



# **Pulsonix Database Connection**

## **Users Guide**

### Copyright Notice

Copyright © WestDev Ltd. 2000-2025

Pulsonix is a Trademark of WestDev Ltd. All rights reserved. E&OE

Copyright in the whole and every part of this software and manual belongs to WestDev Ltd. and may not be used, sold, transferred, copied or reproduced in whole or in part in any manner or in any media to any person, without the prior written consent of WestDev Ltd. If you use this manual you do so at your own risk and on the understanding that neither WestDev Ltd. nor associated companies shall be liable for any loss or damage of any kind.

WestDev Ltd. does not warrant that the software package will function properly in every hardware software environment.

Although WestDev Ltd. has tested the software and reviewed the documentation, WestDev Ltd. makes no warranty or representation, either express or implied, with respect to this software or documentation, their quality, performance, merchantability, or fitness for a particular purpose. This software and documentation are licensed 'as is', and you the licensee, by making use thereof, are assuming the entire risk as to their quality and performance.

In no event will WestDev Ltd. be liable for direct, indirect, special, incidental, or consequential damage arising out of the use or inability to use the software or documentation, even if advised of the possibility of such damages.

WestDev Ltd. reserves the right to alter, modify, correct and upgrade our software programs and publications without notice and without incurring liability.

Microsoft, Windows, Windows NT and Intellimouse are either registered trademarks or trademarks of Microsoft Corporation.

All other trademarks are acknowledged to their respective owners.

Pulsonix, a division of WestDev Ltd.

Printed in the UK Issue date: 03/06/25

### Pulsonix

20 Miller Court  
Severn Drive  
Tewkesbury  
Glos  
GL20 8DN  
United Kingdom

Phone +44 (0)1684 296 551  
Email [sales@pulsonix.com](mailto:sales@pulsonix.com)  
Support +44 (0)1684 296 570  
Email [support@pulsonix.com](mailto:support@pulsonix.com)  
Web [www.pulsonix.com](http://www.pulsonix.com)

# Contents

<b>CHAPTER 1. GETTING STARTED .....</b>	<b>5</b>
About these notes .....	5
System Requirements .....	5
Installing the Database Connection .....	5
Licensing .....	5
Finding More Information .....	6
<b>CHAPTER 2. OVERVIEW .....</b>	<b>7</b>
What is the Pulsonix Database Connection? .....	7
What kind of database can I use? .....	7
What if I don't have a database? .....	7
<b>CHAPTER 3. DATABASE CONNECTION SETUP .....</b>	<b>9</b>
Introduction .....	9
The Setup Dialog .....	9
Database Connection Configuration Files .....	10
Connection To Database .....	10
Change .....	10
Browse .....	11
Server Name and Port Number .....	11
User Name and Password .....	11
DSN .....	11
Conn String .....	11
Connecting to the Database .....	11
Database Tables and Mappings .....	12
Choosing which tables you see .....	12
Mapping table fields to attributes / properties .....	12
Interpreting Alternative Values .....	13
Partname Field, Key Field and Pulsonix Parts .....	15
Replicating your mappings .....	15
Alternative 'Map To' Attribute for Partname .....	15
Saving your settings .....	16
Excluding Tables from the Database Connection Setup .....	16
Overriding the Default Database Query Timeout .....	17
<b>CHAPTER 4. ADDING COMPONENTS .....</b>	<b>19</b>
The Database Bar .....	19
Searching .....	20
Adding search criteria .....	20
Search Results .....	21
Saving Searches .....	21
Tab Searches .....	21
PDC Tab Controls .....	22
Adding Components .....	22
Viewing Database Attributes .....	23
Using Synchronise and Update .....	23
<b>CHAPTER 5. MAINTAINING YOUR DESIGN .....</b>	<b>25</b>
Check Against Database .....	25
What to Check/Update .....	25

Showing the Check Results .....	26
The <Database Status> attribute .....	27
Updating the Design .....	27
How Updating Works .....	27
Updating Local Parts .....	28
Other Options .....	28
Change Part.....	28
Using the Database Bar to Change Part .....	28
Database Information.....	29
Replace Part .....	29
Database Information.....	29
Replace Part for Whole Design.....	30
Database Part Variants .....	30
<b>CHAPTER 6. REPORTING .....</b>	<b>31</b>
Report Maker .....	31
Reporting 'Loaded' Database Information .....	31
Reporting 'Live' Database information .....	31
Running a Database Check from a Report.....	32
Other Reports.....	32
<b>CHAPTER 7. MANAGING LIBRARY DATA.....</b>	<b>33</b>
Local Parts .....	33
Finding Local Parts.....	33
Adding information for local parts to database .....	33
<b>CHAPTER 8. MIGRATING LIBRARY DATA .....</b>	<b>35</b>
Library To/From Database.....	35
Choosing transfer direction.....	35
Which libraries to transfer.....	36
Overwriting existing items.....	36
Transfer from database to libraries.....	36

# Chapter 1. Getting Started

## About these notes

This Users Guide is provided for existing users as a supplement to the main Pulsonix Users Guide, and describes the features of the Pulsonix Database Connection and how to use them.

## System Requirements

The **Pulsonix Database Connection (PDC)** uses **ODBC** (Open Data Base Connectivity), an industry-standard method of interfacing to databases. As such, the Database Connection should work with a large range of database types, including database servers, database files, and also various 'flat' files such as Excel, Text and CSV. In short, anything for which you can set up an ODBC driver.

Other than the presence of the appropriate ODBC driver(s), the Database Connection has no other particular system requirements.

Due to the large number of possible database types, vendors and options, we cannot guarantee that all possible databases will be accessible using the Database Connection. However, the Database Connection has been designed to be as flexible as possible to maximise the range of database types with which it can be used.

## Installing the Database Connection

The Pulsonix Database Connection is installed as an integral part of the standard Pulsonix application. Once you have installed Pulsonix, no additional installation is necessary. The connection to your database will need to be configured, this is covered in the next chapter.

No ODBC drivers are included with Pulsonix. These drivers are generally part of your database software installation, or in some cases included as part of Windows, or may for example be downloaded from the Internet and installed.

## Licensing

Access to the features of the Database Connection is controlled by license. If you have purchased the Database Connection cost option as an addition to your existing Pulsonix installation, you will be provided with a new license file containing the necessary licenses to access the Database Connection features.

### Finding More Information

The next chapter gives you an overview of how the Database Connection fits within your working environment, and then rest of the document describes how to use the various features of the Database Connection to carry out a range of tasks:

- setting up the connection to your database
- adding components to a design
- checking and maintaining your design
- extracting information with reports
- managing libraries in relation to the database

You can also access context-sensitive online help using the F1 key at any point in the application.

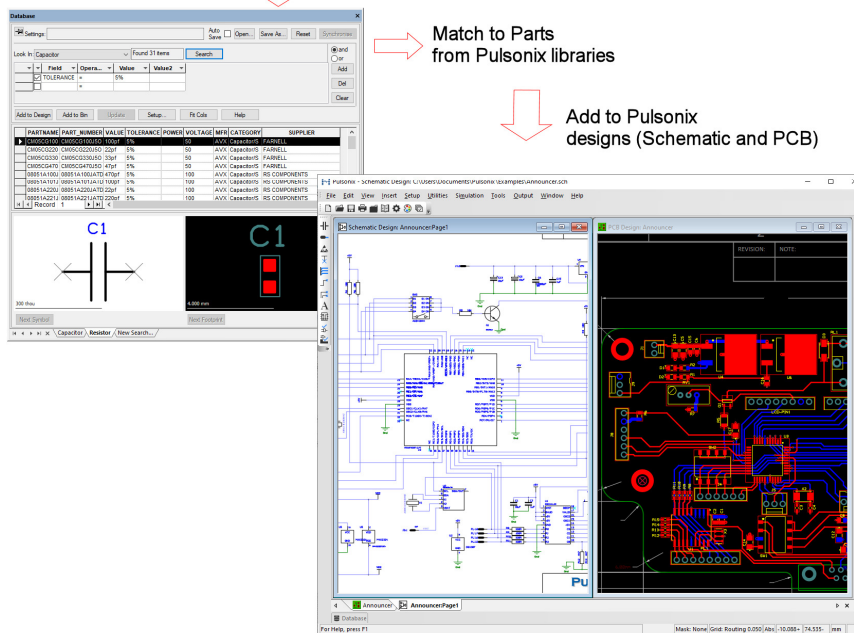
## Chapter 2. Overview

### What is the Pulsonix Database Connection?

The Pulsonix Database Connection (PDC) allows you to ‘hook up’ your Pulsonix program to your corporate component database, to help you find the right components for your designs using parametric searches on any of the properties recorded in the database.

Once you have found the database component you need, the corresponding Pulsonix library items can be added to your design. All the necessary attributes are added to the new component to allow changes in the database details to be tracked.

Search corporate database  
for required components



### What kind of database can I use?

There is really no limit to the type of database you can use with PDC. Various MRP, CIS and other ‘corporate’ type databases can be used, as well as a range of simpler sources of information such as Excel spreadsheets and even text or CSV files. In short, anything that can be connected using the industry-standard ODBC should be suitable for use with PDC.

### What if I don’t have a database?

If you don’t already have a database, you can use the tools provided with PDC to help you create a database from all the parts in your Pulsonix libraries. You will still need to decide on a suitable database to store the information, but the tools will help you ‘populate’ a database as a starting-point for working with PDC. This is described in Chapter 8.





## Chapter 3. Database Connection Setup

### Introduction

Before you can start using the Database Connection, you will need to prepare your system to make the necessary connections and specify how your database information is to be accessed and used.

You do this using the Database Setup dialog, which can be accessed using the **Database Connection** command on the **Setup** menu, or by clicking the **Setup** button on any of the database-related dialogs or dockable bars.

### The Setup Dialog

The **Setup** dialog is divided into sections, each of which is described below:

Database Connection Setup

Database Connection Configuration File  
 C:\Users\Documents\Pulsonix\Databases\PDCdemo.dcc  
 Open... Save As... OK Cancel

Connection To Database  
 Connection Type: Access Change...  
 Path Name: %HERE%\pdcdemo.mdb Browse... OK (No Save)

User Name: admin  
 Password: Disconnect

Database Tables and Mappings  
☐ Specify which tables are visible to searches  
☒ Define table properties and field mappings  
 Apply To All Tables

Table Name: Resistor  
 Key Field: PART\_NUMBER  
 Partname Field: PARTNAME

Field Name	Show	Map To	Load	Critical	Hyperlink	Interpret As
PARTNAME	<input checked="" type="checkbox"/>	PARTNAME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PART_NUMBER	<input checked="" type="checkbox"/>	PART_NUMBER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VALUE	<input checked="" type="checkbox"/>	VALUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TOLERANCE	<input checked="" type="checkbox"/>	TOLERANCE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
POWER	<input checked="" type="checkbox"/>	POWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VOLTAGE	<input checked="" type="checkbox"/>	VOLTAGE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MFR	<input checked="" type="checkbox"/>	MFR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CATEGORY	<input checked="" type="checkbox"/>	CATEGORY	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
IN_STOCK	<input type="checkbox"/>	IN_STOCK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Alternative 'Map To' Attribute for Partname:

## Database Connection Configuration Files

All the information relating to your connection to a particular database is stored in a Database Connection Configuration (DCC) file. You can set up several of these files if you wish, perhaps if you need to connect to different databases. Use the Open button to open an existing DCC file, and the Save As button to save the current configuration to a different file. By default, the Database Connection will attempt to use the same configuration file as last time when it starts up. If you have write permission to this file, any changes made on this Setup dialog will be written back to that file when you click OK.

If you wish to work in a 'controlled' environment, so that Pulsonix users cannot modify how the connection to the database is made, then place the configuration file in a location (a network shared folder, perhaps) where the user does not have write permission. They will then be able to read the file and load its settings, but they will not be able to change them.

When you are first setting up the connection to your database, you can skip this section of the dialog until your connection is established, then use **Save As** to save all the connection and configuration details to a DCC file for later re-use.

## Connection To Database

This section of the **Setup** dialog describes the database to which you will connect, and how to connect to it. The actual entries visible on this part of the dialog will depend on the type of database selected, once you have chosen one from the Connection Type dialog.

For example, if you are connecting to a MySQL database, you would need to specify the location of the MySQL database server and how to connect to the correct database on that server. The following dialog is automatically displayed when a new database type is selected:

The screenshot shows the 'Database Connection Setup' dialog box. It is divided into two main sections. The top section, 'Database Connection Configuration File', contains a text box with the path 'C:\Users\Documents\Pulsonix\Databases\PDC.dcc' and buttons for 'Open...', 'Save As...', 'OK', and 'Cancel'. The bottom section, 'Connection To Database', contains a 'Change...' button and several input fields: 'Connection Type' (MySQL), 'Database' (MPF\_ERM), 'Server Name' (ADMGR), 'Port Number' (3600), 'User Name' (ADM\_MPF), and 'Password' (masked with dots). There are also 'OK (No Save)' and 'Connect' buttons.

## Change

To select the type of database, click the **Change** button to open the Database Type dialog. From this dialog you will be able to choose from a range of pre-defined database types.

If your database type does not appear in the list, you should still be able to connect to your database if you have the appropriate ODBC driver(s) installed on your system. In that case, you will need to choose either "DSN" or "User-defined Connection String",

depending on how your ODBC setup is configured and where the database is stored. Here is an example Connection Strings for a Paradox database server using ODBC 5.x driver:-

```
Driver={Microsoft Paradox Driver (*.db
)};DriverID=538;Fil=Paradox
5.X;DefaultDir=c:\pathToDb\;Dbq=c:\pathToDb\;Collati
ngSequence=ASCII;
```

You should be able to find example connection strings for your particular database either in your database documentation, or on the Internet (try searching for 'ODBC connection string').

## Browse

Some database types (Microsoft Access, for example) will reference a specific file. Use this button to browse to the location of that file.

## Server Name and Port Number

The name (or IP address) of the machine on which the database server is running, and the TCP port number of the database server. Some databases are accessed through a database server 'process', rather than directly to the database file.

## User Name and Password

The user name/password to be used when connecting to (logging into) the database. If no user name and/or password is required, just leave the boxes empty.

## DSN

The Data Source Name (DSN) to connect. DSN's are similar to Connection Strings, but you set them up using the **Data Sources** component of your Windows **Control Panel**, then reference them by name.

## Conn String

The user defined connection string, for those database connections/types that are not specifically provided for in the Database Type dialog. You will need to specify the complete ODBC connection string to hook up to your database, as outlined above.

## Connecting to the Database

On the Setup dialog, the **Connect** button allows you to open a connection to your database. When preparing your database connection for the first time, click **Connect** once you have entered all the configuration details on the dialog. If everything is okay and a connection to the database is established, the dialog will 'extend' downwards to allow you to move to the next stage of the setup process.

However, if there are errors connecting to the database (for example an incorrect password), you will need to make the necessary corrections before you can connect to the database and proceed with the setup.

## Database Tables and Mappings

A database consists of a number of 'tables', with each table having a number of 'fields'. The second part of the setup process allows you to specify which tables you want to use, and how the fields in each table are to be treated.

Having connected to your database, the **Database Tables and Mappings** section of the Setup dialog becomes available. The Database Connection will retrieve information about the structure of your database, and present this on the Setup dialog as a list of available tables within the database, and fields within each table. You do not have to enter this information; the Database Connection will work it out for you once the connection to your database has been made.

### Choosing which tables you see

The next task is to specify which of the possible database tables should be 'visible' when searching for component information. Click the **Specify which tables are visible to searches** radio button if it is not already selected, and you should see a list of the tables in your database.

Alongside each table name is a checkbox that is initially checked to indicate that the table is 'visible', but you can uncheck any of them to 'hide' that table from view.

Table Name	Visible
3M	<input checked="" type="checkbox"/>
Actel	<input checked="" type="checkbox"/>
AD	<input checked="" type="checkbox"/>
Altera	<input checked="" type="checkbox"/>

### Mapping table fields to attributes / properties

As well as deciding which tables are to be visible, you can also specify how each field in a particular table is to be handled. Click the **Define table properties and field mappings** radio button, and the dialog will change to show you information about a particular table.

Field Name	Show	Map To	Load	Critical	Hyperlink	Interpret As
PARTNAME	<input checked="" type="checkbox"/>	PARTNAME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PART_NUMBER	<input checked="" type="checkbox"/>	PART_NUMBER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VALUE	<input checked="" type="checkbox"/>	VALUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TOLERANCE	<input checked="" type="checkbox"/>	TOLERANCE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
POWER	<input checked="" type="checkbox"/>	POWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VOLTAGE	<input checked="" type="checkbox"/>	VOLTAGE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MFR	<input checked="" type="checkbox"/>	MFR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CATEGORY	<input checked="" type="checkbox"/>	CATEGORY	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
IN_STOCK	<input type="checkbox"/>	IN_STOCK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SUPPLIER	<input checked="" type="checkbox"/>	SUPPLIER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

**Table Name** - the name of the database table to set up.

**Key Field** - this is the name of the database field that holds the values that uniquely identify each different component in your database table. This might be a part number, or perhaps a supplier order code. It is this field (combined with the table name) that provides the link back to the database record from the component once it has been added to the design.

**Part Name Field** - this is the name of the database field that holds the name of the Part in your Pulsonix library. The value in this field will be used to retrieve the appropriate part to be added to your design when you choose a particular component record from the database.

**Field Name** - this column in the grid lists the names of all the fields in the currently selected database table.

**Show** - check this box to show this field in the 'search results' grid on the Database Bar.

**Map To** - this is the name of the attribute in the design in which this database field should be stored. The default setting is for them all to be mapped to the same name as the field in the database itself, but you can choose to enter different names if you wish.

**Load** - check this box to load the value of this field into the corresponding attribute on components as they are added to the design. Note that the field selected as the 'Key Field' will always be set automatically to Load=true and you will not be able to deselect it.

**Critical** - check this box to mark this field as one which must have the same value on the component as it has in the database. Note that the field selected as the 'Key Field' will always be set automatically to Critical=true and you will not be able to deselect it.

**Hyperlink** - check this box to specify that this field holds the name of an external document. You will then be able to right click on this field in the 'search results' grid on the Database Bar and choose Execute Hyperlink to view the file.

**Interpret As** – Allows additional meaning to be attached to textual database fields for the purpose of filtering and sorting. Unit and magnitude information specified alongside numeric values will be correctly interpreted when comparing and matching different items. *A more detailed description is in the section below.*

## Interpreting Alternative Values

The **Interpret As** column appears to the right of the dialog.

Field Name	Show	Map To	Load	Critical	Hyperlink	Interpret As
PARTNAME	<input checked="" type="checkbox"/>	PARTNAME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PART_NUMBER	<input checked="" type="checkbox"/>	PART_NUMBER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VALUE	<input checked="" type="checkbox"/>	VALUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TOLERANCE	<input checked="" type="checkbox"/>	TOLERANCE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
POWER	<input checked="" type="checkbox"/>	POWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VOLTAGE	<input checked="" type="checkbox"/>	VOLTAGE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

A blank **Interpret As** entry indicates that it is not in use by the field. This is the default setting for existing users. If you do not wish to use this new feature, do nothing and leave it blank.

A numeric value is identified as one that begins with a digit and may have additional textual unitary information appended to or embedded within the numeric digits.

Using **Interpret As** allows searches such as 'Value between 5K and "10K' to be conducted which as well as finding resistors with values such as '5K' and '7.5 k', will also find those with values such as '8250', 5500R, 6K5 and 0.0095M.

Sorting on a field with an **Interpret As** setting will order items according to their interpreted value rather than in the default alphabetic order.

Below shows the effect on the sort order, setting a Value field to **Interpret As Capacitor Value**. The column to the left shows the standard alphabetic order while the column on the right shows the 'Interpret As' 'Capacitor Value' order.

0.33uf		10pf
100nf		22pf
10pf	-->	330pf
1uf		100nf
22pf		0.33uf
330pf		1uf

The provided **Interpret As** types that may be chosen are **Resistor Value**, **Capacitor Value**, **Inductor Value**, **All Values** or **Number**.

Field Name	Show	Map To	Load	Critical	Hyperlink	Interpret As
PARTNAME	<input checked="" type="checkbox"/>	PARTNAME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PART_NUMBER	<input checked="" type="checkbox"/>	PART_NUMBER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VALUE	<input checked="" type="checkbox"/>	VALUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Capacitor Value
TOLERANCE	<input checked="" type="checkbox"/>	TOLERANCE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
POWER	<input checked="" type="checkbox"/>	POWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Capacitor Value
VOLTAGE	<input checked="" type="checkbox"/>	VOLTAGE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Resistor Value
MFR	<input checked="" type="checkbox"/>	MFR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inductor Value
CATEGORY	<input checked="" type="checkbox"/>	CATEGORY	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Any Value
IN_STOCK	<input type="checkbox"/>	IN_STOCK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Number
SHIPPER	<input checked="" type="checkbox"/>	SHIPPER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

**Resistor Value** will recognise values such as 10, 10R, 10K, 10M, 1R5, 1K5.

**Capacitor Value** will recognise values such as 10mF, 10uF, 10nF, 10pF, 4.7u, 470p, 101, 102, 4n7.

**Inductor Value** will recognise values such as 10mH, 10uH, 10nH, 10pH, 101, 102, 3N3, 33N, N33.

**All Values** is intended for when a single table contains a variety of component types and will recognise any of the previous three types of values. By using an appropriate unit type in the search term the results can be filtered to the required type, e.g. "Value < 5K" will find all items with a resistor type value that evaluates to < 5000.

**Number** will interpret all recognised numeric values as an unmodified number and will disregard any additional unitary information. This is useful for simple numeric ordering.

All **Interpret As** types will recognise unmodified numeric values and will use them in context with other unitary values.

## Partname Field, Key Field and Pulsonix Parts

These two fields are essential to the operation of the Database Connection, and the program will require you to set these up as part of the setup process.

To add components to a design based on information from the database, there **must** be a corresponding Part in the Pulsonix library. The database contains your corporate information (component and supplier details, pricing, and so on), and the Pulsonix libraries contain the information that relates only to Pulsonix (Parts, Symbols and Footprints).

The **Partname** Field is the database field that is used to map database records to those Parts. This relationship may be one-to-one (one database record corresponds to one Pulsonix Part), but it can also be one-to-many with many database records referencing the same Pulsonix Part. The values of the fields on the selected database record that make them different will get added to the component as it is added to the design.

The **Key** Field is also important. It is the field in your database table that uniquely identifies the selected record. Without this field and its value, there is no way to find your way back to the matching database record from a component in a design. If your database is set up with a one-to-one correspondence between records and Parts, you could use the same field for both **Partname** and **Key** fields. Otherwise you will need to choose a different field for the **Key** field, perhaps something like one of the supplier part or order numbers, something that is unique to each record in the table.

## Replicating your mappings

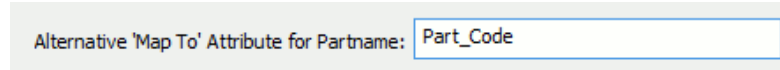
When you have set up the right mappings for the first table, if all your tables have the same 'structure' (i.e. the same field names that you want to map in the same way) you can save time by clicking **Apply To All Tables**, which will copy the settings for the current table across to all the others.

Another shortcut you might find useful is one that allows you to set all the cells in a column to the same setting. For example, to set all the fields to Load=true, check the box on the first one, right click on it and choose **Apply To Entire Column**.

## Alternative 'Map To' Attribute for Partname

There is an alternative means of mapping the database records to Parts in the Pulsonix library using an Attribute name. This can be useful where your database does not contain a field to map to a Pulsonix Partname, but your Pulsonix Parts have an attribute that uniquely identifies the appropriate database record.

On the **Database Connection Setup** dialog in the **Database Tables and Mappings** section, select the **Define table** properties and field mappings radio button. The following new control can be found at the bottom of the dialog:



Alternative 'Map To' Attribute for Partname:

---

NOTE: For the Database Bar to be able to use this alternative mapping, the specified attribute **MUST** also be added to the list of indexed attributes using the **Part Library Manager Index** option.

---

As an example, consider a database whose tables have a unique key field called 'Article\_No' but no field equivalent to the Partname for the parts contained in the Pulsonix

library. However each of the parts in the Pulsonix library, in addition to their Partname, also have an attribute called 'Part\_Code' which contains a value equivalent to the 'Article\_No' field in the database.

The field mappings entry for the 'Article\_No' field should have its **Map To** value set to 'Part\_Code'. By also setting **Alternative 'Map To' attribute for Partname** to 'Part\_Code', you indicate that in the absence of a Partname mapping, records in the database tables should be linked to Pulsonix parts by mapping the 'Article\_No' to the Pulsonix 'Part\_Code' attribute. The **Key Field** for each table must still be set to a uniquely identifying field, but in most cases, as with this one, will be the same field as mapped to the attribute, i.e. 'Article\_No'. As we have said, there is no database field to map to the Pulsonix Partname so the **Part Name Field** should be set the same as the **Key Field**, again 'Article\_No' in this case.

Now, when a database record is selected for adding to the design its 'Article\_No' value will be used to locate the equivalent Pulsonix part by matching its 'Part\_Code' attribute.

### Saving your settings

As mentioned above, once you have set up your desired configuration, you can save all the details in a Database Connection Configuration (DCC) file. Click **Save As** and browse to where you would like to save the file.

If you need to set up a number of users or computers to access the same database, you can place the DCC file in a shared location and have everyone point to the same file. Then you can be sure that everyone is accessing the database and using the information in the same way.

Write-protecting the 'master' DCC file will also prevent accidental changes to the configuration. If the file is write-protected, the Setup dialog will help you out by presenting much of the information as 'read only'.

### Excluding Tables from the Database Connection Setup

By default, all tables in your database will be made available for inclusion in the Pulsonix Database Connection. Even though a given table may not be chosen to be visible it will still be accessed to provide details to the Setup dialog. There can be occasions where it is desirable to exclude certain tables so they are completely ignored by the database connection, for example, a temporary or non-standard table can sometimes prove problematic to the connection process.

To achieve this, it is possible to manually edit the DCC file, using Windows Notepad for example, to add an Exclusions section naming tables to be completely ignored by the database connection. The Exclusions section should be added to the end of the DCC file immediately before the closing bracket as shown in the example below. For brevity, the detail of other sections is shown as '...':



```
(Pulsonix
  (HEADER
    ...
  )
  (Contents
    ...
  )
  (Mappings
    ...
  )
  (Exclusions
    (TableName "TempTab")
    (TableName "User")
  )
)
```

If the reverse situation is true and only a relatively few of the total number of database tables available need to be made accessible, an Inclusions section may be used instead. This is very similar in format to the Exclusions section shown above but is introduced by the keyword Inclusions rather than Exclusions. If an Inclusions section is present, only the tables it names will be accessible to the database connection and all other tables will be completely ignored.

## Overriding the Default Database Query Timeout

By default, all queries sent to the database will time out after a set period to avoid permanently blocking Pulsonix, causing it to freeze. This period will vary and will be defined by system parameters. Sometimes, particularly with a very large database, the time out period can be too short to allow the Database Connection to complete fetching the data it requires. In this case, it is desirable to be able to override the default time out period for Pulsonix without affecting the system wide default setting.

To achieve this, it is possible to manually edit the DCC file, using Windows Notepad for example, to add a Settings section that defines an alternative query timeout period. The Settings section should be added to the end of the DCC file immediately before the closing bracket as shown in the example below. For brevity, the detail of other sections is shown as '...':

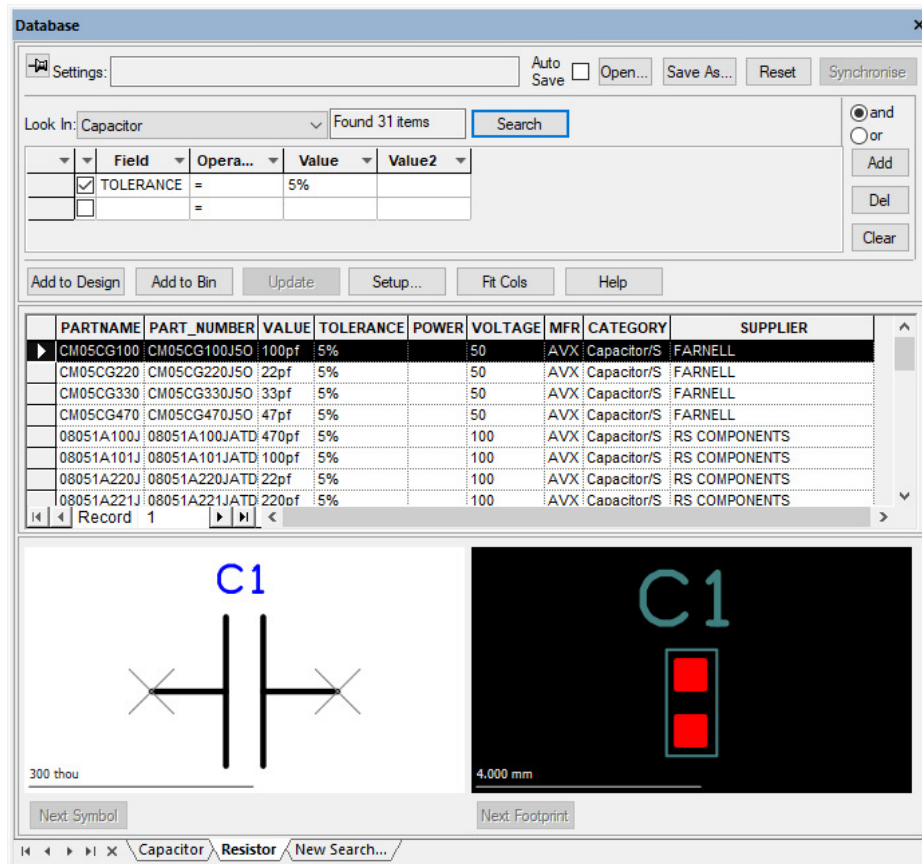
```
(Pulsonix
  (HEADER
    ...
  )
  (Contents
    ...
  )
  (Mappings
    ...
  )
  (Exclusions
    ...
  )
  (Settings
    (QueryTimeout 45)
  )
)
```

## Chapter 4. Adding Components

### The Database Bar

Instead of using the Insert Component dialog to add components directly from the Pulsonix libraries, the Database Bar is where you go to start your 'parametric search' of database information to identify the correct component to add.

The Database Bar is displayed from the **View** men and **Dockable Windows> Database Bar** command or **Component From Database** from the **Insert** menu.



As the Database Bar contains many controls, by default it is arranged 'horizontally' and docked at the bottom of the main Pulsonix application window, but as with other dockable windows you can re-arrange this if you wish. If re-sized for 'vertical' docking, some of the controls will be re-arranged automatically to accommodate the change in shape.

### Searching

The central area of the Database Bar is where you enter the criteria you want to use to search for the desired component in your database. Start by choosing the table to look in from the drop-down list of table names.

Now you can start entering your search criteria. If you press **Search** before entering any criteria, this will match with all possible records in the selected database table, and hence show you all the records in the 'results' grid. So you can either enter search criteria straight away, or start by querying all the records in the table and narrowing down your search from there.

### Adding search criteria

To add to the search criteria, click the **Add** button. This will add a new row to the table of search criteria. Under the **Field** heading, use the drop-down list to choose the field you want to search.

Next, choose the **Operator** from the drop-down list in the next column. A range of operators are available here, the most commonly used will be “=” to search for exact matches, and “begins with” and “contains” to search for parts of text fields.

Now enter the value to look for, in the **Value** column. If you have chosen “between” as the operator, you will need to add the second value under **Value2**, to search for fields that contain data between the two values you have specified.

If you need to add more search criteria, just click **Add** again and repeat the process. Once you have entered all the criteria, simply click **Search** to search the currently selected database table for records that match your criteria.

In this example, you can see we are looking for components with a 5% tolerance (TOLERANCE) and a Power (POWER) of 0.125W:

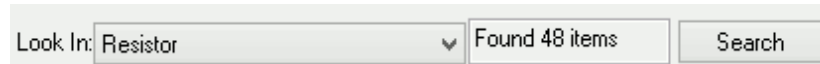
	Field	Operator	Value	Value2
<input checked="" type="checkbox"/>	TOLERANCE	=	5%	
<input checked="" type="checkbox"/>	POWER	=	0.125W	

If you want to remove any of the criteria, just click anywhere on its row, and click the **Del** button.

If instead you want to ‘turn off’ a particular criterion from the search but don’t want to delete it, uncheck the tick box at the left-hand end of the row to ‘disable’ that criterion.

## Search Results

When you press **Search**, the current search criteria are applied to the currently selected table (the table shown at **Look In:**). This searching is done using the ODBC database SQL facility, which provides an efficient method of searching large amounts of data. Depending on the type of database you are using, this search will either take place on your own computer or on the server where the database resides. In either case, the only records from the database that are passed back to the Pulsonix program are those that match the criteria; there is no shuttling back-and-forth of large volumes of data to carry out the actual search.



When the search is complete, the results – the records in the database table that match the specified criteria – are displayed in the lower grid on the Database Bar. Next to the Search button, the results are also displayed as the number of items **Found**.

The results are displayed with one row per matching database record, with the columns displaying the fields as defined in the database setup. You can sort the results on any of the fields by clicking on its column header.

## Saving Searches

If you have particular search criteria that you use frequently, you can save all the details as a Database Connection Search (“DCS”) file. Use the **Save As** button to save the current criteria to a DCS file.

To reload a previously saved DCS file, click **Open** and choose the file you want to load.

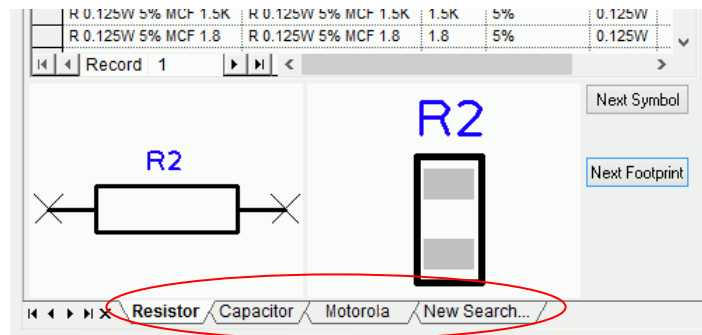
To wipe out all the search criteria currently shown, and start over with a new search, click **New**.

## Tab Searches

A new search can be created simply by clicking on the **New Search** tab. It inherits the name from the table you are searching but can be renamed to be whatever you choose.

To edit the tab name, double click on it and type in the new name. When you’ve finished, press the <Enter> key on the keyboard.

The search tabs can also be re-ordered by dragging and may be saved to a **Pulsonix Database Connection Search file** (.dcs) so they may be reloaded and shared by multiple users.

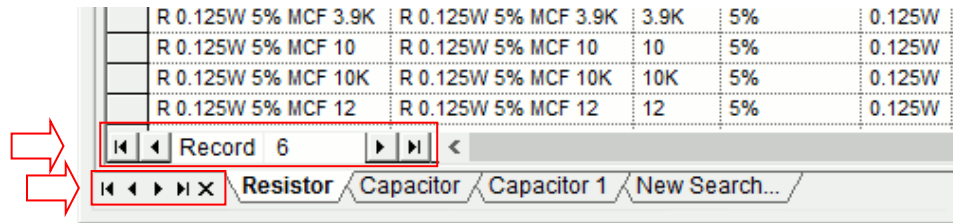


### PDC Tab Controls

There are two sets of controls at the bottom of the PDC dialog. The first displays the record number, effectively the line number in the current search. The arrow buttons either side of the record number allow you to move to the first, previous, next and last records in current search results.

The arrow buttons to the left of the search tabs enable you to scroll the tab display horizontally when there are too many to fit the current window size. The search tabs are only ever activated by clicking the mouse.

The **x** button on the right side of these controls enables you to delete the selected tab.



### Adding Components

Once you have found the component you want to use, selecting its record in the 'results' table should retrieve the part information from that record, and show the corresponding Schematic and PCB components in the preview boxes. If no corresponding Pulsonix library items can be found, a message in the preview box(s) will tell you this. For example, if you choose a database record for which you don't yet have a Pulsonix Part set up, neither preview will show a picture and you will not be able to add the selected component to your design.

Once a valid library Part has been found, the **Add** and **Add to Bin** buttons will become enabled, indicating that you are now able to add the selected component to your design. You can either click **Add** then move the cursor into the design area and place your new component, or move the cursor over one of the preview boxes and 'drag' the component into the design for placement. **Add to Bin** will simply add one instance of the selected component to the component bin for later placement on the design.

As part of the process of adding components to the design, those fields which were designated as 'Load' in the Setup dialog will be added as attributes to the component.

Note that you don't need to 'Load' all the fields from the database record into the design; you would only tend to load those fields that contain values of interest during the design process. Other fields from the database record will still be available to you later on using the **Report Maker** which knows how to retrieve values directly from the database at the time a report is generated.

## Viewing Database Attributes

The database attributes loaded into the design for a given component may be viewed at anytime by displaying its Component Properties dialog and selecting to the **Comp Attributes** tab. You can also choose to display them if you wish by selecting the checkbox alongside their name.

If, generally, you wish to display a particular database field alongside a component in the design, perhaps to aid identification, it may be better to add an appropriately named attribute position to the footprint used by the Pulsonix Part that the database record maps to. Then whenever a component with that database record is added to the design, the database field will be automatically displayed. The attribute position name added needs to be the same as the mapped name for the database field in question as set up in the Database Connection Setup dialog.

## Using Synchronise and Update

The **Synchronise** button, found at the end of the top row of buttons on the Database Bar, is useful when you wish to browse the database for parts similar to one you already have added to your design. It is only available when a single component is selected in the design, and when clicked, it primes the search criteria with the component's database attributes and performs an automatic search.

After using Synchronise you would expect to see the associated database record for the selected component. However, this may not be the case if the attributes and database values have become out of sync. The uniquely identifying **Key Field** is omitted from the search so other similar database records should also be displayed. If the selected component is using a *local* part, an attempt is made to automatically identify possible database matches using the Pulsonix part name from the selected component.

Having performed a synchronise it is then possible to modify the search criteria to widen the search for similar records if required.

The **Update** button works hand in hand with **Synchronise** and is only available when the currently selected record in the Database Bar is compatible with the currently selected component in the design. When clicked, it will update the design component's database link to be the currently selected database record.

The current database record will be compatible with the selected component if it maps to the same Pulsonix Library Part that is used by the component. To change a component's database record to one that maps to a different Pulsonix Library Part use the Change Part or Replace Part function both of which are covered later on in this document.





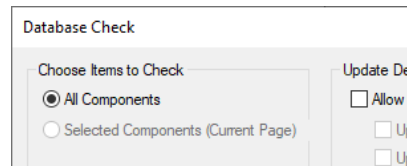
## Chapter 5. Maintaining Your Design

### Check Against Database

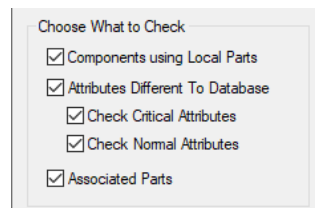
The **Check/Update Against Database** command, available from the **Tools** menu, is used to verify the information in your design against the corresponding records in the database. It will help you identify differences between design and database values, and can be used to 'refresh' the values in the design.

#### What to Check/Update

Under **Choose Items to Check and Update**, you can choose whether to check the whole design, or only the selected components (if any are selected):



Then you can choose from the various checks that are available:



#### Local Parts

**Components using Local Parts** will identify components in the design that are not linked to the corporate database and have been loaded (or reloaded) directly from the Pulsonix Parts library instead, without reference to the database.

These are 'Local Parts', which you may have in your design if you are using a component you have created that has yet to be approved as part of your company purchasing scheme and is thus not yet in the corporate database.

Checking for these Local Parts will help you to flag them up to your purchasing department, so they can start the process of approving them for purchasing and for addition to the database. Alternatively you might need to exchange them for similar parts that *have* been approved. Either way, this check will tell you which ones you need to think about.

### Attribute Differences

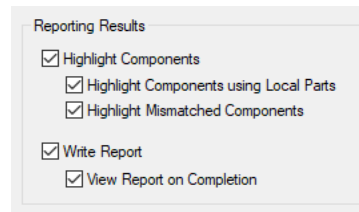
**Attributes Different To Database** allows you to check for components having attribute values that do not match the values defined for the corresponding database record. This check may be further sub-divided into a check on those attributes identified as **Critical** in the Setup, and/or **Normal** (non-critical) attributes.

### Associated Parts

**Associated Parts** additionally allows Associated Parts in the design to be checked to ensure they too are linked to the corporate database. If the **Attributes Different To Database** option is also checked, any associated parts attributes which do not match those values defined for the corresponding database record will be identified.

### Showing the Check Results

A range of options is available under **Reporting Results**:



The **Highlight** options use the three colours on the Highlights page of the Colours dialog to indicate Pass, Warning and Fail status on the components that are checked.

The results of the check and update can also be written to a report which can be automatically displayed on completion if desired. An example report is shown below:

#### Database Check

```
-----
Report Written : 16/04/2023 16:59:42
Design Path   : C:\Pulsonix\Designs\DBTest1.pcb
Design Title  : DBTest1

Checking All Components for:
  Components using Local Parts
  Components with Critical Attribute Mismatch
  Components with Normal Attribute Mismatch

R1      Pass
R2      Fail (TOL% [5:1])
R3      Pass
R5      Unchecked (Component 'R5': Key Field PIN (DB_PIN) =
"07130163" can not be found in database table [Resistor])
R4      Fail (Local Part)

Checking Associated Parts for:
  Local Associated Parts
  Associated Parts with Critical Attribute Mismatch
  Associated Parts with Normal Attribute Mismatch
M2-PLAIN-WASHER Fail (Local Part)
M2-PZPNHD-10 Fail (Local Part)

----- End Of Report -----
```

## The <Database Status> attribute

Performing a Database Check will assign a <Database Status> system attribute to each component covered by the chosen checks indicating the result of the check. The database status values are dependent on what checks are selected but generally indicate the following:-

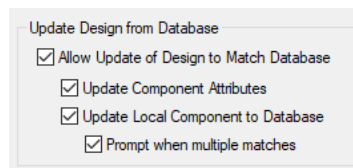
- |                  |   |
|------------------|---|
| <b>Pass</b>      | The component has an associated database record and its values for Critical/Normal attributes being checked matches those of the database record.           |
| <b>Fail</b>      | The component has no associated database record and is using a local part only or has Critical attributes that do not match its associated database record. |
| <b>Warning</b>   | The component has an associated database record but has Normal attributes that do not match its associated database record.                                 |
| <b>Unchecked</b> | The component could not be checked against the database for some reason.  |

The <Database Status> attribute may be used in conjunction with the **Find Bar** to identify and select components with a given database status.

To do this, use the **Attribute** portion of the Find Bar filter, to find items where the attribute <Database Status> is equal to one of the values shown above, e.g. "Fail" (without the quotes). This will list in the Find Bar all components with the given database status. You can then use the Find Bar functionality to browse to each one in turn if desired.

## Updating the Design

Under **Update Design from Database**, you can choose whether or not to 'refresh' your design with the latest values from the database:



## How Updating Works

As well as checking for discrepancies, it is also possible to automatically update components in the design to match the latest definition in the database. If the **Allow Update of Design to Match Database** option is enabled, in addition to the **Check** button, the **Update** button is also available. Clicking **Update** will perform a database check and attempt to update components that do not already match the associated database record.

If **Update Component Attributes** is checked, component attributes in the design that do not already match those in the associated database record will be updated to do so. These may be Critical or Normal attributes or both depending on what has been selected for checking.

If **Associated Parts** have been included in the checks then they will be included in the update as necessary.

The Database Check Report will indicate the updated components alongside the check that was not successful and caused the update. For example:

R2 Fail (TOL% [5:1]) UPDATED

### Updating Local Parts

The **Update Local Component to Database** option will attempt to associate local Pulsonix components to their equivalent record in the corporate database. To do this it will use their Part Name to search for an equivalent database record with a matching Part Name field as defined in the mapping in the Database Connection Setup dialog.

This feature is most applicable when there is a one-to-one relationship between the parts in the Pulsonix library and the records that defined the parts in the database. Where this is not the case and more than one database record maps to a given Pulsonix part then updating will match to the first one it finds in the database.

For more explicit mapping the **Replace Part** or **Change Part** facility should be used instead which will allow the appropriate database record to be identified and selected for each Pulsonix component.

Local components that do not find a database match will be left at 'Fail' status.

### Other Options

The **Default** button will return all dialog options to their default settings.

The **Reset** button provides a way to ensure that all the Database Check information in the design will be resulting from the next check performed, regardless of the settings chosen. It removes all legacy check data, i.e. highlighting and status attributes, from previous database checks ready for a new check.

The **Remove All** button provides the means to strip all database information from components and associated parts in the design effectively switching them all to be *local*. This would be use, for example, if you wished to switch to design to use a different database with a different set of mappings. Having performed a Remove All, the **Update Local Component to Database** option could be used to re-establish component links to the new database (assuming it had a suitable one to one mapping).

### Change Part

The **Change Part** dialog is used to choose an alternative for a given Pulsonix library part. It is invoked from various points in the application whenever a different part needs to be chosen. Using this dialog in the standard way will select the alternative part directly from the Pulsonix Parts Library. So if the part is for a component added through the **Database Bar**, this will result in the component losing its link to the corresponding database record and it will revert to being *Local*.

To avoid this, the Change Part dialog now has an additional **Change Database** button that will allow the alternative part to be selected from the database in the same manner as the original part was added, by using the **Database Bar**. To facilitate this task, the Database Bar is displayed in a restricted modal Change Part mode.

### Using the Database Bar to Change Part

When invoked in this mode, the Database Bar appears as a modal dialog with the title **Database Bar – Change Part**. The bar's standard **Add** and **Add To Bin** buttons are

replaced by **Change** and **Cancel** buttons, but other than this, will operate exactly as when being used to add components to the design allowing the database to be searched and browsed at will.

The component for which the part is to be changed provides a starting point for the searching the database and the search criteria are automatically primed with its database attributes, if it has any, and a search performed. Thus, for an existing database component, on initial display of the modal dialog you would expect to see the existing database record for the originating component. However, this may not be the case if the attributes and database values have become out of sync. The uniquely identifying **Key Field** is omitted from the search so other similar database records should also be displayed as possible replacements. For *local* parts, an attempt is made to automatically identify possible replacements using the Pulsonix part name from the original component.

Of course this initial search is only a starting point and the criteria can be amended as required to search and for a suitable replacement from anywhere the database.

Once an appropriate alternative has been identified the **Change** button should be clicked to confirm the selection, or alternatively if required, the **Cancel** button can be used to abort the database selection process. The modal Database Bar is dismissed and the program is returned back to the dialog that initiated it.

### Database Information

If you are using the Pulsonix Database Connection the Change Part dialog will display the Database Key Field which uniquely identifies the associated database record, rather than the Part Family.

### Replace Part

The existing **Replace Part** function, available from the **Utilities** menu, is used to replace the use of one part for another in a Pulsonix design. Like for Change Part, the standard use of this function will result in components using *Database Parts* losing their link to the database and reverting to being *Local*.

Similarly to Change Part, an additional **Database** button has been added to the Replace Part dialog to allow the replacement part to be selected from the database rather than directly from Pulsonix Library thereby retaining a link to the database.

### Database Information

To indicate that the replacement part has been chosen from the database, the Database Table Name and Key Field values will be shown appended to the Part Name in the dialog.

Similarly, the Replace Part Report will also show this database information as an additional **Database Key:** line in the report.

### Replace Part for Whole Design

Care should be taken when using Replace Part with the **Whole Design** rather than the **Selection Only** option as it will replace the selected Pulsonix Library Part for all components in the design that use it regardless of their associated database record. All the components' database links will be replaced with the database record associated with the chosen replacement part, if it was chosen using the Database button. If the replacement part was chosen directly from the Pulsonix Parts Library rather than from the database, all the affected components' will simply have their database link removed and they will revert to being *Local*.

### Database Part Variants

Pulsonix includes the facility to allow part variants in your design. This is where a given component represents a different part in different design variants. For example the component R1 may be a 5K resistor in the EU variant and a 2K resistor in the USA variant.

Normally, the different parts would be represented by different parts in the Pulsonix Parts Library. However, if you are using the Database Connection, the difference between two variants could be represented as two different records in the database, but with the same Pulsonix Library part. For example, the component R1 could use the Pulsonix part RES\_GEN in both variants, but map to the database record keyed on field PIN=005.00179 which has a field VALUE=5K in the EU variant and map to the database record keyed on field PIN=005.00180 which has a field VALUE=2K in the USA variant.

To enable such variants to be shown in the Variants tab of the Component Properties, it shows an additional **Database** column when the Database Connection is in use. This displays the component's database Key Field information so it is possible to see if this varies across different variants. If a component is *Local* to the Pulsonix Parts Library the Database column will be left empty.

For the case shown above, where a component uses the same Pulsonix Library part but maps to different database record in each variant, it is necessary for the component's variant information to be set up to allow at least different attributes across variants, i.e. have a different **Attribute Shared Identifier** in each variant. This will allow a different database link to be created for the component in each variant.

## Chapter 6. Reporting

### Report Maker

In addition to existing Report Maker commands, several new database specific commands have been added to allow access to the new Database Information that is now available in your Pulsonix design. You can even report 'live' information still held in the database directly from the Pulsonix Report Maker using the database links that you have setup.

#### Reporting 'Loaded' Database Information

The standard Report Maker **Attribute** command can be used to report the database fields 'loaded' in to the design as attributes for a component added from the database via the **Database Bar**. Only those database fields for which Load was set to true in the database mappings will be available to the Report Maker in this way.

Additionally, the system attributes attached to a component to hold its database information are also available to report using the Report Maker **Attribute** command. These include <Database Table>, <Database Field>, <Database Status>.

There is also the <Database Key> attribute which is available for components and which does not contain a value in its own right but will report the value of the attribute identified as the database key field by the component's <Database Field> attribute, if it has one.

There is also a new built-in **Database Status** command that will report the database status generated by the last Database Check that was run.

Database aware versions of the standard Parts List format files are included by default. These will show the Database Key Field value in preference to the Part Name for any component loaded from the database. Local components will continue to show the Part Name as before. These reports are available as **Parts List (Database)** and **Parts List (Database) CSV**.

#### Reporting 'Live' Database information

Fields from a component's associated database record not loaded with the component can still be reported, directly from database itself, using the new **Database Value** command. This command has the added advantage that the reported values will be 'live' and so guaranteed to be up to date. This makes it ideal for reporting information such as stock levels.

### Running a Database Check from a Report

A new **Database Check** command allows a database check to be performed as part of generating a report, in a similar fashion to the existing **Design Rule Check** command. The Database Check (described earlier) is run without using a dialog and using the complete set of checks to detect and report any database discrepancies. It will also update the Database Status of all the components in the design.

Database aware versions of the standard PCB and Schematics Acceptance Report format files are included by default. These will optionally run the Database Check, if required, and are available as **PCB Acceptance (Database)** and **Schematics Acceptance (Database)**.

### Other Reports

*See Checking Your Design.*



## Chapter 7. Managing Library Data

### Local Parts

*Local parts* are those which do not have a corresponding database record. These often occur early in the development cycle of a project, where new components have to be created as Pulsonix Parts and added to the design before they are approved for use and appear in the corporate database. These local parts can only be added directly from the Pulsonix libraries to your designs through the standard methods (Insert Component, Part Browser, etc).

Conversely, *database parts* are those which *do* have entries in the corporate database, and can therefore be added to your design using the Database Bar.

Local parts can be located using the **Check/Update Against Database** command, which will examine your design for components which do not have the necessary attribute information to relate them to records in the database.

### Finding Local Parts

If you use the **Check** command to check for just **Components using Local Parts**, you can use the **Find Bar** to browse your design to find the components in question as described earlier under 'The <Database Status> attribute'.

It is also possible to use the **Find Bar** to locate components using local parts without first performing a **Check/Update Against Database**. To do this, the Attribute filter should be set as <Database Field> Is 'does not exist'.

Alternatively, you can use the Report Maker to report only those components with the appropriate attribute value. This may be a better way to do it, because you could then report the other attribute values that are likely to be needed to add the necessary database records for those parts.

### Adding information for local parts to database

Once you have identified your local parts, you will need to add the necessary records to your database to match those parts. Where these records go, and how they will be added, will of course depend on your database configuration and how your database is managed. Some companies have a separate department or group to manage their database, others allow more general access.

Each record added to the database will need at least the following information:

- Pulsonix Part Name
- Key Field value (the unique identifier for this database record)

You may also need to supply the relevant attribute values to populate the rest of the fields in the database record.



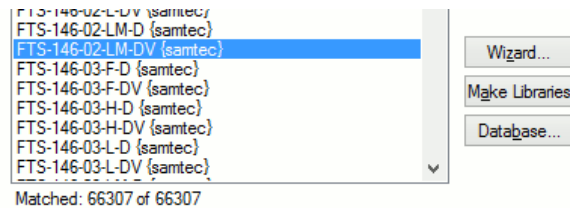
## Chapter 8. Migrating Library Data

You may already have a database to which PDC can connect. However, it is possible that you would like to use PDC but don't yet have a database set up. There are tools provided as part of PDC that will help you do this.

Similarly, if you have a large database of component information and want to start creating the Pulsonix parts to match, the tools will also help you get started on this task.

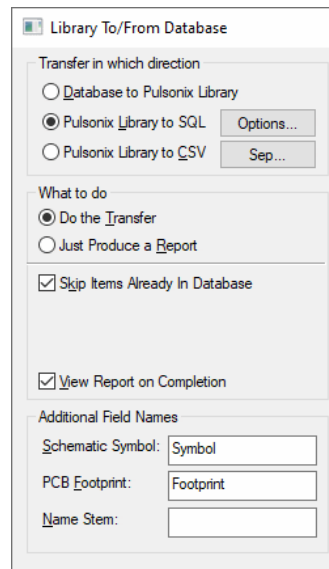
### Library To/From Database

From the **Parts** tab of the **Libraries** dialog, click the **Database** button at the bottom of the middle 'stack' of buttons, this opens the **Library To/From Database** option:



### Choosing transfer direction

The left-hand part of the dialog provides you with settings for controlling the direction in which data is transferred, and various options for how this is done.



Here you can see we have chosen to generate SQL data from the libraries.

This generates a file or set of files containing the SQL 'instructions' that can be processed by many database products to create and populate the database tables.

## Which libraries to transfer

As we have chosen in our example to transfer from Pulsonix parts libraries to database, the list shows us the Parts libraries currently available. The 'All' checkboxes can be used to quickly check/uncheck all the libraries in the list.

'Where to place results' controls whether to create one database table for each library or to lump all part details together into a single table. When transferring to one-per-library, you can also optionally prefix the table names with a text string.

Where to place results

☒ Create One Table Per Library Prefix:

In:  ...

☐ All Parts Into This Table:  New...

Choose Library Names:

- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\3m.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Actel.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\AD.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\ADSP.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Altera.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Altera2.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\AlteraCyclone.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\AMP.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Asmann.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Atmel.pal
- ☒ C:\Users\Documents\Pulsonix\MasterLibraries\Avago.pal

All None

Having made the choices, we can now click OK to begin the transfer. The selected libraries will be processed, and the required SQL files generated ready for loading into a database.

## Overwriting existing items

If you are already connected to your database through PDC, then unchecking the box 'Overwrite/Update Existing Entries' will compare the part names found in each Pulsonix library with the corresponding records in the relevant database tables, and only transfer the details for records that don't already exist in the 'target' file or table.

## Transfer from database to libraries

You can also transfer data from your component database, which will extract component records from your database tables and create 'skeleton' parts in Pulsonix parts libraries.

As only 'basic' component-level information is accessed, the parts created will at best only have part name, attributes, footprint and symbol name specified (assuming you have that information available in your database), but this will be a good starting point to getting the right Pulsonix parts created to use with PDC.