



Linking Existing Components to Your Company Database

Summary

Application Note
AP0134 (v2.0) June 08, 2006

This application note provides detailed information on linking existing Altium Designer components to an external database using Altium Designer's database link feature.

A common requirement for many organizations is to link from the electronic CAD components to the company's central component database. Altium Designer supports this by providing the ability to link existing components to records in the company database, through the creation and use of a database link file. The database link file is added to your design project or library package, allowing you to link from components already placed on schematic sheets or, more typically, defined components in a source schematic library, respectively.

Linking between schematic library components and an external database can only be achieved if the library is part of a library package (*.LibPkg). Choose **File » New » Project » Integrated Library** to create a new library package and then add the source schematic libraries to it.

Once linked, you can transfer mapped information from the database records to design parameters in the components using the **Update Parameters From Database** command. This information can then be included in the final Bill of Materials (BOM), ready for component procurement.

A Word About Linking Choices...

Altium Designer provides two methods of linking from a component to a database – using either a Database Link file (*.DBLink) or Database Library file (*.DBLib) respectively. The former provides an efficient means of linking and keeping the components used in your design (or libraries) synchronized with the data entered in the database. The latter takes this proven linking efficiency and adds to it the freedom to be able to place a component directly from the database – in essence creating the component dynamically from the information stored for it in the corresponding database record.

Before launching into the greater depths of the database link feature, it is worth discussing the differences between these two methods, in regard to the way you use them.

You can also link to a database using an SVN Database Library file (*.SVNDBLib). An extension of the DBLib linking model, this type of library is used in exactly the same way as a DBLib, but the schematic symbol and associated models are stored in libraries under version control (Subversion).

For more information, refer to the [Working with Version-Controlled Database Libraries](#) application note.

Linking using a Database Link file (*.DBLink)

Using this method, the Database Link file defines linkage between the schematic component and a matched record in a database. The record match is established by key field linking, which can be a single key field (for example a part number), or multiple key fields (by defining a **Where** clause).

With this method of linking, the model and parameter information for the component must be predefined as part of the Altium Designer library component. The library component must also include the necessary key field information as part of its definition. Once this has been defined you add a Database Link document to your Library Package or Design project, then you can synchronize the component information (parameters) with the contents of fields in the database.

Although each physical component defined by each database record does not need to map to a unique Altium Designer library component – many database components can share the same component symbol – this method of linking would typically be used in a "one database record-to-one Altium Designer component" fashion. The unique Altium Designer component can either be an instance placed on a schematic sheet, or a unique component in a component library.

With DBLink-style database linking, you include the Database Link file with the project.

Linking using a Database Library file (*.DBLib)

Using this method, the Database Library file also defines the linkage between the schematic component and a matched record in a database. Again, the record match is established by key field linking, which can be a single key field (for example a part number), or multiple key fields (by defining a **Where** clause).

With this method of linking the component symbol, model and parameter information for a component is stored as part of the record definition for that component in the external database. The referenced schematic component (in an underlying component library (*.SchLib)) is simply an empty shell, with a defined symbol only. There are no linked models and no defined design parameters.

When the component is placed, its parameter and model information is created on-the-fly, using the corresponding fields in the matched database record and in accordance with defined mapping. One or more of these parameters will then be used to maintain an ongoing link back to the database, as per the matching criteria defined, enabling future synchronization after placement.

This method of linking, due to its dynamic creation of components at the time of placement, lends itself very well to being used in a "many database records-to-one Altium Designer component fashion.

Unlike the DBLink-style of database linking, whereby the DBLink file must be included with the project, the DBLib file need not be added to the project. The resulting database library simply needs to be made part of the Available Libraries – accessible by the **Libraries** panel. Remember, the Available Libraries can consist of Project Libraries, Installed Libraries or libraries found along specified search paths.

You would typically configure DBLib files in a library-oriented fashion. For example you might have one for all the resistors detailed in your company database, another for capacitors, and so on.



For more detailed information on using this method of linking, refer to the [Using Components Directly from Your Company Database](#) application note.



For information on using an SVN Database Library, refer to the [Working with Version-Controlled Database Libraries](#) application note.

Creating the Database Link File

As mentioned, the backbone of the database link feature is the Database Link file. This file is created and managed using Altium Designer's DatabaseLink Editor (Figure 1).

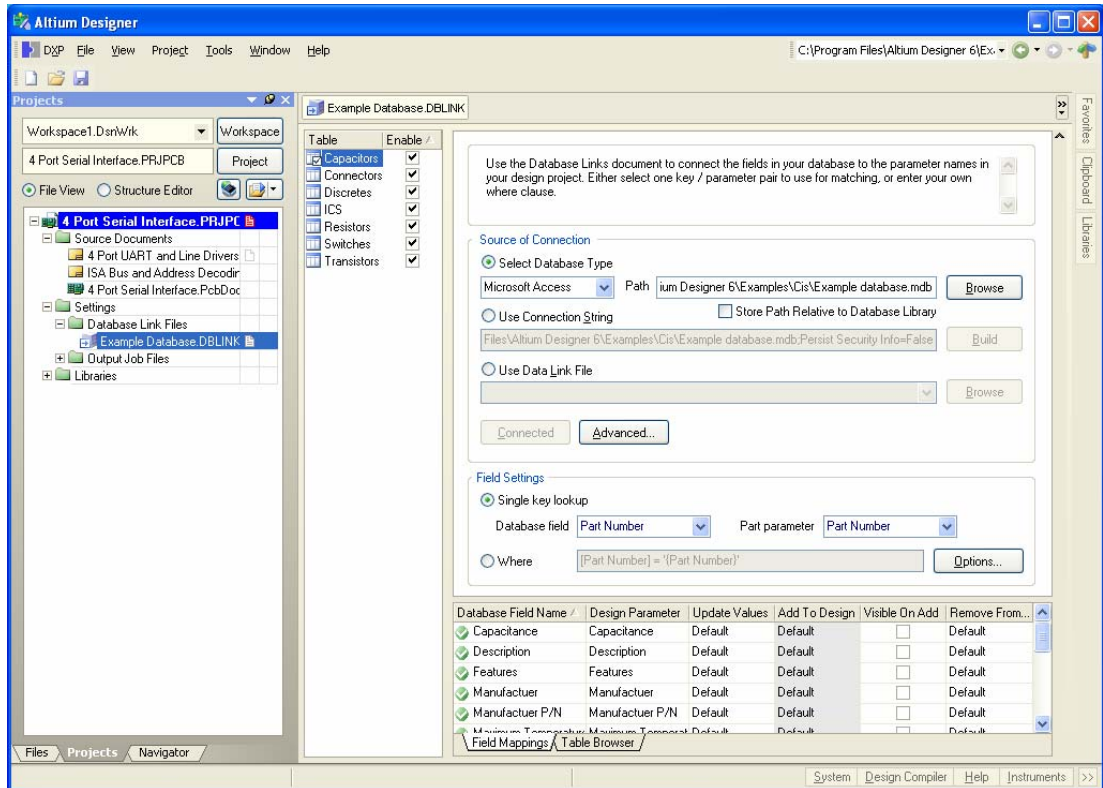


Figure 1. The DatabaseLink Editor.

The DatabaseLink Editor becomes active when the active document in the main design window is a *.DBLink file. Create a new file of this type by:

- using the **File » New » Database Link File** command
- right-clicking on the project's entry in the **Projects** panel and choosing **Add New to Project » Database Link File** from the pop-up menu that appears.

Database Link files, when added to a project, appear in the **Projects** panel under the **Settings\Database Link Files** sub-folder.

Connecting to an External Database

Table and mapping data will only appear in the Editor's main display window after the active Database Link file is successfully connected to the required external database. Connection is defined using the controls provided in the **Source of Connection** region of the window (Figure 2).

Linking Existing Components to Your Company Database

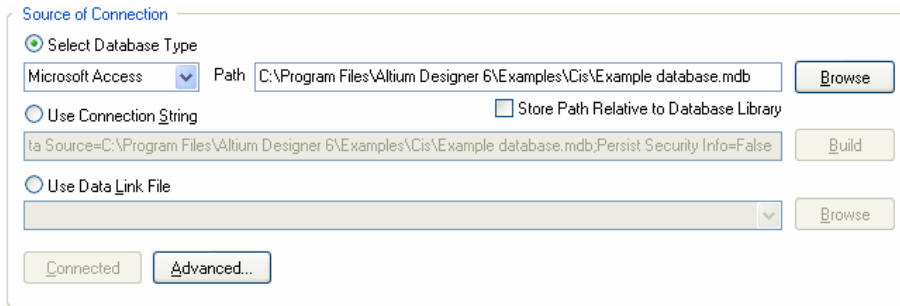


Figure 2. Specifying the connection to the external database.

Any database which provides OLE DB support can be connected to. The options provided in this region of the window each use an OLE DB connection string to connect to the target database. Some databases may not offer OLE DB support.

However, virtually all Database Management Systems in use today can be accessed through the Open Database Connectivity (ODBC) interface. The database link feature uses Microsoft's ODBC provider, which allows an ADO (ActiveX Data Object) to connect to any ODBC data source. The result is that any ODBC database can be connected to. The OLE DB provider for the ODBC database is specified as part of the connection string.

Connection can also be defined on the **Connection** tab of the *Database Connection* dialog, accessed from the main **Tools** menu.

Fast Connection to Access and Excel Databases

The **Select Database Type** option simply offers an expedited method of creating a connection string when the target database has been created using Microsoft Access or Microsoft Excel. Using this option, simply select the database type and then browse to and select the required database file. The corresponding connection string will automatically be composed and entered into the field for the **Use Connection String** option.

The full path can be specified or you can opt to store the path relative to the Database Link file.

Building a Connection String

If your company database is not Access- or Excel-based, and you want to build the connection string explicitly, simply enable the **Use Connection String** option and then click the associated **Build** button to the right. The *Data Link Properties* dialog will appear (Figure 3).

The OLE DB Provider – Microsoft Jet 4.0 – is set by default on the **Provider** tab of the dialog and hence the dialog opens at the **Connection** tab. This is the default provider setting for new Database Link files and is also used to connect to Access database files (*.mdb). Change the provider as necessary.

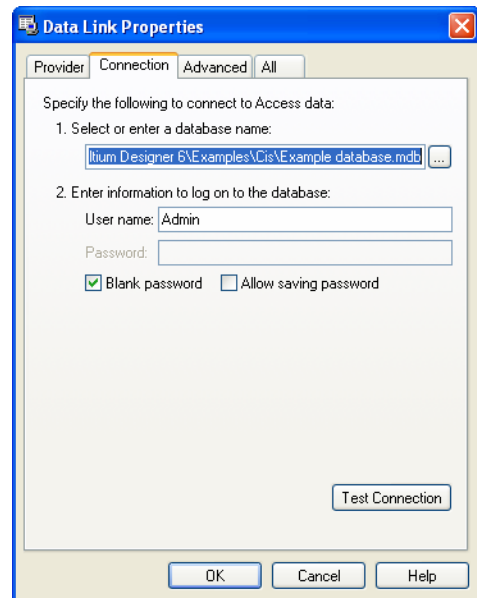


Figure 3. Specifying the connection to the external database.

From the **Connection** tab simply enter the name (including path) of the database you wish to connect to. Alternatively, use the ... button to open a dialog from where you can browse to and open the required file.

If your database requires login permission enter this as required, along with any other advanced settings available from the **Advanced** tab of the dialog. The **All** tab provides a summary listing of link options defined, as well as extended options relating to the chosen OLE DB Provider. Options can be modified as required from this tab.

Once link options have been defined you can check for successful connection by clicking on the **Test Connection** button (on the **Connection** tab). A successful connection will yield a confirmation dialog to that effect.

The *Data Link Properties* dialog is a Microsoft dialog and, as such, pressing **F1** or using the available **Help** button will gain access to the Microsoft Data Link Help file. This file is not part of Altium Designer's documentation set.

Specifying a Data Link file

If the data source to which you wish to connect is described using a Microsoft Data Link file (*.udl), simply enable the third of the connection options – **Use Data Link File** – and click the associated **Browse** button to locate the required file. A Data Link File is essentially a storage vessel for a connection string.

Proceeding with Connection

After defining the connection to the external database, the text of the **Connect** button will become bold, signifying that you can proceed with the connection. If the connection details are correct, the table and mapping information for the target database will be loaded into the Database Link document. The text on the **Connect** button will change to *Connected* and the button will be grayed-out.

If there is a problem with the connection details, for example a connection string is built incorrectly or a path is entered erroneously, connection will fail and a message will appear alerting you to this fact (Figure 4).

If you change the connection settings whilst connected to a database, live connection will be lost and the text on the **Connect** button will change to *Reconnect*. Click to re-establish the connection.

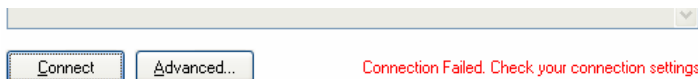


Figure 4. Flagging connection failure.

Check your connection settings and click the **Connect** button again.

After successful initial connection, and after saving the Database Link file, the connection will be made automatically each time the file is opened, provided the target database's location and filename are not changed.

Database Table Listing



After successful connection to the external database, table and mapping data will be loaded. The left-hand side of the display window lists all tables that exist in the connected database (Figure 5).

Table	Enable
<input checked="" type="checkbox"/> Capacitors	<input checked="" type="checkbox"/>
<input type="checkbox"/> Connectors	<input checked="" type="checkbox"/>
<input type="checkbox"/> Discretes	<input checked="" type="checkbox"/>
<input type="checkbox"/> ICS	<input checked="" type="checkbox"/>
<input type="checkbox"/> Resistors	<input checked="" type="checkbox"/>
<input type="checkbox"/> Switches	<input checked="" type="checkbox"/>
<input type="checkbox"/> Transistors	<input checked="" type="checkbox"/>

Figure 5. Tables existing in the connected database.

Linking Existing Components to Your Company Database

The **Enable** option next to each table entry allows you to control whether or not that table is to be used when updating parameters from the database.

As you click on a table entry in the list, its icon changes from  to  in order to distinguish it as being the currently active table. The table – with all its data – appears on the **Table Browser** tab of the display window (Figure 6). This is a non-editable copy of the table and allows you to quickly refer to its contents, without having to launch the external database itself.

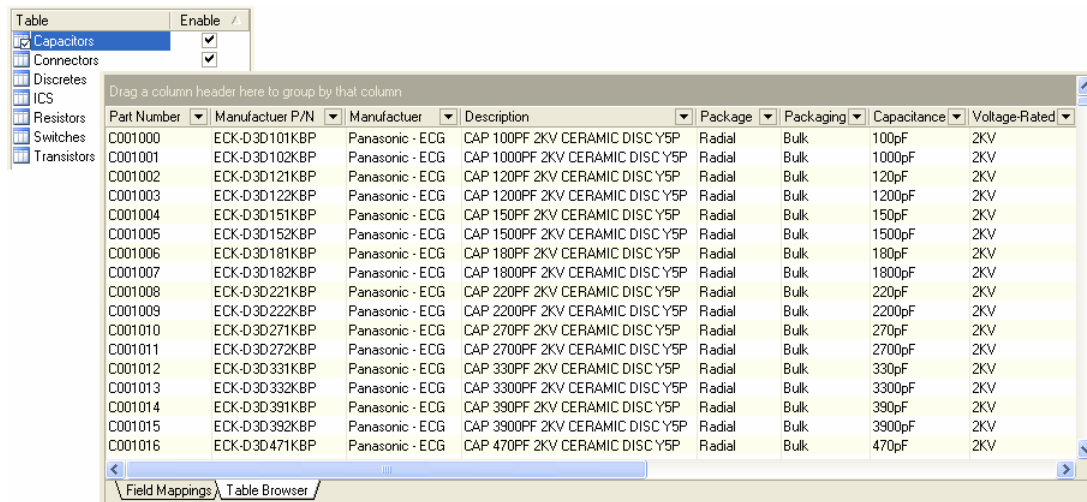


Table	Enable
Capacitors	<input checked="" type="checkbox"/>
Connectors	<input checked="" type="checkbox"/>
Discretes	<input type="checkbox"/>
ICs	<input type="checkbox"/>
Resistors	<input type="checkbox"/>
Switches	<input type="checkbox"/>
Transistors	<input type="checkbox"/>

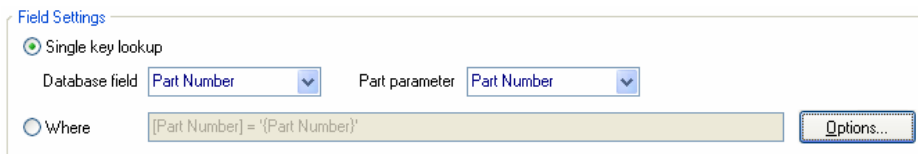
Part Number	Manufacturer P/N	Manufacturer	Description	Package	Packaging	Capacitance	Voltage-Rated
C001000	ECK-D3D101KBP	Panasonic - ECG	CAP 100PF 2KV CERAMIC DISC Y5P	Radial	Bulk	100pF	2KV
C001001	ECK-D3D102KBP	Panasonic - ECG	CAP 1000PF 2KV CERAMIC DISC Y5P	Radial	Bulk	1000pF	2KV
C001002	ECK-D3D121KBP	Panasonic - ECG	CAP 120PF 2KV CERAMIC DISC Y5P	Radial	Bulk	120pF	2KV
C001003	ECK-D3D122KBP	Panasonic - ECG	CAP 1200PF 2KV CERAMIC DISC Y5P	Radial	Bulk	1200pF	2KV
C001004	ECK-D3D151KBP	Panasonic - ECG	CAP 150PF 2KV CERAMIC DISC Y5P	Radial	Bulk	150pF	2KV
C001005	ECK-D3D152KBP	Panasonic - ECG	CAP 1500PF 2KV CERAMIC DISC Y5P	Radial	Bulk	1500pF	2KV
C001006	ECK-D3D181KBP	Panasonic - ECG	CAP 180PF 2KV CERAMIC DISC Y5P	Radial	Bulk	180pF	2KV
C001007	ECK-D3D182KBP	Panasonic - ECG	CAP 1800PF 2KV CERAMIC DISC Y5P	Radial	Bulk	1800pF	2KV
C001008	ECK-D3D221KBP	Panasonic - ECG	CAP 220PF 2KV CERAMIC DISC Y5P	Radial	Bulk	220pF	2KV
C001009	ECK-D3D222KBP	Panasonic - ECG	CAP 2200PF 2KV CERAMIC DISC Y5P	Radial	Bulk	2200pF	2KV
C001010	ECK-D3D271KBP	Panasonic - ECG	CAP 270PF 2KV CERAMIC DISC Y5P	Radial	Bulk	270pF	2KV
C001011	ECK-D3D272KBP	Panasonic - ECG	CAP 2700PF 2KV CERAMIC DISC Y5P	Radial	Bulk	2700pF	2KV
C001012	ECK-D3D331KBP	Panasonic - ECG	CAP 330PF 2KV CERAMIC DISC Y5P	Radial	Bulk	330pF	2KV
C001013	ECK-D3D332KBP	Panasonic - ECG	CAP 3300PF 2KV CERAMIC DISC Y5P	Radial	Bulk	3300pF	2KV
C001014	ECK-D3D391KBP	Panasonic - ECG	CAP 390PF 2KV CERAMIC DISC Y5P	Radial	Bulk	390pF	2KV
C001015	ECK-D3D392KBP	Panasonic - ECG	CAP 3900PF 2KV CERAMIC DISC Y5P	Radial	Bulk	3900pF	2KV
C001016	ECK-D3D471KBP	Panasonic - ECG	CAP 470PF 2KV CERAMIC DISC Y5P	Radial	Bulk	470pF	2KV

Figure 6. Browsing source tables in the connected database.

Specifying Matching Criteria

In order to update part parameter information with that stored in fields of the external database, each part in the design must first be 'matched' with one row from one table in the database. The **Field Settings** region of the main display window (Figure 7) allows you to define the matching criteria – either a simple, single key lookup, or a more advanced match using a **Where** clause.

Matching criteria is specified on a per-table basis.



Field Settings

Single key lookup

Database field: Part parameter:

Where:

Figure 7. Controlling matching criteria.

Single Key Lookup

If the **Single key lookup** option is enabled (default) the **Database field** and **Part parameter** fields become available. The former lists all of the available field names (column headers) in the active table of the database. The latter lists all of the parameters found across all parts in the design – both system and user-defined parameters. Note that the drop-down list for the **Part parameter** field will only become populated after compiling the parent project.

Use these fields to specify the matching between one field in the table and one parameter in a part (and typically common to all parts). Typically, the matching field in both cases is something that uniquely identifies each component in the external database, such as a Part Number. The chosen lookup field is distinguished on the **Field Mappings** tab of the window by the **Design Parameter** entry shown as grayed-out.

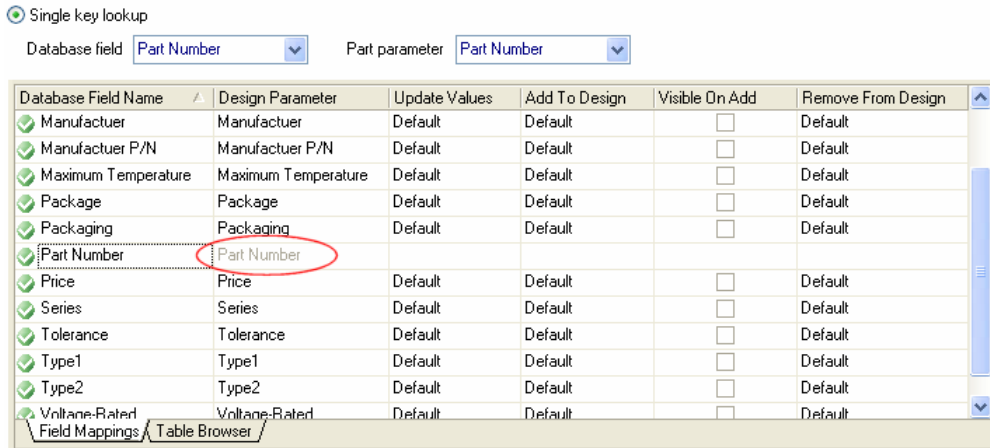


Figure 8. Single key mapping by Part Number.

When using the **Update Parameters From Database** feature, information is read from the chosen key parameter in the placed schematic components and then searched for in the chosen (key) field of the database – across all enabled tables. When there is a match, information from other cells in that record of the parent table can then be taken back to the mapped parameters in the schematic component.

Advanced Matching – the Where Clause

While the **Single key lookup** option works well if there is a unique part number/id to match on, it is not so effective when matching by a parameter that is not unique, such as capacitance or resistance. In this case the more advanced **Where** clause should be used, enabling you to specify multiple key matching in order to link the schematic component to the required row in the relevant database table.

In its simplest form the **Where** clause (written using SQL syntax) reflects the chosen entries that define the single key lookup. For example if the **Database field** was chosen to be `Part Number` and the unique **Part parameter** (existing for all parts in the design) was chosen to be `Part Number`, the entry for the **Where** clause would be:

```
[Part Number] = '{Part Number}'
```

The square brackets around the database field (table column) are quote characters, as specified in the **Advanced** tab of the **Database Connection** dialog (Figure 9). Access this dialog either by clicking on the **Advanced** button in the **Source of Connection** region of the window, or from the main **Tools** menu.

Linking Existing Components to Your Company Database

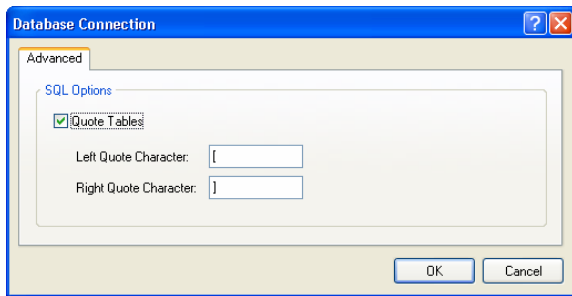


Figure 9. SQL quote control.

Using standard SQL syntax, the **Where** clause can then be extended to match using multiple Database field/Part parameter entries, for example:

```
[Capacitance] = '{Capacitance}' AND  
[Tolerance] = {Tolerance} AND [Manufacturer]  
= '{Manufacturer}'
```

In this case a single record in the relevant table of the database would be linked to using three different design parameters. Notice that the entry for the `Tolerance` design parameter is not quoted. This means that the column type in the associated table of the database is Number and not String.

Using standard SQL syntax you can conceivably make the **Where** clause as simple or as complicated as you like. An example of a more complicated clause might be:

```
[Tolerance] BETWEEN {Min Tolerance} AND {Max Tolerance} AND [Price] <= {Max  
Price} AND [Manufacturer] IN ('{Preferred Manufacturer 1}', '{Preferred  
Manufacturer 2}')
```

In this case, the table would be queried for matching parts based on three database fields – `Tolerance`, `Price` and `Manufacturer`. The tolerance must lie in the range defined by the `Min Tolerance` and `Max Tolerance` design parameters. The price must be less than or equal to that specified in the `Max Price` design parameter and the manufacturer can be any of the two candidates specified by the `Preferred Manufacturer 1` and `Preferred Manufacturer 2` design parameters.

Creating a Part Parameter

If no suitable parameter exists in your Altium Designer components – for use as the **Part parameter** for matching purposes in the DBLink file – you will need to create one. Addition of the 'base' parameter can be achieved quickly using either the Parameter Manager or the relevant Inspector panel.

Adding a Parameter using the Parameter Manager

To add a new parameter using the Parameter Manager, from a schematic document (or schematic library document) choose **Tools » Parameter Manager** from the main menus. The *Parameter Editor Options* dialog will appear. To add the new parameter to all parts, ensure that the **Parts** option is enabled in the **Include Parameters Owned By** region, and `All Objects` is set as the criteria (Figure 10).

The curly brackets (braces) specify that the entry being referenced is a design parameter. The single quotes are used to specify the design parameter be treated as a string, as opposed to a number (no quotes). The type matching is very important, as SQL is type sensitive. The design parameter should be made the same type as the column in the database.

When quoting tables, the specific quote characters used will depend on the database you are using. For example, square brackets [] are only usable in Microsoft databases like Access, Excel via ADO, or MSSQL (later versions). MYSQL would use the ' character for quoting. You really only need to quote column names, in any database, if they include spaces or are reserved words (for that database). Check the documentation for your particular database software to see which quote characters are used (if any).

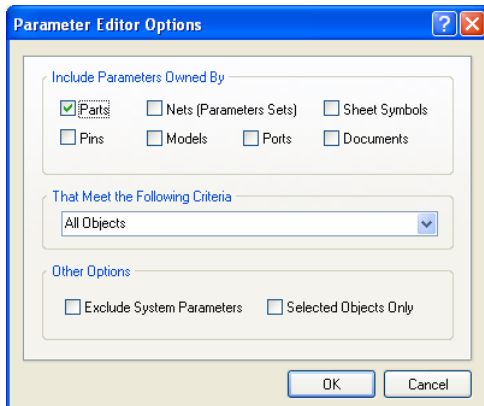


Figure 10. Setting parameter management options.

Clicking **OK** will present the *Parameter Table Editor* dialog, showing all parameters currently ‘owned’ by all part objects on schematic source documents in the active project (or schematic components in the active schematic library). Simply click on the **Add Column** button and use the *Add Parameter* dialog that appears to define the name for the parameter (Figure 11). Ensure that the **Add to all objects** option is enabled, so that the new parameter will be added to all parts. It is best to leave the Value for the parameter unassigned – at this stage you just want to get the parameter added to each part. Bear in mind however, that to effectively link to a corresponding record in the database, the value for the parameter will need to be defined afterwards on a per-part basis.

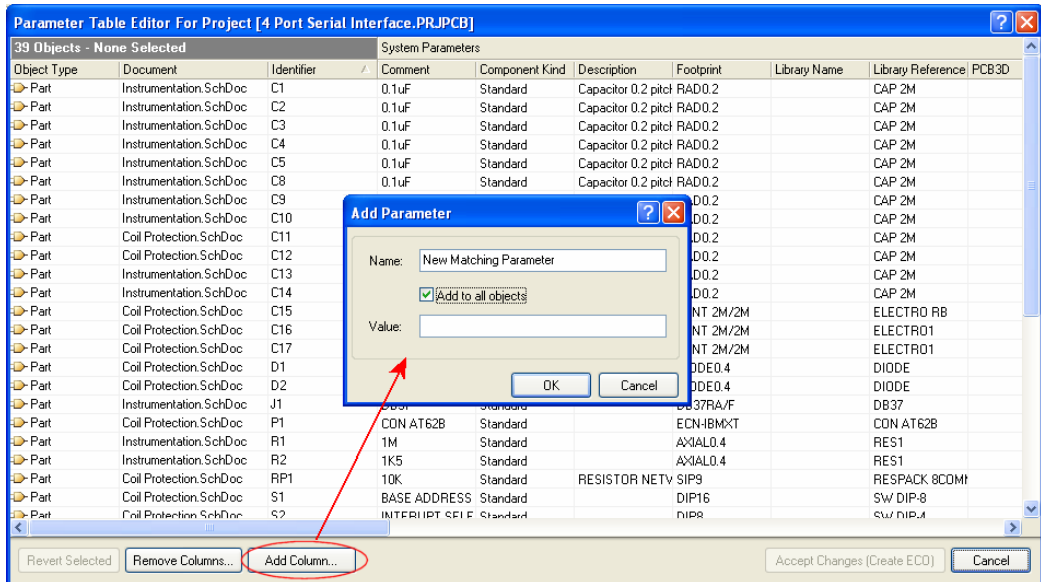


Figure 11. Adding a new parameter to all parts.

Once the new column has been added, click the **Accept Changes (Create ECO)** button. Use the *Engineering Change Order* dialog that appears to validate and then execute the changes – adding the parameter to each of the required parts.

Linking Existing Components to Your Company Database

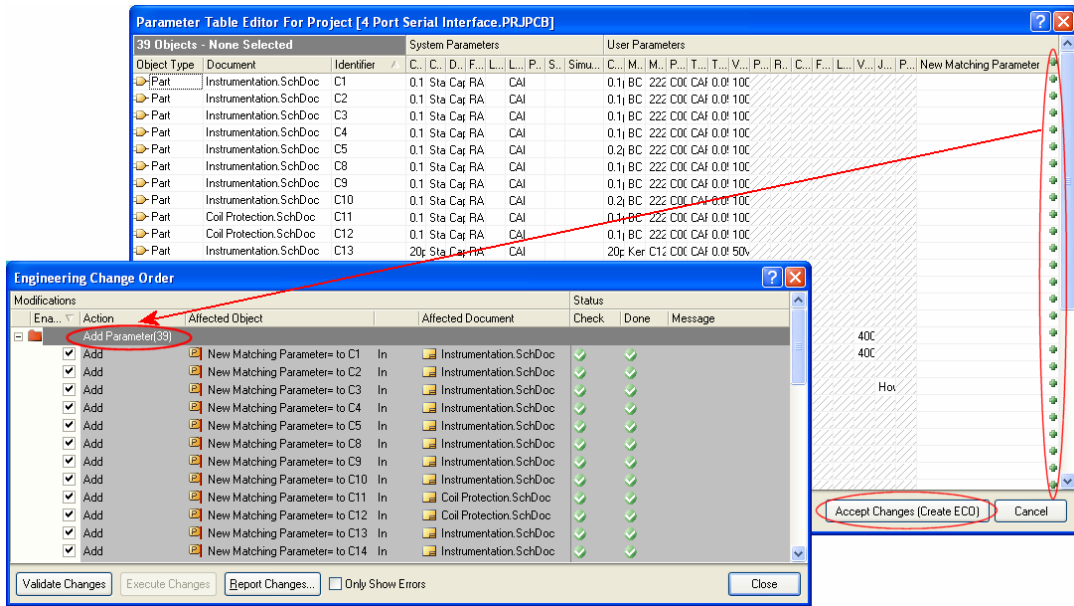


Figure 12. Executing the parameter changes.

Adding a Parameter using the Inspector Panel

A new parameter can be added to selected parts using an Inspector-based panel. The following procedure outlines the case when you wish to add the same unique parameter to all components in the active schematic library document, using the **SCHLIB Filter** and **SCHLIB Inspector** panels.

Firstly, select all components in the library, using the **SCH Filter** panel and a query of `IsPart`. Ensure that the **Whole Library** option is enabled, otherwise only the active library component will be considered. In the **SCHLIB Inspector** panel, you can verify that all components have been selected, then add the required parameter using the **Parameters** region of the panel.

Simply click inside the field to the

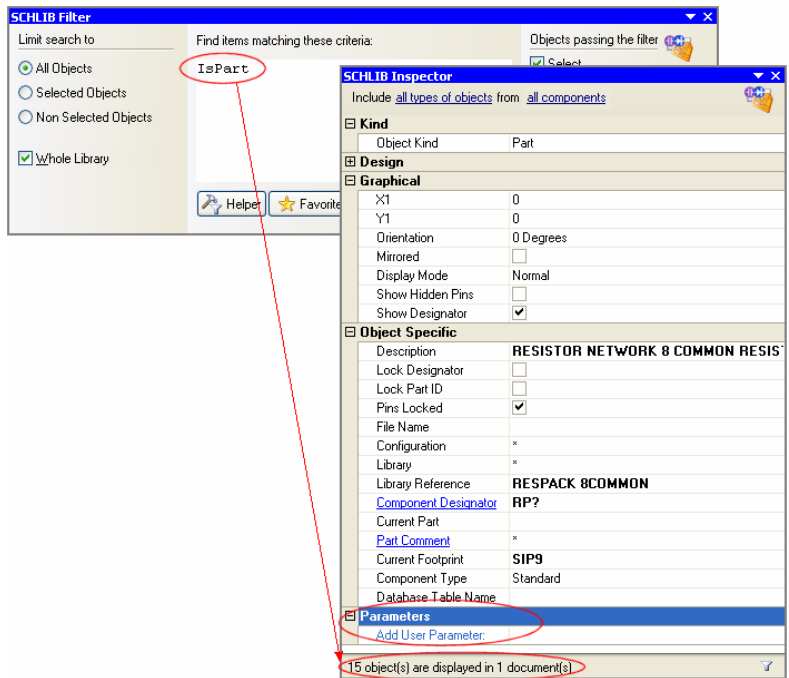


Figure 13. Parameter creation from an Inspector-based panel.

right of the Add User Parameter entry and, not entering a value, press **Enter**. In the dialog that appears, enter the name for the required parameter. After clicking **OK**, the parameter will be added to the selected components, and appear as a parameter entry in the **Parameters** region of the **SCHLIB Inspector** panel (Figure 14).

The procedure is similar when adding a parameter to placed parts on schematic source documents. In this case, the **SCH Filter** and **SCH Inspector** panels will be used.

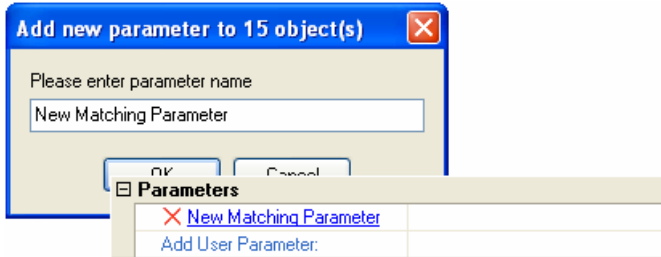


Figure 14. New parameter added to the selected components.

Mapping Database Fields to Design Parameters

Design parameters for existing components – either those placed on the source schematic sheets of a design project, or those defined in a schematic component library that has been added to a library package – are updated with externally-stored information from a company database using the **Update Parameters From Database** command, available from the main Tools menu when the schematic document/schematic library document is active. When a match occurs between a design component and a record in a table of the linked database, what happens next is determined by mapping and update information specified in the Database Link file. These settings are performed on the **Field Mappings** tab of the DatabaseLink Editor's main display window (Figure 15).

Mapping and update options are specified on a per-table basis.

Database Field Name	Design Parameter	Update Values	Add To Design	Visible On Add	Remove From Design
✓ Capacitance	Capacitance	Default	Default	<input type="checkbox"/>	Default
✓ Description	Description	Default	Default	<input type="checkbox"/>	Default
✓ Features	Features	Default	Default	<input type="checkbox"/>	Default
✓ Manufacturer	Manufacturer	Default	Default	<input type="checkbox"/>	Default
✓ Manufacturer P/N	Manufacturer P/N	Default	Default	<input type="checkbox"/>	Default
✓ Maximum Temperature	Maximum Temperature	Default	Default	<input type="checkbox"/>	Default
✓ Package	Package	Default	Default	<input type="checkbox"/>	Default
✓ Packaging	Packaging	Default	Default	<input type="checkbox"/>	Default
✓ Part Number	Part Number				
✓ Price	Price	Default	Default	<input type="checkbox"/>	Default
✓ Series	Series	Default	Default	<input type="checkbox"/>	Default
✓ Tolerance	Tolerance	Default	Default	<input type="checkbox"/>	Default
✓ Type1	Type1	Default	Default	<input type="checkbox"/>	Default
✓ Type2	Type2	Default	Default	<input type="checkbox"/>	Default
✓ Voltage-Rated	Voltage-Rated	Default	Default	<input type="checkbox"/>	Default

Field Mappings / Table Browser

Figure 15. Specifying parameter mapping and update options.

Parameter Mapping

The first two columns (from the left) on the **Field Mappings** tab allow you to control which information from the database is to be mapped to the component's parameters.

Linking Existing Components to Your Company Database

The **Database Field Name** column lists all field (column) names in the currently active table of the database. The **Design Parameter** column allows you to specify the design parameter that should be mapped to the corresponding field in the database.

Initial mapping is performed automatically upon connection to the database, with all database fields mapped to design parameters using the same names. For example, if a field in the database is called `Tolerance`, a design parameter with the name `Tolerance` will be mapped to it. You can change the name for a design parameter simply by:

- clicking in its cell and typing the new name directly
- using the cell's associated drop-down list to choose an existing design parameter (provided the parent project has been compiled beforehand).

You may have a large number of data fields associated with a component in the database, not all of which you will want, or even need, added as design parameters to the existing design component. Much of this information may only be required when generating a Bill of Materials. The Report Manager includes an option that allows you to add parameter information to a BOM, directly from a linked database – allowing you to reduce the amount of information that gets 'carried' with the schematic source documents. For more information, refer to the section [Adding Database Information Directly to a BOM](#).

For fields that you explicitly do not want mapped from the database, set the **Design Parameter** entry to `[None]`. Unmapped database fields are distinguished on the tab by the use of a red cross icon (✘). Mapped database fields are distinguished by a green tick icon (✔).

To quickly remap an unmapped field, click inside the row for that field and use the **Ctrl + D** keyboard shortcut. The name for the design parameter will be identical to that of the corresponding **Database Field Name**.

Parameter Update Options

The remaining columns on the **Field Mappings** tab (Figure 15 previously) allow you to specify the actions taken for parameters, when updating a component using the **Update Parameters From Database** command. In the following descriptions, the term Altium Designer component is used to represent a placed component on a schematic sheet, or a defined component in a schematic component library (which is part of a library package).

The four columns are as follows:

- **Update Values** – a cell in this column is used to determine the action that should be taken if the parameter exists for the Altium Designer component and in the database, but the values are currently different. Choose to update the parameter of the Altium Designer component with the value stored in the database, or not to update at all.
- **Add To Design** – a cell in this column is used to determine the action that should be taken if the parameter is found in the database but does not exist for the Altium Designer component. You can choose to add/not to add the parameter or add the parameter only if it has a value assigned to it in the database.
- **Visible On Add** – a checkbox in this column is used to determine whether a newly added parameter is made visible for the Altium Designer component (enabled) or not (disabled).
- **Remove From Design** – a cell in this column offers the converse of the Add To Design field, i.e. what action to take if the parameter is found to exist for the Altium Designer component, but not in

the database. You can choose to not remove the parameter at all, or only remove it if it has no value assigned to it in the database.

Initially, the **Update Values**, **Add To Design** and **Remove From Design** fields for each mapped database field will be set to the entry `Default`, and the **Visible On Add** option will be disabled, as illustrated in Figure 16.

Database Field Name	Design Parameter	Update Values	Add To Design	Visible On Add	Remove From Design
✓ Capacitance	Capacitance	Default	Default	<input type="checkbox"/>	Default
✗ Description	[None]				
✗ Features	[None]				
✓ Manufacturer	Manufacturer	Default	Default	<input type="checkbox"/>	Default
✓ Manufacturer P/N	Manufacturer P/N	Default	Default	<input type="checkbox"/>	Default
✓ Maximum Temperature	Maximum Temperature	Default	Default	<input type="checkbox"/>	Default
✓ Package	Package	Default	Default	<input type="checkbox"/>	Default
✓ Packaging	Packaging	Default	Default	<input type="checkbox"/>	Default
✓ Part Number	Part Number				
✓ Price	Price	Default	Default	<input type="checkbox"/>	Default
✓ Series	Series	Default	Default	<input type="checkbox"/>	Default
✓ Tolerance	Tolerance	Default	Default	<input type="checkbox"/>	Default
✓ Type1	Type1	Default	Default	<input type="checkbox"/>	Default
✓ Type2	Type2	Default	Default	<input type="checkbox"/>	Default
✓ Voltage-Rated	Voltage-Rated	Default	Default	<input type="checkbox"/>	Default

Figure 16. Specifying parameter update options.

Looking at Figure 16 there are three important points to make regarding update options:

- Unmapped database fields will have no associated update options.
- The key field (e.g. Part Number in Figure 16) will have no associated update options. This field is solely used for matching purposes.
- A setting of `Default` causes an update option to follow its corresponding default definition, as specified in the *Database Link Options* dialog (**Tools » Options**). This dialog can also be accessed by clicking the **Options** button in the **Field Settings** region of the main window.

The third point is beneficial in that it allows you to specify update options from a central location and then point to that location when defining the update options for each mapped field. That is why the `Default` entry is loaded automatically into the relevant update fields upon mapping a database field to a design parameter.

Should you wish to override the default setting for an update option, simply click inside the relevant update field on the **Field Mappings** tab and then click again to access a drop-

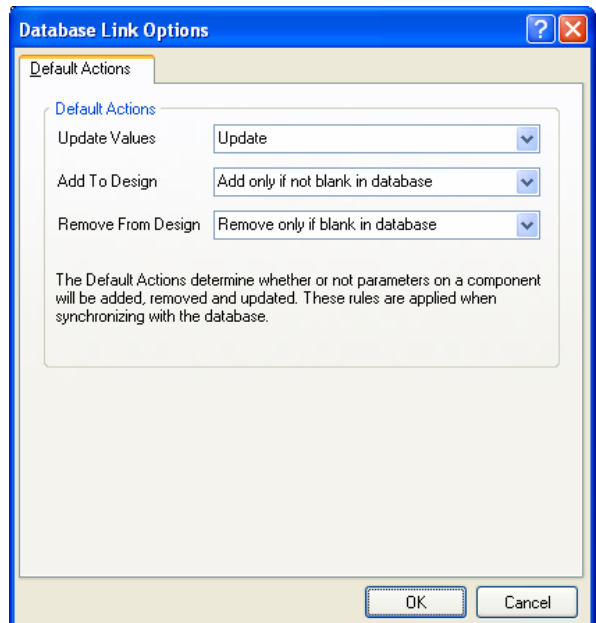


Figure 17. Setting default parameter update options.

Linking Existing Components to Your Company Database

down providing the applicable update choices. (Figure 18).


✔ Price	Price	Default	Default	<input type="checkbox"/>	Default
✔ Series	Series	Default	Default	<input type="checkbox"/>	Default
✔ Tolerance	Tolerance	Default	Default	<input type="checkbox"/>	Default
✔ Type1	Type1	Default	Default	<input type="checkbox"/>	Default
✔ Type2	Type2	Default	Default	<input type="checkbox"/>	Default
✔ Voltage-Rated	Voltage-Rated	Default	Default	<input type="checkbox"/>	Default

Figure 18. Manually overriding a parameter update option.

In this way, you have full control over how the parameters in the design are updated. You can set all fields to `Default` and make the required update decisions from the *Database Link Options* dialog, set each update field individually, or have a mixture of the two – the decision is entirely yours to make as you see fit. The update, when performed, is carried out through use of an *Engineering Change Order* dialog. If at this stage there are updates that you would prefer not to make, you can simply opt to not include those particular changes – giving you the final and ultimate say in which design parameters get updated.

Ensuring Synchronicity

After the mapping and update options have been specified in the Database Link file as required, and the file saved, you will be in a position to pass information from the database to the mapped parameters of Altium Designer components. Remember, the chosen key field parameter is used to ensure that an Altium Designer component retains its link to the corresponding record for that component in the external database. This means that at any stage in the future, changes to parameter information in the database can be easily passed back to the Altium Designer component, synchronizing the two. Updates are performed using the **Update Parameters From Database** command, available from the Schematic Editor's main **Tools** menu.

 For further information on using this parameter update tool, refer to the [Keeping Components Up-To-Date](#) application note.

Dual Synchronization – DBLib/SVNDBLib and DBLink

You may have an existing design project where the majority of the placed parts have been linked to an external database using a DBLink file. Design changes may result in additional circuitry, the components for which might be placed using the database library feature. The associated DBLib/SVNDBLib file could, quite conceivably, point to a different external database.


When using the **Update Parameters From Database** command, all linked parameters for placed components will be queried – across all linked databases, irrespective of the linking method used – and detected differences for those parameters displayed in the *Select Parameter Changes* dialog.

If the same database field has been used for matching in both the DBLink and DBLib/SVNDBLib files, the database linked by the DBLink file will be searched first for a match, followed by the database linked to by the DBLib/SVNDBLib file. If the component is present in both databases, you could quite possibly match and update from the wrong external record.

Adding Database Information Directly to a BOM

Source information for a Bill of Materials (BOM) has, in the past, been taken from the parameter information of the placed components for the design. But that can lead to a lot of information attached to a schematic that is only ever used for the BOM. If your placed Altium Designer components are linked to an external database using a DBLink file, the BOM Generator is able to extract any other record information that has not been added to the components as design parameters.

Parameters can be included from an external database, irrespective of the method employed to link to that database – DBLink, DBLib, or SVNDBLib.

When configuring the Bill of Materials report using the *Report Manager*, simply enable the **Include Parameters from Database** option. This option will only be available if one or more components in your design are linked to an external database. In the parameter listing, the  icon is used to distinguish a parameter that exists for one or more placed components in a linked external database.

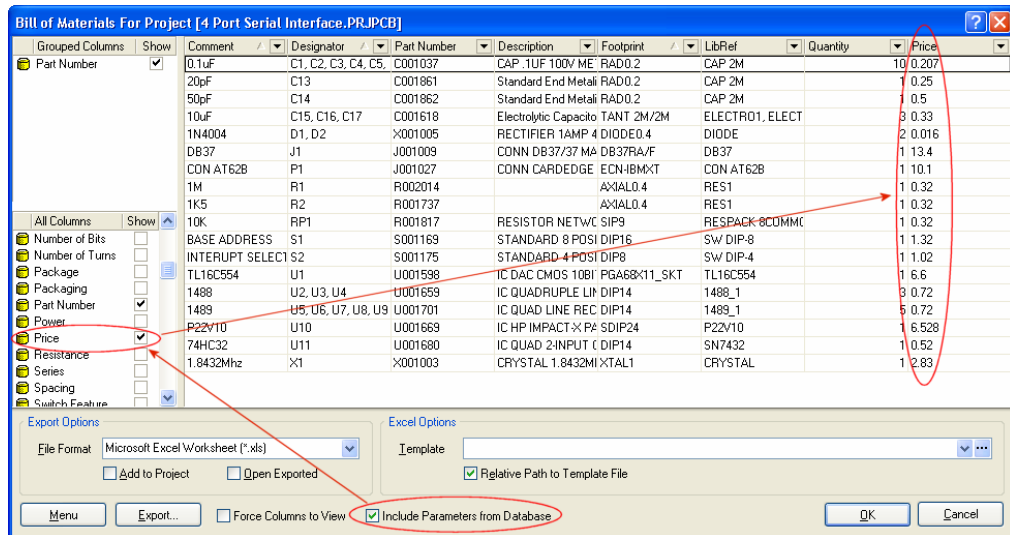


Figure 19. Include additional component information that exists only in an external database.

Revision History

Date	Version No.	Revision
29-Nov-2005	1.0	Initial release
08-Jun-2006	2.0	Updated for Altium Designer 6.3. Information on Update Parameters From Database moved to document AP0144 Keeping Components Up-To-Date. Added links to AP0145 Working with Version-Controlled Database Libraries.

Software, hardware, documentation and related materials:

Copyright © 2006 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment. Altium, Altium Designer, Board Insight, CAMtastic, CircuitStudio, Design Explorer, DXP, LiveDesign, NanoBoard, NanoTalk, Nexar, nVisage, P-CAD, Protel, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.