

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 PCB 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.

For more information on linking existing Altium Designer components (placed on schematics or part of a schematic component library), refer to the [Linking Existing Components to Your Company Database](#).

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 information on using an SVN Database Library file (*.SVNDBLib), whereby the schematic symbols and associated models are stored in libraries under version control, refer to the [Working with Version-Controlled Database Libraries](#) document.

Creating the Database Library

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

An example Database Library – VishayCapacitor.DBLib – can be found in the \Examples\Cis\Example DBLib folder of the installation.

For background information on components and libraries, refer to the article, [Component, Model and Library Concepts](#).



Figure 1. The DatabaseLib Editor.

The DatabaseLib Editor becomes active when the active document in the main design window is a *.DBLib file. Create a new file of this type by using the **File » New » Library » Database Library** command.

Connecting to an External Database

Table and mapping data will only appear in the Editor's main display window after the active Database Library 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).

Source of Connection

Select Database Type

Microsoft Access Path C:\Program Files\Altium Designer\Examples\Cis\Example database.mdb Browse

Use Connection String Store Path Relative to Database Library

ta Source=C:\Program Files\Altium Designer 6\Examples\Cis\Example database.mdb;Persist Security Info=False Build

Use Data Link File

Browse

Connected Advanced...

Figure 2. Specifying the connection to the external database.

Tip:

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

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

library 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.

Fast Connection to Access and Excel Databases

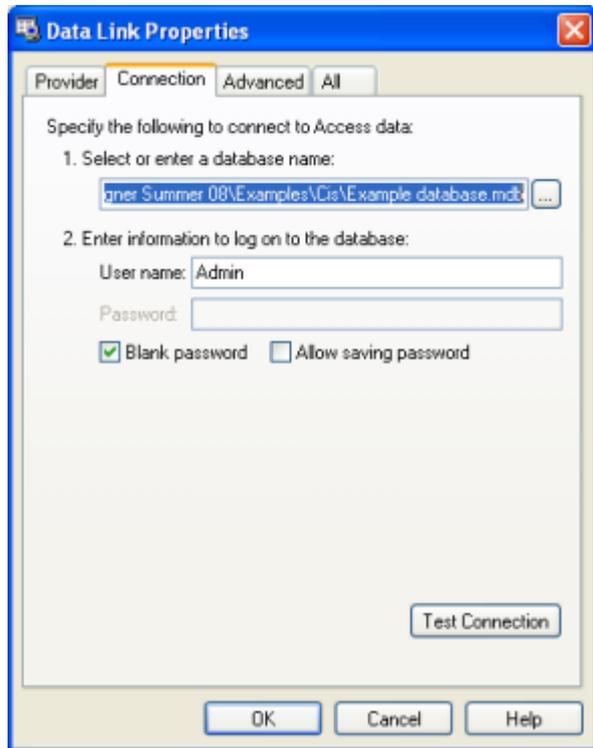


Figure 3. Specifying the connection to the external database.

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 Library 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 Library files and is also used to connect to Access database files (*.mdb). Change the provider as necessary. 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.

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 Library 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).



Figure 4. Flagging connection failure.

Tip:

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.

Check your connection settings and click the Connect button again.

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

A DBLib can also be created from an integrated library, using the **Integrated Library to Database Library Translator Wizard** . For more information, refer to the [Database Library Migration Tools](#) document.

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).

 If the target database has been created using multiple Excel spreadsheet files (*.xls), there is a limit of 64 sheets that can connect, due to ODBC driver limitations.

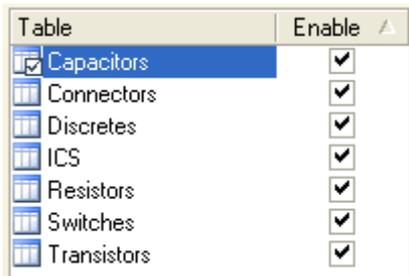


Figure 5. Tables existing in the connected database.

The Enable option next to each table entry allows you to control whether or not that table is to be part of the resulting database library. When the database library is added to the list of Available Libraries for browsing in the Libraries panel, each table will appear as a separate library entity. So although only one database library is added in real terms, from the perspective of the Libraries panel it is as though you have added multiple, distinct libraries. For more information, see Adding the Database Library.

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.

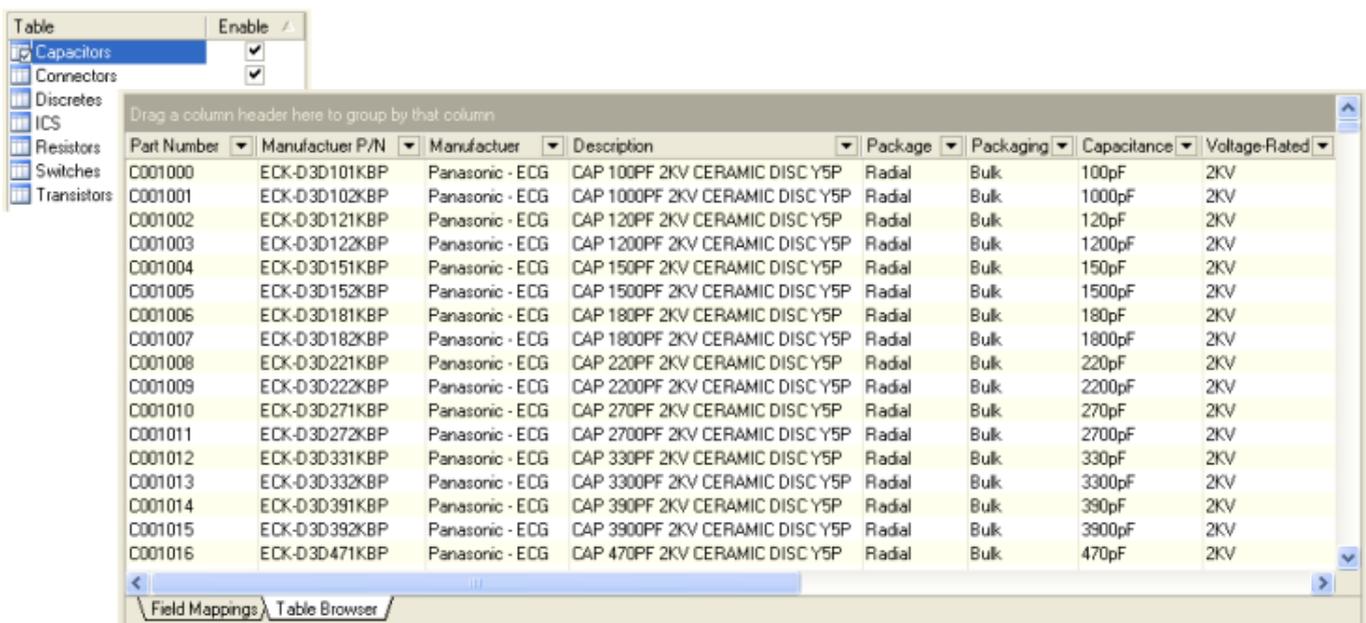


Figure 6. Browsing source tables in the connected database.

Specifying Matching Criteria

After a component has been placed from the external database, there needs to be some way of retaining the link between that placed component and the database record whose information was used to create it. In essence, the two need to be matched.

When a component is placed, its parameter information is created on-the-fly, using the corresponding fields in the database record. The post-placement link between the schematic component and the database record is established using one or more of these parameters. 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.



Figure 7. Controlling post-placement 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. As the parameters for a schematic component are added as it is placed, the Part parameter field will simply reflect whatever database field is chosen.

Typically, the lookup key field used 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.

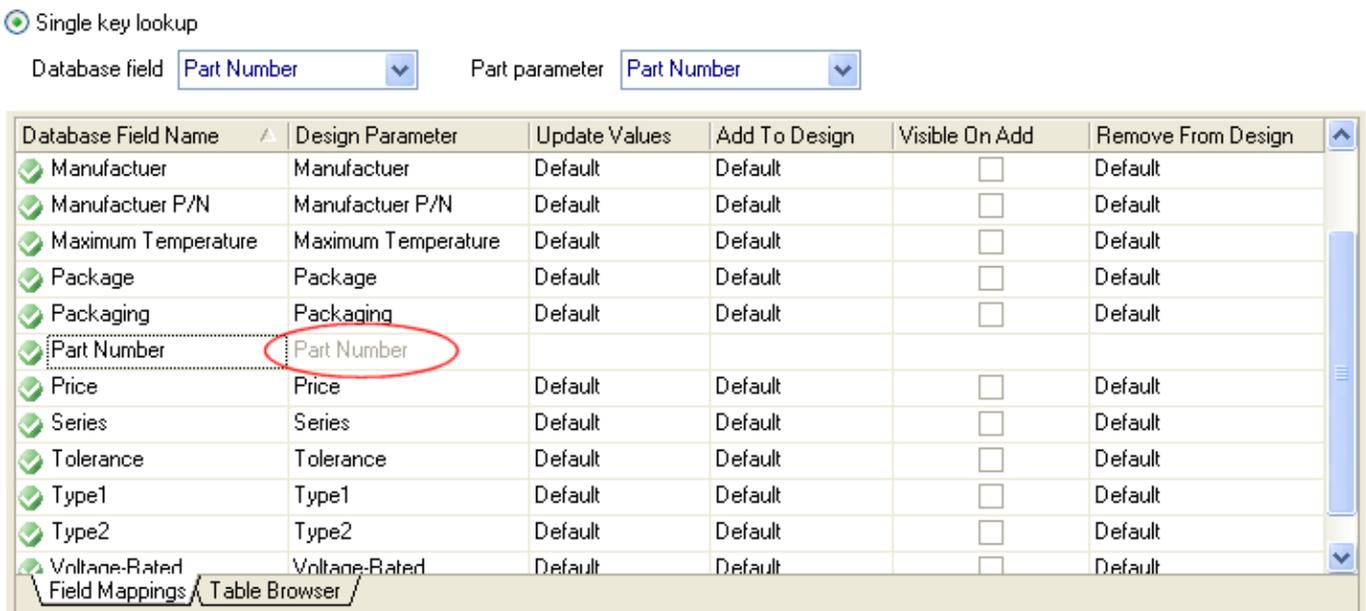


Figure 8. Single key mapping by Part Number.

When using the Update Parameters From Database feature at some stage after placement, 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

Tip:

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 ` (reverse apostrophe) character for quoting. You 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).

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 its source database record.

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 – the default – the Part parameter field would automatically be set to Part Number also, and 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 Tools menu.



Figure 9. SQL quote control.

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.

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.

Mapping Database Fields to Design Parameters

Design parameters for a component placed from a database library are created/added at the time of placement. Which parameters are actually created and the options used to update their information after placement, using the Update Parameters From Database command, is determined by mapping and update information specified in the Database Library file. These settings are performed on the Field Mappings tab of the DatabaseLib Editor's main display window (Figure 10).

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
✔ Component Type	Component Type	Default	Default	<input type="checkbox"/>	Default
✔ ComponentLink1Description	ComponentLink1Description	Default	Default	<input type="checkbox"/>	Default
✔ ComponentLink1URL	ComponentLink1URL	Default	Default	<input type="checkbox"/>	Default
✔ ComponentLink2Description	ComponentLink2Description	Default	Default	<input type="checkbox"/>	Default
✔ ComponentLink2URL	ComponentLink2URL	Default	Default	<input type="checkbox"/>	Default
✔ DatasheetDocument	DatasheetDocument	Default	Default	<input type="checkbox"/>	Default
✔ Designator	Designator	Default	Default	<input type="checkbox"/>	Default
✔ Footprint	Footprint	Default	Default	<input type="checkbox"/>	Default
✔ Footprint Path	[Footprint Path]				
✔ Footprint Ref	[Footprint Ref]				
✔ Library Path	[Library Path]				
✔ Library Ref	[Library Ref]				
✔ PackageReference	PackageReference	Default	Default	<input type="checkbox"/>	Default
✔ Part Number	Part Number				
✔ Pin Count	Pin Count	Default	Default	<input type="checkbox"/>	Default
✔ Published	Published	Default	Default	<input type="checkbox"/>	Default
✔ Publisher	Publisher	Default	Default	<input type="checkbox"/>	Default
✔ Signal Integrity	Signal Integrity	Default	Default	<input type="checkbox"/>	Default
✔ Simulation	Simulation	Default	Default	<input type="checkbox"/>	Default
✔ Value	Value	Default	Default	<input type="checkbox"/>	Default

Figure 10. Specifying parameter mapping and update options.

Model and Parameter Mapping

Tip:

To quickly remap an unmapped field, click inside the row for that field and use the CTRL + D keyboard shortcut. Note that for model mappings you will need to manually select from the associated Design Parameter drop-down.

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 models and parameters.

The Database Field Name column lists all field (column) names in the currently active table of the database. The Design Parameter column defines how each corresponding field in the database is to be used - whether it is used to source a schematic component, link-in a particular model, or to be attached to the component as a mapped design parameter.

Initial mapping is performed automatically upon connection to the database, with all database fields mapped.

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 (✔).

Models

If the database field name is one of the following reserved names, the corresponding model mapping entries will be automatically set in the Design Parameter field:

Tip:

Unlimited footprint and PCB3D model references (and paths) can be specified in a database table and mapped in the DBLib file. In the reserved names on the left, n represents a positive integer, starting from 2.

Description → [Description]
Footprint Ref → [Footprint Ref]
Footprint Path → [Footprint Path]
Footprint Ref n → [Footprint Ref n]
Footprint Path n → [Footprint Path n]
Library Ref → [Library Ref]
Library Path → [Library Path]
PCB3D Ref → [PCB3D Ref]
PCB3D Path → [PCB3D Path]
PCB3D Ref n → [PCB3D Ref n]
PCB3D Path n → [PCB3D Path n]

Tip:

Only one simulation model link can be defined for a component in an external database. Typically there will only ever be a single simulation model linked to a component. Should you wish to set up multiple simulation model links, the other links will need to be defined and stored with that component in the source schematic library file.

Sim Description → [Sim Description]
Sim Excluded Parts → [Sim Excluded Parts]
Sim File → [Sim File]
Sim Kind → [Sim Kind]
Sim Model Name → [Sim Model Name]
Sim Netlist → [Sim Netlist]
Sim Parameters → [Sim Parameters]
Sim Port Map → [Sim Port Map]
Sim Spice Prefix → [Sim Spice Prefix]
Sim SubKind → [Sim SubKind]

Note : References to PCB3D refer to the legacy 3D viewer. The current 3D viewer can render component bodies and imported STEP model files associated with component footprints. These mappings define the symbol and model information for the component. When the component is placed, the schematic symbol specified by the corresponding database record's Library Ref field will be extracted from the specified schematic library. PCB footprint, PCB3D and Simulation model information stored in the record will be added to the component as linked footprint, PCB3D and simulation models respectively.

For detailed information regarding the simulation model link fields that can be added to the external database, refer to the [Linking a Simulation Model to a Schematic Component](#) document.

The [Library Ref] entry must exist in the Design Parameter column mapped to the required Database Field Name in order to be able to place a component in that table from the Database Library. If the database table contains symbol reference information under a different field name, for example SCH

Symbol, you will need to set the associated Design Parameter entry for this field to [Library Ref], using the available drop-down list for that cell (Figure 11). Likewise, if model reference information is entered into the database using different field naming, you will need to manually map by choosing the appropriate Design Parameter entry ([Description], [Footprint Path], [Footprint Ref], [Footprint Ref n], [PCB3D Ref], [PCB3D Ref n], [Sim Model Name] and so on) from the drop-down list, for each field in turn.

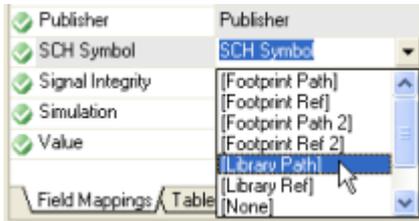


Figure 11. Manually defining the model mapping.

Library and model path mappings ([Library Path], [Footprint Path], [Footprint Path n], [PCB3D Path], [PCB3D Path n], [Sim File]) are optional. For more information see Specifying Symbol and Model Search Paths.

Parameters

All other database field names will be automatically 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. It is these design parameter names that will appear in the Parameters region of the component's associated properties dialog, once it has been placed on a schematic sheet.

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 component when placed on a schematic sheet. 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.

Parameter Update Options

The remaining columns on the Field Mappings tab (Figure 11 previously) allow you to specify the actions taken for parameters, when placing a component from the database library for the first time, or updating a component after it has been placed using the Update Parameters From Database command.

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 both on a schematic sheet and in the database, but the values are currently different. Choose to update the parameter of the placed component with the value stored in the database, or not to update at all. This option is obeyed when using the Update Parameters From Database command, after the component is placed.
- 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 placed 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. This option is obeyed both when initially placing the component from the database library

and when using the Update Parameters From Database command, after the component has been placed.

- Visible On Add – a checkbox in this column is used to determine whether a newly added parameter – resulting from initial placement or update after placement – is made visible for the component on the schematic sheet (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 placed 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. This option is obeyed when using the Update Parameters From Database command, after the component has been placed.

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 12.

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 12. Specifying parameter update options.

Looking at Figure 12, there are four important points to make regarding update options:

- Unmapped database fields will have no associated update options.
- Symbol and Model based mappings will have no associated update options, as these are not design parameters.
- The key field (e.g. Part Number in Figure 12) 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 on the Default Actions tab (Figure 13) of the Database Library Options dialog (**Tools » Options**). This dialog can also be accessed by clicking the Options button in the Field Settings region of the main window.

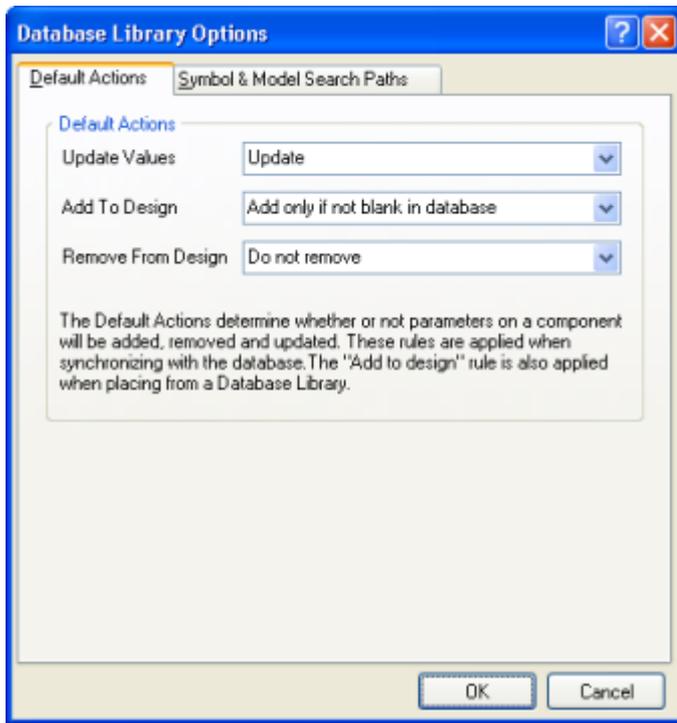


Figure 13. Setting default parameter update options.

The fourth 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-down providing the applicable update choices. (Figure 14).

Database Field Name	Design Parameter	Update Values	Add To Design	Visible On Add	Remove From Design
✓ DatasheetDocument	DatasheetDocument	Default	Default	<input type="checkbox"/>	Default
✓ Designator	Designator	Default	Default		Default
✓ Footprint	Footprint	Default	Do not add		Default
✓ Footprint Path	[Footprint Path]		Add Add only if not blank in database		

Figure 14. 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 Library 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. For placed components 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.

Specifying Symbol and Model Search Paths

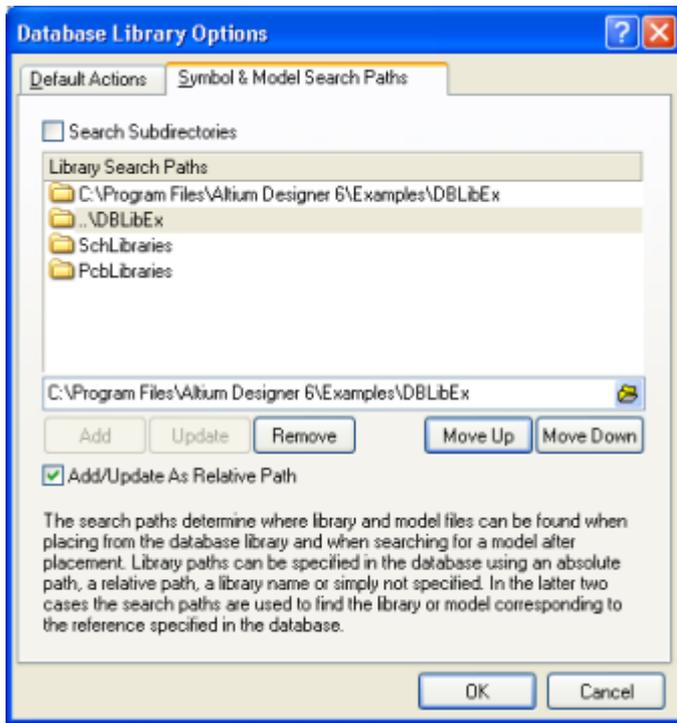


Figure 15. Specifying search paths for symbol and model libraries.

When you place a component from a database library its symbol – specified by the [Library Ref] mapping is extracted from the specified schematic library (*.SchLib). Similarly, any model references (footprint, PCB3D, simulation) specified in the database will reside in underlying PCB Library (*.PcbLib), PCB3D Library (*.PCB3Dlib) and Simulation Model (*.mdl, *.ckt) files. The paths to these files can be specified explicitly in the database by:

- Entering an absolute path to the file (e.g. C:\DBLibs\Precision\SchLibs\Capacitors.SchLib)
- Entering a relative path to the file (e.g. SchLibs\Capacitors.SchLib).

If you have defined fields in your database for path information, these fields need to be mapped to the appropriate design parameters – [Library Path], [Footprint Path], [PCB3D Path], [Sim File], and so on (refer back to Model and Parameter Mapping).

Entering paths – even relative – in a database table can be a little restrictive. If you move the location of a library or model file, you would need to update the database table accordingly. To give you even greater freedom, the database library feature provides the ability to specify library search paths as part of the Database Library file (Figure 15). This allows you to simply specify the name of the source library or model file in the database or, better yet, not to define it at all!

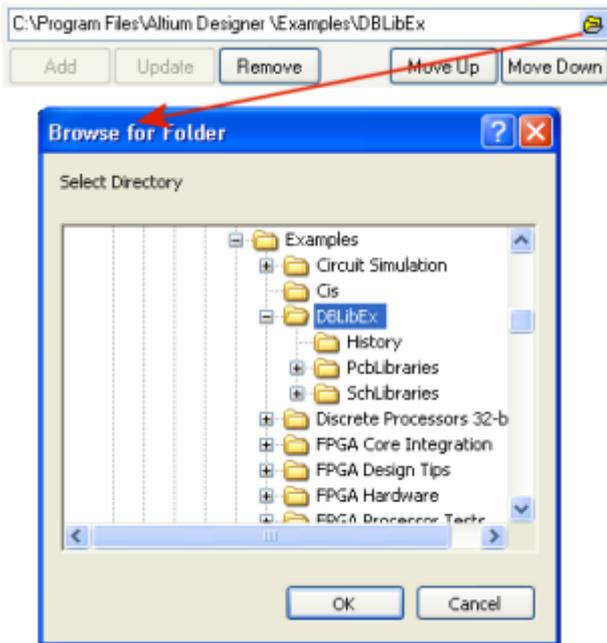


Figure 16. Entering a search path.

As can be seen from Figure 15, library search paths are defined on the Symbol and Model Search Paths tab of the Database Library Options dialog (**Tools » Options**). This dialog can also be accessed by clicking the Options button in the Field Settings region of the main window.

To add a path to the list, either type the path directly into the available field, or click on the  button to access the Browse for Folder dialog, from where you can locate the directory in which the required library/model file(s) reside (Figure 16).

After specifying the required path, add it to the search paths list by clicking the Add button. You can add either as a full path, or as a relative path (relative to the location of the DBLib file). For the latter, ensure that the Add/Update As Relative Path option is enabled.

For direct entry, if you specify an incorrect path (e.g. to a folder that does not exist) the entry can still be added but will appear grayed-out in the list, to distinguish that it is an invalid search path.

Continue adding search paths as required. If you find that you have entered a path erroneously, you can select it in the list and either click the Remove button, or modify its path definition and click the Update button.

Locating the Library Files

Tip:

For a simulation model, path information is entered in the Sim File field. Model reference information is entered in the Sim Model Name field.

The library search paths determine where library and model files can be found when placing from the database library and when searching for a model after placement. The specific schematic symbol, footprint model(s), PCB3D model(s), or simulation model used will depend on how you have set up your library search paths and whether you have added specific library information into your database. The searching will proceed as follows:

- If a full path exists in the mapped path field for the symbol or model, use that library/model file and extract the symbol or model specified in the applicable reference field.

- If a relative path exists in the mapped path field for the symbol or model, use that library/model file and extract the symbol or model specified in the applicable reference field.
- If only a library/model file name exists in the mapped path field for the symbol or model, use the search paths to locate the first library/model file that matches the specified name and which contains a match for the symbol or model specified in the applicable reference field.
- If no library/model file information exists in the database, use the search paths to locate the first library/model file containing a match for the symbol or model specified in the applicable reference field.

In the last two scenarios, the order of the search paths plays an important role. Use the Move Up and Move Down buttons on the Symbol and Model Search Paths tab of the Database Library Options dialog to order the list of search paths as required.

Countdown to Placement

Once you have defined the mapping and update options in your Database Library file, you are basically a hop-and-a-skip away from utilizing the fruits of your labor. Once you have saved your DBLib file you are ready to begin...

Adding the Database Library

Like any other library, a database library is made available in the Libraries panel by adding it to the Available Libraries list. From the Libraries panel, click the Libraries button to access the Available Libraries dialog.

The Database Library can be added as a project library or as part of the installed libraries - making it available to all projects and not just the active one. Alternatively, you can specify a search path to the folder in which the DBLib file resides. Figure 17 illustrates addition of a Database Library to the installed libraries list.

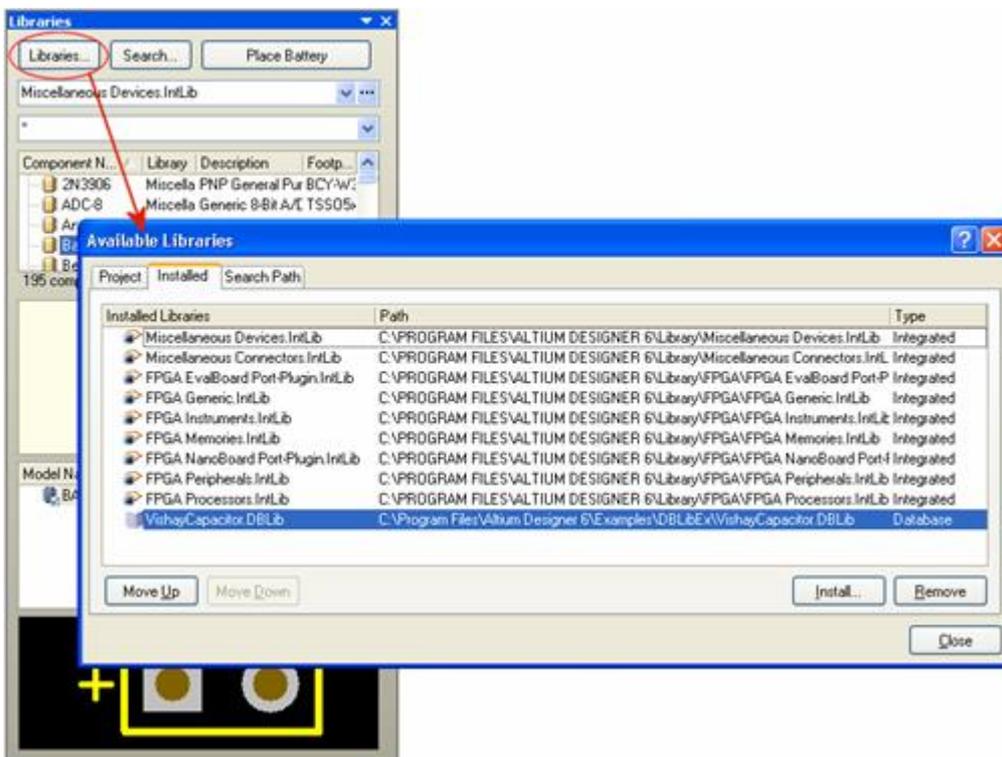


Figure 17. Making the Database Library available for use.

Browsing the Database Components

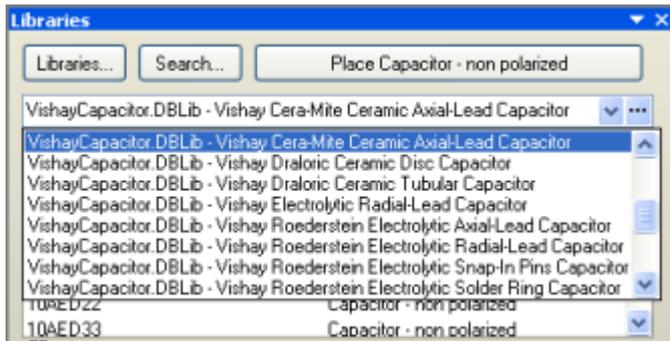


Figure 18. DBLib loaded.

Once the database library has been added to the Available Libraries list, the components in the database will become available for browsing in the Libraries panel. Although only a single DBLib file has been added, each table in the linked database will present as if it were a distinct library.

If the target database has been created using multiple Excel spreadsheet files (*.xls), there is a limit of 64 sheets that can connect, due to ODBC driver limitations.

The top drop-down list in the panel will populate with entries of the form:

LibraryName.DBLib – TableName

Each component entry in the panel corresponds to a record in that particular table of the database. In fact when browsing loaded database libraries, the Libraries panel behaves like a direct database browser.

Symbol and model information is populated from the underlying symbol and model libraries pointed to by the relevant fields of the database (and in conjunction with any search paths defined).



Figure 19. Browsing database components.

By default, only the Part Number and Library Reference fields are displayed in the component listing section of the panel. To 'expose' other fields in the database table, simply right-click in the component listing and choose Select Columns. This will give you access to the Select Parameter Columns dialog, from where you can enable the display of any of the additional fields in that table.

For more information on the Libraries panel, press F1 when the cursor is over the panel.

Searching for a Component

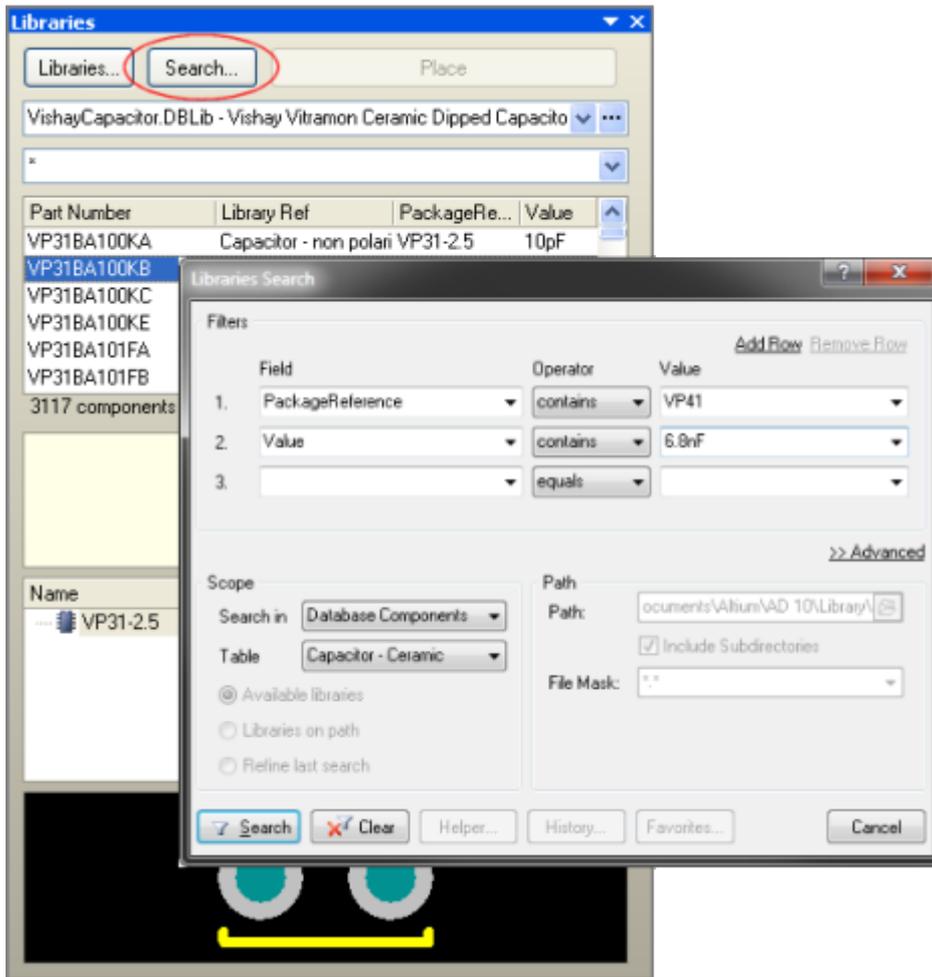


Figure 20. Accessing the libraries search facility.

The external database that you have connected to may include a number of tables, each with a sizeable amount of component records. Being able to place directly from a database is one thing, locating the specific component that you want to place is quite another. The latter is made efficiently manageable through the search facility of the Libraries panel.

Click the Search button at the top of the panel to open the Libraries Search dialog (Figure 20), from where you can conduct a search of database components in a single table of an installed DBLib file, or across all tables in all installed DBLib files.

To enable searching of database components, set the Search Type field to Database Components. Additional options for searching database library files will appear. The Scope and Path regions of the dialog become grayed-out, as their options are not relevant when searching database libraries.

Figure 21. Database Components - search options.

When searching database libraries, the search facility essentially offers three 'levels' of searching - Simple, Altium Designer-based querying and SQL-based querying.

Simple Search

Tip:

The query results will only be displayed in the Libraries panel if the Components browse mode option for the panel is enabled.

This is a 'coarse' search that will return all database components found in either a specific table or all tables. Use the available drop-down field (also referred to as the table scope) to select a specific table that you know the required component to be in, or leave the scope set to <Search All Tables>.

Click the Search button to begin the search. The Libraries Search dialog will close and the results of the search will be listed in the Libraries panel. When searching a single, specific table, the libraries drop-down list will contain the entry Query Results. If you have searched all tables, a separate entry will appear in the list for each table. Each entry in the drop-down list will be of the form: Query Results - TableName (n) where (n) is the number of matching components found.

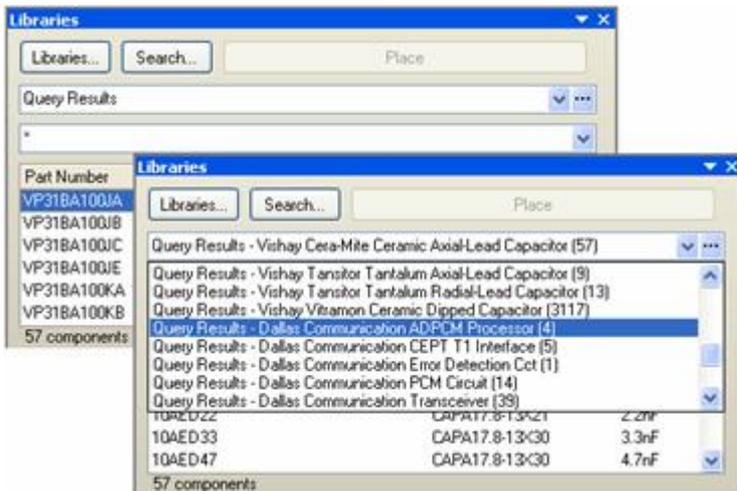


Figure 22. Search results for a single table or all tables.

Altium Designer Query Search

This is a 'fine' search that will return all database components that fall under the scope of a defined Altium Designer query expression. Again, you can apply the query to a specific table or all tables. By using query expressions you can narrow a search and therefore minimize the list of resulting items returned by that search. For example, you may want to search for database components with a specific package reference or pin count.

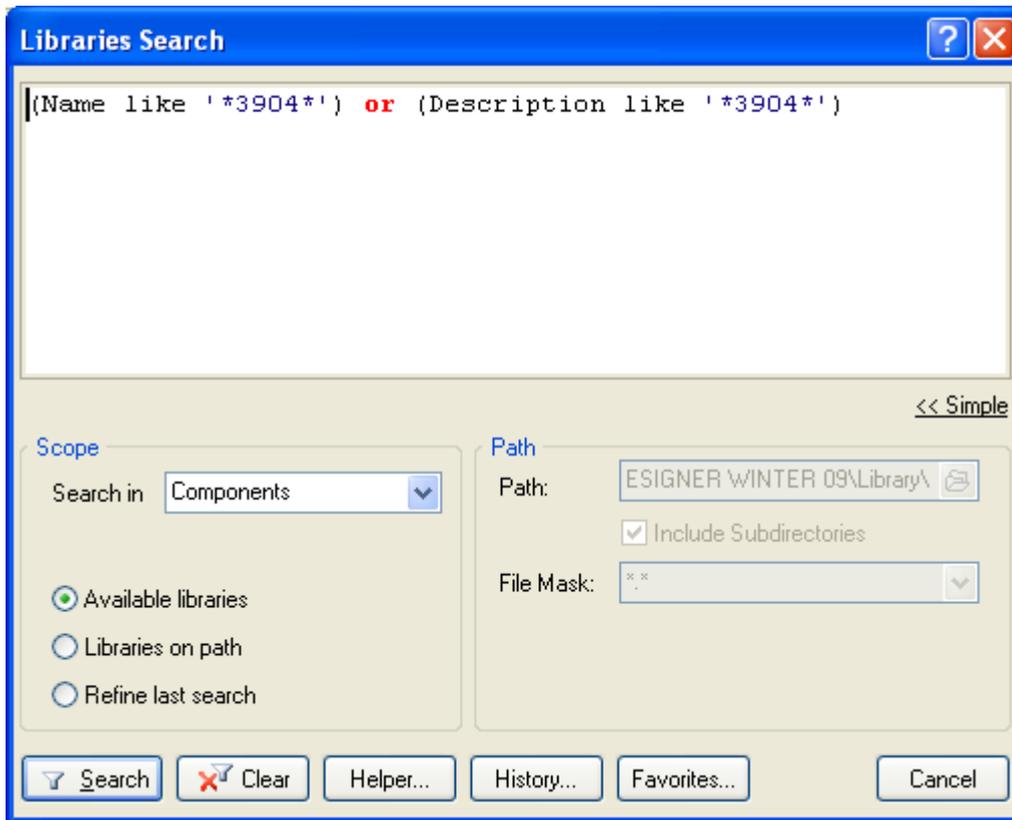


Figure 23. Example Altium Designer query.

The top section of the Libraries Search dialog (also referred to as the query editor section) allows you to construct filters through the entry of logical queries. You can type a query directly into the field. Figure 23 shows an example of such a search – searching for components across all tables, where only matches against the specified query expression should be returned, in this case all components whose Package Reference field contains VP41 and whose value is 6.8nF. After defining the query and setting the table scope as required, click Search to proceed. The results will be loaded into the Libraries panel. If searching all tables, separate query results will be available for each table where a match was found.

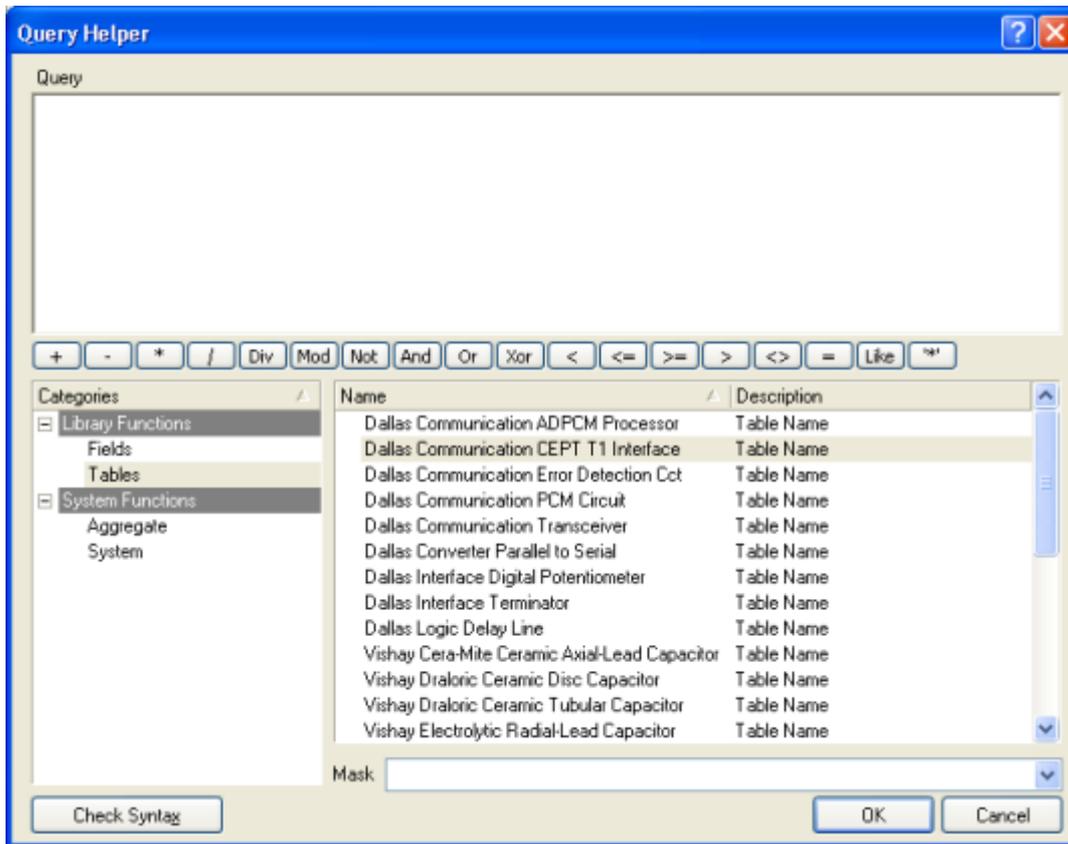


Figure 24. Use the Query Helper to create a query expression.

A facility is available to provide aid in the creation of queries – the Query Helper. This facility can be very useful if you are unsure of the syntax of a query or the possible keywords that you may want to use. To use the Query Helper, click the Helper button in the Libraries Search dialog. The Query Helper dialog will appear.

Use the top section of the dialog to compose a query expression, using the available Library Functions and System Functions. The former contains two sub-lists:

- Fields – all unique fields found across all tables in all installed database libraries.
- Tables – all table names found in all installed database libraries.

The mid-section of the dialog provides a range of operators for use when constructing an expression.

For conceptual information about the Altium Designer query language, refer to the [Introduction to the Query Language](#) article.

For more detailed information about using the query language, refer to the article [An Insider's Guide to the Query Language](#).

For information about a specific keyword in the Query Helper dialog, either click inside the keyword and press F1, or select the field entry in a list and press F1.

SQL Query Search

This level of searching also offers a 'fine' search based on a specified query expression, but instead of using the Altium Designer query language, you can enter a direct SQL query. Querying of a single,

specific table only, is supported using this method.

If you are conversant with SQL, you can simply enable the SQL Query option and type your expression directly into the query editor section of the Libraries Search dialog.

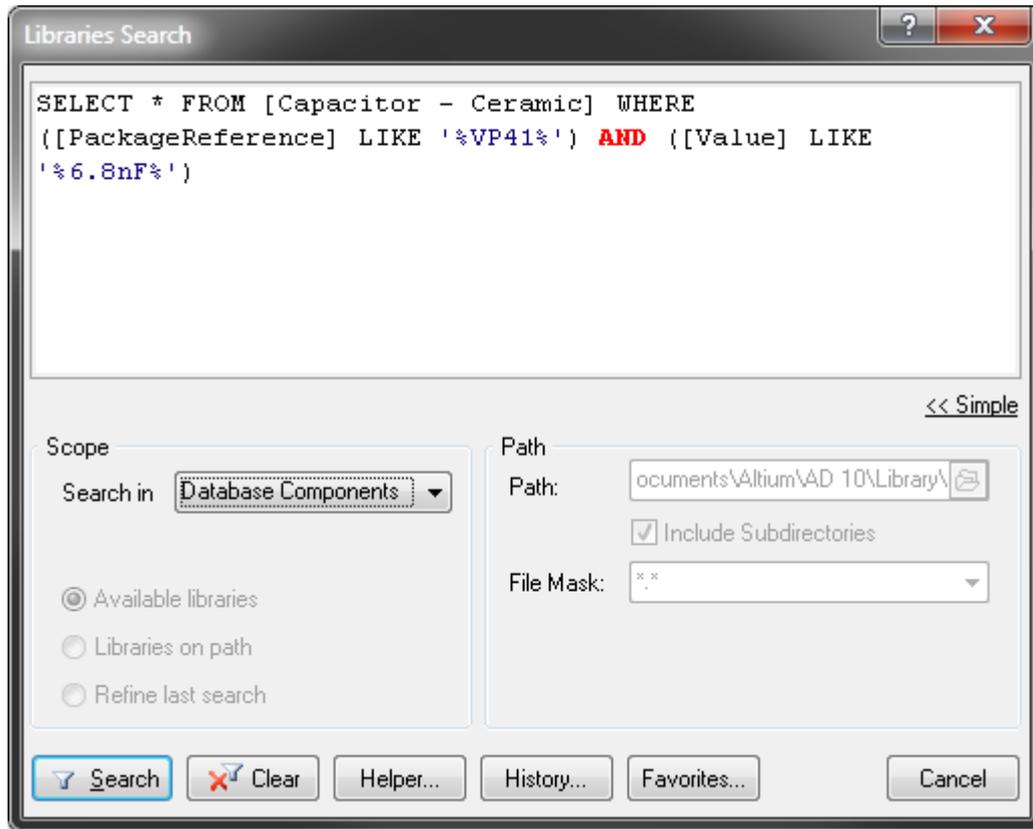


Figure 25. Directly specifying an SQL search query.

Alternatively, you can define an expression using the Altium Designer query language, then enable the SQL Query option - the query will be translated into SQL (Figure 26). For best translation results, ensure that the table scope setting is set to a specific table before enabling the SQL Query option.

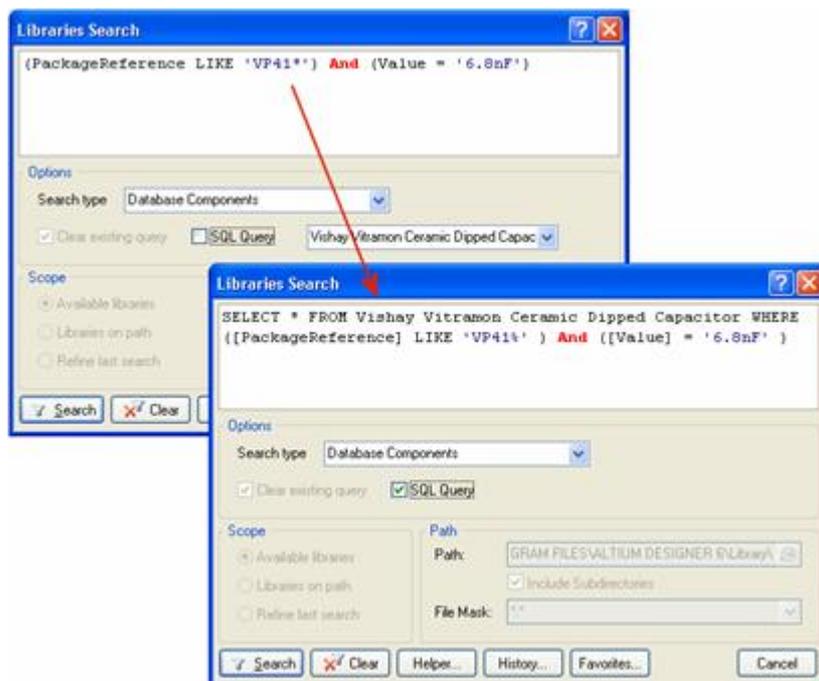


Figure 26. Converting an Altium Designer query to an SQL query.

Placing a Component

Now comes the "piece de resistance" – the moment where you place a component directly from your linked database. With the target schematic sheet open as the active document in the main design window, simply select the component you want to place from the Libraries panel and either click the Place button or drag-and-drop the component directly onto the sheet. At this point the following actions take place:

- The schematic symbol specified in the database record is retrieved
- Any referenced footprint, PCB3D and simulation models are linked in
- The design parameters specified in the DBLib file are added (in accordance with the associated Add To Design update setting) to the component.

After placement, double-click on the component on the schematic to access its properties dialog (Figure 27). You will see the design parameters added to the Parameters region of the dialog, the linked models added to the Models region and another region – Library Link – Database Component.

This region provides the following information:

- The parent Database Library file.
- The specific database table in which the component resides.
- The value for the Physical Component. This is the chosen key field defined in the Field Settings region of the DBLib file, and is typically the part number.
- The Logical Symbol for the component. This is the schematic symbol specified for the component.



Figure 27. Information added to the component upon placement.

- You can change the placed component to another from the same table by clicking the Choose button. This will open the Browse Libraries dialog, initially listing all components in the same table. In fact, you can change to a component in a different table of the same DBLib file, or browse to one in a different DBLib file altogether. The region will update accordingly with the new information for the chosen database component.

Ensuring Synchronicity

Tip:

In the PCB Editor, use the Tools » Update From PCB Libraries command to update placed footprints with the latest information stored in the source libraries.

After placement, the chosen key field parameter is used to ensure that the placed component on the schematic 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 placed component, synchronizing the two.

If you simply want to update parameter information, use the Update Parameters From Database command, available from the Schematic Editor's main Tools menu.

To perform a full update, including parameters, model and graphical attributes of schematic symbols, use the Update From Libraries command (also available from the Schematic Editor's main Tools menu).

For further information on using the various update tools, refer to the [Keeping Components Up-To-Date](#) document.

Dual Synchronization - DBLib 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 (DBLib) feature. The associated DBLib 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 files, the database linked by the DBLink file will be searched first for a match, followed by the database linked to by the DBLib 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

Tip:

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

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 you place components from a Database Library the BOM Generator is able to extract any other record information that has not been added as design parameters at the time of placement.

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.

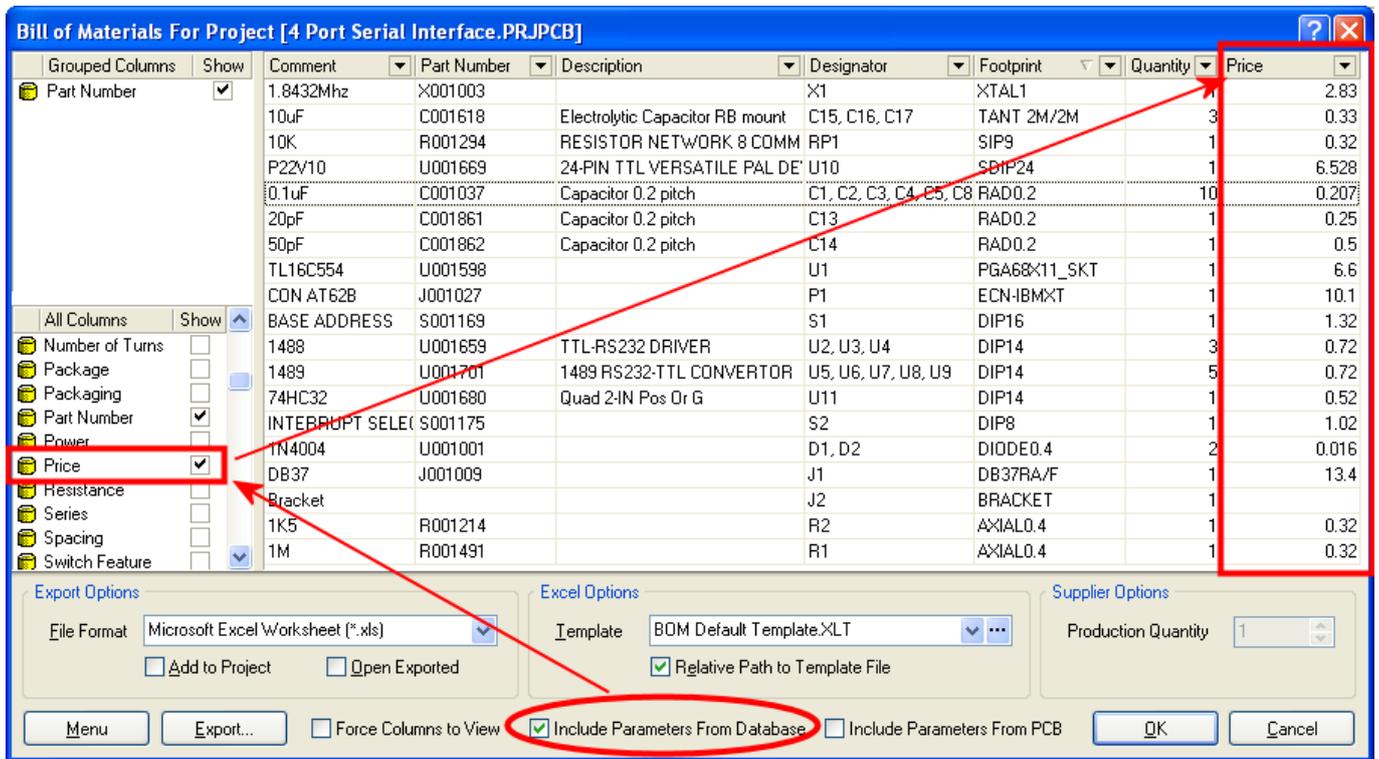


Figure 28. Include additional component information that exists only in an external database.

Source URL:

<http://techdocs.altium.com/display/ADOH/Using+Components+Directly+from+Your+Company+Database>