

概要

Article
AR0104 (v2.3) April 21, 2008

このアーティクルでは、コンポーネント、モデル、ライブラリと、それらの関係について説明します。コンポーネント - ライブラリ間の関係を定義し、管理するための手法が解説されています。また、モデルの検索順序に関する説明や、より効果的に検索を行うオプションについても説明しています。

コンポーネントは、エレクトロニクス製品の基本ブロックです。デザインキャプチャとインプリメンテーションのプロセスでは、コンポーネントはいろんな風に表現できます。回路図では論理的なシンボル、PCB ではフットプリント、シミュレーション用には SPICE 定義、シグナルインテグリティ解析には適切な解析用の記述、3D コンポーネントや PCB ビジュアライゼーションには 3 次元表示用の記述が使用されます。これらの表現すべてがコンポーネントに必要であるとは限りませんが、少なくとも、コンポーネントが回路図に配置される前に要求される、最小限度の記述というものが存在します。最初のコンポーネントが作成されるときに何ができるか、また、プロパティやパラメータはどのように追加されるのか、さまざまな設計プロセスのパーツとしてモデルはどのようにコンポーネントをインプリメントするのか、について説明します。

コンポーネント (部品) - 基本のビルディングブロック

”コンポーネント (部品)”は、エレクトロニクスのデザイン (回路図、PCB レイアウト) 上に配置される構成要素に与えられる一般的な名前です。デザインキャプチャのフェーズごとに、コンポーネントはいくつもの記述を持つ場合があります。したがって、インプリメンテーションの進展に依存して、異なる状況から名前を参照されることも珍しくありません。コンポーネントの定義には、柔軟性が要求されるので、ある設計フェーズから別の設計フェーズへと、そのすべての情報とリンクは容易に言い替えたり、振り替えたりできなくてはなりません。

なぜなら、回路図のキャプチャに始まるエレクトロニクス設計において、どんな回路図に配置するにせよ、コンポーネントには常に最低限の要求があるからです。少なくとも、回路図ライブラリ上でコンポーネントの名前を省くわけには行きません。それは単一かマルチパートの形式で、ピンやグラフィックシンボルを備えており、別の表示オプションを持っていることもあります。

いったん回路図上に配置され、グラフィック表示が割り当てられると、コンポーネントはより一般的な”シンボル”という名前と呼ばれるようになります。そこでは、グラフィカルに表示されているので、シンボルには物理的な形状を定義する描画オブジェクトと、電気的な接続ポイントや”論理”を定義するピンが含まれることになります。回路図設計の際に、”論理シンボル”と称されるシンボルのことはご存知かも知れません。

この最初の、ごく単純な定義は、コンポーネントをより簡単、より柔軟に言い替えることで、非常に複雑なエレクトロニクスの構成を表現しています。あるコンポーネント、たとえば、レジスタネットワークやリレーなどは、回路図上に別々に配置できる一連のパーツとして描くことができます。これらは、”マルチパートコンポーネント”として参照され、各ピースは単純に”パーツ”と呼ばれます。

別な言葉としては”ドメイン”があります。これは、特別なタイプ、グループ、エリアを表します。Altium Designer の環境の有効なドメインには、PCB レイアウト、SPICE シミュレーション、シグナルインテグリティ解析、そして 3D が含まれます。

すでに述べたとおり、状況によって区別されるいくつかの用語が、PCB のレイアウトや設計において使われています。”フットプリント”は、PCB レイアウト上のコンポーネントを表すモデルのことです。したがって、フットプリントは、ボードレイアウト上でコンポーネントを配置したり、接続したりするのに必要なスペースを定義するコンポーネントの形状や、PCB のパッド一式がグループ化されたものを表します。いったん PCB 上に配置されたコンポーネントは、以後、”物理コンポーネント”とみなされます。

ついでですが、ほとんどの場合、論理シンボルもまた物理コンポーネントであると言えます。この場合、それぞれの関係性もほぼ同様であると言えるでしょう。しかし、特定の例外もあります。それは、コンポーネントがデータベースライブラリから配置された場合で、外部データベース内の記録が物理コンポーネント(またはシンボル、モデル)を表します。

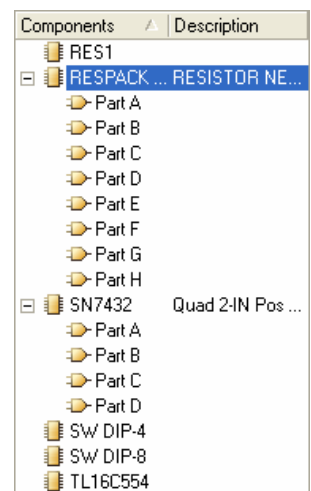


図 1 コンポーネントは一つ以上のパーツを持つことができます。

Component, Model and Library Concepts

さて、ここから、コンポーネントがどのように定義され、記述されるのか、異なるタイプの表現はありえるのか、そして、より特殊化されたタイプはどのようにサポートされているのか、もう少し詳しく見ていくことにしましょう。

コンポーネントの属性

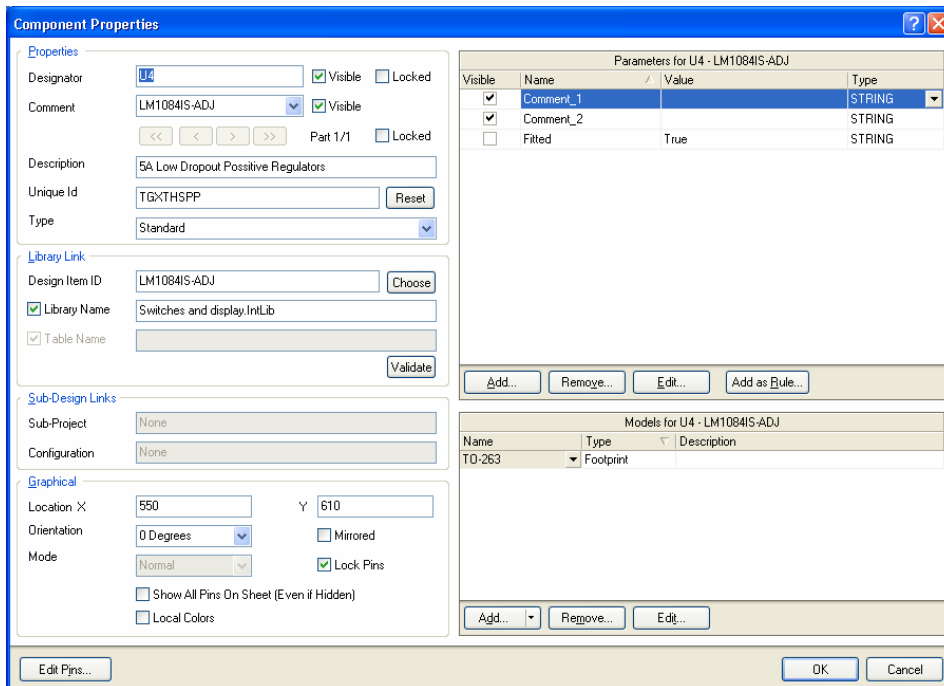


図2 配置されたコンポーネントをダブルクリックすると、Component Properties のダイアログが表示されます。ここで、各コンポーネントについての特定の属性、ライブラリへのリンク、グラフィック表示、検索場所などを定義できます。

Component Properties ダイアログは使用しているデザインエディタ(回路図、PCB、その他)によって、表示が少し異なります。

コンポーネントのタイプ

コンポーネントを作るには、異なるいくつかの方法があります。標準コンポーネントと非標準コンポーネントのタイプ、また、マルチパートのコンポーネントについてみてみましょう。

同じグラフィックで異なるコンポーネント

物理コンポーネントのそれぞれに対して、単一のコンポーネントシンボルがある場合

このタイプの表わしかたは、論理シンボルが集積回路のような物理コンポーネントと同じである場合、どんなコンポーネントにとっても理想的です。コンポーネントは特別な表現、または PCB フットプリント、シミュレーション、3D モデリング情報などのモデル(詳細は以下)を含んでいます。

グラフィック的に等価なコンポーネントに対する単一のシンボル

論理的には等価であるが、やや異なった仕様になっているコンポーネントがあります。例としては、さまざまな論理ファミリ(74ACT32 や 74HC32)に使用可能な論理ゲートがあります。この場合、シンボルはいったん描かれ、その後、それぞれの等価コンポーネントの必要に応じて別の名前や“エイリアス”が定義されます。コンポーネントのエイリアスは Schematic Library のエディタパネルに追加されます。コンポーネントエイリアスは名前の違う種類のコンポーネントであることもできます。

あるタイプのコンポーネントに対して単一のシンボル

設計の物理レイアウトの要求に従って、特定のコンポーネントが複数の PCB フットプリントにリンクされる必要があるとき、PCB コンポーネントにはコンポーネントタイプに関連していくつかの特別な考慮が求められます。例としては、回路図上に配置されるときに定義された値属性を持つ、レジスタなどのディスクリートコンポーネントが挙げられます。

同じコンポーネントで異なるグラフィック

Altium Designer は同じコンポーネントに対する複数のシンボルをサポートしています。たとえば、伝統的な描画形状を使って、シンボルを描くように求める顧客がいる一方で、別の顧客は標準仕様に従ったシンボルを求めたりすることがあります。ある

Altium Designer のコンポーネントにとって、シンボルについての最小限の定義とリンクが説明の出発点になります。これにより、後ほど、設計やキャプチャの別プロセスで、違った表現でコンポーネントを説明するのに大幅な柔軟性が生まれます。

Altium Designer は、ユーザの求める設計仕様に応じて、コンポーネントを構築するための異なるアプローチをサポートしています。

様々な属性、リンク、記述が、どんなコンポーネントにも容易に定義でき、そのためのユニークな“プロパティ(属性)”を Component Properties ダイアログから作成できます。これによって、ユーザは正確に表現タイプを作成でき、設計のあらゆる段階で必要なインプリメンテーションができるようになります。

いは、多くの異なるシンボルが同じコンポーネントのために要求されるかもしれません。新しい“モード”の追加によって、あるいは、回路図ライブラリエディタの **Tool** メニューから、または、**Mode** ツールバーの使用によって、コンポーネントとともに格納されているシンボル用として、追加のグラフィック表現を定義することができます。

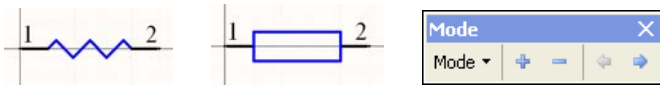


図3 Mode 機能を使って、複数のグラフィック表現を同じコンポーネントに定義します。最初のモードは、“Normal”と呼ばれます。それに続くモードには“Alternate 1”が付けられています。作成されるどのモードにも、コンポーネントが自動的に格納されます。

マルチパートコンポーネント

いくつかのインスタンスでは、ある物理コンポーネントを表現するのに複数のシンボルを用いるのが適切です。たとえば、レジスタネットワーク中の各レジスタとか、コイルやリレーの接点などの場合です。

追加パーツは、ライブラリエディタ内の **Tools** メニューで追加または削除されます。各パーツはそれぞれに描画され、対応したピンが追加されます。

標準でないコンポーネント形式

すべてのコンポーネントが PCB に実装されたり、部品表(BOM)で必要になったりするわけではありません。また、PCB に実装されるすべてのアイテムが回路図上に表される必要があるわけではありません。Altium Designer は標準でないコンポーネントタイプを (ライブラリエディタまたは回路図エディタの) **Component Properties** ダイアログにある **Component Type** 属性の設定でサポートしています。

例えば、PCB を接続するようなシャーシに組み込む部品を回路図に含めることで回路図がわかりやすくなります。このコンポーネントが PCB の BOM で必要なければ、コンポーネントタイプを **Graphical** に設定することができます。グラフィックコンポーネントは、回路図の電気的な検証や BOM には含まれず、回路図を PCB に同期させる間もチェックされません。この場合、**Component Type** を **Graphical** に設定します。

コンポーネントの別の特別な種類としては、テストポイントがあります。このコンポーネントは、回路図と PCB の両方で必要となりデザインの同期中にチェックされますが、BOM には含まれません。この場合、**Component Type** を **Standard (No BOM)** に設定します。

特別なコンポーネントの種類別の例として、ヒートシンクがあります。一般的に、それは回路図には表示されず、回路図の電気的な検証でチェックする必要はありませんが、BOM に含める必要があります。この場合、**Component Type** を **Mechanical** に設定します。

コンポーネントパラメータ

パラメータは定義方法を提供し、コンポーネントにいろいろなテキスト情報を結びつけています。これには、コンポーネントの電気的な詳細情報 (例えば、ワット数または許容差)、コンポーネントの購入先、在庫状況、設計者による注記、コンポーネントのデータシートの参照情報などを含めることができます。この情報はパラメータの追加によってコンポーネントに含められます。つまり、ライブラリエディタでコンポーネントを作成しているときや、一度コンポーネントが回路図に配置されたとき(DBLink ファイルを使用)、またはデータベースライブラリ(DBLib または SYNDBLib)から配置しているとき自動的に含められます。

特定のコンポーネントにパラメータを追加

Parameters for U5 - M8_B2B1			
Visible	Name	Value	Type
<input type="checkbox"/>	ArchiveName	GENERIC	STRING
<input type="checkbox"/>	Core Revision	1.00.00	STRING
<input checked="" type="checkbox"/>	Published	18-Nov-2003	STRING
<input type="checkbox"/>	Publisher	Altium Limited	STRING

図5 名前や値をコンポーネントのパラメータとして定義することができます。また、**Component Properties** からワークスペースにパラメータ情報がどのように現れるかについてのグラフィック属性を定義することができます。

パラメータセクション内に定義されたどんなパラメータでも、**Annotate** ダイアログ上の **Match By Parameters** 領域に作るすることができます。これは、定義済みで詳細情報込みのユニークなパラメータを使って、マルチパートコンポーネント用の特別なパーツ群として、後でまとめた場合に便利です。

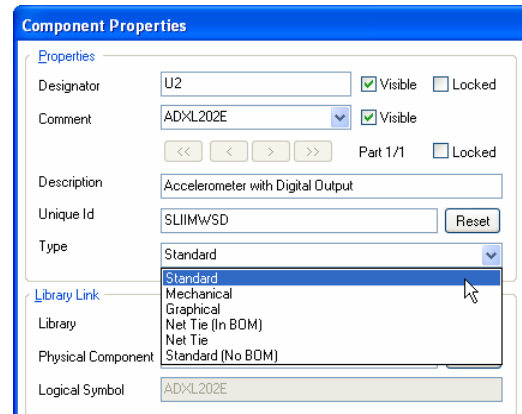


図4 特別なコンポーネントは必要に応じてコンポーネントタイプを設定。ここでは、マルチパート用に、どのパーツを今、見ているのかについても確認できるように注意してください。

グラフィックコンポーネントは、回路図の電気的な検証や BOM には含まれず、回路図を PCB に同期させる間もチェックされません。この場合、**Component Type** を **Graphical** に設定します。

コンポーネントの別の特別な種類としては、テストポイントがあります。このコンポーネントは、回路図と PCB の両方で必要となりデザインの同期中にチェックされますが、BOM には含まれません。この場合、**Component Type** を **Standard (No BOM)** に設定します。

特別なコンポーネントの種類別の例として、ヒートシンクがあります。一般的に、それは回路図には表示されず、回路図の電気的な検証でチェックする必要はありませんが、BOM に含める必要があります。この場合、**Component Type** を **Mechanical** に設定します。

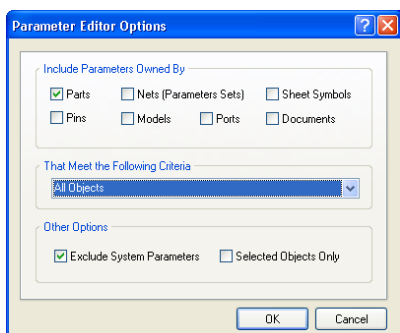
コンポーネントパラメータ

パラメータは定義方法を提供し、コンポーネントにいろいろなテキスト情報を結びつけています。これには、コンポーネントの電気的な詳細情報 (例えば、ワット数または許容差)、コンポーネントの購入先、在庫状況、設計者による注記、コンポーネントのデータシートの参照情報などを含めることができます。この情報はパラメータの追加によってコンポーネントに含められます。つまり、ライブラリエディタでコンポーネントを作成しているときや、一度コンポーネントが回路図に配置されたとき(DBLink ファイルを使用)、またはデータベースライブラリ(DBLib または SYNDBLib)から配置しているとき自動的に含められます。

特定のコンポーネントにパラメータを追加

Component Properties ダイアログを使うと、特定のコンポーネントへのパラメータの追加が容易に行えます。

コンポーネントライブラリにパラメータを追加



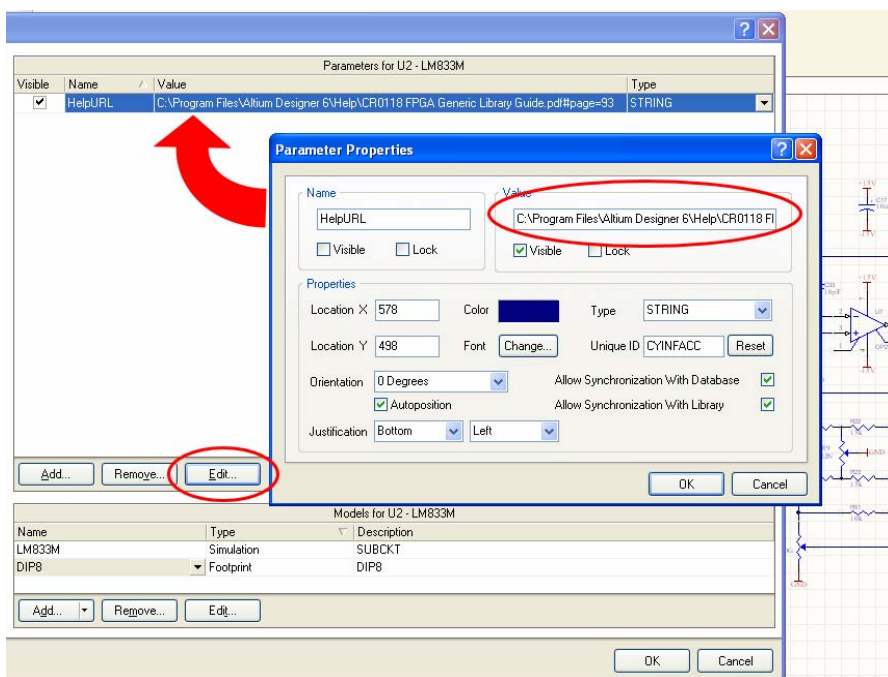
上記の例は、Component Properties ダイアログを使ったマニュアル操作による、各ライブラリコンポーネントへのパラメータの追加を示しています。コンポーネントライブラリにパラメータを追加するには、もっと効率的なアプローチも追求できます。このようなインスタンスでは、Parameter Manager ダイアログのほうがよいオプション

図6 回路図ライブラリを開いて、Parameter Editor ダイアログ (Tool » Parameter Manager) に進みます。複数コンポーネントのパラメータ値を効果的に設定することができます。

パラメータとしてデータシートを参照

コンポーネントのデータシートを使った設計プロジェクトから、参照マテリアルへのアクセスが必要になるかもしれません。Altium Designer は二つのオプションを提供することで、回路図シート上のコンポーネントから、コンポーネントパラメータが付加されている参照データシートへのリンクに対応しています。最初のオプションでは、F1 ボタンを使って、特定の参照ドキュメントにアクセスできます。二つ目は、複数参照のためのオプションです。右クリックを使ってメニューを開いてください。

1つのドキュメントへのリンク - F1 キーによるアクセス



あるコンポーネントが、HelpURL のシステム予約名を使ったパラメータを含んでいる場合、カーソルがコンポーネント上か、ライブラリパネルのエントリ上にあるときに F1 キーを押すと、目的の URL にアクセスできます。URL には実際のウェブアドレス、テキストファイル、PDF ファイルが指定できます。

パラメータの値として、参照するドキュメントだけでなく、PDF のページ番号も指定することができます。

図7 ここで、Component Properties ダイアログから、HelpURL パラメータが回路図シンボルに追加されています。カーソルがコンポーネント上にあるときに F1 キーを押すと、与えられた値、" \Help\CR0118 FPGA Generic Library Guide.pdf#page=93 " が参照先 PDF ファイルの 93 ページを開きます。

リンクされた複数のドキュメント - 右クリックによるアクセス

この二つ目の機能により、右クリックで表示されるメニューから、ペアになったパラメータと ComponentLinkURL のシステム予約名を使って、一つ以上の参照先ドキュメントへのマルチリンクの定義とサポートができるようになります。

	パラメータ名	パラメータ値の例
1 st パラメータ	ComponentLink1URL	C:\MyDatasheets\XYZDdatasheet.pdf
2 nd パラメータ	ComponentLink1Description	Datasheet for XYZ
1 st パラメータ	ComponentLink2URL	C:\MyDatasheets\AlternateXYZDdatasheet.pdf
2 nd パラメータ	ComponentLink2Description	Datasheet for Alternate XYZ

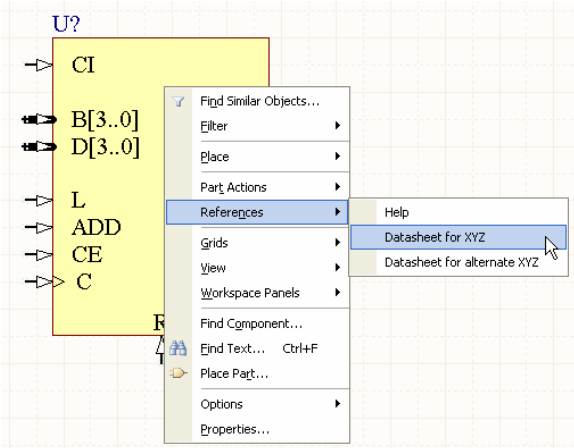


図8 配されたシンボル上で右クリックすると、データシートのリンクにアクセスします。

同じパラメータのペアを使って、リンクはいくつでも定義できます。ただし、インクリメンタルな指定はできません。データシートリンクを使ったコンポーネント上で右クリックすると、**Reference** メニューのエントリが表示されます。メニューから各コンポーネントリンク用のエントリを見つけてください。

Libraries パネルでコンポーネントを閲覧するときにも、コンポーネントとデータシートのリンクが使用できます。 - F1 キーまたはコンポーネント名の上で右クリックすると、リンク先ドキュメント/URL にアクセスします。

コンポーネントパラメータの他のタイプを追加する場合の詳細は、[TU0103 ライブラリコンポーネントの作成](#)を参照してください。

モデル - コンポーネントの特別な表示方

先に述べたとおり、ドメインは、Altium Designer における設計プロセスの一部であるコンポーネント表現のタイプ、グループ、エリアとみなすこともできます。したがって、“モデル”は、特定ドメイン用に便利なコンポーネントをインプリメンテーションする際の表わし方ということができます。これは、PCB ではフットプリント、シミュレーションでは SPICE 定義、シグナルインテグリティ解析では適切な解析用の記述、旧式の 3D ビューアや DirectX ベースの 3D エンジンにおける 3D 視覚化では 3 次元モデル、と呼びかえることができるかもしれません。コンポーネントには、単独で回路図に配置するために添付されるモデルは要求されません。しかし、モデルが無いと、どんなドメインにもインプリメントできません。

Model Libraries は、コンポーネント表現の集合体です。これについては、以下のライブラリのセクションでさらに詳しく説明します。モデルマッピング情報がコンポーネントと共にどのように格納されていくか、その基本について説明しましょう。

モデルマッピング情報の基本

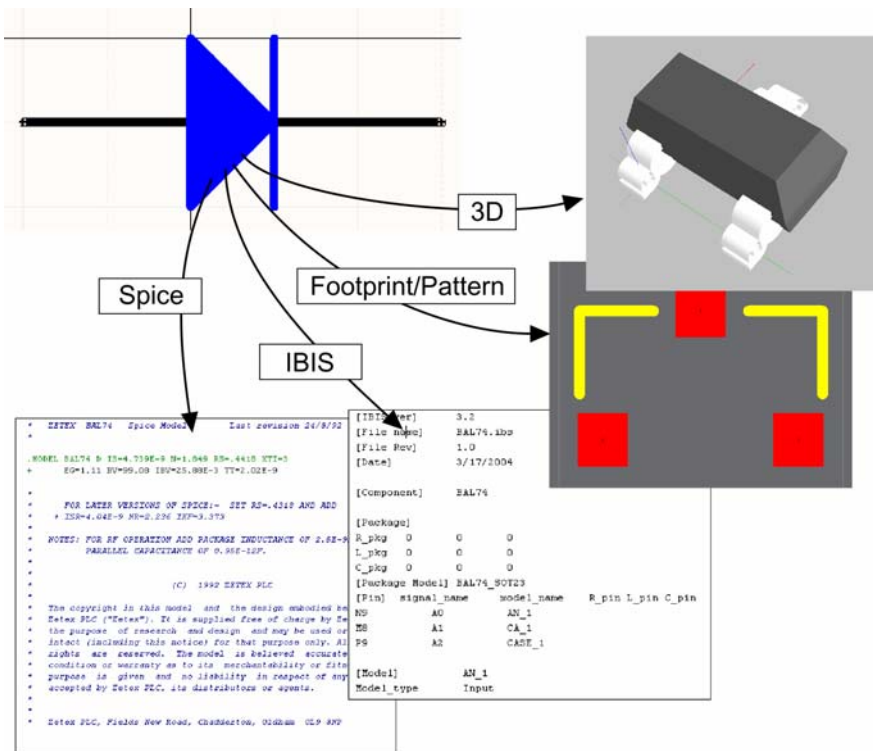
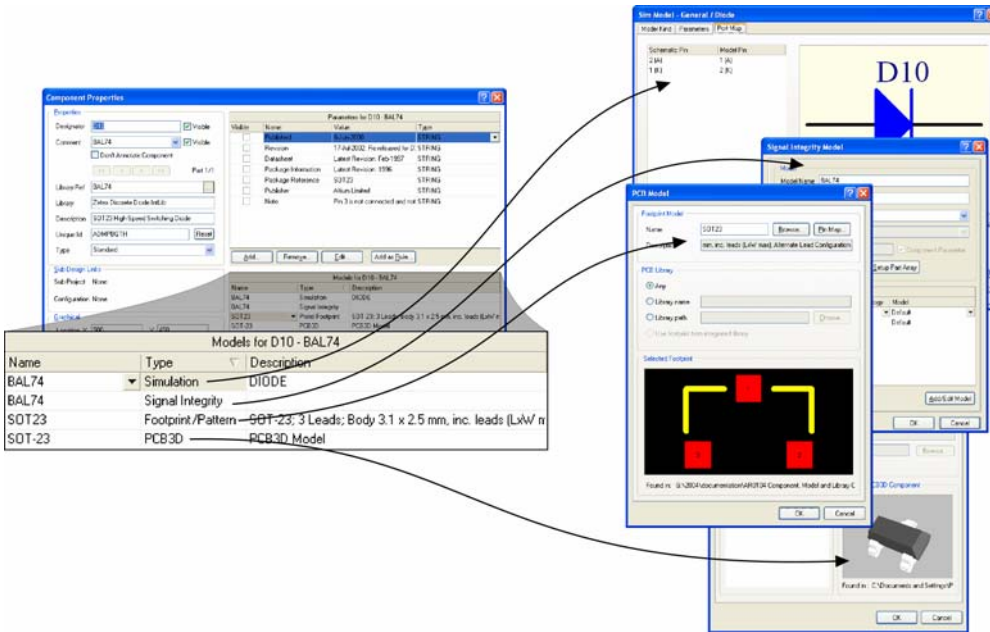


図9 各ドメインにあるコンポーネントをどのようにモデル化するかは、モデルファイルに保存されています。どのようにしてシンボルがそれぞれのインプリメンテーションモデルに接続されるか、確認できます。

回路図の段階では、デザインは論理的に接続されたコンポーネントの集まりです。デザインをテストしインプリメントするには、シミュレーション、PCB レイアウト、シグナルインテグリティ解析など、別のモデル化するドメインへそれらを移行させる必要があります。

各ドメインには、コンポーネントに関する情報と回路図シンボルのピンに情報をマップする手段が必要となります。ドメイン情報のいくつかは、モデルファイルの中にあり、そのフォーマットはたいてい、事前に定義されています。例としては、IBIS、MDL、CKT があります。いくつかの情報はモデルファイルには存在しません。例えば、SPICE のピンマッピングやネットリスト生成のデータはシステムで保存、管理されています。

必要なドメイン情報はすべて、回路図コンポーネントに含まれていて、追加される各モデルへの別のインターフェースを格納しています。実際、完成したモデルは、コンポーネントに格納されたモデルマッピング情報やモデルライブラリに格納されたドメインモデル情報の組み合わせです。



コンポーネントは、マルチドメイン用のモデルを複数持つことができます。また、各ドメインは複数のモデルを持つことができます。それらのうちの1つが現在のモデルとして設定され

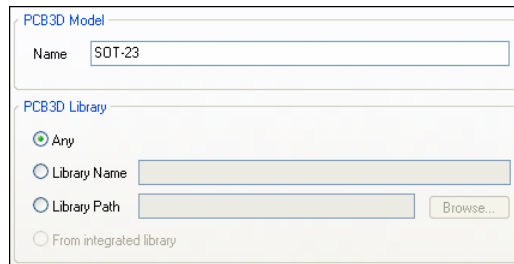
IBISシグナルインテグリティモデルやVRMLまたはIGES 3Dモデルは、使用する前にAltium Designerフォーマットモデルにインポートする必要があります。IBISモデルは、Signal Integrity Modelダイアログ（SIモデルをコンポーネントに追加する時に開きます）で直接インポートされます。VRMLとIGESモデルは、それが回路コンポーネントに追加される前にPCB3DLibにイ

図 10 各モデルと各マッピングへの必要なリンクは、それぞれの Model ダイアログで定義されます。

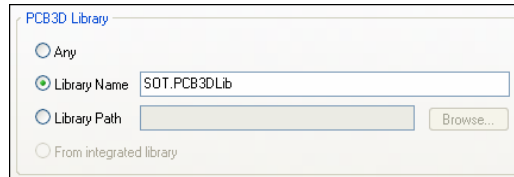
モデル参照のオプション

コンポーネントに対してモデルを追加する場合は常に、どの程度の強さでモデル検索を制御したいかを定義するオプションがあります。一つのモデルタイプから別のタイプへ、わずかに変化する程度ですが、モデルエディタのダイアログにはこれらのオプションが含まれています：**Any**、**Library Name**、**Library Path**、**From integrated library**。

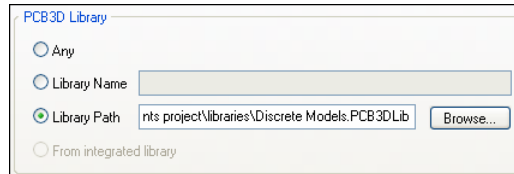
Any – 一致するモデルについて、すべてのライブラリを検索。



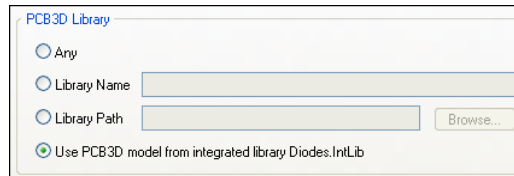
Library name – 一致するモデルについて、この名称のライブラリのみ検索。



Library path – 一致するモデルについて、この保存場所のこの名称のライブラリのみ検索。



Integrated library – このコンポーネントを配置するために使用した統合ライブラリから直接、モデルを引き抜きます。統合ライブラリは、有効な場所で利用することができます。



ライブラリ - コンポーネントとモデルの集合体

コンポーネントとモデルは、ほぼ、いつも、集合体として格納されています。つまり、“ライブラリ”です。設計プロセス中の、より効率的な利用のためです。ライブラリは、構成要素を含んでいるだけの存在といえるかもしれませんが。コンポーネントを作り上げるだけで、完全な定義を必要としません。たとえば、説明的な名前である“Modelライブラリ”は、そのライブラリがモデルだけを含んでいることを示しています。一方、“回路図ライブラリ”には回路図シンボルだけが含まれます。

しばしば、ライブラリの構成要素は Altium Designer のファイル用ではない、またはバージョン管理がされていない、不特定のフォーマットで格納されていることがあります。これは、一人以上のエンジニアがひとつの設計に従事している場合や、違った場所で設計を行っている場合に、より顕著になります。databaseライブラリは、すべてのシンボル参照、モデルリンク、パラメータ情報が ODBC や ADO に準拠したデータベース、または Excel のスプレッドシートで格納されている場所を意味します。“バージョン管理の行き届いているデータベース”ライブラリは、シンプルな拡張が可能です。そこでは、シンボルやモデルがバージョン管理の元でデータベースライブラリに格納されているからです。

それぞれのライブラリや適切なファイルの拡張については、以降で詳しく説明します。

ライブラリタイプ

モデルライブラリ(*.MDL, *.CKT, *.PcbLib)

各ドメインタイプのモデルは、モデルライブラリ、または *Model container* に格納されます。モデルのグループ化と組織化はドメイン間で異なります。いくつかのドメインでは、SPICE(*.MDL, *.CKT)のように、1 ファイルにつき 1 モデルになっています。別のドメインでは、モデルは通常、パッケージ形式のライブラリ(*.PcbLib)内にグループ化された PCB フットプリントのようなユーザ指定の分類に従ってライブラリファイル内にグループ化されます。

回路図ライブラリ (*.SchLib)

これらのライブラリには回路図シンボルや、別のモデルライブラリに存在するモデル定義への固有のリンク(実際のモデルそれ自体ではなく、そこへのポイントみたいなもの)が含まれています。

アクティブプロジェクトの回路図ソースライブラリの生成

作成済みのデザインを収めておけるちゃんとしたライブラリやアーカイブを作成したいなら、進行中のプロジェクトのソースとなる回路図ドキュメントに配置されたすべてのコンポーネントが格納された回路図ソースライブラリを作成する能力があれば、とても便利です **Design » Make Schematic Library** の操作により、実際のプロジェクトで使用するすべての回路図ソースドキュメントを開くことができます。開く準備ができていないときは、ライブラリドキュメント `ProjectName.SCHLIB` が自動的に作成され、プロジェクトに追加されます。

統合ライブラリ (*.IntLib)

統合ライブラリは、回路図ライブラリが参照されるモデルライブラリとともに結合され、“コンパイル”されて、単一のライブラリファイルになったものです。一つの統合ライブラリにコンパイルすることの利点は、すべてのコンポーネント情報が、単一の持ち運びできるファイルとして利用できるからです。単一の統合ライブラリ内にあるコンポーネントやモデルは、デコンパイル(*.IntLib ファイルを開いて、ソースを解凍)しないと、編集できません。すべてのモデルはただひとつの統合ライブラリとしてまとめられているので、必要に応じてプロジェクトや、他のプロジェクトで使用されます。

ワーキングライブラリまたは終了済み設計アーカイブの生成または再利用

進行中のプロジェクトで使われた回路図ドキュメントに配置されたすべてのコンポーネントとリンクされたモデルが格納された統合ライブラリも生成できます。とても簡単に、プロジェクトライブラリから、またはサードパーティから送ってもらって作成できます。回路図設計が進行中の場合、**Design » Make Integrated Library** の操作でもアクセスできます。

データベースライブラリ (*.DBLib, *.SVNDBLib)

これらのライブラリには、シンボル情報、モデルへのリンク、モデル、パラメータ情報が ODBC ベースや ADO ベース、あるいは Excel スプレッドシートなど、Altium Designer に属さないフォーマットで格納されています。

つまり、データベース内のそれぞれの記録は、モデルリンク、データシート参照、あるいはその他のコンポーネント情報といったパラメータのすべてが格納されたコンポーネントを表しています。その情報には、在庫管理やその他の業務用データへのリンクなどを含めることができます。

データベースライブラリには、バージョン管理なし(*.DBLib)と、バージョン管理あり(*.SVNDBLib)との、二つのタイプがあります。違いは、参照先のシンボルやモデルを含むシンボルやモデルライブラリの場所です。この違いは、以下のようにまとめられます。

Component, Model and Library Concepts

- **Database Library (*.DBLib)** - シンボルとモデルのライブラリは、使用している PC のハードディスクか、その他のローカル/ネットワークメディアに格納されます。
- **SVN Database Library (*.SVNDBLib)** - シンボルとモデルのライブラリは、サブバージョンリポジトリ内のバージョン管理の下に置かれます。

データベースライブラリには、外部ベンダのデータベースからコンポーネントを直接に配置できる機能があります。データベースライブラリの中に、コンポーネントを完全に構成する一切の情報が、データベースそれ自体に格納されています。つまり、シンボルは、ただのグラフィック表現、あるいは、その他のモデルということです。

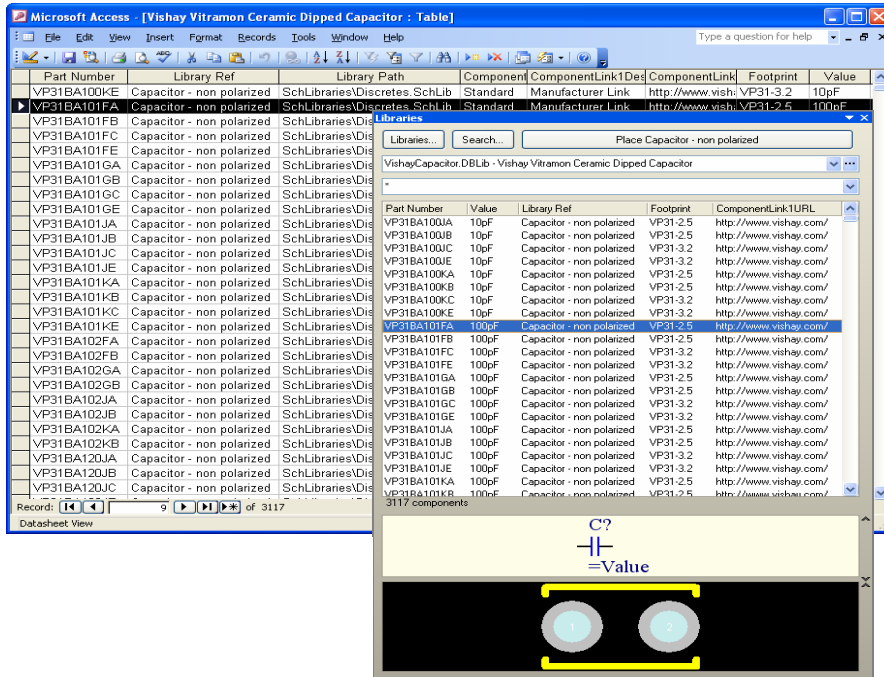


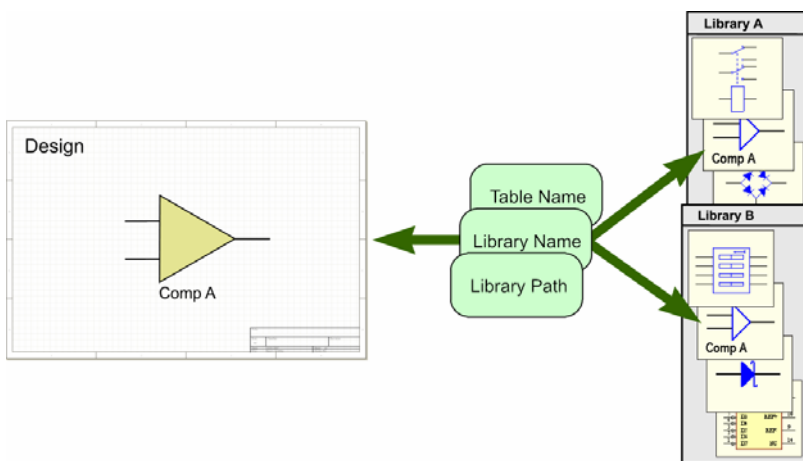
図 11 データベースライブラリにはすべての情報が格納されており、コンポーネントを配置する際に読み出されます。

データベースライブラリからコンポーネントが配置されると、適合するデータベースの記録にある関連フィールドと、中間リンクファイルにある定義されたマッピングの一致を使って、そのコンポーネント用のパラメータとモデルの情報がその場ですぐに作成されます。一つ以上のこれらのパラメータは、データベースにさかのぼる現在のリンクを保守するのに使われ、配置後は、将来の設計のために同期が取られることとなります。

コンポーネントの検索と識別

デザインやライブラリに対する可搬性は一般的な要求であるといえます。ライブラリがデザインそれ自体とは別な場所にあったり、二つ以上のライブラリのセットが単一の設計で使われたりすることが普通になっています。あるいは、単に、会社ライブラリのローカルコピーが使える自宅に作業を持って帰りたいだけかもしれません。なぜなら、ソースライブラリを参照できるその柔軟性が必要とされているからです。コンポーネントのソースを管理し、正しいコンポーネントを選べることも重要なのです。

ライブラリからコンポーネントを配置する際に、どのライブラリ（DBLibの場合はテーブル）からコンポーネントを取り出したか、覚えておかななくていけない、というのがその理由です。コンポーネントを更新しようとしてみたことが一度でもあるなら、また、オリジナルのライブラリを持っておらず、限定的な設計環境で作業を強いられたことがあるなら、このような問題がどんなにフラストレーションを引き起こすか、よくご存知のはずです。したがって、どんな作業環境の構成にも適合できる定義可能な管理レベル、そして、メンテナンスが容易な設計コンポーネントの統合方法、この二つを提供するソリューションが必要です。



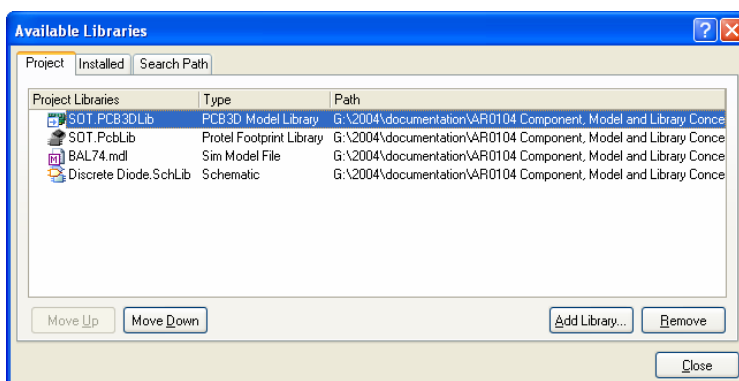
Altium Designer は、柔軟性と管理のしやすさの両方を提供することで、参照ライブラリの場所を容易に切り替えることができます。また、デザインから正しいコンポーネントを取り出して使っていることを識別し、確認します。

図 12 コンポーネント A がライブラリ A とライブラリ B の朗報から参照される同じコンポーネントであることを示しています。ライブラリパス、ライブラリ名を変えることで、コンポーネントや、DBLink の Table Name のソースを切り替えることができます。

検索場所の確認

コンポーネントを確認するプロセスの一環として、操作が必要な時はいつでも、Altium Designer は特別な場所でモデルやライブラリを検索します。検索については、すでに検討したように、モデルエディタからのモデル参照をどのようにコントロールするかを指定するオプションのことも考慮されています。たとえば、サーキットシミュレーションを行う場合、各コンポーネントにリンクした SPICE モデルが読み出され、XSPAICE シミュレーションエンジンによって使用されます。他には、回路図キャプチャから PCB レイアウトにデザインを移行する場合が挙げられます。この過程の間は、各コンポーネントのフットプリントが検索され、PCB に配置されます。これらの場所にあるどんなモデル/ライブラリも、*available libraries* (利用可能なライブラリ) として参照されます。つまり、どんなモデル/ライブラリのセットでも、プロジェクトや設計環境で使えることを意味しています。d d お

Available Libraries ダイアログ



現在のプロジェクトで利用できるモデル/ライブラリを *Available Libraries* ダイアログで確認することができます。Libraries パネルで Libraries ボタンをクリックするか、または Design » Add/Remove Library を選択してダイアログを開きます (図 13)。

図 13 すべてのモデルが統合ライブラリに結び付けられているわけではないので、検索の順序は左から右へ、Available Libraries ダイアログ (Project, Installed, Search Path) のタブを通じて行われます。実際は、available libraries はこのダイアログ内の上から順に存在しているはずなので、全体の検索順序は直感的に、また、簡単にセットアップできます。

Component, Model and Library Concepts

検索可能なモデル/ライブラリが存在する場所

- **Project** – これらのモデル/ライブラリは、プロジェクトとそのドキュメントだけにリンクされ、利用可能になります。この方法の利点は、プロジェクトを開く時はいつでもモデル/ライブラリを利用できることです。不利な点は、もし、プロジェクトファイルのある PC から他の PC へ移動した場合、Project フォルダ構造に保存されていないモデル/ライブラリを忘れてしまうことです。
- **Installed** – これらのモデル/ライブラリは、Altium Designer 環境と関連しているので、コンポーネントは開いているすべてのプロジェクトで利用できます。Installed libraries のリストに追加されたどんなモデル/ライブラリも、指定されたパスに従って、インストールできます（詳細は以下）。
- **Search Path** – これらのモデル/ライブラリは、Options for Project ダイアログの Search Path タブで検索パスを指定すると（または、Available Libraries ダイアログにある Search Path タブの Paths ボタンから便利にアクセスすると）、プロジェクトで利用できます。各検索パスではフォルダを定義し、もし、Recursive オプションが有効ならばサブフォルダを含めることができます。Search Path で見つかったすべてのモデルとライブラリファイルは、有効になります。Search Path フォルダに多くのファイルがある場合、検索パスを使用してモデルを検索することは、遅くなる傾向があることに注意してください。

Altium Designerの環境が、モデル/ライブラリの場所について、柔軟なコントロールを提供できるように、各モデルのタイプには正しいファイル拡張子をお使いください。たとえば、フットプリントには、*.lib または *.pcb.lib という拡張子が必要です。同様に、SPICE .subckt には、*.ckt ファイルで、SPICE .model には *.mdl ファイルである必要があります。モデル検索がうまくマッチしないときは、Message! ヘルプエラーが表示されます。

ライブラリを相対パスでインストール

Installed Library のリストに追加されているどんなライブラリでも、指定パスに関連付けてインストールすることができます。Available Libraries ダイアログの Installed タブが有効なら、異なるライブラリのセットを簡単に切り替え、デザイン内部のコンポーネントのソースを管理できるようになります。パスのエントリを変更すると、自動的にこれら既存のライブラリが新しい場所で見つかったリストにリロードされます。

ライブラリの起動と終了

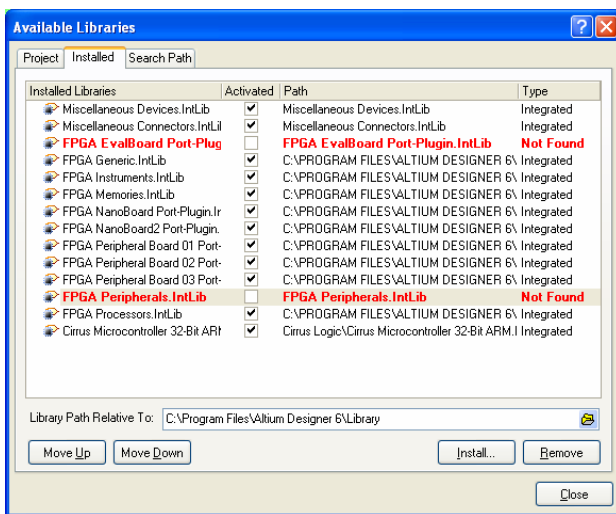


図 14 新しい場所で見つかったリスト内のライブラリは赤くハイライト表示されます。

現在、Installed Libraries のリスト（Available Libraries ダイアログの Installed タブからアクセス）に追加されている各ライブラリは、'Activated' または 'Deactivated' になります。これにより、どのライブラリのセットを設計に使っているのか、視覚的にすばやく、特定できるようになります。Activated オプションで、シンプルに切り替えることができます。

アクティベートされていないライブラリは、アンインストールされているものとして扱われていますが、リストには残っていますので、デザインの必要に応じて、すばやくアクティベートできます。特定の関連パス上に発見されなかったライブラリは、アクティベートできません。

Component Properties (コンポーネント属性) ダイアログ

Available Libraries ダイアログからのコンポーネントの検索と特定に加え、Component Properties ダイアログにも、一段上の管理方法があります。

コンポーネントレベルでライブラリ名を変更

コンポーネントを特定するための次の下位レベルは、ライブラリ名をコンポーネントそれ自体に変えてしまうことです。

Component Property ダイアログには、コンポーネントを特定する 3 つのレベルがあります。

- **Design Item ID** - 既存のアクティベートライブラリのセットの中で、そのコンポーネント名がシート上のデザインコンポーネントにマッチする最初に見つかったライブラリコンポーネントを使用します。
- **Use Library Name** - アクティベートされているライブラリの中で、特定のライブラリ名にマッチし、コンポーネント名がシート上のデザインコンポーネントにマッチする最初に見つかったライブラリコンポーネントを使用します。

- **Use Database Table Name** - 特定のライブラリ名にマッチするアクティベートされているライブラリの中で、または、特定のテーブル名にマッチするデータベースのテーブルのライブラリの中で、コンポーネント名がシート上のデザインコンポーネントにマッチする、最初に見つかったライブラリコンポーネントを使用します。

コンポーネント名 (Design Item ID)、Use Library Name、Use Database Table Name はすべて、コンポーネントを配置する属性ダイアログの Library Link の領域で、特定することができます。Library Name または Table Name の使用は、選択オプションで有効/無効を切り替えることができます。

DBLibからのテーブル名の変更

統合ライブラリから独自に配置された設計コンポーネントのため、参照ライブラリのベースを変更することができます。デザイン中のすべてのコンポーネントを選択すると、新しくコンパイルされた DBLib、または、SVNDBLib に変更できます。

- **Use Library Name** オプションを無効にします (新しい DBLib/SVNDBLib が Install Library のリストに追加され、アクティブで、以前の統合ライブラリが削除、またはアクティベートされていないことを確認してください)。
- **Use Library Name** オプションが有効のまま、代わりに、DBLib/SVNDBLib の名前を入力します。選択されたコンポーネントがすべて、同じテーブル名に所属している場合、あるいは、blank/disable のままになっている場合、Table Name は特定できます。つまり、データベースの中のいずれかのテーブルで、最初に見つかったものがそれぞれのケースで使用されます。

正しいライブラリが実際に、デザインコンポーネントのためのリファレンスとして使用されていることを確認するには、**Validate** ボタン - *Component Properties* ダイアログの **Library Link** 領域にあります。デザインコンポーネントとの最初のマッチが見つかったパスとライブラリが表示されたダイアログが表示されます。

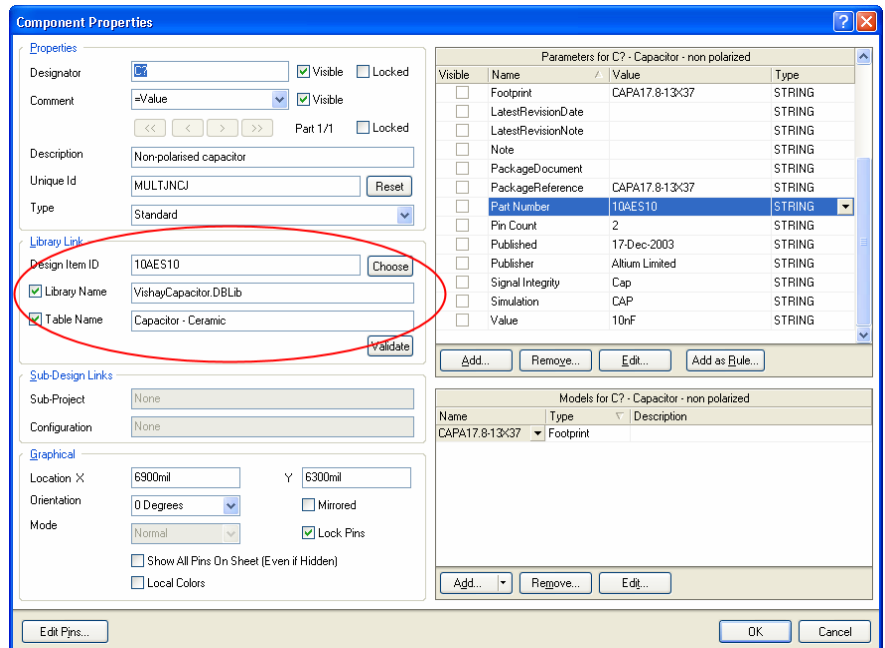


図 15 *Component Properties* ダイアログから、異なるレベルのコンポーネントの特定を管理する

更新履歴

Date	Version No.	Revision
9-Dec-2003	1.0	New product release
28-Jul-04	1.1	Added images and extra detail to text
01-Jul-2005	1.2	Updated for Altium Designer SP4
30-Nov-2005	1.3	Updated for Altium Designer 6.0, database libraries added
07-Jun-2006	2.0	Updated for Altium Designer 6.3
21-May-2007	2.1	Updated for Altium Designer 6.7
13-Nov-2007	2.2	References to 3D visualization added for 6.8
21-April-2008	2.3	Converted to A4

ソフトウェア、ハードウェア、文書、および関連資料

Copyright © 2008 Altium Limited.

All rights reserved. この文書の印刷は、(1) 個人的使用に限定し、ネットワークコンピュータやあらゆる種類の媒体にコピーや送付を行わない、かつ (2) 文書の変更をまったく行わない、という条件でのみ行うことができます。Altium Limited の事前の書面による許可なく、本書の全体または一部を問わず、機械的または電子的な複製、他言語への翻訳を禁じます。ただし、公表するレビュー目的での抜粋を除きます。本書の無許可の複製は、各国の法律でも禁止されています。違反者は、罰金や実刑を含む刑罰と民事罰両方の対象となることがあります。Altium、Altium Designer、Board Insight、CAMtastic、CircuitStudio、Design Explorer、DXP、LiveDesign、NanoBoard、NanoTalk、Nexar、nVisage、P-- CAD、Protel、SimCode、Situs、TASKING、Topological Autorouting、およびそれぞれに対応するロゴは、Altium Limited またはその子会社の商標または登録商標です。本書に記載されているそれ以外の登録商標や商標はそれぞれの所有者の財産であり、商標権を主張するものではありません。