



# Keeping Components Up-To-Date

## Summary

Application Note  
AP0144 (v1.0) June 09, 2006

This application note provides detailed information on updating placed components with changes made to those components in source libraries or a linked external database.

You've placed your components on the schematic, maybe even fully wired, compiled and taken the design over to the board layout stage. There will undoubtedly come a time when a change to a component is required – a new parameter added, a change to a model link, or maybe a change to a symbol or footprint at the graphical-level.

The logical place to make a change is at the source. The nature of this source depends on how the component was placed:


**From an Integrated Library:** the source libraries are extracted, the change made and then the integrated library package compiled to generate the revised integrated library.


**From a Database Library:** for a parameter or a symbol/model reference, the change is made to the corresponding record for the component, in the linked external database. If the change is a graphical one – to a symbol or model – it is carried out within the referenced source library.


**From a Schematic/PCB Library:** the change is made to the component or footprint in the corresponding library from which the component/footprint was placed.


You may also have linked a placed component to an external database, using a Database Link file (DBLink). The symbol and model changes are made in the source libraries. For parameters, changes are made in the database.

Once changes are made, they need to be pushed across to the design. Altium Designer provides a variety of update tools that allow you to quickly and efficiently pass component changes to your Altium Designer components. This document details those tools – tools that enable you to always retain synchronicity between your component source and the placed instances of those components.

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

 For more information on integrated libraries, refer to the [Enhanced Library Management Using Integrated Libraries](#) article.

 For more information on direct placement of components from a database, refer to the [Using Components Directly from Your Company Database](#) and [Working with Version-Controlled Database Libraries](#) application notes.

 For more information on linking existing components – placed or part of a library – to a database, refer to the [Linking Existing Components to Your Company Database](#) application note.

## Updating Parameters from a Database

Altium Designer provides a tool for updating the parameters of Altium Designer components with information stored in an external database.

The components must be linked to the corresponding component records in the external database, for update to be made possible. Linkage is performed through the use of an intermediary linking file, which can be one of the following:

- **Database Link** file (\*.DBLink) – used when linking existing placed components to an external database or, more typically, defined components in a source schematic library.
- **Database Library** file (\*.DBLib) – used when placing components directly onto a schematic sheet from an external database.
- **SVN Database Library** file (\*.SVNDBLib) – as for DBLib, but with symbol and model libraries stored under version control.

Updates are performed from a schematic document using the **Update Parameters From Database** command, available from the main **Tools** menu. Launching this command will bring up the *Update Parameters From Database* dialog (Figure 1). Use this dialog to choose which schematic documents and component types you wish to include in the update.

In the context of this tool, the term 'Altium Designer component' is used to describe a component instance placed on a schematic sheet, or a component in a source schematic library.

The **Update Parameters From Database** command is also available in the Schematic Library Editor. The source library must be part of a library package (\*.LibPkg) and a DBLink file used to provide the linking from library components to component records in the external database.

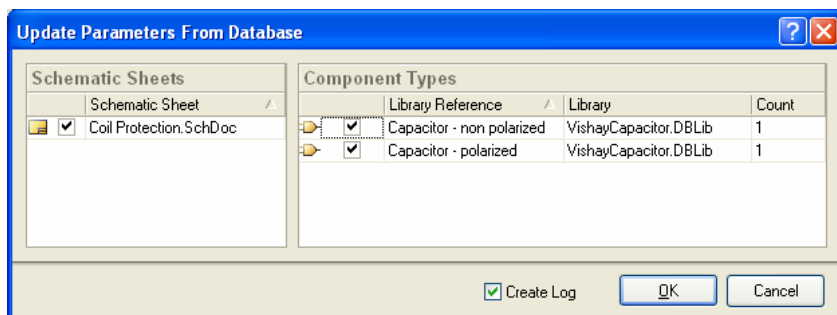


Figure 1. Specifying the scope of the update.

After defining the scope of the update, click **OK**. The external database will be queried for matching components. If there are parameter differences between the Altium Designer components and the matching records in the database, you will be taken to the *Select Parameter Changes* dialog (Figure 2).

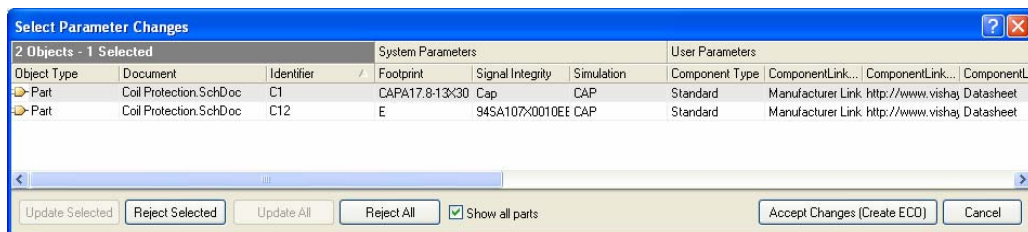



Figure 2. Specifying updates at the parameter-level.

This dialog lists all parameters that exist in the database records for linked Altium Designer components falling under the scope of the update. Any parameters that are defined for an Altium Designer component, but which are not a field in a database table, will not appear listed. For example, you may have placed a component directly from a database using the database library feature, then added one or more parameters after placement.

Only those parameters that are mapped – between the external database and the placed component instance – will be listed. Parameter mapping is performed on the **Field Mappings** tab of the intermediary link file.


The dialog will initially show proposed updates to bring the Altium Designer component parameters into sync with those in the database, based on the update actions you have defined in the intermediary link file (DBLink, DBLib, SVNDBLib).

 For information on setting update options for parameters in an intermediary link file, refer to the section *Parameter Update Options*, in either the [Linking Existing Components to Your Company Database](#), or [Using Components Directly from Your Company Database](#) application notes.

Parameter differences are distinguished by the use of a unique icon inserted in the relevant cell. For example, a blue triangle in the corner of a cell (Figure 3) means that a difference has been detected between the value of a parameter in the Altium Designer component and the same parameter in the linked database record.

ComponentLink...	DatasheetDocu...	Designator
http://www.vishay	12-Nov-2005	10AED47
http://www.vishay	12-Nov-2005	94SA107X0010EE

Figure 3. Detected parameter differences.

 For a full listing of all available cell states in the *Select Parameter Changes* dialog, see the section *Parameter Update States*, later in this document.

The controls provided in the *Select Parameter Changes* dialog allow you to fully control which updates to proceed with and which to reject. You can reject updates to all parameters for a selected component, or for specific parameters of that component. To reject a proposed update for a specific parameter, simply select the relevant cell and click the **Reject Selected** button. To reinstate the update, simply click the **Update Selected** button.

A parameter associated with an Altium Designer component can be prevented from being updated. See [Preventing Parameter Updates](#).

When you are satisfied with the update solution, click the **Accept Changes (Create ECO)** button. Use the *Engineering Change Order* dialog that appears (Figure 4) to validate and then execute the updates accordingly. If you realize there is an update you really don't want to proceed with, simply disable the applicable change order entry.

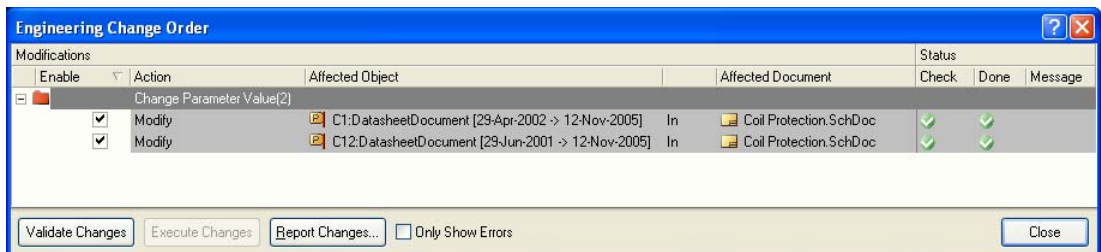


Figure 4. Executing an ECO to perform the updates.

# Updating Schematic Components from Libraries

Altium Designer provides a tool for updating placed components on a schematic sheet with modified information from a source library. This includes Schematic Component Libraries (\*.SchLib), Integrated Libraries (\*.IntLib) and Database Libraries (\*.DBLib, \*.SVNDBLib).

The update feature allows you to pass changes to parameters, as well as model and graphical information.

Updates are performed from a schematic document using the **Update From Libraries** command, available from the main **Tools** menu. Launching this command will bring up the *Update From Library* dialog (Figure 5).

Although DBLib and SVNDBLib files present as libraries in the **Libraries** panel, they are only a source of connection and field mapping – not libraries in the true sense of the word. The update feature passes changes to parameters, symbol and model references in the external database, as well as graphical modifications made in referenced symbol and model libraries.

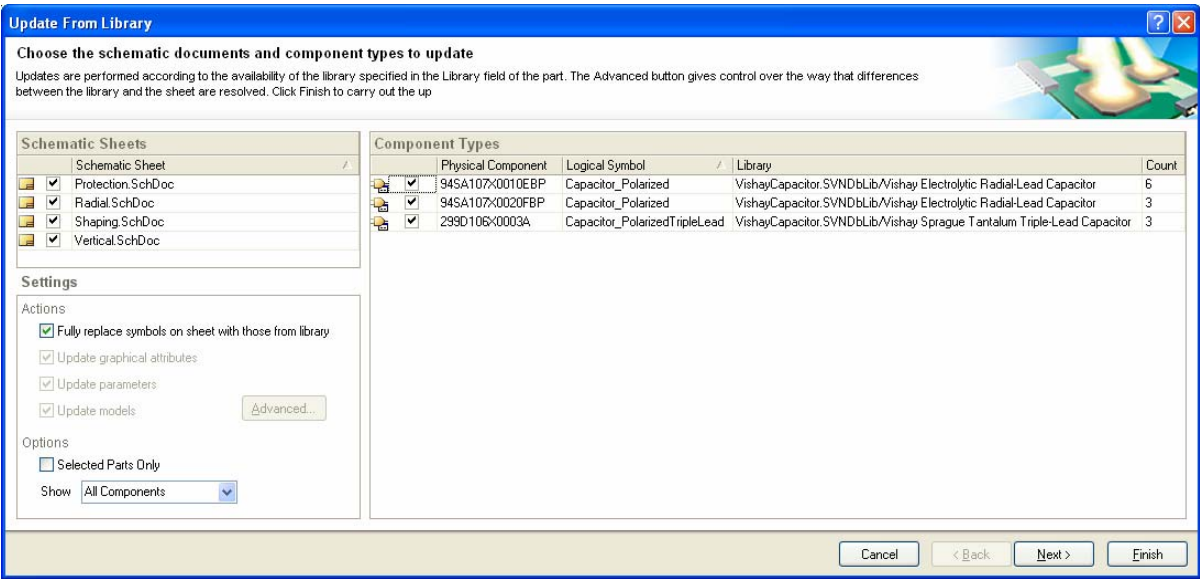


Figure 5. Specifying the scope and default actions for the update.

## Defining the Scope of the Update

The first page of the dialog deals with the scope of the update – which source schematic documents are to be included in the update and the specific component types contained thereon.

The **Schematic Sheets** region of the dialog will load all possible schematic documents to which the update can be applied. This can be either a single, free schematic document, or all schematic sheets within the active project. Enable those documents whose components you wish to be considered in the update.

As you enable/disable entries in the **Schematic Sheets** list, the constituent physical components will be listed in the **Component Types** region of the dialog. Each component is listed in terms of:

- Physical component name

- Associated logical symbol
- Source library from which it was placed
- Number of instances currently placed across all enabled schematic documents.

Inclusion of components in the list is in accordance with the options available at the bottom left of the page (Figure 6). Use the **Show** field to determine the types of components included in the update. By default, **All Components** is selected, but you can choose to update just:

- **Standard Components** – those placed from a schematic library or integrated library or
- **Database Components** – those placed from an external database using a DBLib or SVNDBLib.

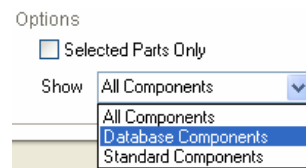


Figure 6. Specifying component types to include in the update.

All component types are initially enabled for inclusion in the update. Disable any that you definitely don't want to update. Alternatively, select the exact components that you wish to update – directly on the schematic sheet(s) – prior to entering the dialog. Then simply enable the **Selected Parts Only** option. Only those components in your selection will appear listed.

The main thing to remember is that you are always in full control over what gets included in the update.

### Defining Default Update Actions

The **Settings** region of the page is where you can define the default level of update required. The default update actions you define here will be applied to all component instances of the enabled component types.

The simplest form of update is to fully replace the components on the schematic sheet(s) with those defined in the source library. Graphical attributes, parameters and model links are all updated directly with the information that exists in the source library. Remember that if the source library is a DBLib or SVNDBLib, the parameter and model link information will come from the corresponding component records in the external database, while the graphical attribute changes will come from the referenced symbol libraries.

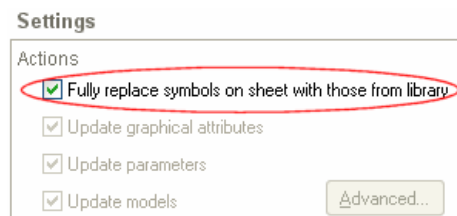


Figure 7. Fully replace by default.

Full replacement is enabled by default (Figure 7). Should you wish a little more control over what is updated, disable the full replacement option. The individual update options (graphical, parameter, model) will become available for inclusion/exclusion as required.

For parameter and model update actions, still further control is afforded through the

**Library Update Settings** dialog (Figure 8). Access this dialog by clicking the **Advanced** button. Not only can you define the default, global update actions for parameters and models using this dialog, but also control which specific parameters and models are included in the update.

Definitions on the first page of the *Update From Library* dialog, and the *Library Update Settings* dialog are persistent. They are stored in the project file upon saving.

## Keeping Components Up-To-Date

### Actions

- ☐ Fully replace symbols on sheet with those from library
  - ☒ Update graphical attributes
  - ☒ Update parameters
  - ☒ Update models
- Advanced...

**Library Update Settings**

**Parameter Settings**

Library parameters not in sheet: Add

Sheet parameters not in library: Remove

Parameters to update:

Parameter	Update
Comment	<input checked="" type="checkbox"/>
Component Kind	<input checked="" type="checkbox"/>
Component Type	<input checked="" type="checkbox"/>
ComponentLink1Description	<input checked="" type="checkbox"/>
ComponentLink1URL	<input checked="" type="checkbox"/>
ComponentLink2Description	<input checked="" type="checkbox"/>
ComponentLink2URL	<input checked="" type="checkbox"/>
DatasheetDocument	<input checked="" type="checkbox"/>
Description	<input checked="" type="checkbox"/>
Designator	<input checked="" type="checkbox"/>
DrillSize	<input checked="" type="checkbox"/>
Library Reference	<input checked="" type="checkbox"/>
Note	<input checked="" type="checkbox"/>
PackageReference	<input checked="" type="checkbox"/>
Part Number	<input checked="" type="checkbox"/>
PCB_Layout	<input checked="" type="checkbox"/>
Pin Count	<input checked="" type="checkbox"/>
Published	<input checked="" type="checkbox"/>
Publisher	<input checked="" type="checkbox"/>
Value	<input checked="" type="checkbox"/>

**Model Settings**

☒ Add models found in library but not in sheet

☒ Remove models found in sheet but not in library

☒ Update which models are the current models

Models to update:

Model Name	Update
Footprint	<input checked="" type="checkbox"/>
299D-A	<input checked="" type="checkbox"/>
E	<input checked="" type="checkbox"/>
F	<input checked="" type="checkbox"/>

OK Cancel

Figure 8. Defining default update actions for parameters and models.

After defining the scope of the update as required, and the default actions to be carried out, you could simply click **Finish**, then review and execute the changes to be implemented in the subsequent Engineering Change Order that is generated. Should you wish to further refine the update on a per-component basis, prior to generating the ECO, click **Next** to access the second page of the dialog, detailed in the next section.

## Controlling the Update by Component Instance

The second page of the *Update From Library* dialog presents you with a detailed grid, listing all components instances involved in the intended update process (Figure 9). The previous page of the dialog allowed you to specify, at a coarse level, which physical component types get included in the update. This page allows you to fine-tune exactly which component instances of those types get updated.

Definitions on the second page of the *Update From Library* dialog are not persistent. They will be lost if you go back to the first page or close the dialog.



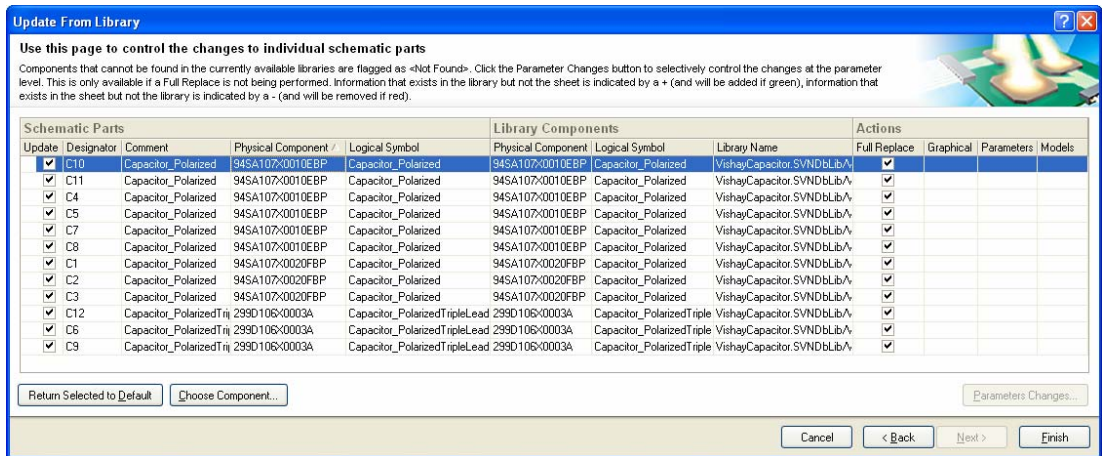


Figure 9. Controlling the update by component instance.

The grid is partitioned into three regions:

- **Schematic Parts** – information about each placed component on the schematic sheet(s)
- **Library Components** – information about the specific source library component that will be used in the update.
- **Actions** – the specific update actions required for each component instance. These will initially be set in accordance with the default action settings defined on the first page of the dialog.

For each component, the entry for the source library component will initially be that used to place the component instance in the first place. This information is taken from the **Library Link** region of the placed component's associated properties dialog, an example of which is shown in Figure 10.

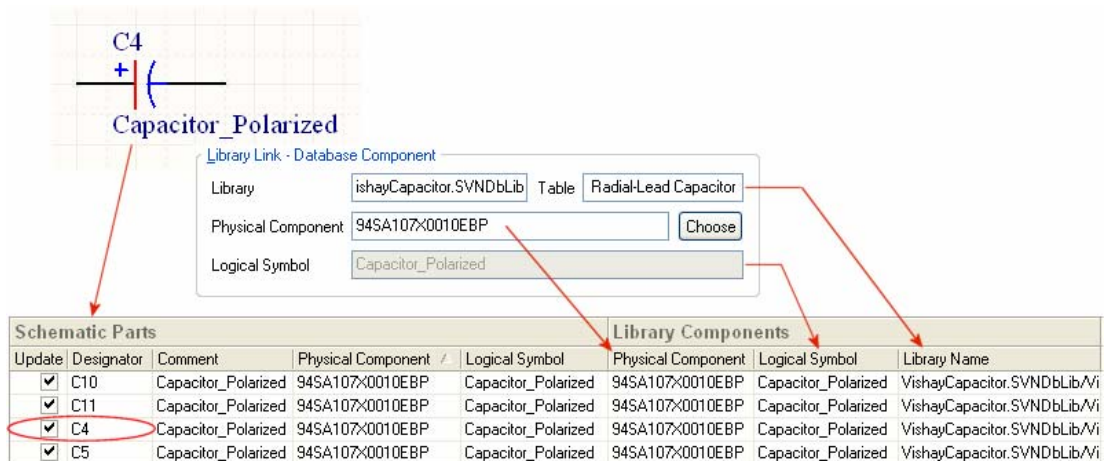


Figure 10. Retrieving source library information.

## Specifying an Alternate Component

Typically, the update will involve passing on changes made to the original physical component in the source library, to the placed instance(s) of that same physical component on the schematic sheet(s). There may be occasion where you want to change the physical component that is placed in the design altogether. This can be readily specified as part of the update.

You can specify an alternate component for a component instance by directly editing the corresponding **Physical Component** field (in the **Library Components** region). If the specified physical component does not exist in any of the libraries currently added to the project's Available Libraries list, the entry <Not Found> will be entered into the **Logical Symbol** and **Library Name** fields. Update in this case will not be possible, for that component instance.

Alternatively, select a component instance in the grid and click the **Choose Component** button. The *Browse Libraries* dialog will appear (Figure 11), from where you can browse for the required replacement component in any of the currently Available Libraries for the project. The dialog also provides a search facility, which can be used to search for a component within the Available Libraries, or along any nominated search path.

When searching for a database component, the search can only be across currently loaded DBLib and SVNDBLib files (Available Libraries). You cannot search for database components along a specified search path.

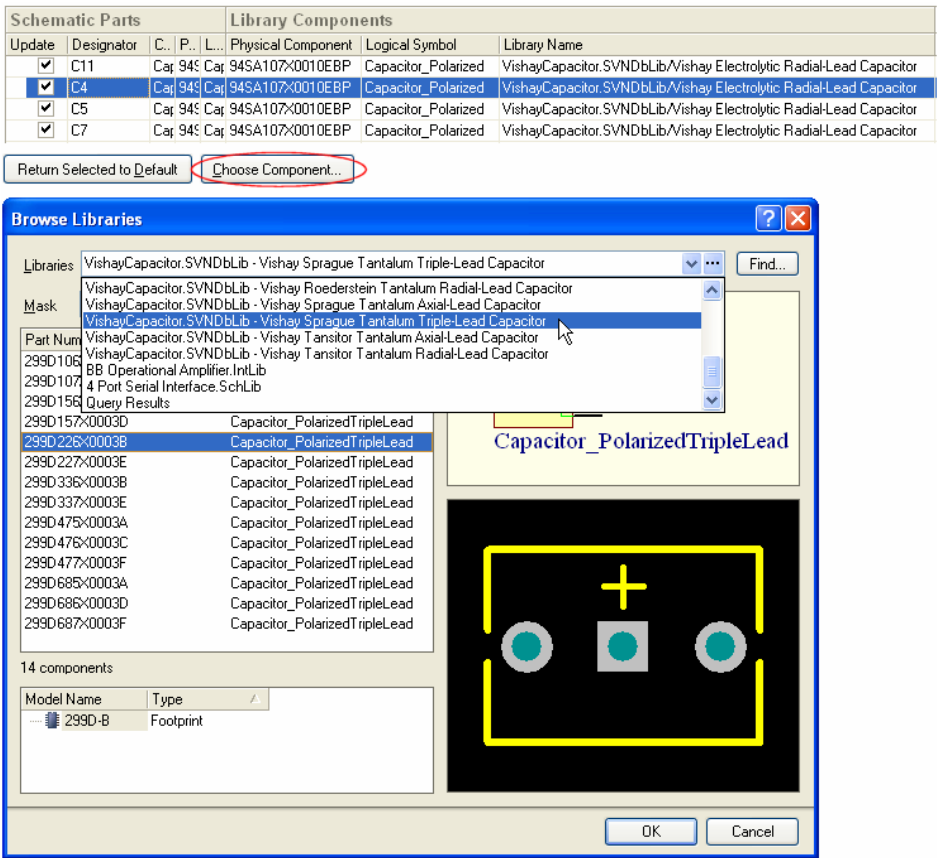


Figure 11. Browsing for an alternate component to use.



When a valid component is chosen (Figure 11), the relevant information for the chosen component will be entered into the **Library Components** region of the page, overwriting the original source library component (Figure 12).

Library Components		
Physical Component	Logical Symbol	Library Name
94SA107X0010EBP	Capacitor_Polarized	VishayCapacitor.SVNDbLib/Vishay Electrolytic Radial-Lead Capacitor
299D 226X0003B	Capacitor_PolarizedTripleLead	VishayCapacitor.SVNDbLib/Vishay Sprague Tantalum Triple-Lead Capacitor
94SA107X0010EBP	Capacitor_Polarized	VishayCapacitor.SVNDbLib/Vishay Electrolytic Radial-Lead Capacitor

Figure 12. Information for the chosen replacement component.

Should you wish to revert to the original source library – keeping the original physical component – select the component instance in the grid and click the **Return Selected to Default** button.

Controlling Parameter Changes

Should you wish to browse individual parameter changes proposed by the update, click on the **Parameter Changes** button. The *Select Parameter Changes* dialog will appear, summarizing the parameter changes for those component instances with a parameter update action enabled. Figure 13 illustrates this – notice that component instances C1, C3, C7 and C10 have their parameter update action disabled and so do not appear in the dialog.

Individual parameter-level changes for a component instance will only be shown provided the **Full Replace** option is disabled and the **Parameters** option is enabled, in the **Actions** region of the page.

Update From Library

Use this page to control the changes to individual schematic parts

Components that cannot be found in the currently available libraries are flagged as «Not Found». Click the Parameter Changes button to selectively control the changes at the parameter level. This is only available if a Full Replace is not being performed. Information that exists in the library but not the sheet is indicated by a + (and will be added if green), information that exists in the sheet but not the library is indicated by a -

Schematic Parts				Library Components		Actions					
Update	Design...	Comment	Physical Compon...	Logical Symbol	Physical Component	Logical Symbol	Library Name	Full Replace	Graphical	Parameters	Models
<input checked="" type="checkbox"/>	C1	Capacitor_I	94SA107X0020FBP	Capacitor_Polari	94SA107X0020FBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C2	Capacitor_I	94SA107X0020FBP	Capacitor_Polari	94SA107X0020FBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C3	Capacitor_I	94SA107X0020FBP	Capacitor_Polari	94SA107X0020FBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C4	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C5	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C6	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C7	Capacitor_I	299D 106X0003A	Capacitor_Polari	299D 106X0003A	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C8	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C9	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C10	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C11	Capacitor_I	299D 106X0003A	Capacitor_Polari	299D 106X0003A	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C12	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C13	Capacitor_I	94SA107X0010EBP	Capacitor_Polari	94SA107X0010EBP	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C14	Capacitor_I	299D 106X0003A	Capacitor_Polari	299D 106X0003A	Capacitor_Polarize	VishayCapacitor.SVNDbLib	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Return Selected to Default Choose Component...

Parameters Changes...

Cancel < Back Next > Finish

Select Parameter Changes

10 Objects - None Selected

Object Type	Document	Identifier	C...	C...	D...	L...	C...	C...	C...	C...	C...	Datasheet/Document	Designator	D...	Note	Package/Reference	P...	P...	Pin Count	Published
Part	Protection.SchDoc	C2	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0020FBP	Nor	Part Numbers show F	94S	Cor	2	19-Mar-2004	
Part	Radial.SchDoc	C4	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Radial.SchDoc	C5	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Radial.SchDoc	C6	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Radial.SchDoc	C8	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Shaping.SchDoc	C9	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Shaping.SchDoc	C11	Cap.	Star	Resi	Cap.	Sta	Ma	htp	Dat	htz	28-Mar-2002	299D 106X0003A	See	Data Sheet for	29S	3		19-Mar-2004	
Part	Vertical.SchDoc	C12	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Vertical.SchDoc	C13	Cap.	Star	Solk	Cap.	Sta	Ma	htp	Dat	htz	29-Jun-2001	94SA107X0010EBP	Nor	Part Numbers show E	94S	Cor	2	19-Mar-2004	
Part	Vertical.SchDoc	C14	Cap.	Star	Resi	Cap.	Sta	Ma	htp	Dat	htz	28-Mar-2002	299D 106X0003A	See	Data Sheet for	29S	3		19-Mar-2004	

Update Selected Reject Selected Update All Reject All Show all parts OK

Figure 13. Controlling parameter-level updates on an individual component basis.

## Keeping Components Up-To-Date

As with the **Update Parameters From Database** tool, the dialog will list all mapped parameters for those component instances linked to an external database. In addition, all parameters found in the placed component instances involved in the parameter update will also be listed. This is a key difference when accessing the *Select Parameter Changes* dialog between these two update tools.

Again, the dialog shows proposed changes to be made in order to bring the parameters of the placed components back into sync with those for the components in the relevant source library or database. The proposed updates are in accordance with the update actions defined in:

- the **Parameter Settings** region of the *Library Update Settings* dialog – when the source of the update is a schematic or integrated library
- the intermediary link file (\*.DBLink, \*.DBLib, \*.SVNDBLib) – when the source of the update is a database.

Parameter differences are highlighted by the use of a unique icon inserted in the relevant cell. See the next section for a full listing of all possible cell states.

Use the available controls in the dialog (buttons or right-click menu) to determine which parameter changes to proceed with and which to reject, giving you the power to override the default update conditions at the individual parameter-level. Remember that rejecting a proposed update in a cell simply means you do not wish to have any changes made to that parameter for that particular component instance on the schematic sheet.

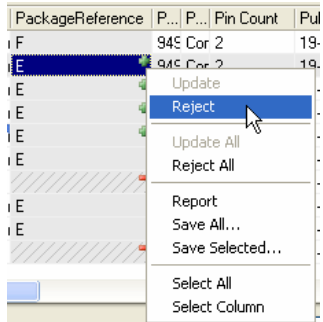










Figure 14. Rejecting a proposed parameter update.

### Parameter Update States

There are a number of different update states into which a cell in the *Select Parameter Changes* dialog can be placed, depending on the exact differences detected between the placed components and the source library/database, and the applicable parameter update actions specified. **Error! Reference source not found.** lists the various possible cell states.

Table 1. Parameter cell states.

Cell State	Description
Manufacturer Link	the Altium Designer component and the component in the library/database both possess the parameter and the value is the same. Update will cause no change.
	the Altium Designer component and the component in the library/database both possess the parameter, but it has no value assigned. Update will cause no change.
	the Altium Designer component and the component in the library/database do not possess the parameter. Update will cause no change.  This state can also appear if the parameter in the Altium Component is prevented from being included in an update – by disabling the <b>Allow Synchronization With Library</b> and/or <b>Allow Synchronization With Database</b> options, in the associated <i>Parameter Properties</i> dialog, respectively. See <a href="#">Preventing Parameter Updates</a> .

	the Altium Designer component and the component in the library/database both possess the parameter, but with different values. Update will assign the value from the database.
	the Altium Designer component and the component in the library/database both possess the parameter. For the Altium Designer component it has a value assigned, in the database it does not. Update will assign the database value, in this case no value.
	the Altium Designer component does not possess the parameter, but the component in the library/database does. Update will add the parameter to the Altium Designer component with the value shown in the cell.
	the Altium Designer component does not possess the parameter, but the component in the library/database does. Update will add the parameter to the Altium Designer component with no value initially assigned.
	the Altium Designer component possesses the parameter, but the component in the library/database does not. Update will remove the parameter from the Altium Designer component.  <b>Note:</b> If the component instance is being updated from an external database and the relevant database field-to-design parameter mapping is set to [None], the parameter will be treated as though it does not exist in the database record, even if it does.
	there is a difference between the parameter values for the Altium Designer component and the component in the library/database. The proposal to update to the value in the database has been rejected. No change will occur.
	the Altium Designer component does not possess the parameter, but the component in the library/database does. The proposed update to add the parameter to the Altium Designer component has been rejected. No change will occur.
	the Altium Designer component possesses the parameter, but the component in the library/database does not. The proposed update to remove the parameter from the Altium Designer component has been rejected. No change will occur.

## Preventing Parameter Updates

Whether a design parameter gets updated or not can also be controlled on an individual parameter basis, directly from within the design. Two options are available from the associated *Parameter Properties* dialog for a parameter – **Allow Synchronization With Library** and **Allow Synchronization With Database** respectively (Figure 15). By default, these are enabled, to always allow synchronization with the source library/database. Disable an option to prevent that parameter from being included in an update process.

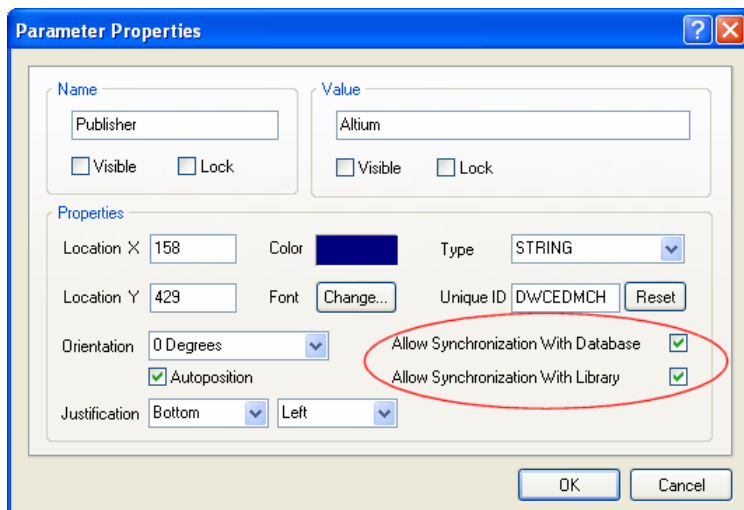


Figure 15. Control a parameter's inclusion in an update process.

## Updating PCB Footprints from Libraries

The **Update From Libraries** command in the previous section is solely used to update placed components on schematic sheets. Although models can be included as part of that update, it is the model linking that is being considered, not the actual graphical attributes of the linked model. To pass on changes to the graphical attributes of a PCB footprint, from the source PCB Footprint library, requires the use of another tool from Altium Designer's update arsenal – **Update From PCB Libraries**.

This update tool checks the component footprints on a board against the corresponding footprints in source PCB Footprint libraries, by performing a physical comparison of each footprint's primitives.

To illustrate the use of this tool, consider the component footprints in Figure 16. These are existing footprints on a PCB document. To the right are the modified footprints that exist in the source PCB Footprint library (\*.PcbLib). The changes, although subtle – the top footprint having the '+' sign moved; the bottom having increased width for the '+' sign and arc – are sufficient for the purposes of our simple illustration.

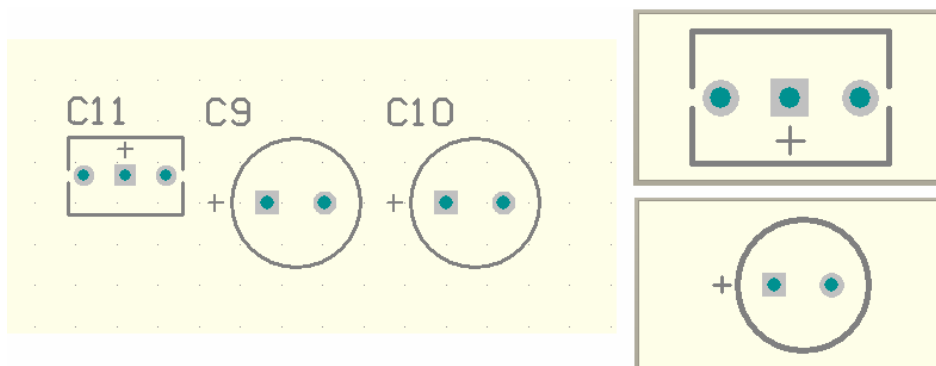


Figure 16. Initial footprints (left) and modified library versions (right).

## Defining Update Options

Updates are performed from the PCB document using the **Update From PCB Libraries** command, available from the main **Tools** menu. Launching this command will bring up the *Update From PCB Libraries - Options* dialog (Figure 17).

Use this dialog to determine the following options to be used by the comparison tool:

- The measurement unit used when performing the physical comparison of the footprint primitives. Imperial units are used by default. Note also that this setting does not follow the units defined for the PCB document itself.
- The tolerance to be applied. This is the maximum permissible distance discrepancy allowed when comparing locations of primitives. If the difference in a primitive's location is greater than this tolerance, the footprints are considered not to match, and are candidates for update. By default, this is set to 0.001 mil.
- Which layers to include in the comparison. By default, all are enabled.

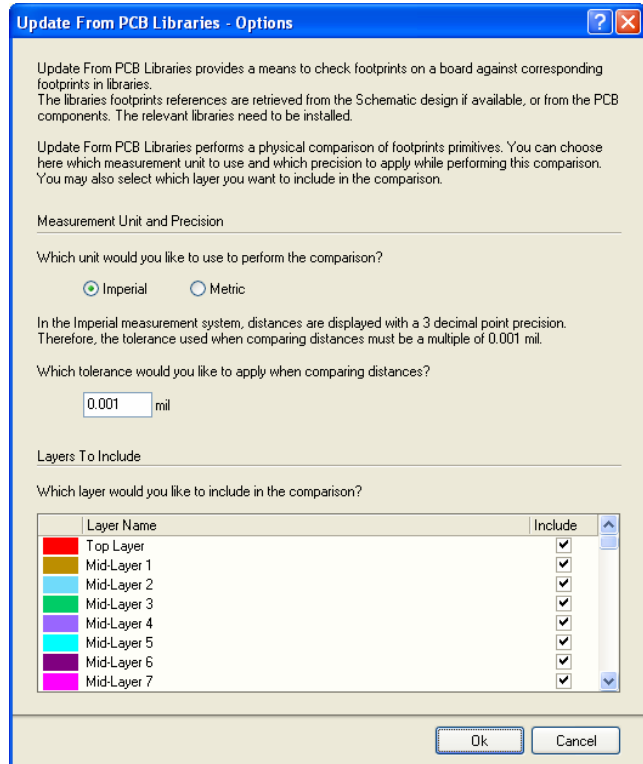


Figure 17. Setting footprint comparison options.

## Viewing Comparison Results

After defining the options in the *Update From PCB Libraries - Options* dialog as required, click **OK** – the comparison will proceed, with the results appearing in the subsequent *Update From PCB Libraries* dialog (Figure 18).

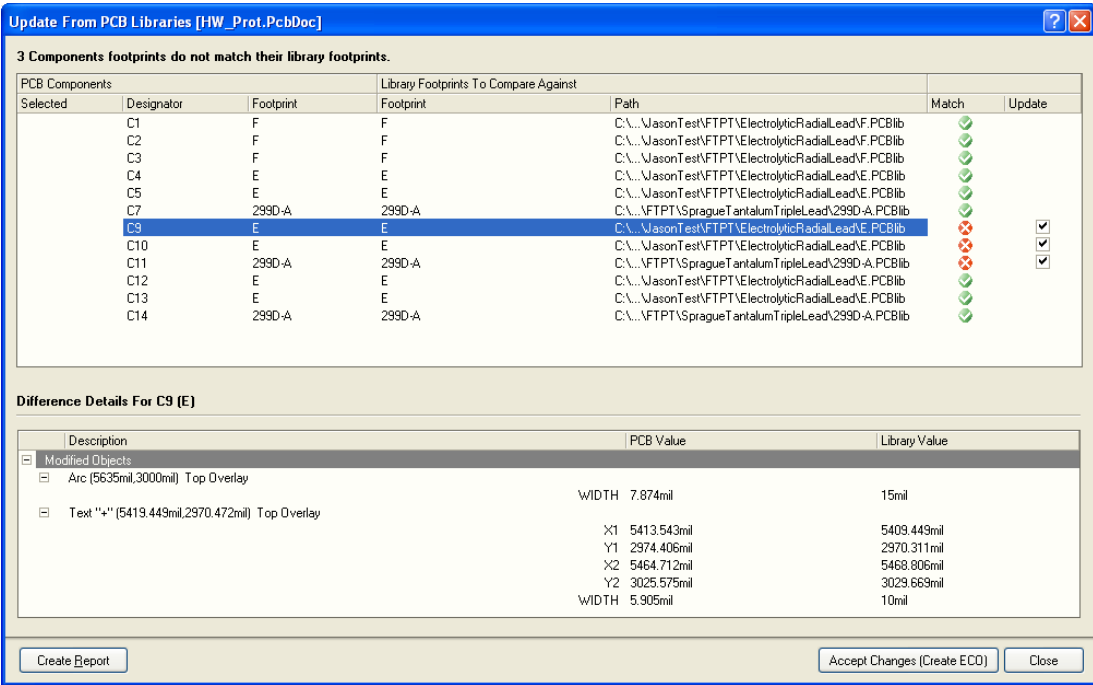


Figure 18. Reviewing the results of the footprint comparison.

The dialog is divided into two main regions. The upper region presents the comparison results. The footprint of each component on the PCB is compared against the corresponding footprint in the indicated source library. If the comparison of each primitive in the footprint is within the tolerance specified, the footprints will be deemed to match, with a green tick icon (✓) appearing in the **Match** column. No update is required in the case of a match.

If, however, one or more primitives in a footprint are outside of the tolerance – in terms of their locations within the footprint – then the footprints are flagged as not matching. A red cross icon (✗) will be inserted into the **Match** column. An update is required to bring the placed component's footprint and source library footprint back into sync.

There will be an associated **Update** option for each component with a non-matching footprint. By default, each component that fails to match is automatically included in the update. Control, as is the norm in Altium Designer, is always yours, and you may choose to exclude any non-matching components from the update process, should you wish.

The lower region of the dialog lists the differences found between primitives for the currently selected (and non-matching) component entry in the region above.

## Generating a Comparison Report

Should you wish to create a Footprint Comparison Report (Footprint Comparison - PCBDocumentName.html), simply click the **Create Report** button, at the bottom left of the *Update From PCB Libraries* dialog. The report provides:

The source PCB library is identified from the PCB model link associated to the corresponding component in the source schematic. The source libraries must be part of the Available Libraries.



- A list of unmatched components
- A list of matched components
- Primitive difference details for each component in the unmatched components list.

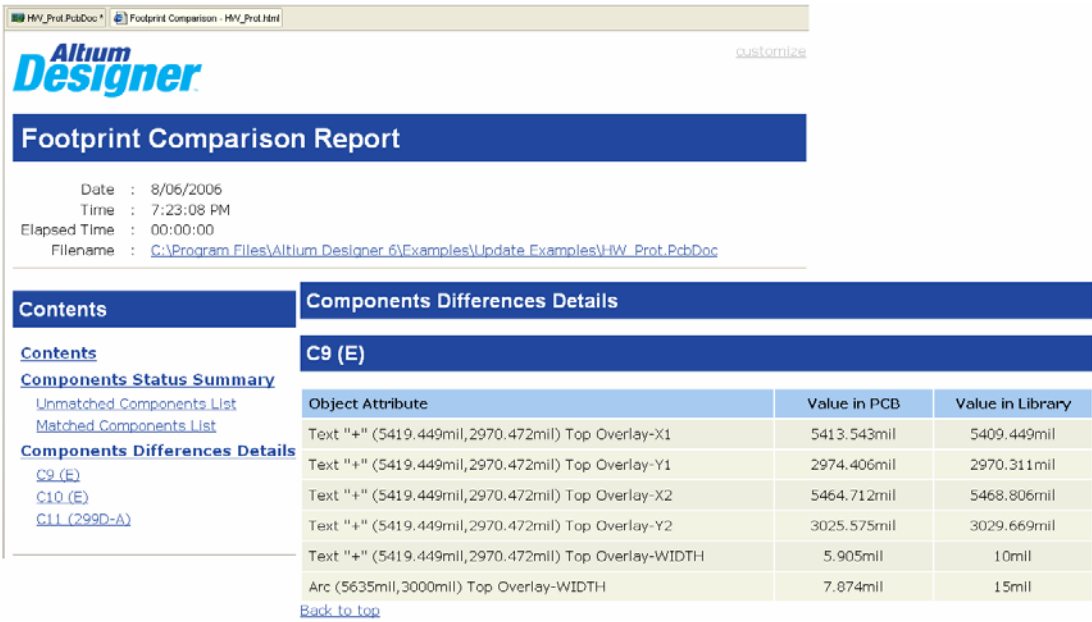



Figure 19. Browsing footprint differences in a generated report.

 Generating a comparison report will close the *Update From PCB Libraries* dialog. You will have to launch the command again.

## Proceeding with the Update

When you are satisfied with the update solution, click the **Accept Changes (Create ECO)** button. Use the *Engineering Change Order* dialog that appears to validate and then execute the updates accordingly. Disable any change that you do not wish to proceed with. The changes will be made and the components included in the update will have their footprints updated to those in the source libraries (Figure 20).

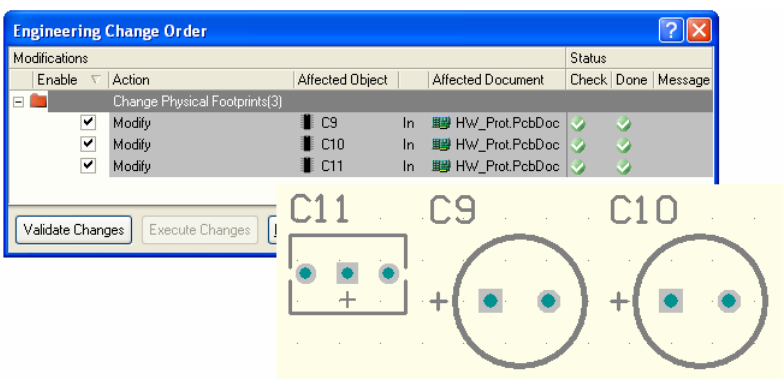


Figure 20. Finalizing the update through execution of an ECO.

## Updating Directly from a Source Library Document

Direct update of placed schematic components and PCB footprints can be performed from within a source schematic or PCB library respectively. This method of updating is particularly suited to the update of components that have been placed from isolated schematic component (\*.SchLib) or PCB Footprint (\*.PcbLib) libraries and where those libraries are made part of the Available Libraries for the project.

### Applying Changes from the Schematic Library Editor

Modifications made to components in a source schematic library are passed to the schematic document(s) using the **Update Schematics** command, available from the Schematic Library Editor's main **Tools** menu. Alternatively, changes to selected components can be applied by selecting those components in the **SCH Library** panel and using the update command from the right-click menu (Figure 21).

All instances of the placed component(s), in all open schematic documents, will be updated, regardless of the project to which a document belongs. Essentially, if a placed component has the same entry for its **Physical Component** field as that of the library component, it will be updated. You will be notified as to how many components (and in how many documents) have been modified by the update process.

The update is a full replacement of each existing component instance with that in the library.

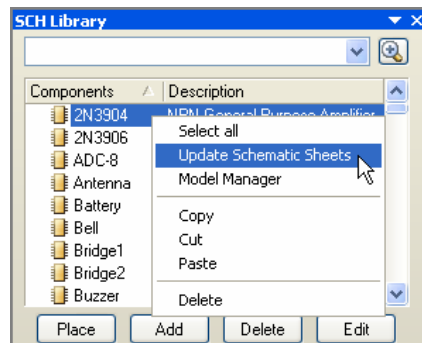


Figure 21. Updating a selected schematic component.



If a component instance on a schematic has been placed from an integrated library (IntLib) or database library (DBLib, SVNDBLib), using this command fully replaces the component and therefore the original library link will be replaced by a link back to the source library involved in the update. This can be especially dangerous for a component placed from a database, as the link to the corresponding record in that database will be severed. It is therefore advisable (and perhaps safer) to use the [Update From Libraries](#) command.

### Applying Changes from the PCB Library Editor

In much the same way, changes made to one or more component footprints in a source PCB library can be passed directly to placed instances of those footprints on the PCB. The main **Tools** menu in the PCB Library Editor offers commands for updating just the current footprint (**Update PCB With Current Footprint**), or all footprints (**Update PCB With All Footprints**). These commands can also be accessed from the **PCB Library** panel's right-click menu (Figure 22).

All instance of the placed component footprint(s), in all open PCB documents, will be updated, regardless of parent project. Essentially, if a placed component footprint has the same name as that of the library component, it will be updated.

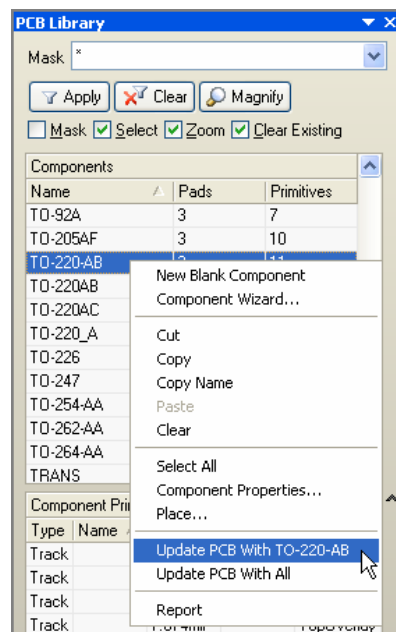


Figure 22. Updating a selected PCB component footprint.

## Verifying an Update

After updating the placed schematic components, schematic library components, or PCB Footprints, you will of course want to verify that the update proceeded as planned. The following sections look at how you can quickly check that parameters, model links, and graphical attributes have updated successfully.

### Checking Parameter Updates

You can check that the parameters have been updated correctly either by running the *Parameter Manager* (**Tools » Parameter Manager**), or by interrogating the parameters for a component either in its associated properties dialog, or in the relevant inspector-based panel for the editor you are in (**SCH Inspector** or **SCHLIB Inspector**).

In Figure 23, the component C1 has had the value of its *DatasheetDocument* parameter changed to 12-Nov-2005 as part of an update process.

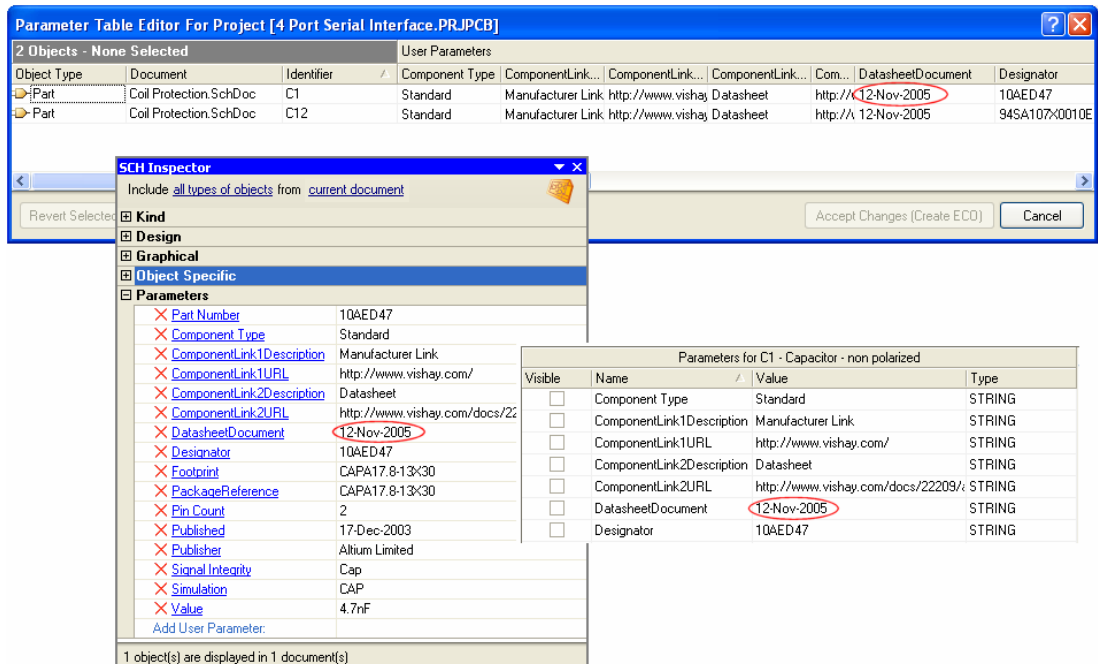


Figure 23. Verifying parameter updates.

### Checking Model Link Updates

Updates to model links for a component can be verified from the **Models** region of the associated *Component Properties* dialog. In Figure 24, component Q1 is updated with an additional footprint model (TO-220-AB) that has been added at some stage to the corresponding component in the source schematic component library. This model has also been set as the current model out of the two footprint models linked to the component.

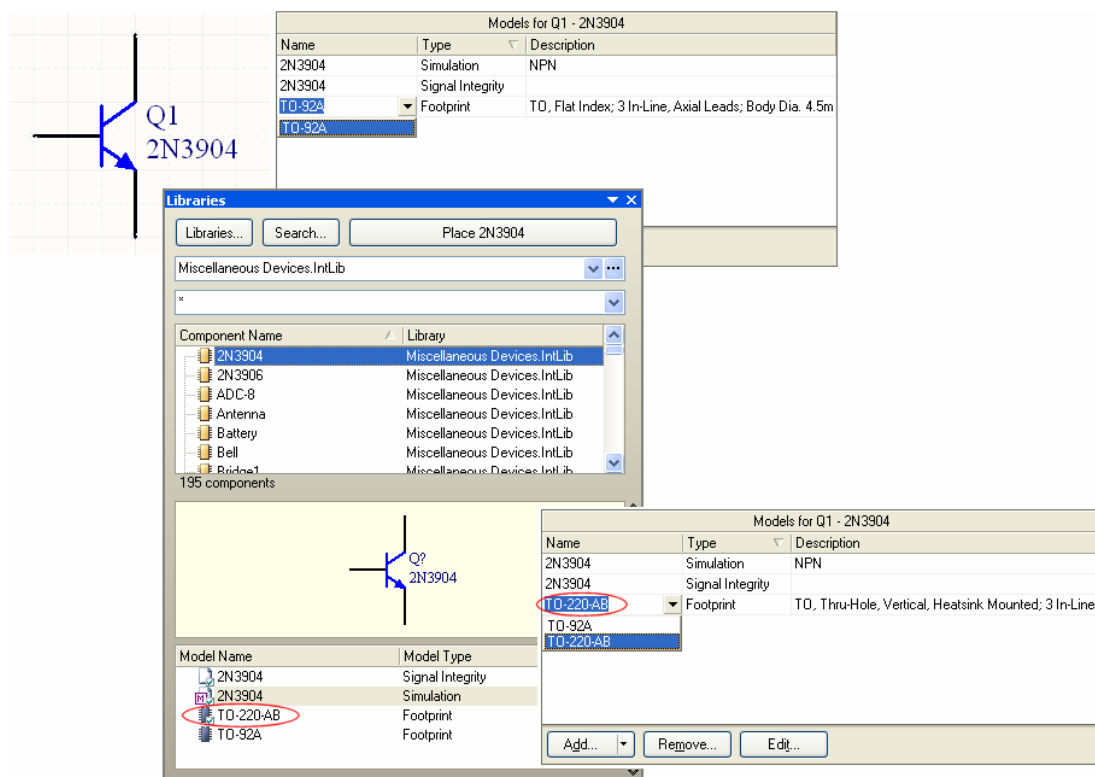


Figure 24. Verifying model updates.

## Checking Graphical Updates

The simplest way to check graphical updates is to compare the design component on the sheet against the component in the source library. The latter can easily be displayed using the **Libraries** panel, allowing direct side-by-side comparison. Figure 25 illustrates a change to the symbol graphics of a placed component.

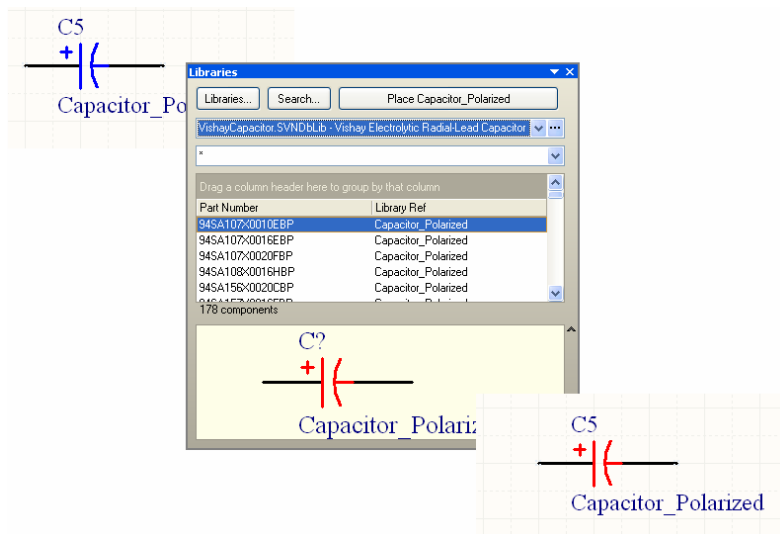


Figure 25. Verifying graphical updates.

## Revision History

---

Date	Version No.	Revision
09-Jun-2006	1.0	Initial release

Software, hardware, documentation and related materials:

Copyright © 2006 Altium Limited.

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