsystems [<http://java.sun.com/>] 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 the 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 PARTICULAR PURPOSE OR NON-INFRINGEMENT. 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 PARTIES, 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 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 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.