コンポーネント データベースからの直接使用

概要

このアプリケーションノートでは、Altium Designer のデータベースライブラリ機能を使ってデータベースからコンポーネントを使用するための詳細情報を記載しています。

Application Note
AP0133 (v2.5) April 18, 2008

Altium Designer では、データベース ライブラリ (*.DBLib)を作成し、それを使用することで社内のデータベースから直接コンポーネントを配置できます。配置は、データベースライブラリをインストール後、**Libraries** パネルからデータベースを参照するような形で行えます。

配置後、デザインパラメータの情報は、配置したコンポーネントとそれに対応するデータベースのレコード間で、Update Parameters From Database コマンドを使用して同期を取ることができます。グラフィカルシンボル、参照するモデル、パラメータを含む、コンポーネントすべての情報は、Update From Libraries コマンドで更新可能です。

リンクを超えたダイレクトな配置

Altium Designer では、コンポーネントとデータベースをリンクする方法として、データベース リンクファイル(`*.DBLink`) を使用する方法と、データベース ライブラリファイル Database Library file (`*.DBLib`)使用する方法の 2 通りが用意されています。前者は、デザイン（またはライブラリ）に使用されているコンポーネントとデータベースに入力されているデータを効率的にリンクする為の手段として用意されています。後者は、この効率的なリンクを使ってコンポーネントを配置する際、データベースのデータを自由に追加し、データベースから直接配置することができます。これは実質的に対応するデータベースレコードに保持されている情報からコンポーネントをダイナミックに作成しています。

データベースライブラリの機能詳細を紹介する前に、これらの 2 つの方法の違いを説明します。

SVN データベース ライブラリファイル (*.SVNDBLib)を使用することでもデータベースとのリンクが取れます。DBLib の拡張版である、このタイプのライブラリは、DBLib と同じように使用されますが、回路図シンボルと関連しているモデルはバージョンコントロール(Subversion)が管理するライブラリに保存されます。

より詳細な情報は、アプリケーション ノート *Working with Version-Controlled Database Libraries* を参照してください。

データベース リンクファイル(`*.DBLink`)を使用したリンク

この方法を使用するには、データベース リンクファイルで回路図コンポーネントとデータベース間で一致しているレコードのリンクを定義します。レコードの一致は、ひとつの（例えば、Part Number）キーフィールドによるリンクで定義するか、あるいは（**Where** 句の定義によって）複数のキーフィールドの一致によって確立されます。

このリンク方法では、コンポーネントのモデルとパラメータ情報を Altium Designer のライブラリコンポーネントの一部として、事前に定義する必要があります。これを定義し、データベースリンクドキュメントを統合ライブラリプロジェクトか、PCB プロジェクトに追加することでコンポーネント情報（パラメータ）とデータベースのフィールド内容との同期が取れます。

多くのデータベースコンポーネントは、同じコンポーネントシンボルを共有することができるので、各データベース レコードで定義された実際のコンポーネントをそれぞれ異なる Altium Designer ライブラリのコンポーネントに割り当てる必要はありません。このリンク方法は通常、「データベースレコードと Altium Designer コンポーネントを 1 対 1 で対応させる」形で使用されます。この Altium Designer コンポーネントとは、回路図シートに置かれたインスタンス、またはコンポーネントライブラリのコンポーネントのどちらでも可能です。

DBLink 形式のデータベースリンクでは、データベース リンクファイルをプロジェクトに含める必要があります。

既存のAltium Designer コンポーネント（回路図に配置されたコンポーネントや回路図コンポーネントライブラリの一部）とのリンクの詳細は、アプリケーションノート *Linking Existing Components to Your Company Database* を参照してください。

データベース ライブラリファイル(*.DBLib) を使ったリンク

この方法を使用する場合も、データベース ライブラリファイルで回路図コンポーネントとデータベース間で一致しているレコードのリンクを定義します。また、レコードの一致は、ひとつの（例えば、Part Number）キーフィールドによるリンクで定義するか、あるいは（**Where** 句の定義による）複数のキーフィールドの一致によって確立されます。

この方法では、コンポーネントのシンボル、モデル、パラメータ情報は、外部データベースで定義されたコンポーネントの一部のレコードとして定義されたものがリンクされます。参照される回路図コンポーネント（コンポーネントライブラリ (*.SchLib)）は単に定義されたシンボルだけがある空のシェルです。モデルやデザインパラメータとはリンクされていません。

コンポーネントが配置される時に定義したマッピングに従い、一致しているデータベース レコードに対応しているフィールドを使ってパラメータとモデル情報がその時点で作成されます。これらのパラメータは、配置した後で同期が取れるよう、データベースのリンクを維持するため一致条件として使用されます。

このリンク方法では、配置を行う時点でコンポーネントがダイナミックに作成されるので、「複数のデータベースレコードをひとつの Altium Designer コンポーネント」に割り当てる形がよく使われます。

DBLink ファイルは、プロジェクトに含める必要がありますが、DBLib ファイルは、DBLink 形式のデータベースリンクとは異なり、プロジェクトに追加する必要はありません。データベース ライブラリは、**Libraries** パネルを呼び出し、利用可能なライブラリとして登録するだけです。利用可能なライブラリとは、Project Libraries（プロジェクト内）、Installed Libraries （導入済み）、指定した検索パスで検索できるライブラリです。

DBLib ファイルは、ライブラリ指向のファイルになります。例えば、一つのシンボルで社内データベースのすべての抵抗やコンデンサなどを持つことができます。

バージョンコントロールにより、回路図シンボルやモデルを保存するSVN データベースライブラリファイル (*.SVNDBLib)についての詳細は、アプリケーションノート *Working with Version-Controlled Database Libraries* を参照してください。

データベースライブラリの作成

先に説明したように、データベースライブラリ機能のバックボーンはデータベースライブラリファイルです。このファイルは、Altium DesignerのDatabaseLib エディタ(図 1)を使用することで作成、管理ができます。

サンプルのデータベースライブラリ VishayCapacitor.DBLib は、\Examples\Cis\Example DBLib フォルダにインストールされています。

コンポーネント及びライブラリに関する基本的な情報については、アーティクル *Component, Model and Library Concepts* を参照してください。

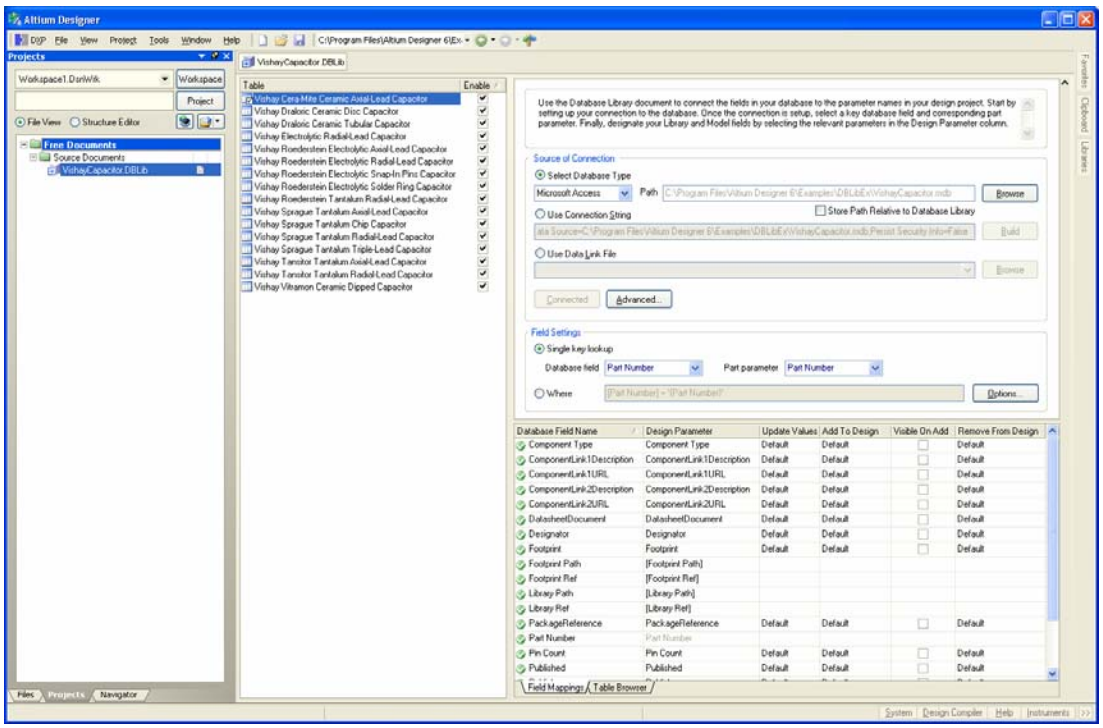


図 1. DatabaseLib エディタ

メインのデザインウィンドウに、*.DBLib ファイルを読み込めば DatabaseLib エディタがアクティブになります。このファイルは、**File » New » Library » Database Library** コマンドで新規に作成できます。

外部データベースとの接続

テーブルとマッピングデータは、アクティブなデータベース ライブラリファイルが必要とする外部データベースと接続が確立した後、エディタのメイン表示ウィンドウにのみ現れます。接続は、ウィンドウの**Source of Connection** (図 2) のところで定義できます。

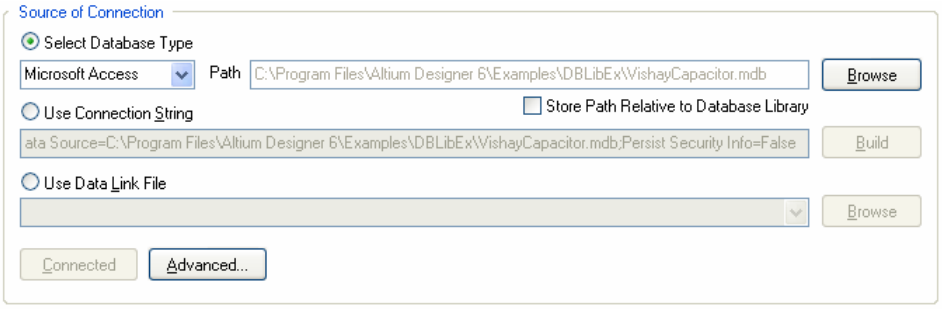


図 2. 外部データベースとの接続を指定

接続は、Tools メニューから *Database Connection* ダイアログに呼び出し、**Connection** タブで行ないます。

OLE DB をサポートしているデータベースであれば、すべて接続が可能です。ウィンドウのこの領域で提供されているオプションは、接続の対象となるデータベースの OLE DB コネクションストリングが使用されます。いくつかのデータベースは、OLE DB をサポートしていない可能性がありますが、現在使用されているほとんどのデータベース管理システムは、Open Database Connectivity (ODBC)インターフェースを経由してアクセスできます。データベース ライブラリ の機能は、Microsoft の ODBC provider を使用します。これは、ODBC データソースへ ADO (ActiveX Data Object)を使用して接続します。 これによって、どの ODBC データベースとの接続が可能です。OLE DB provider の ODBC データベースは、コネクションストリングの一部として指定します。

Access 、Excel データベースをすぐに接続

Select Database Type は、接続対象のデータベースが Microsoft Access か Microsoft Excel の場合、コネクションストリングをすぐ作成できるように用意されたオプションです。このオプションを使用するとデータベースのタイプが選択でき、必要なデータベースを指定します。対応したコネクションストリングが、自動的に作成され、**Use Connection String** オプションのフィールドに入力されます。

データベース ライブラリファイルは、絶対パス、または、相対パスでの指定が可能です。

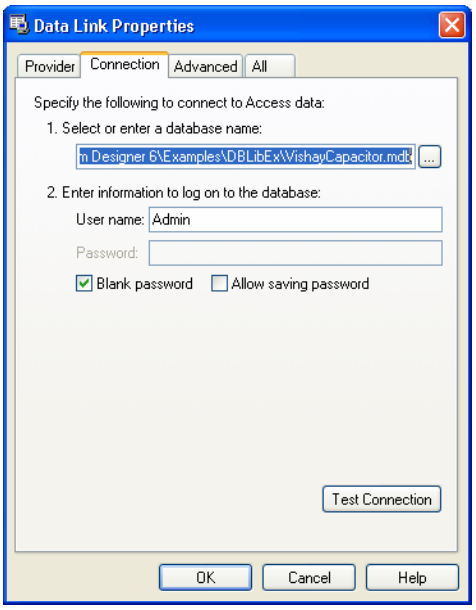


図 3. 接続する外部データベースの指定

コネクションストリングの構築

社内データベースが Access や Excel で無い場合やコネクションストリングを自身で作成したい場合は、**Use Connection String** オプションを有効にし、右側にある **Build** ボタンをクリックします。*Data Link Properties* ダイアログ(図 3)が表示されます。

Connection タブがオープンされますが、このダイアログの **Provider** タブでは、OLE DB Provider – Microsoft Jet 4.0 がデフォルトで設定されています。これが新規データベース ライブラリファイルを作成した場合のデフォルトのプロバイダ設定となり、Access データベースファイル(`*.mdb`)を接続するために使用されます。必要に応じ、プロバイダは変更してください。

Connection タブから接続したいデータベースの（パスを含んだ）名前を入力します。あるいは、**...** ボタンを使ってダイアログをオープンし、必要なファイルを参照し、オープンしてください。

お使いのデータベースで、ログインする必要があればここで入力します。その他の詳細設定は、ダイアログの **Advanced** タブから設定できます。**All** タブでは、リンクオプションの全リストが用意され、選択した OLE DB Provider に関する拡張オプションも用意されています。

リンクオプションが定義されると、(**Connection** タブにある) **Test Connection** ボタンをクリックすることで正しく接続されているかチェックできます。接続がうまくいけば、ダイアログが表示されます。

Data Link Properties ダイアログは、Microsoft のダイアログです。**F1** キーや **Help** ボタンを使うことで Microsoft Data Link Help が表示されます。このファイルは、Altium Designer のドキュメントではありません。

データリンク ファイルの指定

接続したいデータソースが Microsoft Data Link ファイル(`*.udl`)を使って記述されていれば、3 番目の接続オプションである、**Use Data Link File** を有効にし、**Browse** ボタンをクリックし、必要なファイルを指定します。データリンクファイルは、基本的にコネクションストリングを保存するための器です。

接続の実行

外部データベースとの接続を定義した後は、**Connect** ボタンの文字が太文字になり、接続を実行できることを示します。接続の詳細が正しければ、対象となるデータベースのテーブルとマッピング情報がデータベース ライブラリドキュメントに読み込まれます。**Connect** ボタンの文字は、`Connected` に変わり、ボタンはグレイアウトします。

コネクションストリングが正しく無い場合や入力されたパスが間違っている場合など、接続の詳細に問題がある場合は、これに関する警告のメッセージ (図 4) が表示されます。

図 4. 接続に失敗した状態

接続していたデータベースの接続設定を変更した場合、接続が切断され、**Connect** ボタンの文字が `Reconnect` に変わります。この場合、ボタンをクリックし、再度、接続を確立してください。

コネクション設定をチェックし、再度 **Connect** ボタンを押します。

初期接続がうまくいった後は、データベース ライブラリファイルを保存します。ファイルを保存した後は、ファイルをオープンしたときに対象となるデータベースファイルの保存場所やファイル名が変更されていなければ、自動的に接続が行われます。

DBLibは、統合ライブラリから**Integrated Library to Database Library Translator Wizard** を使って作成することも可能です。詳細については、アプリケーションノート *Database Library Migration Tools* を参照してください。

データベース テーブルリスト

外部データベースとの接続が成功すれば、テーブルとマッピングデータが読み込まれます。表示ウィンドウの左側には、接続したデータベースに存在する全テーブルがリストアップされます。(図 5)

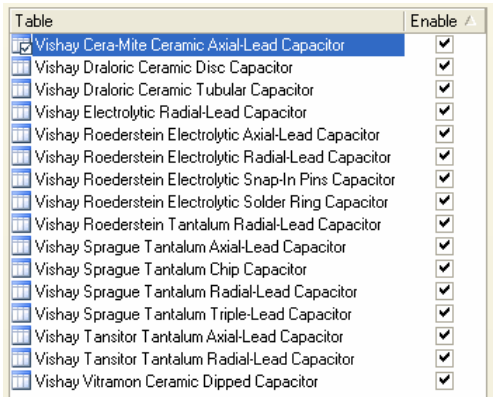


図 5. 接続したデータベースに存在するテーブル

対象となるデータベースが Excel ファイル(`*.xls`)で作成されている場合、ODBC ドライバの制限により 64 シートまで接続が可能です

各テーブル名の横にある**Enable**オプションで、そのテーブルをデータベース ライブラリとして使用するかどうかをコントロールします。データベース ライブラリを**Libraries**パネルで参照できるようにAvailable Libraries のリストに追加すると、各テーブルは、別のライブラリとして表示されます。実際には、ひとつのデータベース ライブラリが追加されただけですが、**Libraries**パネルから見ると、複数の異なるライブラリが追加された状態になります。詳細は*データベースライブラリの追加* を参照してください。

リスト上のテーブル名をクリックすると、現在アクティブになっているテーブルを区別する為に、アイコンが から へ変わります。そのテーブルにある全データが、表示ウィンドウの **Table Browser** タブに現れます(図 6)。これはテーブルのコピーで編集はできませんが、外部のデータベースそのものを起動せず、内容をすぐに確認できます。

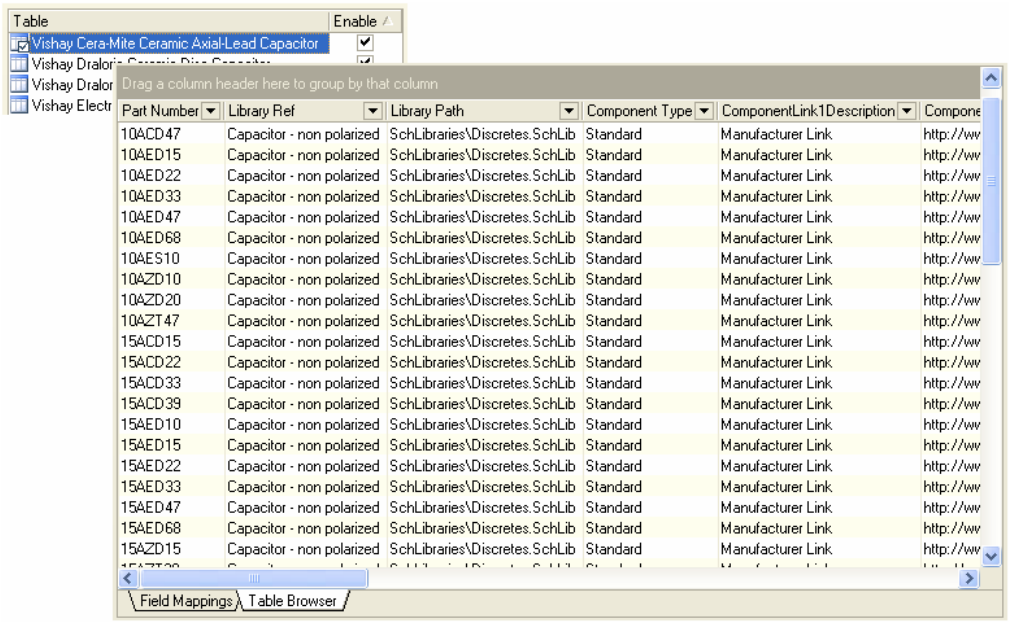


図 6. 接続したデータベースのソーステーブルを参照

一致条件の指定

コンポーネントが外部のデータベースから配置された後、そのコンポーネントとそれを作成する為に使用されたデータベース レコードの情報は、リンクを維持するために何らかの方法が必要です。基本的にその二つが一致している必要があります。

コンポーネントが配置されるとき、そのパラメータ情報はデータベースレコードの対応するフィールドを使って、その時点で作成されます。配置後、回路図コンポーネントとデータベースレコード間でのリンクこれらの一つのパラメータ、または複数パラメータを使ってリンクが確立されます。メインの表示ウィンドウ(図 7)にある**Field Settings**エリアで簡単な単キーの参照、または、**Where** 句を使って、より高度な一致条件を定義することができます。

一致条件の指定は、各テーブルで指定を行います。

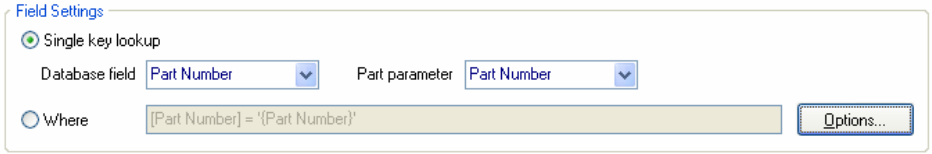


図 7. 配置後の一致条件をコントロール

単キー参照

Single key lookup オプションが有効（デフォルト）であれば、**Database field** 及び **Part parameter** フィールドが使用できます。前者はデータベースのアクティブなテーブルで利用可能な、すべてのフィールド名(コラムヘッダー)がリストアップされます。回路図コンポーネントのパラメータは、配置されるときに追加されるので、**Part parameter** のフィールドは、データベースのフィールドで選択されたものを反映しているだけです。

通常、参照キーフィールドは、Part Number のように、外部データベースにおいて各コンポーネントを区別できるものが使用されます。選択された参照フィールドは、ウィンドウの **Field Mappings** タブで区別され、**Design Parameter** の入力は、グレイアウトで表示されます。

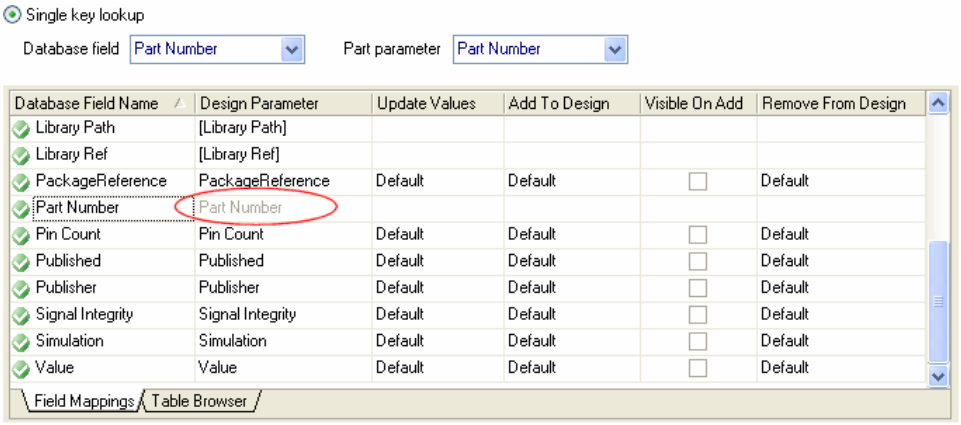


図 8. Part Number による単キー マッピング

配置後のある時点で Update Parameters From Database 機能が使用されるとき、情報は、配置された回路図コンポーネントで選択されたキーパラメータから読み込まれ、有効なすべてのテーブルにあるデータベースの選択された（キー）フィールドが検索されます。一致しているものがあれば、親テーブルのレコードにある他のセルからの情報を回路図コンポーネントにマッピングされているパラメータに戻すことができます。

高度な一致 – Where 句

一致させるために値が重複しないパートナンバー／ID があれば、**Single key lookup** オプションがうまく動作するはずですが、コンデンサの容量値や抵抗値など、同じ値が存在するパラメータによって一致させる場合は、これは効果的ではありません。このケースでは、回路図コンポーネントとソースのデータベースレコードとリンクを取るために、より高度な **Where** 句を使用して複数のキーを指定することができます。

単キー参照を定義すると一番簡単な形式の **Where** 句 (SQL 構文を使用して記述) が反映されます。例えば、**Database field** で `Part Number` が選ばれると **Part parameter** フィールドも自動的に `Part Number` に設定され、**Where** 句には、次のような値が入力されます：

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

データベースフィールド（テーブルカラム）を囲んでいる [ ] 括弧は、引用文字で、**Database Connection** ダイアログ(図 9)の **Advanced** タブで指定します。このダイアログにアクセスするには、ウィンドウの **Source of Connection** のところにある、**Advanced** ボタンをクリックするか、または、**Tools** メニューを使用します。

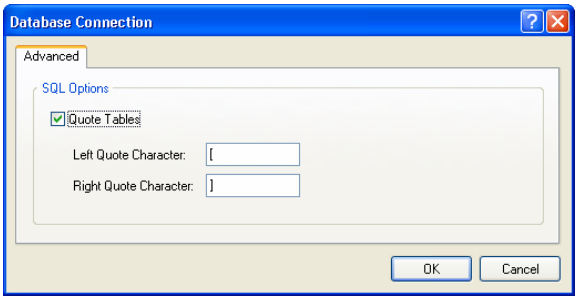


図 9. SQL 引用コントロール

{ }括弧は、入力がデザインパラメータとして参照されることを指定します。シングルクォートは、デザインパラメータを文字列として扱うことを指定するために使用されます。数字として扱う場合は、クォートは不要です。SQL では、型を認識するので型を一致させることは非常に重要です。デザインパラメータは、データベースのカラムと同じ型で作成しておく必要があります。

標準の SQL 構文を使い、**Where** 句によって複数のデータベースフィールド／パートパラメータを使って一致させることができます。例えば：

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

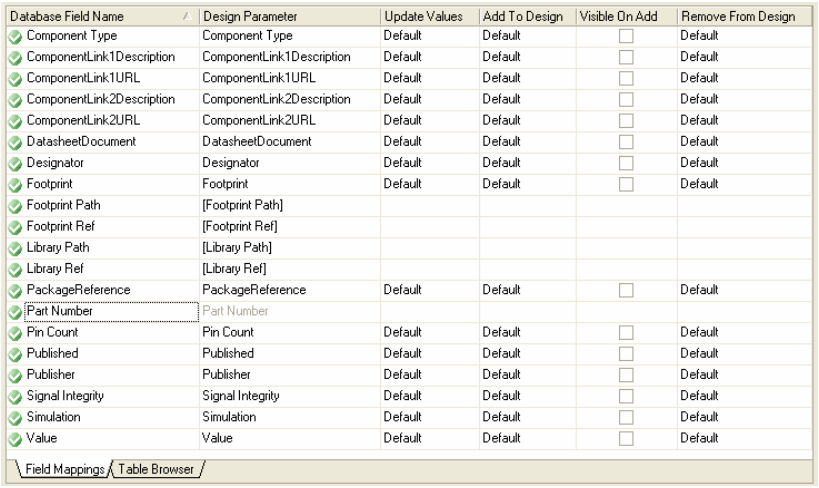
このケースでは、データベースの関連するテーブルのひとつのレコードを 3 つの異なるデザインパラメータを使ってリンクさせることになります。デザインパラメータ `Tolerance` にはクォートが無いことに注意してください。これは、データベースのテーブルのカラムの型が文字列ではなく、数値であることを意味しています。

標準の SQL 構文を使って、使用する人の好みに応じて簡単な、または、複雑な **Where** 句を作成することができます。

データベースフィールドをデザインパラメータにマッピング

データベースライブラリから配置されたコンポーネントのデザインパラメータは、配置時に作成／追加されます。これらのパラメータは、データベースライブラリファイルでマッピングと更新情報を指定し、配置をした後、**Update Parameters From Database**コマンドを使って、それらの情報を更新するためのオプションとして作成されます。これらの設定は、データベースライブラリのエディタのメインディスプレイウィンドウの**Field Mappings**タブ(図 10)で行われます。

マッピングと更新のオプションは、テーブル別に指定します。



Database Field Name	Design Parameter	Update Values	Add To Design	Visible On Add	Remove From Design
Component Type	Component Type	Default	Default	☐	Default
ComponentLink1Description	ComponentLink1Description	Default	Default	☐	Default
ComponentLink1URL	ComponentLink1URL	Default	Default	☐	Default
ComponentLink2Description	ComponentLink2Description	Default	Default	☐	Default
ComponentLink2URL	ComponentLink2URL	Default	Default	☐	Default
DatasheetDocument	DatasheetDocument	Default	Default	☐	Default
Designator	Designator	Default	Default	☐	Default
Footprint	Footprint	Default	Default	☐	Default
Footprint Path	[Footprint Path]				
Footprint Ref	[Footprint Ref]				
Library Path	[Library Path]				
Library Ref	[Library Ref]				
PackageReference	PackageReference	Default	Default	☐	Default
Part Number	Part Number				
Pin Count	Pin Count	Default	Default	☐	Default
Published	Published	Default	Default	☐	Default
Publisher	Publisher	Default	Default	☐	Default
Signal Integrity	Signal Integrity	Default	Default	☐	Default
Simulation	Simulation	Default	Default	☐	Default
Value	Value	Default	Default	☐	Default

Field Mappings | Table Browser

図 10. パラメータマッピングとアップデートオプションの指定

モデルとパラメータのマッピング

Field Mappings タブの最初（左）の 2 つのカラムで、データベースからの情報とコンポーネントのモデル、及びパラメータとのマッピングが設定できます。

Database Field Name のカラムには、現在アクティブなデータベースのテーブルのすべてのフィールド（カラム）名が一覧表示されます。**Design Parameter** のカラムでは、データベースで使用されている各フィールドを、デザインパラメータとしてマッピングし、回路図コンポーネントのソースとして使用するか、特定のモデルにリンクするか、コンポーネントに付属するかを定義します。

イニシャルのマッピングは、接続したデータベースのすべてのフィールドを自動的にマッピングします。

データベースからマッピングさせたくないフィールドは、**Design Parameter** の入力を [None] にします。マッピングされないデータベースフィールドは、このタブ上で赤いクロスのアイコン(✖)が使用されます。マッピングされたデータベースフィールドは、緑のチェックマークのアイコン(✔)になります。

マッピングされていないフィールドを再度、マッピングするには、フィールドを選択し、ショートカットキーCTRL + D を使用します。モデルのマッピングは、関連する Design Parameter のドロップダウンリストからマニュアルで選択する必要があります。

モデル

データベースフィールド名が、次に記載されている予約された名前の一つであれば、対応したモデルマッピングは、自動的に **Design Parameter** フィールドに設定されます：

```
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]
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]
```

フットプリント及びPCB3Dモデルのリファレンス(及びパス)は、DBLibファイルのデータベーステーブルに無制限に指定及びマッピングできます。左記の予約された名称にあるnは、2から始まる整数です。

外部データベースのコンポーネントでリンクできるシミュレーションモデルは、一つだけです。通常、コンポーネントには、一つのシミュレーションモデルをリンクします。複数のシミュレーションモデルをリンクさせたい場合、他のリンクは、ソースの回路図ライブラリファイルに定義しておく必要があります。

注意：ここに記載されているPCB3Dは、レガシー3Dビューアで使用されます。現在の3Dビューアは、コンポーネントフットプリントに関連付けしたコンポーネント外形とインポートしたSTEPモデルを3次元表示することができます。

これらのマッピング定義は、コンポーネントのシンボルとモデル情報を定義します。コンポーンネントが配置されると、データベースレコードのLibrary Refフィールドで指定された回路図シンボルが回路図ライブラリから抽出されます。レコードに保存されたPCBフットプリント、PCB3D、シミュレーションモデルの情報は、コンポーネントに追加され、それぞれフットプリント、PCB3D、シミュレーションモデルとしてリンクされます。

外部データベースに追加できるシミュレーションモデルリンクフィールドについては、アプリケーションノート *Linking a Simulation Model to a Schematic Component* を参照してください。

`[Library Ref]` エントリは、**Design Parameter**カラムに存在する必要があり、データベースライブラリからコンポーネントが配置できるようにする為、適切な**Database Field Name**にマッピングする必要があります。例えば、`SCH Symbol`のようにデータベーステーブルに異なるフィールド名でシンボルへの参照情報が含まれている場合は、関連付けを行う**Design Parameter**の入力をセル(図 11)で利用できるドロップダウンリストを使って`[Library Ref]`に設定する必要があります。同じくデータベースでモデルへの参照情報が異なるフィールド名で入力されている場合は、適切な**Design Parameter**を選び、(`[Description]`, `[Footprint Path]`, `[Footprint Ref]`, `[Footprint Ref n]`, `[PCB3D Ref]`, `[PCB3D Ref n]`, `[Sim Model Name]` など)各フィールドを順に選び、ドロップダウンリストからマニュアルでマッピングする必要があります。

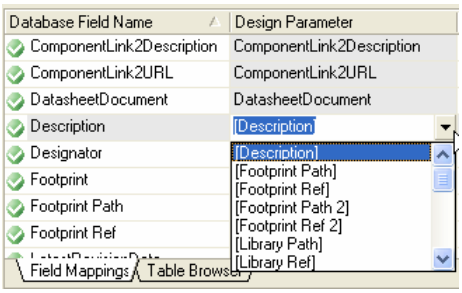


図 11. マニュアルでのモデルマッピング定義

ライブラリとモデルパスのマッピング(`[Library Path]`, `[Footprint Path]`, `[Footprint Path n]`, `[PCB3D Path]`, `[PCB3D Path n]`, `[Sim File]`) は、オプションです。詳細は、*Specifying Symbol and Model Search Paths*を参照してください。

パラメータ

その他、すべてのデータベースフィールド名は同じ名前を使い、自動的にデザインパラメータにマッピングされます。例えば、データベースのフィールドが `Tolerance` という名前であれば、デザインパラメータは、それを `Tolerance` として割り当てます。デザインパラメータの名前は、セルをクリックし、新しい名前を直接入力することで変更できます。これらのデザインパラメータの名前は、回路図シートに配置されたコンポーネントの属性を示すダイアログのパラメータのところに表示されます。

お使いのデータベースには、コンポーネントに関係するデータフィールドを非常にたくさん持たせているケースがあります。これらは回路図シートにコンポーネントが配置されるときに、デザインパラメータとして追加されます。すべての項目が必要ない場合もあります。あるいは、必要であってもこれらのほとんどの情報は、部品表を作成する場合にだけ必要となるものです。レポートマネージャでは、BOMにリンクしたデータベースのから直接、パラメータ情報を追加することができ、これによって、ソースとなる回路図ドキュメントに持たせる情報量を減らすことが可能となります。詳細は、*データベースの情報を直接BOM（部品表）に追加* のセクションを参照してください。

パラメータの更新オプション

Field Mappings タブ(前にある図 11)の残りのカラムで、データベースライブラリからコンポーネントを初めて配置したとき、あるいは、配置後、**Update Parameters From Database** コマンドを使って更新を行った際、パラメータに対する動作を指定することができます。

4 つのカラムは、次のようになります：

- **Update Values** – このカラムにあるセルは、パラメータが回路図シートとデータベースの両方に存在し、現在の値が異なっている場合、どのような動作をするか設定します。配置されたコンポーネントのパラメータをデータベースの保存されている値で更新するか、更新しないかを選びます。このオプションは、コンポーネントを配置した後、Update Parameters From Database コマンドを実行した場合に適用されます。
- **Add To Design** – このカラムにあるセルは、配置されたコンポーネントに存在しないパラメータがデータベースで見つかった場合にどのような動作をするか設定します。パラメータを追加する／追加しない、あるいは、データベースで値が割り当てられているパラメータだけ追加する設定が選択できます。このオプションは、最初にコンポーネントをデータベースから配置する場合と Update Parameters From Database コマンドを実行した場合の両方で適用されます。
- **Visible On Add** – このカラムにあるチェックボックスは、コンポーネントを初めて配置する際、あるいは、配置後に更新を行った際に、新たに追加されたパラメータを回路図シート上で表示する(enabled)か、しない(disabled)か、を設定します。
- **Remove From Design** – このカラムにあるセルはフィールドを追加するのとは反対に、配置されたコンポーネントにパラメータが存在し、データベースに無い場合、どのようなアクションを取るか指定します。パラメータを削除しない、または、データベースで値が割り当てられていない場合にだけ削除するかを選ぶことができます。このオプションは、コンポーネントを配置した後、Update Parameters From Database コマンドを実行した場合に適用されます。

図 12にあるように、マップされた各データベースフィールドの**Update Values**, **Add To Design**, **Remove From Design** オプションは`Default`に、**Visible On Add**オプションは無効になっています。

Database Field Name	Design Parameter	Update Values	Add To Design	Visible On Add	Remove From Design
ComponentLink1Description	ComponentLink1Description	Default	Default	☐	Default
ComponentLink1URL	ComponentLink1URL	Default	Default	☐	Default
ComponentLink2Description	[None]				
ComponentLink2URL	[None]				
DatasheetDocument	DatasheetDocument	Default	Default	☐	Default
Designator	Designator	Default	Default	☐	Default
Footprint	Footprint	Default	Default	☐	Default
Footprint Path	[Footprint Path]				
Footprint Ref	[Footprint Ref]				
Library Path	[Library Path]				
Library Ref	[Library Ref]				
PackageReference	PackageReference	Default	Default	☐	Default
Part Number	Part Number				
Pin Count	Pin Count	Default	Default	☐	Default

図12. パラメータのアップデートオプションの指定

図 12を見ると、アップデートオプションに関して、4 つの重要なポイントがあります：

- マッピングがされていないデータベースフィールドは、アップデートオプションがありません。
- シンボルとモデルに関係するマッピングは、デザインパラメータでは無いのでアップデートオプションはありません。
- キーフィールド (図 12の例では、Part Number) は、アップデートオプションはありません。このフィールドは、一致をさせる目的で使用されます。

- `Default` に設定することで、*Database Library Options* ダイアログ(**Tools » Options**)の **Default Actions** タブ (図 13) で指定した、デフォルト定義のアップデートが実行されます。このダイアログは、メインウィンドウの **Field Settings** にある **Options** ボタンをクリックすることでもアクセスできます。

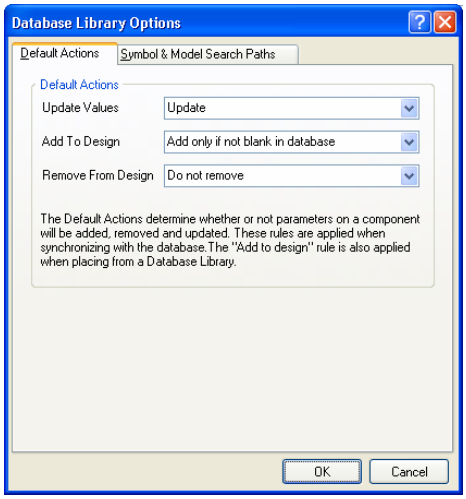


図 13. パラメータアップデートオプションのデフォルト設定

4 番目のポイントは、アップデートオプションの指定を簡単にするため、集中的に指定を行い、それから各マッピングフィールド別にアップデートオプションを定義することができます。このことから、`Default` が自動的に入力されています。

アップデートオプションのデフォルト設定を無視したい場合は、**Field Mappings** タブのアップデートを行うフィールドをクリックし、ドロップダウンリストから利用できるオプションを選びます。(図 14)

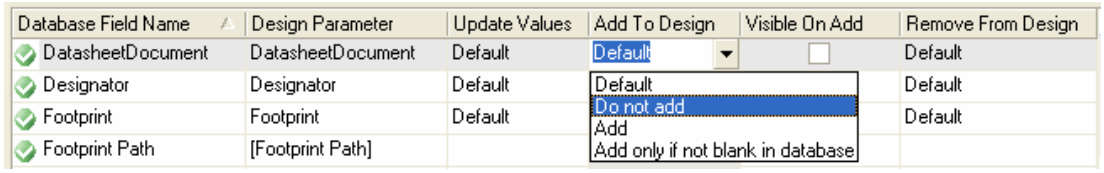


図 14. パラメータアップデートオプションをマニュアルで設定

この方法でデザインのパラメータをどのように更新するかを完全にコントロールできます。すべてのフィールドを `Default` に設定し、必要な更新の決定を *Database Library Options* ダイアログから作成することができます。各更新フィールドを個別に設定するか、または、これらの 2 つを組み合わせるなど、全体の決定を適した方法で作成できます。配置されたコンポーネントの更新は、*Database Library Options* ダイアログから実行されます。この段階で決めたくない更新がある場合は、個別には変更せず、最終的にデザインパラメータを更新する前に決定することができます。

シンボル及びモデル検索パスの指定

データベースライブラリからコンポーネントを配置するとき、そのシンボルは、`[Library Ref]`のマッピングによって指定され、指定した回路図ライブラリ(`*.SchLib`)から抽出されます。これと同じく、データベースで指定された、すべてのモデル(footprint, PCB3D, simulation)は、PCB ライブラリ(`*.PcbLib`), PCB3D ライブラリ(`*.PCB3Dlib`) 、シミュレーションモデル(`*.mdl`, `*.ckt`)に存在します。これらのファイルへのパスは、データベース内で明示的に指定できます：

- ファイルへの絶対パスを入力(例：`C:\DBLibs\Precision\SchLibs\Capacitors.SchLib`)
- ファイルへの相対パスを入力(例：`SchLibs\Capacitors.SchLib`).

データベース内でパス情報のフィールドが定義されていれば、`[Library Path]`, `[Footprint Path]`, `[PCB3D Path]`, `[Sim File]`などのフィールドは、適切なデザインパラメータにマッピングする必要があります。(前の *モデルとパラメータのマッピング*を参照してください。)

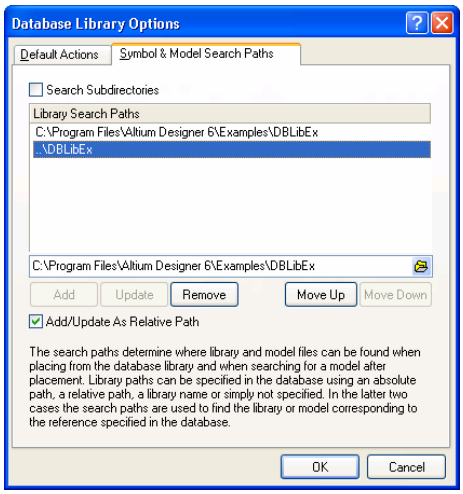


図 15. シンボルとモデルライブラリの検索パス指定

パスの入力は、相対であってもデータベーステーブル内では、若干制限があります。ライブラリやモデルファイルを移動した場合、それに従ってデータベースのテーブルを更新する必要があります。高い自由度が与えられていますが、データベースライブラリは、検索パスをデータベー

スライブラリファイルの一部として指定する機能(図 15)が供給されています。これによってデータベース内ですべてを一度に定義せずに、ソースライブラリやモデルファイルの名称だけを指定することが可能です。

図 15 で見られるライブラリ検索パスは、*Database Library Options* ダイアログ(**Tools » Options**)の **Symbol and Model Search Paths** タブで定義します。このダイアログは、メインウィンドウの **Field Settings** にある **Options** ボタンをクリックすることでもアクセスできます。

パスをリストに追加するには、フィールドにパスを直接入力するか、ボタンをクリックし、*Browse for Folder* ダイアログを表示し、必要なライブラリ／モデルファイルが保存されているディレクトリを指定します。(図 16)

必要なパスを指定した後、それを検索パスリストに追加するには、**Add** ボタンをクリックします。絶対パス、または、(DBLib ファイルへの)相対パスのどちらでも追加することができます。相対パスを使用するには、**Add/Update As Relative Path** オプションが有効になっていることを確認してください。

直接入力で間違ったパス(例えば、存在しないフォルダ)を指定した場合、追加はされますが、その検索パスが無効であることが判るように、リストではグレイアウトされて表示されます。

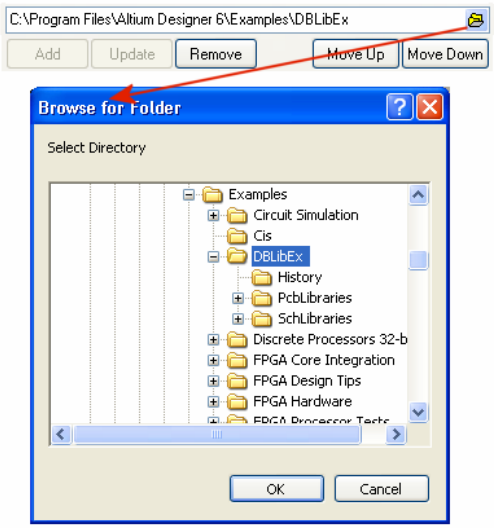


図16. 検索パスの入力

必要に応じて検索パスを追加します。入力したパスを間違えた場合は、それをリスト上で選択し、**Remove** ボタンをクリックするか、そのパスの定義を修正し、**Update** ボタンをクリックします。

ライブラリファイルの保存場所

ライブラリ検索パスは、データベースライブラリから配置を行う際、ライブラリとモデルファイルを検索する場所を指定します。指定した回路図シンボル、フットプリントモデル、PCB3D モデル、シミュレーションモデルが使用は、ライブラリ検索パスをどのように設定したか、また、データベースでライブラリ情報を指定したかによります。検索は、次のように実行されます：

シミュレーションモデル用のパス情報は、`Sim File` フィールドに入力します。モデルリファレンス情報は `Sim Model Name` フィールドに入力します。

- シンボルやモデル用にマッピングされたパスフィールドに絶対パスがあれば、そのライブラリ／モデルファイルを使用して、該当するフィールドで指定されたシンボルやモデルを抽出します。
- シンボルやモデル用にマッピングされたパスフィールドに相対パスがあれば、そのライブラリ／モデルファイルを使用して、該当するフィールドで指定されたシンボルやモデルを抽出します。
- シンボルやモデル用にマッピングされたパスフィールドにライブラリ／モデルファイル名だけある場合、検索パスは、指定した名前と該当するリファレンスフィールドで指定されたシンボル及びモデルが一致するものを含んだ最初のライブラリを検出するために使用されます。
- ライブラリ／モデル情報がデータベースに存在しない場合、検索パスは、該当するリファレンスフィールドで指定されたシンボル及びモデルが一致するものを含んだ最初のライブラリを検出するために使用されます。

最後の 2 つの場合は、検索パスでの順序が重要な役割を持っています。必要に応じ、*Database Library Options* ダイアログの **Symbol and Model Search Paths** タブの **Move Up** 及び **Move Down** ボタンを使って検索リストの順序を変更してください。

配置へのカウントダウン

データベース ライブラリファイルでマッピングとアップデートのオプションを定義し、DBLib ファイルを保存すれば、準備は完了です。これで配置が可能です。

データベース ライブラリの追加

他のライブラリと同じ様に、データベースライブラリは、**Libraries** パネルから利用可能なライブラリのリストとして追加することができます。*Available Libraries* ダイアログにアクセスするには、**Libraries** パネルから **Libraries** ボタンをクリックします。

データベースライブラリは、プロジェクトライブラリとして追加するができます。あるいは、アクティブなプロジェクトだけではなく、すべてのプロジェクトで利用できるように導入済み（Installed）ライブラリとして追加することができます。更には、DBLibファイルが保存されているフォルダを検索パスとして指定することも可能です。図 17では、データベースライブラリがライブラリのリストに追加された状態を示しています。

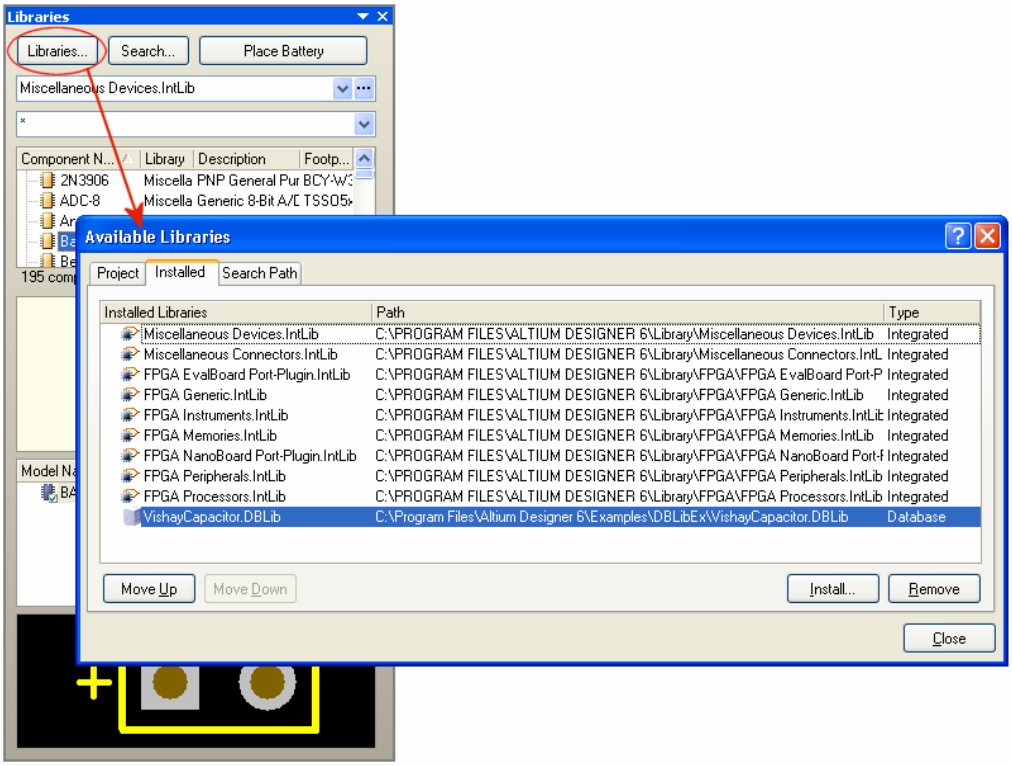


図 17. データベースライブラリを使用可能に

データベースコンポーネントの参照

Available Libraries リストにデータベースライブラリが追加されると、**Libraries** パネルでデータベースにあるコンポーネントが参照可能となります。一つの DBLib ファイルが追加されただけですが、リンクされたデータベースの各テーブルは、別のライブラリのように表示されます。

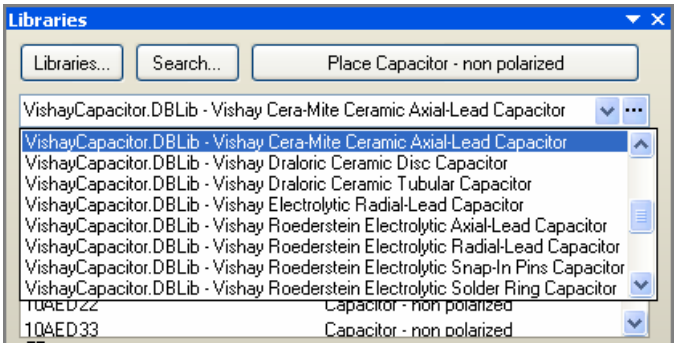


図 18. DBLib が読み込まれた状態

データベースが Excel ファイル(`*.xls`) で作成されている場合、ODBC ドライバの制限により 64 シートまで接続が可能です。

上側のドロップダウンリストは、次のような形式で表示されます：

LibraryName`.DBLib - `*TableName*

パネルでの各コンポーネントの入力はデータベースの特定のテーブルのレコードに対応しています。実際、読み込まれたデータベースライブラリを参照すると、**Libraries** パネルは、ダイレクトなデータベースブラウザのように動作します。

シンボルとモデル情報はデータベースの関連フィールド(及び検索パスで指定したもの)で指定されたシンボルライブラリとモデルライブラリから集められます。

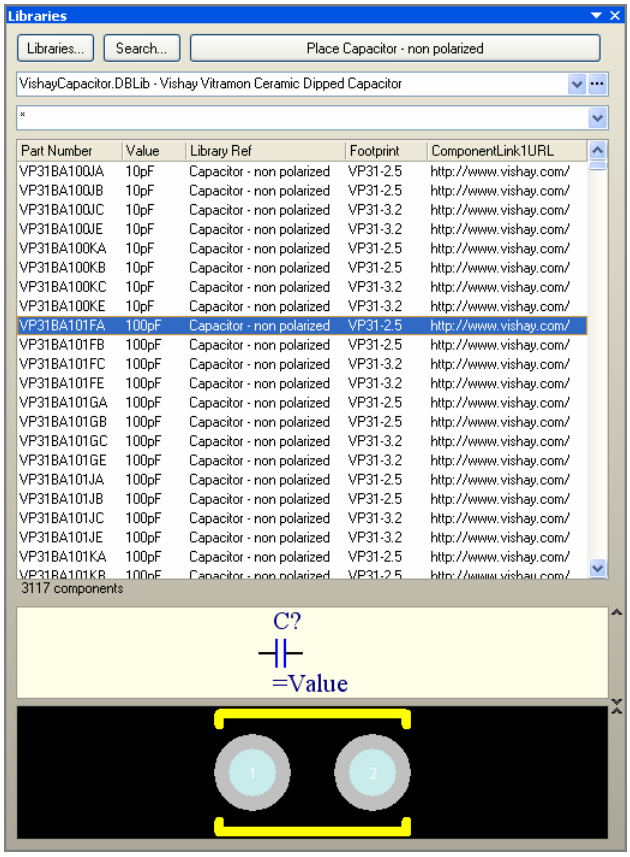


図 19. データベースコンポーネントの参照

デフォルトでは、Part Number と Library Reference フィールドがパネルのコンポーネントのリストセクションに表示されています。データベーステーブルのその他のテーブルを表示するには、コンポーネントリスト上で右クリックを実行し、**Select Columns** を実行します。これによって、*Select Parameter Columns* ダイアログが表示され、そのテーブルあるフィールドを追加することができます。

Libraries パネルの詳細については、カーソルをパネルの上に移動し、**F1** キーを押してください。

コンポーネントの検索

接続した外部データベースには複数のテーブルが含まれ、それぞれ、いくつかのコンポーネントレコードを持っています。データベースから直接、配置できるのは、ひとつのことですが、配置したいコンポーネントを特定するというのは、まったく別のことになります。後者は、**Libraries** パネルの検索機能から効率的に管理ができます。

パネル上部にある **Search** ボタンをクリックすると、*Libraries Search* ダイアログ(図 20)が表示され、ここからインストールされている DBLib ファイルのひとつのテーブルから、あるいは、すべてのテーブルからデータベースコンポーネントの検索を指示することができます。

データベースコンポーネントの検索を有効にするには、**Search Type** フィールドを `Database Components` に設定します。データベースライブラリファイルの検索用の追加オプションが現れます。データベースライブラリの検索時に、これらのオプションは関連しないので、ダイアログの **Scope** 及び **Path** の部分はグレイアウトになります。

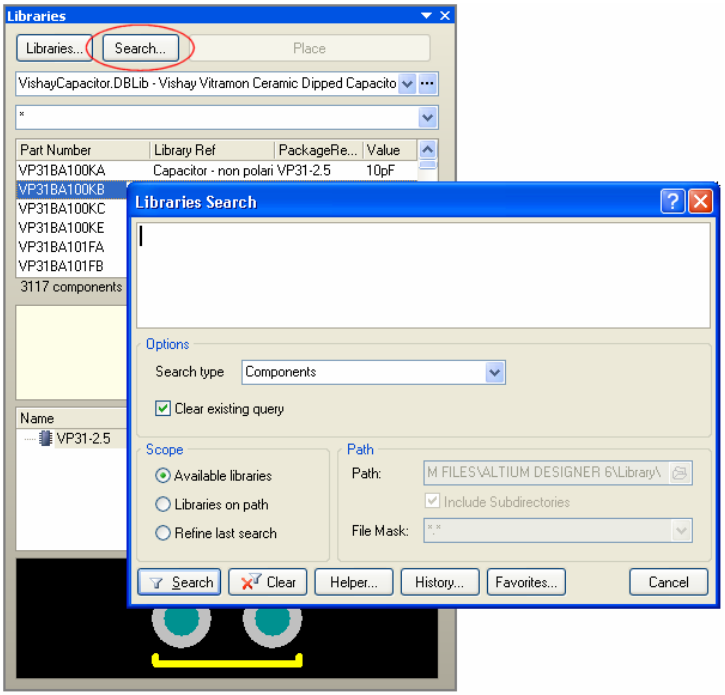


図 20. ライブラリ検索機能へアクセス

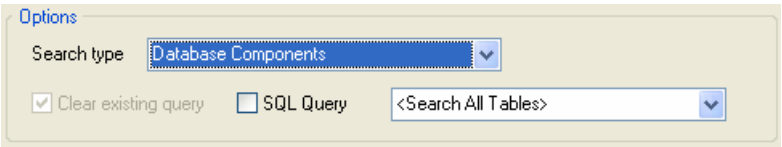


図 21. データベースコンポーネント – 検索オプション

データベースライブラリを検索するとき、シンプル、Altium Designer ベースのクエリ、SQL ベースのクエリという 3 つのレベルの検索機能が用意されています。

シンプルな検索

これは「粗い」検索で、指定したテーブル、あるいは、すべてのテーブルで検索されたすべてのデータベースコンポーネントを返します。必要とするコンポーネントがどこにあるか判っている場合は、ドロップダウンリスト(これは、テーブルスコープとしても利用されます)から特定のテーブルを選択します。あるいは、スコープを `<Search All Tables>` のままにしておきます。

参照モードオプションで **Components** が有効な場合、検索結果は **Libraries** パネルでのみ表示されます。

Search ボタンをクリックすると検索が開始されます。*Libraries Search* ダイアログはクローズされ、検索結果が **Libraries** パネルに表示されます。検索が一つの特定のテーブルのとき、ライブラリのドロップダウンリストには、`Query Results` という項目が表示されます。すべてのテーブルを検索した場合は、各テーブルが別々に表示されます。ドロップダウンリストでの各エントリは、次のような形式で表示されます：

```
Query Results - TableName (n)
```

この `(n)` は、検索されたコンポーネントの数を示しています。

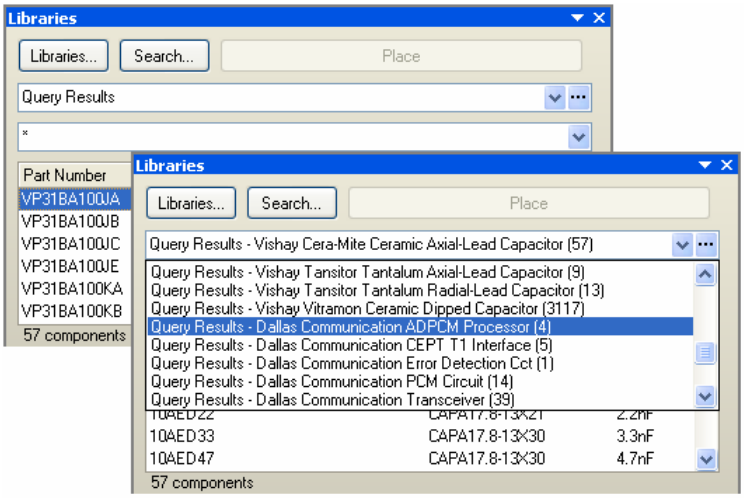


図 22. 検索結果

Altium Designerクエリ検索

これは「細かい」検索で、Altium Designer のクエリで指定したスコープに相当するすべてのデータベースコンポーネントを返します。クエリは、特定のテーブル、あるいは、全テーブルに対して適用させることができます。クエリを使うことで、データベースコンポーネントの検索を、特定のパッケージやピン数で検索範囲を絞り込むことができ、検索によって返される結果項目を最小限にできます。

Libraries Search ダイアログの上側(クエリエディタ セクション) でクエリを入力することで、フィルタを作成することができます。このフィールドにクエリを直接入力することができます。 図 23 に、すべてのテーブルで、指定したクエリに一致したコンポーネントを返す検索例をサンプルとして示しています。このケースは、Package Reference フィールドに `VP41` が含まれ、その値は `6.8nF` のすべてのコンポーネントです。

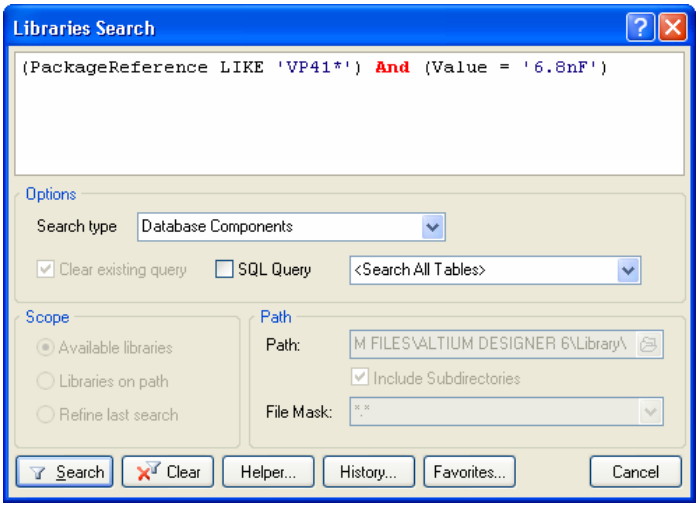


図 23. Altium Designer クエリの例.

クエリを定義し、必要に応じてテーブルスコープを設定した後、**Search** ボタンをクリックすると検索が始まります。結果は、**Libraries** パネルに読み込まれます。全テーブルを検索した場合、検索結果は、一致したものが検索された各テーブル別で表示されます。

クエリの作成をアシストする Query Helper の機能が用意されています。この機能は、クエリの構文や使用したいキーワードがあるかどうか不明な場合にとても便利です。Query Helper を使用するには、*Libraries Search* ダイアログの **Helper** ボタンをクリックします。*Query Helper* ダイアログが表示されます。

ダイアログの上側で、Library Functions 及び System Functions を使ってクエリを作成します。Library Functions には、2 つのサブリストが用意されています：

- **Fields** – 導入されている全データベースライブラリの全テーブルにあるユニークな全フィールド
- **Tables** – 導入されている全データベースライブラリの全テーブル名

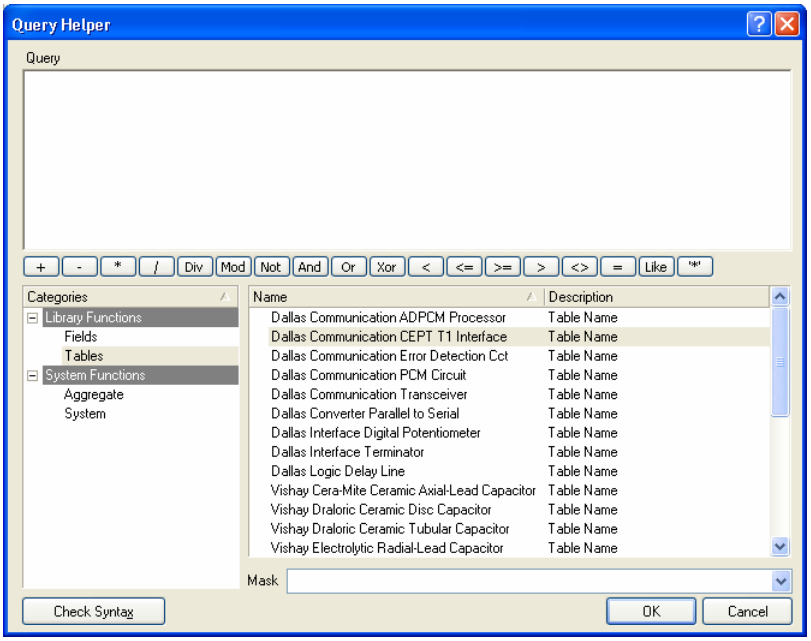


図 24. クエリを作成するために使用するクエリヘルパー

ダイアログの中ほどのセクションでは、式を作成するために使用する演算子が用意されています。

Altium Designerのクエリ言語についての概念については、アーティクル *Introduction to the Query Language* を参照してください。

クエリ言語の使用方法についての詳細は、アーティクル*An Insider's Guide to the Query Language*を参照してください。

特定のキーワードについての情報は見るには、*Query Helper* ダイアログでキーワードを選択し、**F1** キーを押すか、入力フィールドでキーワードを選択し、**F1** キーを押します。

SQL クエリ検索

このレベルの検索も「細かい」検索で、指定したクエリに基づくものですが、Altium Designer のクエリ言語の代わりに SQL クエリを直接入力することができます。この方法を使用する場合、一つの、特定のテーブルだけを指定するクエリがサポートされています。

SQL について精通されている方は、**SQL Query** オプションを有効にして、*Libraries Search* ダイアログのクエリエディタのセクションに直接入力してください。

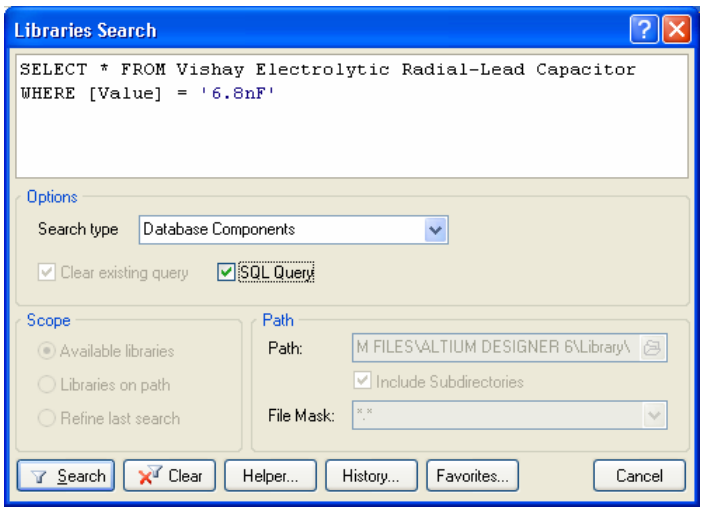


図 25. SQL 検索クエリを直接指定

あるいは、Altium Designerのクエリ言語を使って表現を定義し、**SQL Query**オプションを有効にすることで、クエリをSQLに変換することができます(図 26)。最も良い変換結果を得るには、**SQL Query**オプションを有効にする前に、テーブルスコープが特定のテーブルに設定されていることを確認してください。

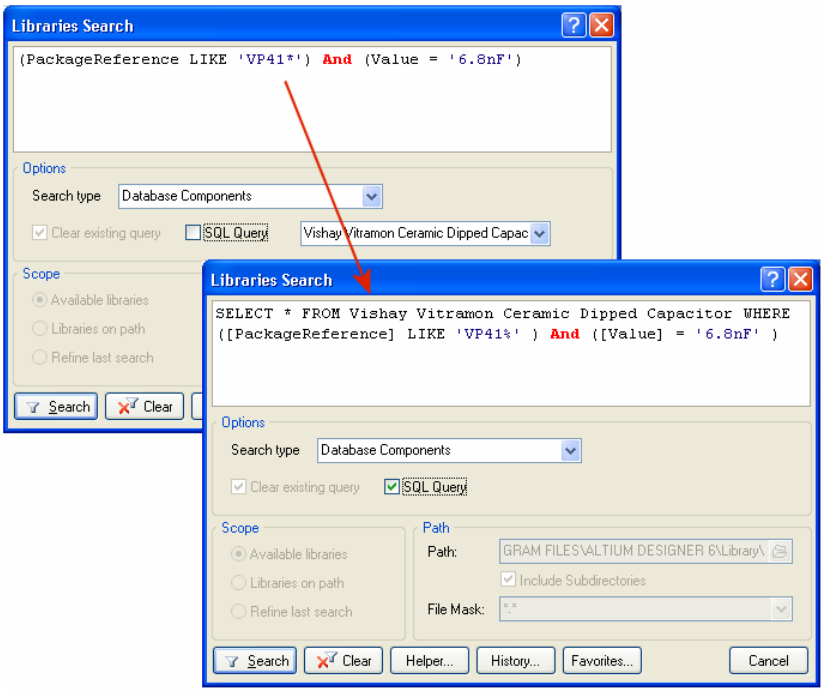


図 26. Altium Designer クエリを SQL クエリに変換.

コンポーネントの配置

いよいよリンクしたデータベースから直接コンポーネントを配置する時がきました。目的とする回路図シートをオープンし、メインのデザインウィンドウでドキュメントをアクティブにします。配置したいコンポーネントを **Libraries** パネルで選択し、**Place** ボタンをクリックするか、コンポーネントをシート上に直接ドラッグ＆ドロップします。この時、次の動作が実行されます：

- データベースレコードで指定された回路図シンボルを呼び出します。
- すべてのフットプリント、PCB3D、シミュレーションモデルがリンクされます。
- DBLib ファイルで指定されたデザインパラメータが(関連する **Add To Design** の設定に従って)コンポーネントに追加されます。

配置後、回路図上のコンポーネントをダブルクリックすると、プロパティダイアログ (図 27) が表示されます。ダイアログの **Parameter** には、追加されたパラメータが、リンクされたモデルは **Models** のエリアで確認できます。また、別の **Library Link – Database Component** のところでは、次の情報が表示されています：

- 親データベース ライブラリファイル
- コンポーネントが保存されていたデータベーステーブル
- Physical Component（実コンポーネント）の値。これは、DBLib ファイルの Field Settings で定義したキーフィールドの値で、通常、パートナンバーです。
- コンポーネントの Logical Symbol（論理シンボル）これはコンポーネントで指定した回路図シンボルです。

配置したコンポーネントは、**Choose** ボタンをクリックし、同じテーブルから別のものに変更することができます。これによって、*Browse Libraries* ダイアログがオープンし、まず、同じテーブルのすべてのコンポーネントが表示されます。実際には同じ DBLib ファイルの別テーブルのコンポーネントや、別の DBLib ファイルを参照して変更することもできます。この領域は、選択したデータベースコンポーネントの新しい情報によって更新されます。

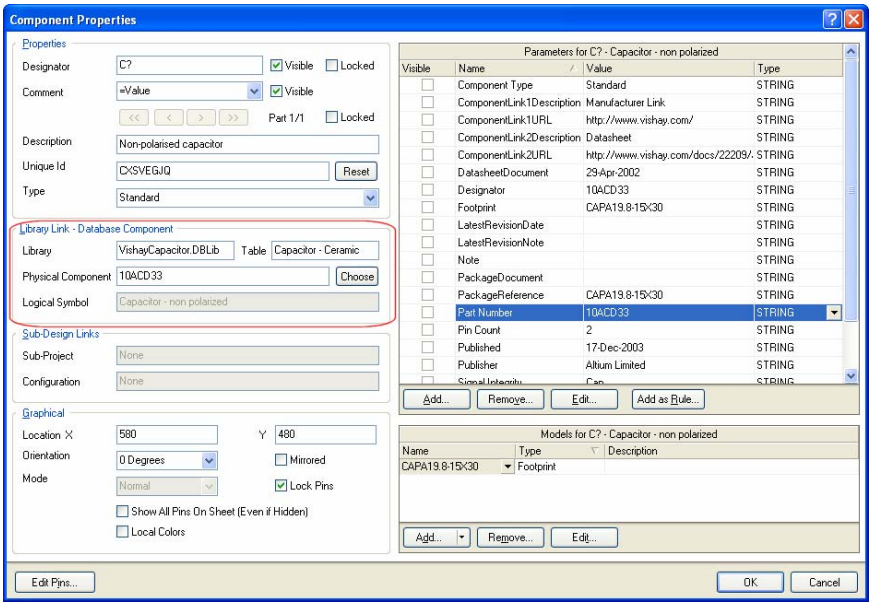


図 27. 配置によってコンポーネントに追加された情報

確実な同期

配置後、選択されたキーフィールドパラメータは、回路図に配置されたコンポーネントとそれに対応する外部データベースのコンポーネントとのリンクを取るために使用されます。これは将来的にデータベース内のパラメータ情報が変更された場合に配置されたコンポーネントに簡単に戻すことができ、どの段階でも両者の同期が取れること意味します。

単にパラメータ情報を更新する場合は、回路図エディタの **Tools** メニューにある **Update Parameters From Database** コマンドを使用します。

回路図シンボルのパラメータ、モデル及びグラフィカルシンボルのすべてを更新するには、**Update From Libraries** コマンド (これも回路図エディタの **Tools** メニューにあります。) を使用します。

配置されたフットプリントをソースライブラリに保存されている最新情報を元に更新するには、PCB エディタで、**Tools » Update From PCB Libraries** コマンドを使用します。

アップデートツール使用する場合の詳細情報は、アプリケーションノート*Keeping Components Up-To-Date* を参照してください。

DBLibとDBLinkとのデュアル シンクロナイズ

既存のデザインプロジェクトがある場合、DBLink ファイルを使って配置されている部品と外部のデータベースをリンクしている方がほとんどだと思います。デザインの変更した結果、回路が追加され、データベースライブラリ(DBLib)の機能を使って配置されたコンポーネントが使用されているかもしれません。この DBLib ファイルは、まったく別の外部データベースであることが考えられます。

Update Parameters From Database コマンドを使用するとき、配置されているコンポーネントのリンクされている全パラメータは、リンクの方法に関係なく、リンクされているすべてのデータベースを検索し、これらのパラメータの違いを検出し、*Select Parameter Changes* ダイアログに表示します。

DBLink と DBLib ファイルの両方で同じデータベースフィールドがマッチングで使用されている場合は、DBLink ファイルによってリンクされたデータベースリンクを最初に検索し、次に DBLib ファイルによってリンクされているデータベースを検索します。コンポーネントが両方のデータベースに存在する場合は、間違った外部レコードから一致と更新を行う可能性があります。

データベースの情報を直接部品表（BOM）に追加

部品表(BOM)用のソース情報は、先にデザインに配置されたコンポーネントのパラメータ情報を元にします。それは多くの情報を回路図に付属することができますが、それは BOM にだけ使用されます。データベースライブラリからコンポーネントを配置した場合、部品表作成機能は、配置の時点でデザインパラメータとして追加されていない、その他のレコード情報を抽出することができます。

Report Manager を使って部品表の設定するときに、**Include Parameters from Database** オプションを有効にします。このオプションは、デザイン内に外部データベースから配置されたコンポーネントがある場合にだけ表示されます。パラメータリストで、アイコンは、外部データベースへリンクされたコンポーネントのパラメータを区別するために使われます。

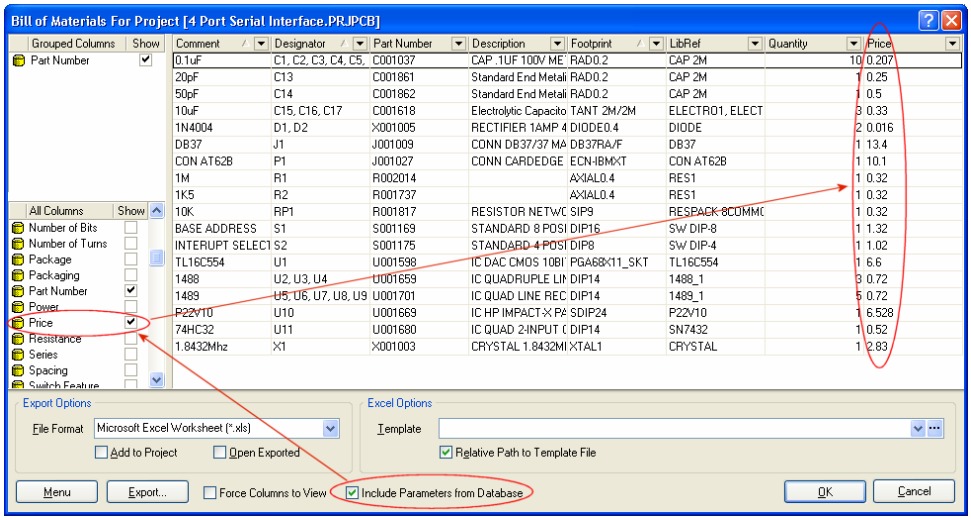


図 28. 外部データベースにだけ存在する追加のコンポーネント情報を含める

更新履歷

Date	Version No.	Revision
4-Dec-2005	1.0	Initial release
23-Jan-2006	1.1	Added information on Integrated Library to Database Library Translation Wizard.
10-Apr-2006	1.2	Added information on linked simulation model information and a section on OrCAD CIS support.
16-Jun-2006	2.0	Updated for Altium Designer 6.3. Information on Integrated Library to Database Library Translation Wizard, and OrCAD CIS support moved to new document – AP0143 Database Library Migration Tools. Information regarding Updating Parameters From Database moved to new document – AP0144 Keeping Components Up-To-Date. Links added to the application note: AP0145 Working with Version-Controlled Database Libraries.
12-Feb-2007	2.1	Information added on 64-sheet limit when using a DBLib and linking to an Excel target database.
24-Aug-2007	2.2	Updated Quote character for my SQL to ` (reverse apostrophe) instead of ‘.
11-Dec-2007	2.3	Information added for Design Parameter, [Description].
21-Dec-2007	2.4	PCB3D legacy reference added. Cross-references to figures added.
18-Apr-2008	2.5	Updated PageSize to A4.