Creating Library Components

The tutorial was created for releases prior to Altium Designer 10. Path information, Footprints, Models may not be included with Altium Designer 10. It is possible to Download Examples and Reference Designs and to Download Libraries from the previous versions from our Wiki.

Contents

Altium

This tutorial covers the creation of schematic components and PCB footprints, including adding 3D body objects, using the Schematic and PCB Library Editors in Altium Designer. A glossary of terms used in this tutorial is included as an appendix.

In this tutorial, we will cover the following topics:

- creating new libraries
- creating schematic components with single and multiple parts
- checking the components using Schematic Library Editor reports
- creating PCB component footprints manually and using the **PCB Component Wizard**
- handling other special footprint requirements, including irregular pad shapes
- including three-dimensional component detail (3D bodies)
- checking the component footprints using PCB Library Editor reports
- creating an integrated library of the new components and models.

This tutorial presumes you have a working understanding of the Schematic and PCB Editor environments and are familiar with placing and editing components. The example components and libraries used in this tutorial are available in the *Creating Components* folder of your Altium Designer installation.

Schematic Libraries, Models and Integrated Libraries

Schematic component symbols are created in schematic libraries (*.SchLib). The components in these libraries then reference footprints and other models defined in separate footprint libraries and model files. As a designer, you can place components from these discrete component libraries or you can compile the symbol libraries, footprint libraries and model files into integrated libraries (*.IntLib).

The advantages of Integrated libraries are that they are portable (everything is in one file) and the components and models in them cannot be edited. The bulk of Altium Designer components (around 70,000 ISO compliant components) are supplied in integrated libraries, which you will find in the *Library* folder of your Altium Designer installation. You can extract the source libraries out of an integrated library, to do this open the integrated library and choose **Extract Sources** to extract the source libraries, which will then be opened for editing. For more information, refer to the Building an Integrated Library tutorial.
You can also create a schematic library of all the components that have been placed in the schematic documents of the active project by using the **Design » Make Schematic Library** command.

Creating Schematic Components

The Schematic Library Editor is used to create and modify schematic components, and manage component libraries. It is similar to the Schematic Editor and shares the same graphical design objects, with the addition of the Pin tool.
Components are created with the design objects in the Schematic Library Editor. Components can be copied and pasted from one schematic library to another or from the schematic editor to the schematic library editor.

Creating a New Library Package and Schematic Library

Before we start creating components, we need a new schematic library to store them in. This library could be created as a stand-alone library, referencing models in separate files. An alternate approach is to create the new schematic library with the intention of compiling it and the referenced models into an integrated library package. This means that before we create the library we need to create a new library package. A library package (.LibPkg) is the basis of an integrated library - it binds together the separate schematic libraries, footprint libraries and model files that are ultimately compiled into the single integrated library file.

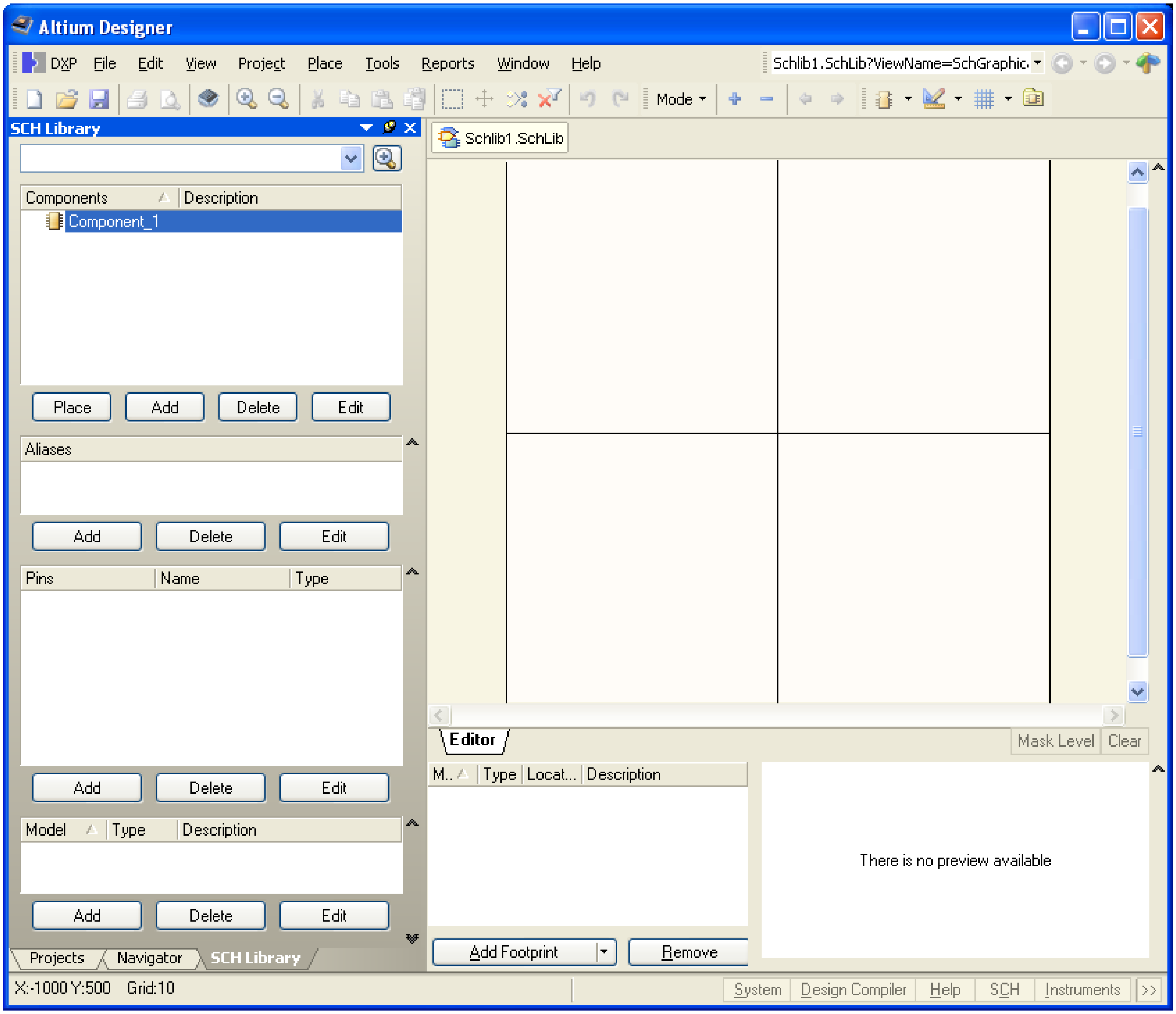


Figure 1. The new library, open at the default Component_1.

To create a new integrated library package and an empty schematic library, complete the following steps.
1. Select *File » New* **» Project » Integrated Library** . A new library package is created, named *Integrated_Library 1.LibPkg* . This is displayed in the **Projects** panel.
2. Right-click on the library package name in the **Projects** panel and select **Save Project As** from the floating context menu. Browse to a suitable location, type in the name New Library.LibPkg, and click the **Save** button. Note that the extension will be added automatically if you do not type it in.
3.To add an empty schematic library select **File » New » Library » Schematic Library** . A new library, named Schlib1.SchLib, is created and an empty component sheet, Component_1, displays in the design window.
4.Select **File » Save As** and save the library as Schematic Components.SchLib.
5. Click on the **SCH Library** tab to open the **SCH Library** panel.

Creating a New Schematic Component

To create a new schematic component in an existing library, you would normally select **Tools » New Component .** However, since a new library always contains one empty component sheet , we'll simply rename Component_1 to get started on creating our first component, an NPN transistor.

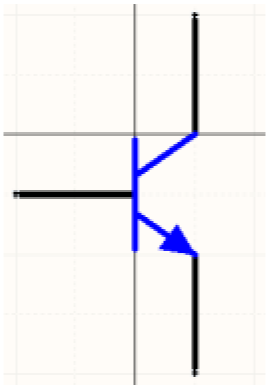
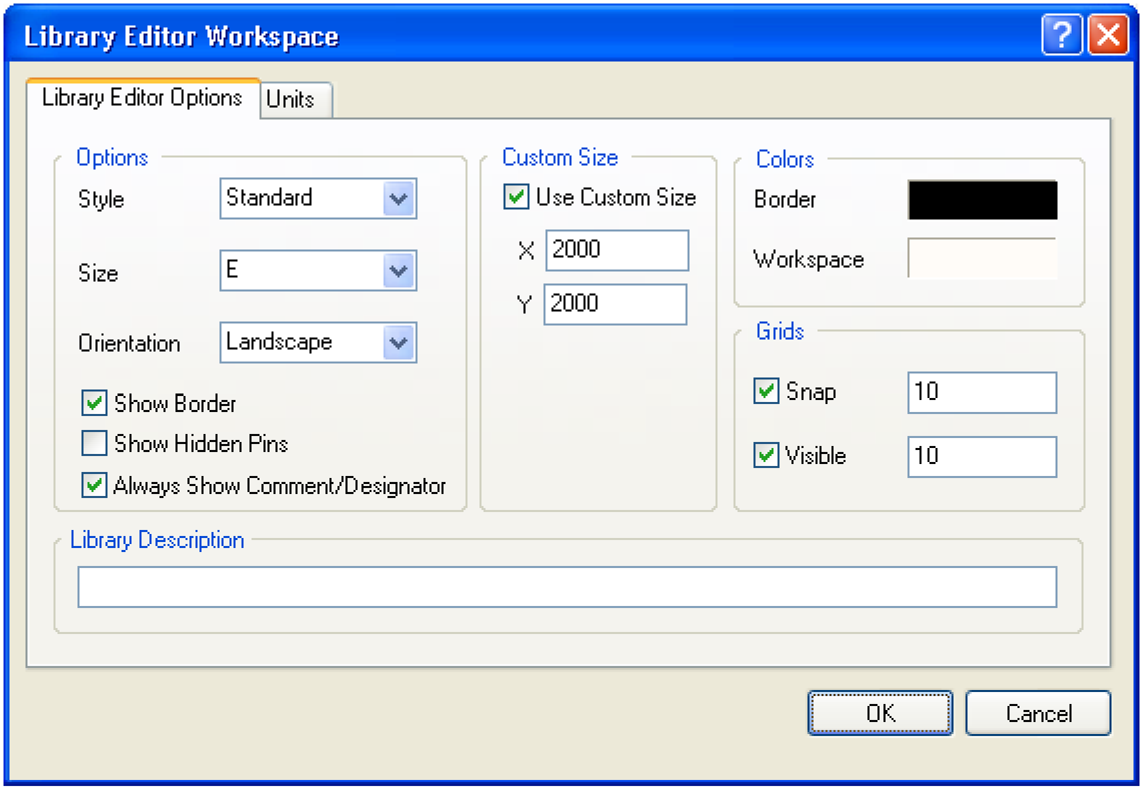


Figure 2. Symbol for an NPN transistor.

1. Select *Component_1* from the Components list in the **SCH Library** panel and select **Tools » Rename Component** . Type the new component name that uniquely identifies it, e.g. NPN, in the *Rename Component* dialog and click **OK** .
2. If necessary, relocate the origin of the sheet to the center of the design window by selecting **Edit » Jump » Origi n** [shortcut: **J** , **O**]. Check the Status line at the bottom left of the screen to confirm that you have the cursor at the origin. Components supplied by Altium are created around this point, marked with a crosshair through the center of

the sheet. You should always create your components close to this origin. When you place a component on the schematic, the component will be 'held' by the electrical hot spot (pin end) that is nearest this origin.



Library Editor Workspace

Library Editor Options | Units

Options

Style: Standard

Size: E

Orientation: Landscape

☑ Show Border

☐ Show Hidden Pins

☑ Always Show Comment/Designator

Custom Size

☑ Use Custom Size

X: 2000

Y: 2000

Colors

Border: ■

Workspace: □

Grids

☑ Snap: 10

☑ Visible: 10

Library Description

Figure 3. Set the units and other sheet properties in the Library Editor Workspace dialog.

3. The units, and snap and visible grids can be set in the *Library Editor Workspace* dialog (**Tools » Document Options** , [shortcut **T** , **D**]).
You can enable the Always Show Comment/Designator option from the Library Editor Workspace dialog to display the Comment/Designator strings for the current component in your library document.
Rather than opening the *Library Editor Workspace* dialog whenever you need to change the grid, simply press G on the keyboard to quickly cycle through and set the Snap Grid to 1, 5 or 10 units.

Rather than opening the Library Editor Workspace dialog whenever you need to change the grid, simply press G on the keyboard to quickly cycle through and set the Snap Grid to 1, 5 or 10 units. These three settings can be configured in the Schematic - Grids page of the Preferences dialog.

You can enable the Always Show Comment/Designator option from the Library Editor Workspace dialog to display the Comment/Designator strings for the current component in your library document.

These three settings can be configured in the Schematic - Grids page of the *Preferences* dialog.
Set the Library Editor Workspace options as per Figure 3. Then click on the **Units** tab and enable **Imperial Units** , using **DXP Defaults** as the units. Click **OK** to close the dialog. If the schematic library editor grid is not visible when the dialog is closed, press **PAGE UP** to zoom in until it is visible. Note that zooming occurs around the cursor, so keep the cursor close to the origin as you zoom in.

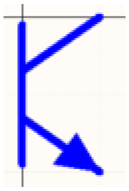
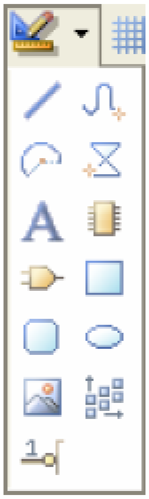


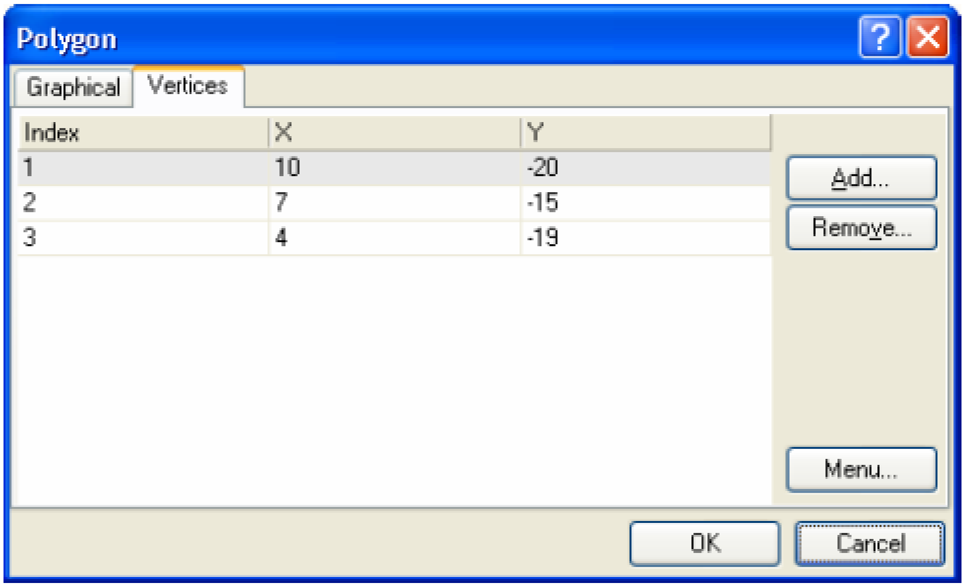
Figure 4. NPN body

4. To create the NPN transistor we will first define the component body. Select **Place » Line** [shortcut: **P** , **L**], or click on the **Place Line** toolbar button (**Utilities** toolbar).
If required, you can press the **TAB** key to set the line properties in the *PolyLine* dialog. Using Figure 4 as a guide (use the grid lines to help you), place the vertical line.
Click once to define the first end of the line, move the mouse to the location of the other end and click to define it, then right-click or press **ESC** to end placement of this line. Note that you are still in line placement mode, as indicated by the crosshair on the cursor.
5. Now create the other two lines. For this transistor they are placed on a non-regular angle, when you start placing the line you may find that the line is constrained to horizontal/vertical, or 45°. Press **SHIFT** + **SPACEBAR** while you are placing a line to cycle through the different placement modes. One of the modes is any angle, this mode will allow you to define the lines correctly. After defining these two lines you will need to press **ESC** once more to drop out of line placement mode.
The exact location of graphical lines is not critical. What is critical in component design is the pin location, or more specifically what is referred to as the *hot* end of the pin. This is the point that creates the electrical connectivity - so it is the pins you should always place on a grid that is suitable for wiring.
6. The small arrow head is created out of a closed polygon. Select **Place » Polygon** [shortcut: **P** , **Y**] or click on the **Place Polygon** toolbar button (**Utilities** toolbar). Before you start placing the polygon press the **TAB** key to set the polygon properties in the *Polygon* dialog. Set the **Border Width** to **Smallest ,** enable **Draw Solid** and set the fill and border colors to the same color (basic color 229), then click **OK** to close the dialog. Click to define each vertex of the triangle and right-click to end. Right-click or press **ESC** to end polygon placement mode. Figure 5 shows the coordinates of the polygon vertices.



Utilities toolbar

> The exact location of graphical lines is not critical. What is critical in component design is the pin location, or more specifically what is referred to as the hot end of the pin. This is the point that creates the electrical connectivity - so it is the pins you should always place on a grid that is suitable for wiring.



Index	X	Y
1	10	-20
2	7	-15
3	4	-19

Figure 5. Use the coordinate information to confirm that your arrow head is correct.

7. Save the component [shortcut: **CTRL** + **S**].

Adding Pins to the Schematic Component

Component pins give a component its electrical properties and define connection points on the component. They also have graphical properties.
To place pins on the component:
1. Select **Place » Pin** [shortcut: **P** , **P**] or click on the toolbar button. The pin appears floating on the cursor, held by the electrical end that must be placed away from the component body, also referred to as the *hot* end.
2. Before placing the pin, press the **TAB** key during placement to edit the pin's properties. The *Pin Properties* dialog (Figure 8) displays. If you define the pin attributes before you place it, the settings you define will become the defaults and the pin numbers and any numeric pin names will auto-increment when you place them.
3. In the *Pin Properties* dialog, type in a pin name in the **Display Name** field (1 for the first NPN pin), and a unique pin number (also 1) in the **Designator** field. Enable the **Visible** options if you want the pin name and designator visible when you place the component on a schematic sheet.
4. Set the **Electrical Type** of the pin from the drop-down list. This type is used when compiling a project or analyzing a schematic document to detect electrical connection errors in a schematic sheet. In this component example, all pins are have their **Electrical Type** set to **Passive** .
5. Set the length of this pin (all pins in this component are set to 20), and click **OK** .
6. Press the **SPACEBAR** to rotate the pin in 90º increments while it is floating on the cursor. Remember that only

one end of a pin is electrical (referred to as the *hot end*) and you must place the pin with this end out from the component body. The non-electrical end of the pin has the pin name next to it.

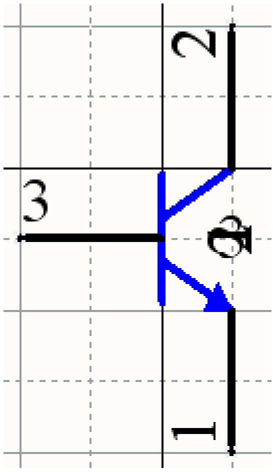
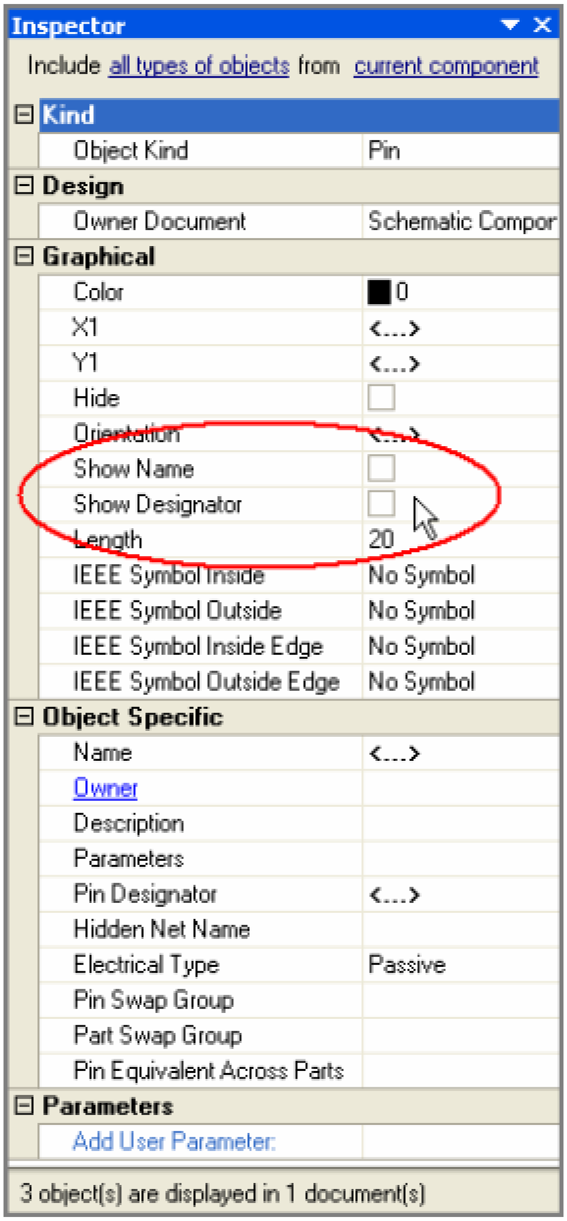


Figure 6. NPN symbol with pin name and number visible.

7. Continue to add the pins required to finish the component, making sure the pin names, numbers, symbols and electrical types are correct as shown on the NPN symbol shown in Figure 6.



Inspector

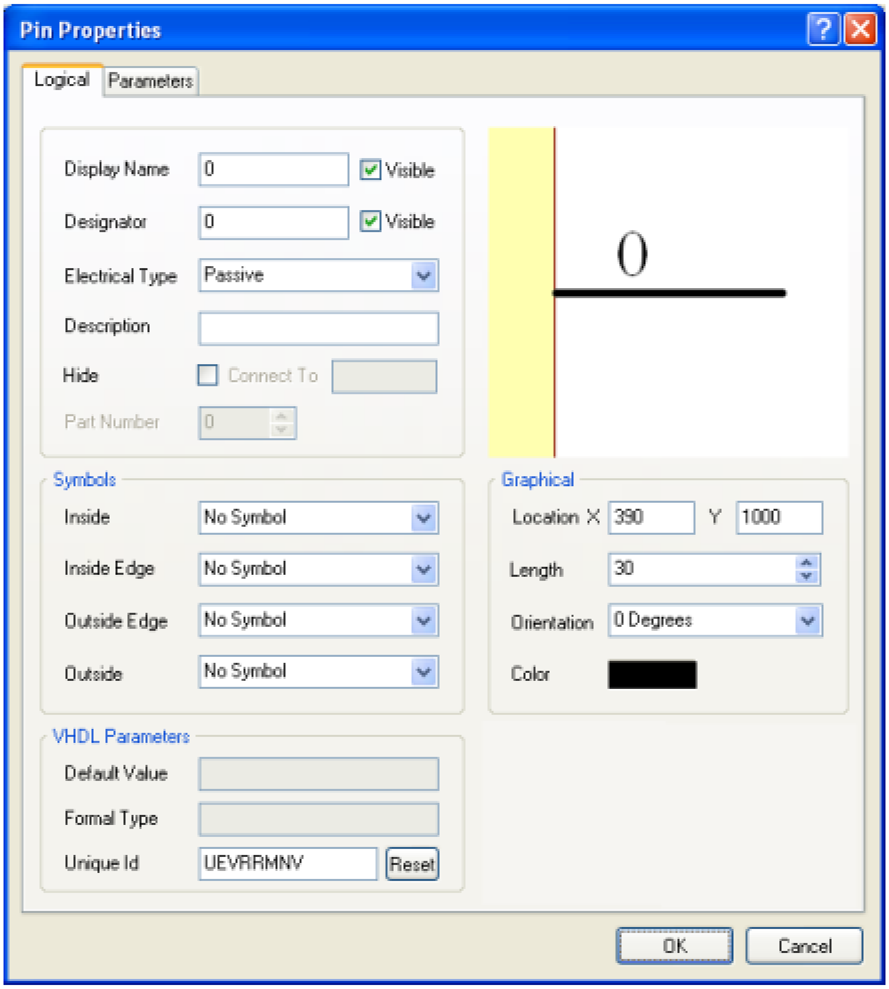
Include all types of objects from current component

⊟ **Kind**	
Object Kind	Pin
⊟ **Design**	
Owner Document	Schematic Compor
⊟ **Graphical**	
Color	■ 0
X1	<...>
Y1	<...>
Hide	☐
Orientation	<...>
Show Name	☐
Show Designator	☐
Length	20
IEEE Symbol Inside	No Symbol
IEEE Symbol Outside	No Symbol
IEEE Symbol Inside Edge	No Symbol
IEEE Symbol Outside Edge	No Symbol
⊟ **Object Specific**	
Name	<...>
Owner	
Description	
Parameters	
Pin Designator	<...>
Hidden Net Name	
Electrical Type	Passive
Pin Swap Group	
Part Swap Group	
Pin Equivalent Across Parts	
⊟ **Parameters**	
Add User Parameter:	

3 object(s) are displayed in 1 document(s)

Figure 7. Editing the visibility of the pin name and number for all 3 pins.

If you wish to alter the distance (in hundredths of an inch) between the pin name or number and the body of the component, select **Tools » Schematic Preferences** and change the Pin Margin options in the Schematic — General page.



Pin Properties

Logical | Parameters

Display Name: 0 ☑ Visible

Designator: 0 ☑ Visible

Electrical Type: Passive

Description:

Hide □ Connect To

Part Number: 0

Symbols

Inside: No Symbol

Inside Edge: No Symbol

Outside Edge: No Symbol

Outside: No Symbol

Graphical

Location X: 390 Y: 1000

Length: 30

Orientation: 0 Degrees

Color: ■

VHDL Parameters

Default Value:

Formal Type:

Unique Id: UEVRRMNV Reset

Figure 8. Set the properties of the Pin before placing it.

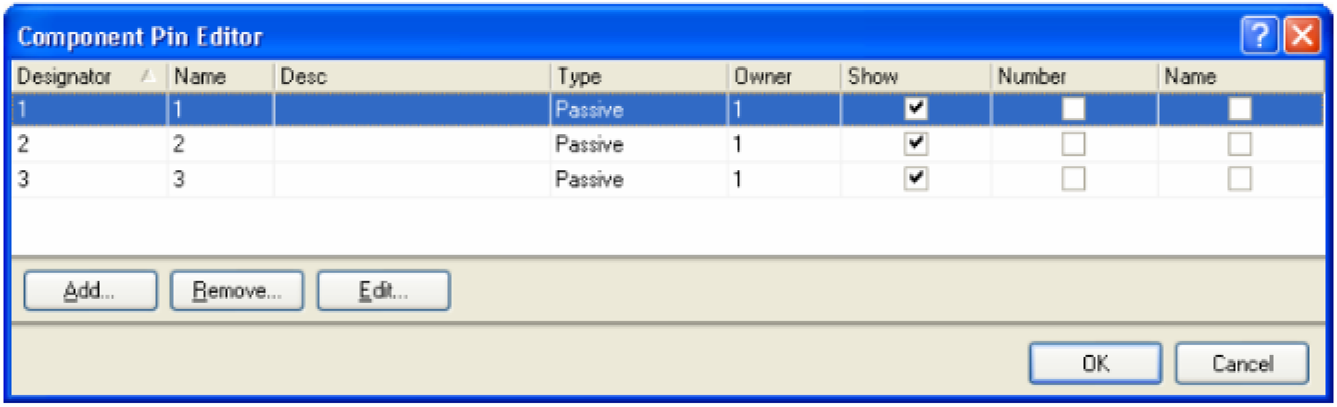
If you wish to alter the distance (in hundredths of an inch) between the pin name or number and the body of the component, select **Tools » Schematic Preferences** and change the Pin Margin options in the Schematic — General page.

8. If you have placed all the pins with the Name and Number visible, you can easily change the display state of all in a single editing action. To do this select *just* the 3 pins (**SHIFT** + Click on each pin), press **F11** to display the **Inspector** panel, and disable the **Show Name** and **Show Designator** options, as shown in Figure 7.
9. You have now finished drawing your component, select **File » Save** to save it.

Notes on Adding Pins

- To set pin properties after placing the pin, double-click on the pin, or double-click on the pin in the **Pins** list in the **SCH Library** panel to open the *Pin Properties* dialog. Alternatively, edit multiple pins in the **Inspector** , as described above.
- Use a ** (backslash) after a letter to define an overscored letter in a pin name, e.g. M\C\L\R\/VPP will display as .
- If you wish to hide the power and ground pins on a component, enable the **Hide** option. When they are hidden, these pins will be connected to power and ground nets as specified in the **Connect To** field, eg. the pin for VCC will connect to net VCC when placed.
- To view hidden pins or hidden pin names/numbers, select **View » Show Hidden Pins**.

- You can also edit pin properties directly in the *Component Pin Editor* dialog, without having to edit each pin through its corresponding *Pin Properties* dialog. Click on **Edit Pins** in the *Library Component Properties* dialog (double-click on a component name in the **SCH Library** panel to open this dialog) to display the *Component Pin Editor* dialog, as shown in Figure 9.



Designator	Name	Desc	Type	Owner	Show	Number	Name
1	1		Passive	1	☑	□	□
2	2		Passive	1	☑	□	□
3	3		Passive	1	☑	□	□

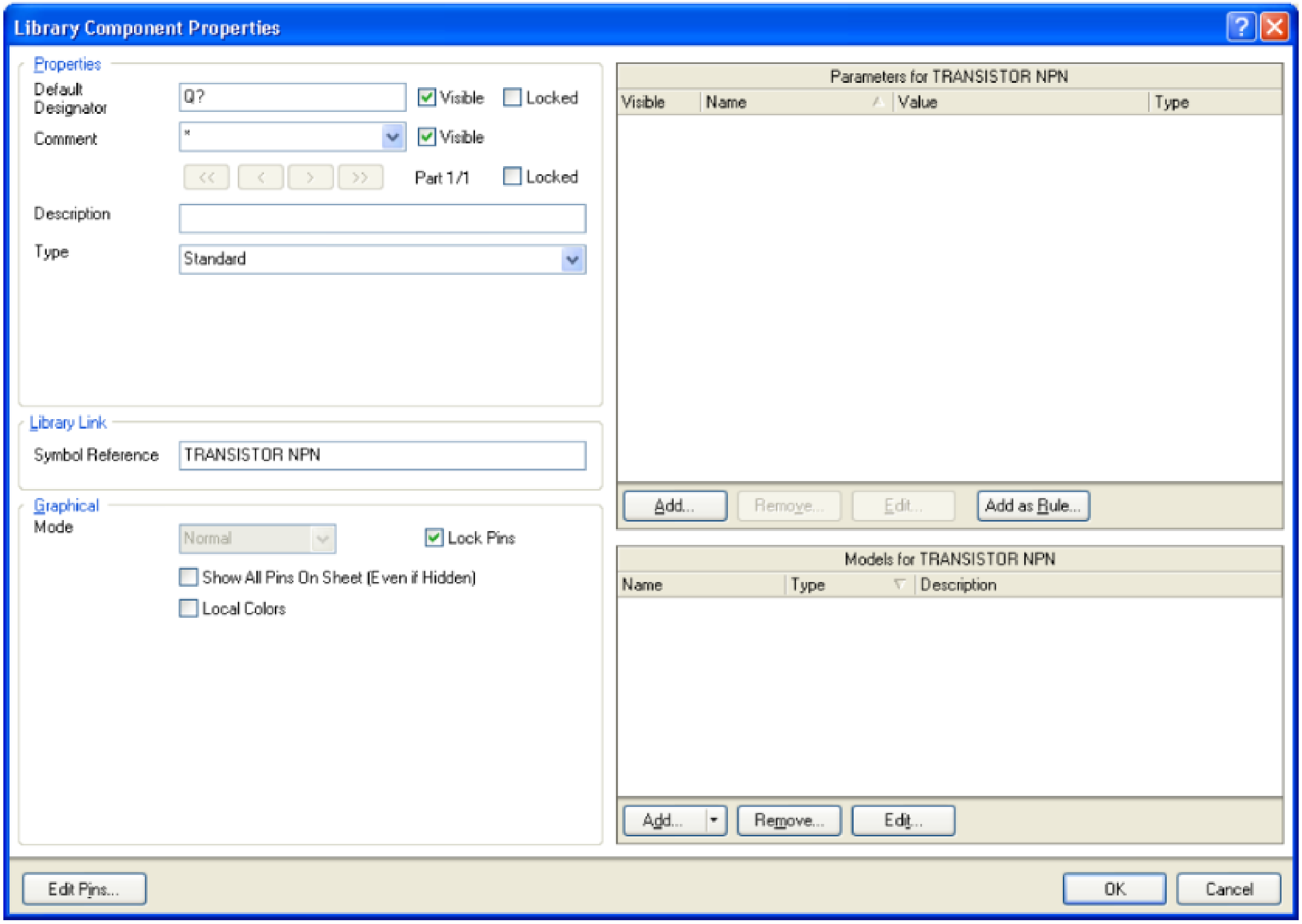
Figure 9. Review and edit all pins in the Component Pin Editor dialog.

- For a multi-part component, the relevant pins for the selected part will be highlighted with a white background in the *Component Pin Editor* dialog. All pins of other parts are grayed. You are, however, still able to edit the pins of these non-selected parts. Select a pin and click **Edit** to display the *Pin Properties* dialog for that pin.

Setting the Schematic Component's Properties

Each component has properties associated with it such as the default designator, the PCB footprint and/or other models, and any parameters that have been defined for the component. To set the component's properties:

1. Select the component in the Components list of the **SCH Library** panel and click the **Edit** button, or double-click on the component name. The *Library Component Properties* dialog (Figure 10) displays.



Library Component Properties

Properties

Default Designator: Q? ☑ Visible □ Locked

Comment: * ☑ Visible

Part 1/1 □ Locked

Description:

Type: Standard

Library Link

Symbol Reference: TRANSISTOR NPN

Graphical

Mode: Normal ☑ Lock Pins

□ Show All Pins On Sheet (Even if Hidden)

□ Local Colors

Visible	Name	Value	Type

Parameters for TRANSISTOR NPN

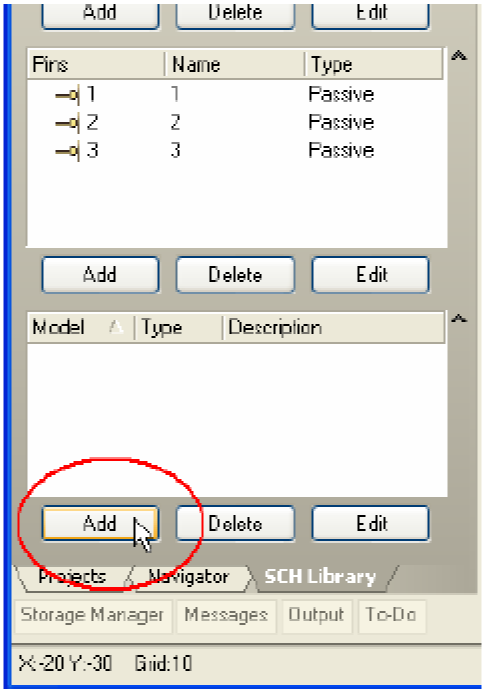
Name	Type	Description

Models for TRANSISTOR NPN

Figure 10. Basic component properties are defined in the Library Component dialog.

2. Type in the **Default Designator** , eg. Q? Including the question mark will allow the designator number to auto-increment on placement, e.g. Q1, Q2, if the designator is defined before placing the component (press **T AB** w hile placing to edit an object before placement).
3. Enter a **Comment** that will display when the component is placed on a schematic sheet, e.g. NPN. Make sure the **Visible** options for the **Designator** and **Comment** fields are enabled. If the **Comment** field is left blank it will automatically be populated with the Library Reference when the component is placed.
4.Enter a string into the **Description** field that describes the transistor, eg. Transistor, NPN Generic. This string is searched during a library search, and is displayed in the **Libraries** panel.
5.Leave the other fields at their default values while we add models and parameters, as required.

Adding Models to the Schematic Component



Pins	Name	Type
1	1	Passive
2	2	Passive
3	3	Passive

Figure 11. Add models via the Sch Library panel.

You can add any number of PCB footprint models to a schematic component, as well as model files that are used for circuit simulation and signal integrity analysis. If a component has multiple models, for example multiple footprints, you can select the appropriate model in the *Component Properties* dialog when you place the component on a schematic.

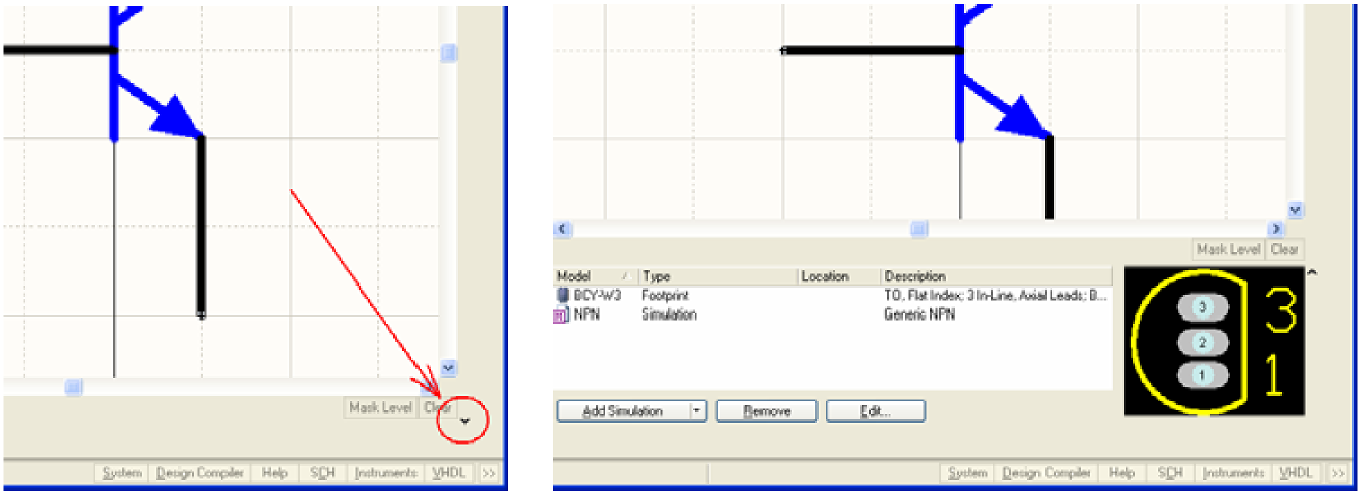
In terms of sourcing the models, you can create your own, use models from the existing Altium libraries, or download a vendor's model file from the web.

Supplied PCB footprint models are located in the *C:\Program Files_ _Altium Designer \Library\Pcb* folder in the form of PCB libraries (*.PcbLib files). PCB libraries can include any number of PCB footprints.

Wherever possible, SPICE models used for circuit simulation (.ckt and .mdl files) are included in the supplied integrated libraries in the *Library* folder of your Altium Designer installation. If you are creating a new component you would typically source the Spice model from the device vendor's website. You can also use the **XSpice Model Wizard** to create certain Spice model types to add to the component (**Tools » XSpice Model Wizard**).

The Schematic library editor's *Model Manager* dialog allows you to overview and organize your component models, for example you can add the same model to multiple, selected components. To access the *Model Manager* dialog, select **Tools » Model Manager** .

Alternatively, you can add models to the current component via the **SCH Library** panel by clicking on the **Add** button below the **Model** list, as shown in Figure 11, or from the **Model** region of the Schematic Library Editor workspace. Click the arrow located at the lower right side of the workspace to show the Models view, as shown in Figure 12.



Model	Type	Location	Description
BCY-W3	Footprint		TO, Flat Index; 3 In-Line, Axial Leads; B...
NPN	Simulation		Generic NPN

Figure 12. Click the arrow to display the Models view at the bottom of the workspace, then use the Add button to select a model-kind and add a new model.

1. Display the **Model** view section in the schematic library editor workspace by clicking the upside-down arrow/caret symbol as shown in Figure 12.

Search Locations for Model Files

When you add a model to a component in the Schematic Library Editor the model is linked, the model data is not copied and stored in the schematic component. This means the linked models must be available both during library creation, and when the component is placed on a schematic sheet.
When you are working in the library editor, the link from the component to the model information is resolved using the following valid search locations:
1.Libraries that are included in the current library package project are searched first.
2. PCB libraries (but not integrated libraries) that are available in the currently Installed Libraries list are searched next. **Note** : The list of libraries can be ordered.
3.Finally, any model libraries that are located down the Project search paths are searched. Search paths are defined in the *Options for Project* dialog (**Project » Project Options**). **Note**: Libraries that are down the search path cannot be browsed to locate a model, however, the compiler does include them when searching for a model.

> **Refer to the Components, Models and Library Concepts article for more information about the way models are searched for in the Schematic Library Editor and the Schematic Editor.**

In this tutorial we will use different methods of linking the components and its model files. When the library package is compiled to create the integrated library the various models are copied from their source locations into the final integrated library.

Adding Footprint Models to a Schematic Component

First, we will add the model that represents the component in the PCB Editor - the footprint (also known as a "pattern" or "decal" in other design tools). The footprint required for our schematic component is named BCY-W3.

Note : When linking a PCB footprint model to a schematic component in the Schematic Library Editor, the model must exist in a PCB library, not an integrated library.
1. Click the small arrow on the **Add** button in the **Model s** region of the *Library Component Properties* dialog, and select **Footprint** from the list, as shown in Figure 13.

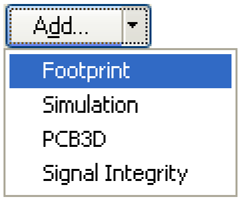
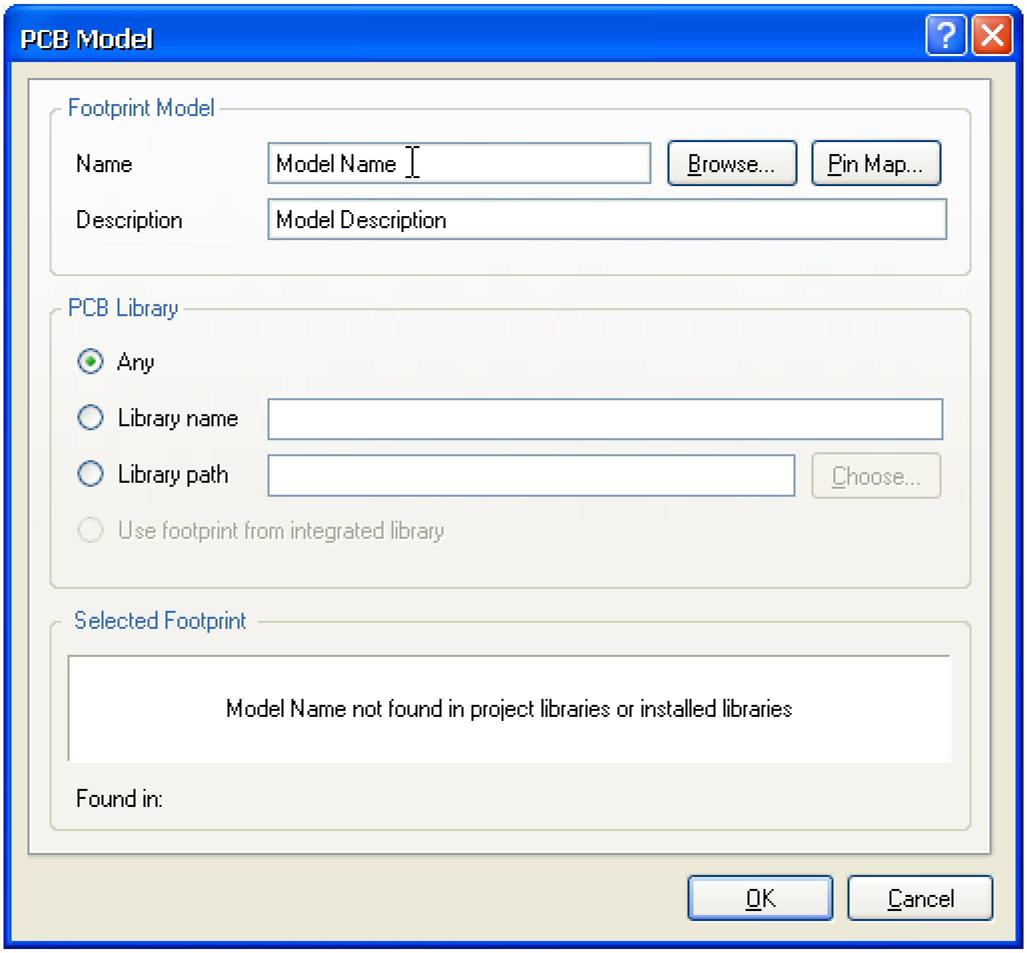


Figure 13. Add a footprint model to the component.

2. The *PCB Model* dialog (Figure 14) displays.


PCB Model

Footprint Model

Name: Model Name

Description: Model Description

PCB Library

● Any

○ Library name

○ Library path

○ Use footprint from integrated library

Selected Footprint

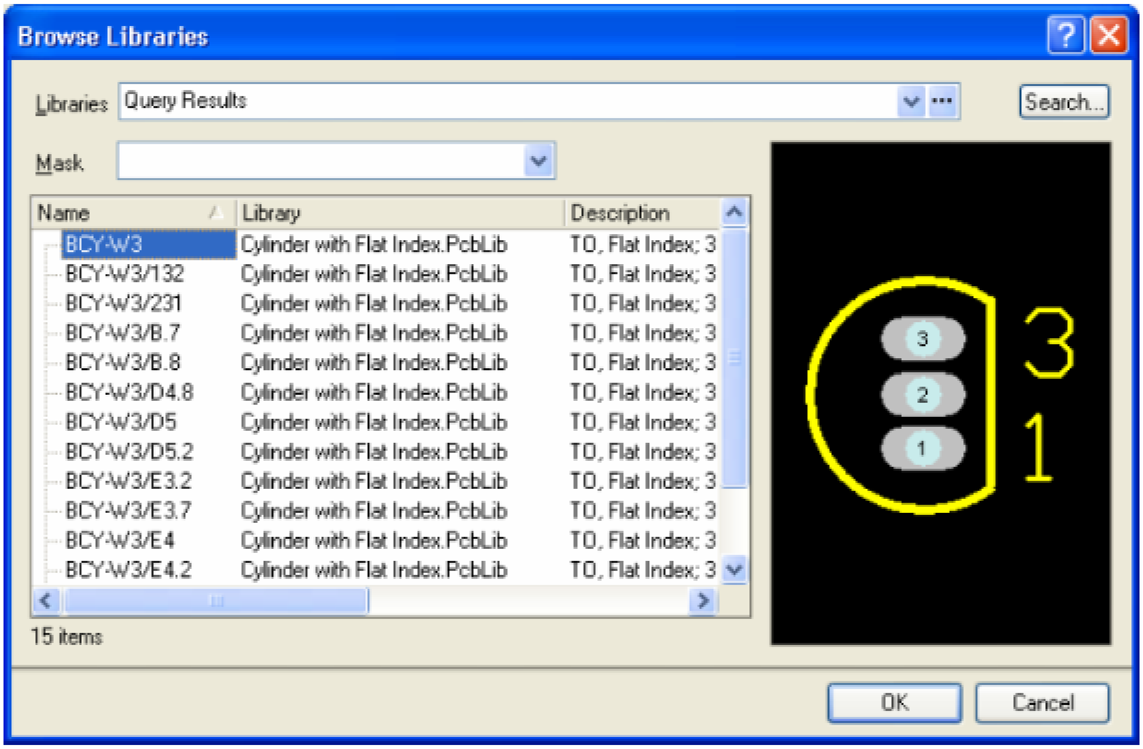
Model Name not found in project libraries or installed libraries

Found in:

Figure 14. Assign PCB models to schematic components

3. Click the **Browse** button to open the *Browse Libraries* dialog, this dialog allows you to browse footprint libraries that have been added into your library project, or have been added to the installed libraries list.
4.If the footprint is not available in any of the current libraries we will need to search for it. To do this click the **Find** button in the *Browse Libraries* dialog. The *Libraries Search* dialog displays, as shown in Figure 15.

Figure 15. Searching the supplied footprint libraries for the footprint.



Libraries: Query Results

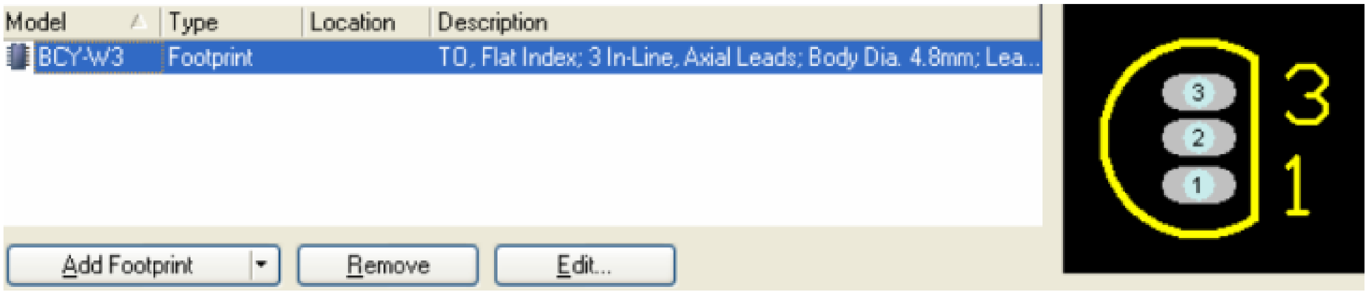
Mask:

Name	Library	Description
BCY-W3	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/132	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/231	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/B.7	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/B.8	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/D4.8	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/D5	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/D5.2	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/E3.2	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/E3.7	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/E4	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3
BCY-W3/E4.2	Cylinder with Flat Index.PcbLib	TO, Flat Index; 3

15 items

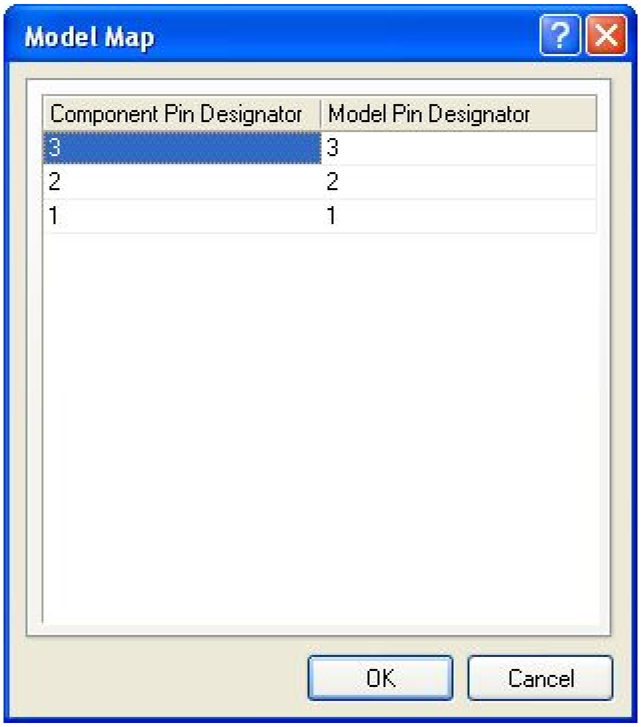
Figure 16. Search results_ _for the BCY-W3 footprint.

5. Set the Scope to **Libraries on Path** and the **Path** to the *Library\Pcb* folder of your Altium Designer installation. Make sure **Include Subdirectories** option is enabled.
6. In the query field at the top of the dialog, type BCY-W3 and click on **Search** .
7. You should get a number of results from the PCB library Cylinder with Flat Index.PcbLib listed in the *Browse Libraries* dialog, as shown in Figure 15. Select BCY-W3 from this library and click **OK** to return to the *PCB Model* dialog.
8. Since this is the first time you have used this library, you will be asked to confirm the installation of this library which will make it available for use. Click **Yes** in the *Confirm* dialog. The *PCB Model* dialog is updated with the footprint model information.
9. Click **OK** to add the model. It will appear in the **Model** region at the bottom of the workspace, as shown in Figure 17.
10. After you added the model it is a good idea to check the PIN - PAD mapping. Open the properties for the model to see the PCB Model properties, Figure 14. Press the Button **Pin Map** to open the Window Model Map. Figure 17a. Check and if necessary modify the mapping for the Pin - Pad pairs.



Model	Type	Location	Description
BCY-W3	Footprint		TO, Flat Index; 3 In-Line, Axial Leads; Body Dia. 4.8mm; Lea...

Figure 17. The footprint model has been added to the component.



Component Pin Designator	Model Pin Designator
3	3
2	2
1	1

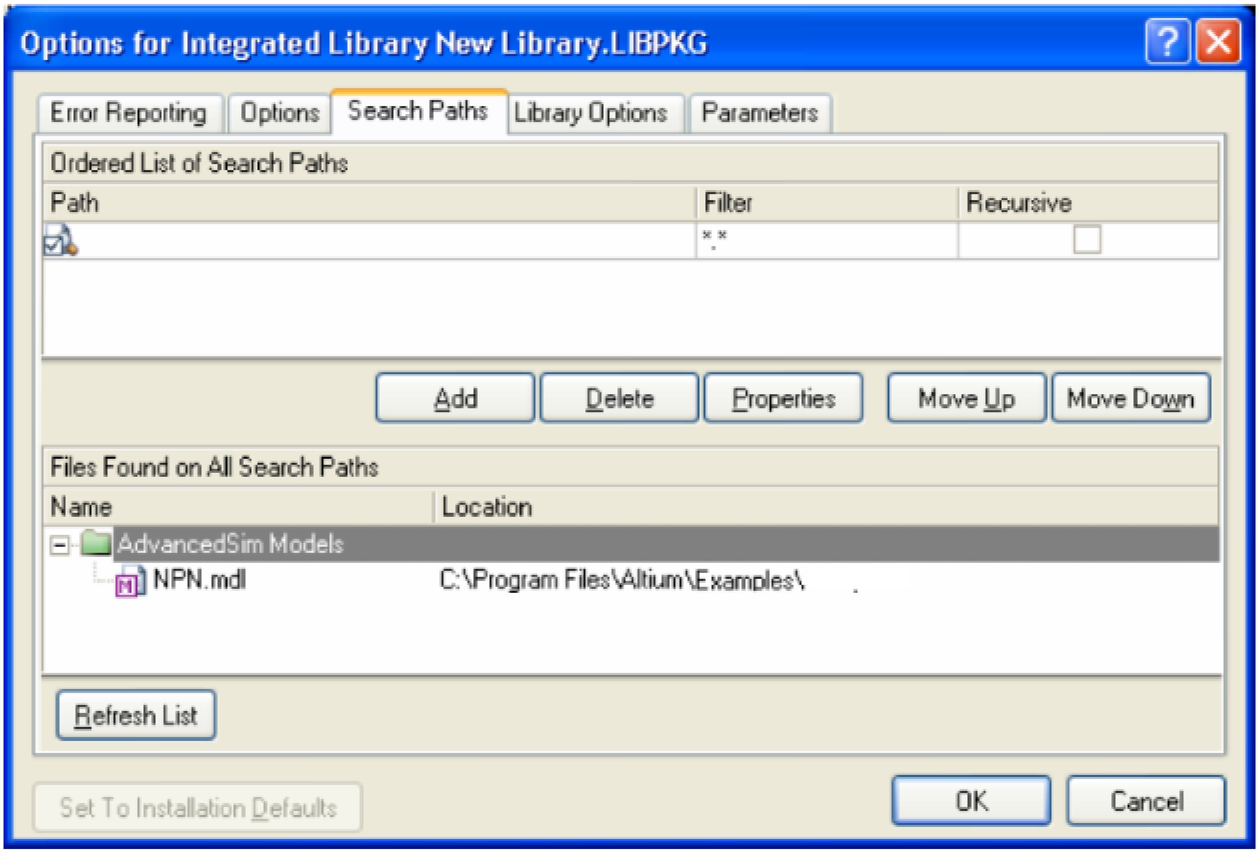
Figure 17a. the Pin - Pad mapping for the model .

Adding a Circuit Simulation Model

SPICE models are used for circuit simulation (.ckt and .mdl files) and are typically sourced from the device vendor website. If the device you are planning to use is available in one of the supplied libraries it will include a Spice model if one was available when the library was created. For the purpose of this tutorial you can find a suitable generic NPN model in the
C:\Program Files_ _Altium Designer \Examples\Tutorials\Creating Components folder, copy this model to the folder where you have saved your library.
1.How the Spice model is referenced in the project is up to you (remember that models are linked). You could add it as another source file to the project (right-click on the Project filename in the **Projects** panel and select **Add Existing to Project**). However, if you view the model file as a reference source file that should not easily be edited, then you would not include it as a normal project file. In this case the most appropriate way of referencing it would be to define a search path to the folder that contains the simulation model files. To do this select **Project » Project Options** from the menus, then click on the **Search Paths** tab.
2.Click the **Add** button to define a new search path, the *Edit Search Path* dialog will appear.
3.Unless specifically required, it is suggested that you always disable the **Include sub-folders in search** option as

this can greatly slow the searching process.

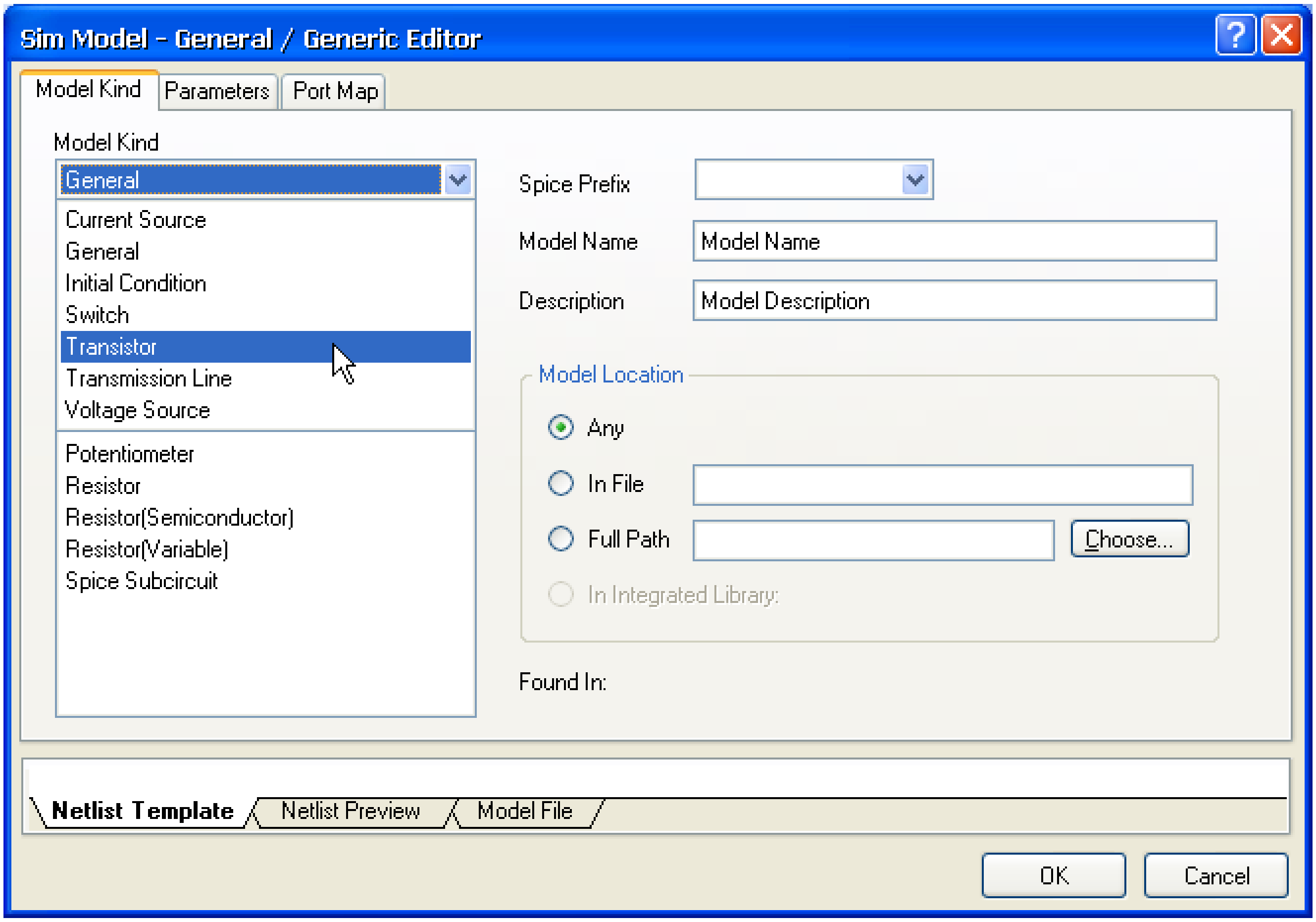
4. The default search path is the current folder, since you have copied the model to your working folder you can simply click **OK** . To confirm that the model will be found click the **Refresh List** button in the *Options* dialog (Figure 19). Now that the search path is defined any model stored in the working folder will automatically be found.



NPN.mdl C:\Program Files\Altium\Examples\

Figure 18. Search path to the current working folder.

5. Now that the model is available to the project you are creating (the library package New Library.LibPkg), it is time to add the simulation model to the NPN component. To do this follow the same steps you followed to add the footprint model, except choosing **Simulation** model this time. The *SIM Model - General / Generic Editor* dialog displays (Figure 19).



Model Kind: General

Spice Prefix:

Model Name: Model Name

Description: Model Description

Model Location

● Any

○ In File

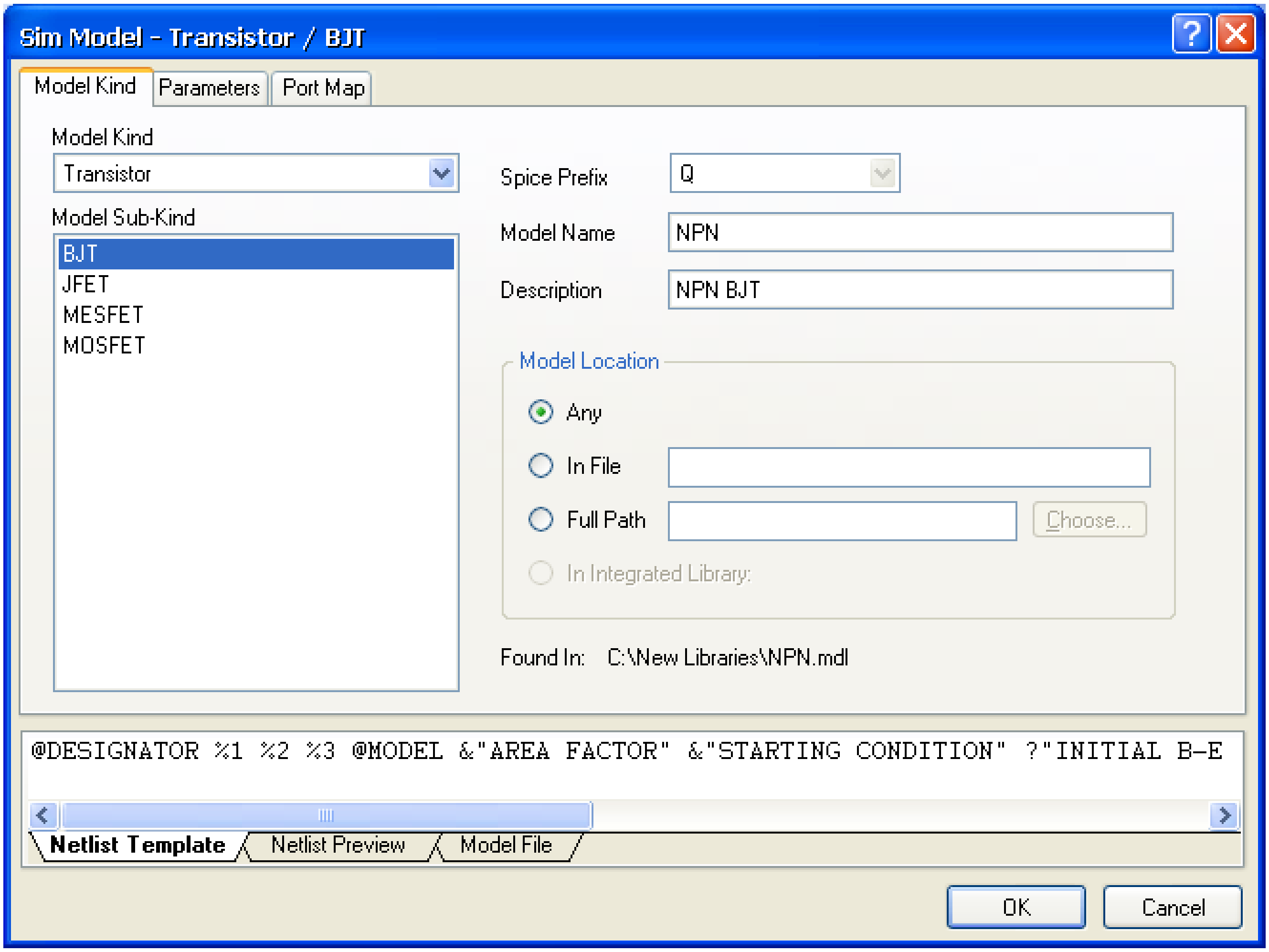
○ Full Path

○ In Integrated Library:

Found In:

Figure 19. Specifying and configuring all Spice model kinds is done in the Sim Model dialog.

6.Since the NPN is a transistor, select **Transistor** from the **Model Kind** list. The dialog becomes the _Sim Model - Transistor/BJT dialog (Figure 20).



Model Kind: Transistor

Model Sub-Kind: BJT (selected), JFET, MESFET, MOSFET

Spice Prefix: Q

Model Name: NPN

Description: NPN BJT

Model Location:
● Any
○ In File
○ Full Path
○ In Integrated Library:

Found In: C:\New Libraries\NPN.mdl

```
@DESIGNATOR %1 %2 %3 @MODEL &"AREA FACTOR" &"STARTING CONDITION" ?"INITIAL B-E
```

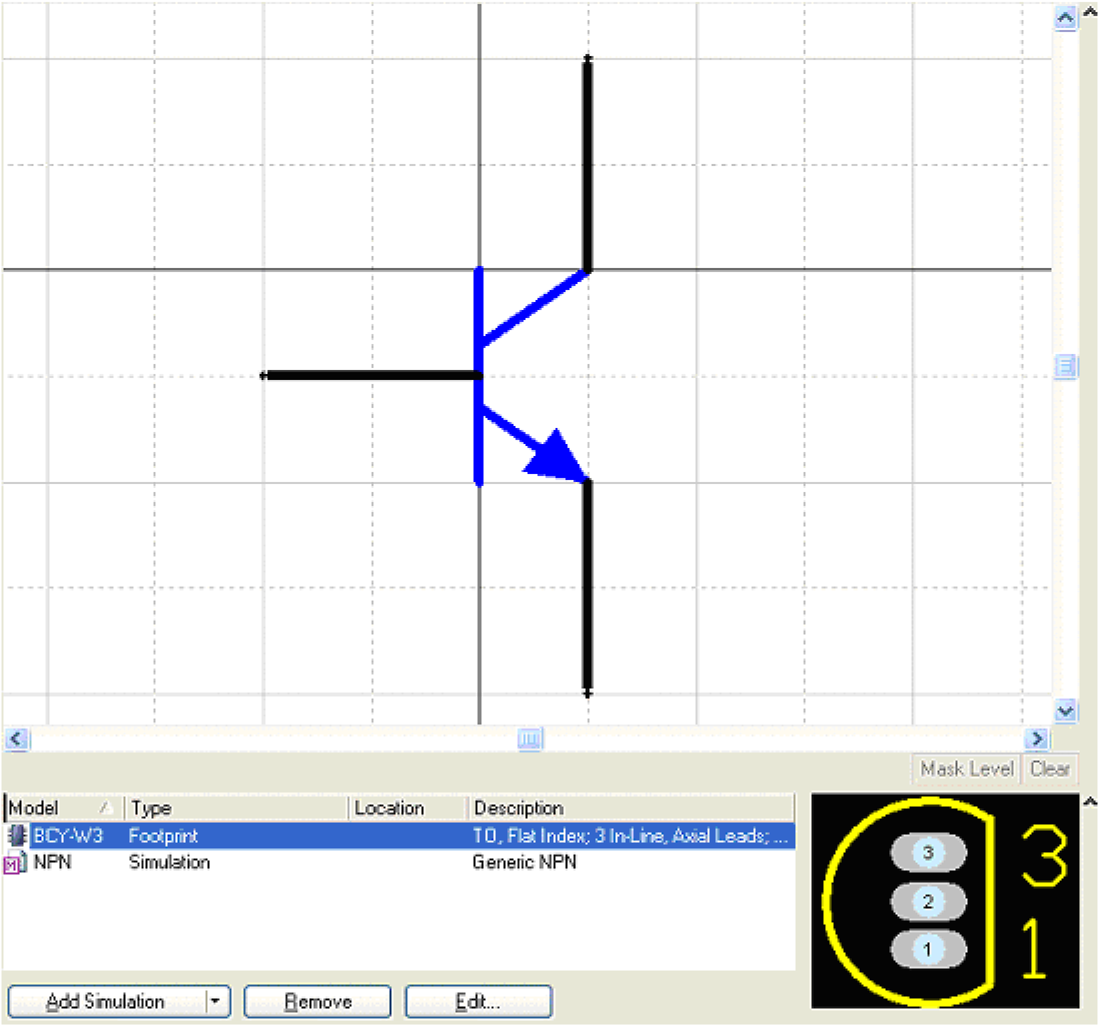
Figure 20. Configuring the NPN model.

7. Make sure **BJT** is selected as the **Model Sub-Kind** .

> The Model Name is the link to the SIM model file, so make sure it is a valid model filename (without the extension).
>
> Note: the Found In region of the dialog - when a valid model is found it is listed here.

8.Enter the name of the model file in the **Model Name** field, for this tutorial it is NPN (for the model file NPN.mdl), as soon as you finish typing the name the model should be detected, if it is found the path/name will be shown in the **F ound In** region of the dialog.
The Model Name is the link to the SIM model file, so make sure it is a valid model filename (without the extension).
Note: the Found In region of the dialog - when a valid model is found it is listed here.

9.Type in a suitable **Description** , e.g. Generic NPN.
If you do not have a model file (.mdl), you could click the **Create** button to run the **Spice Model Wizard** that will step you through the creation of a simulation model file for your component.
10.Click **OK** to return to the *Library Component Properties* dialog where the model NPN has been added to the Models list, as shown in Figure 21.



Model	Type	Location	Description
BCY-W3	Footprint		TO, Flat Index; 3 In-Line, Axial Leads; ...
NPN	Simulation		Generic NPN

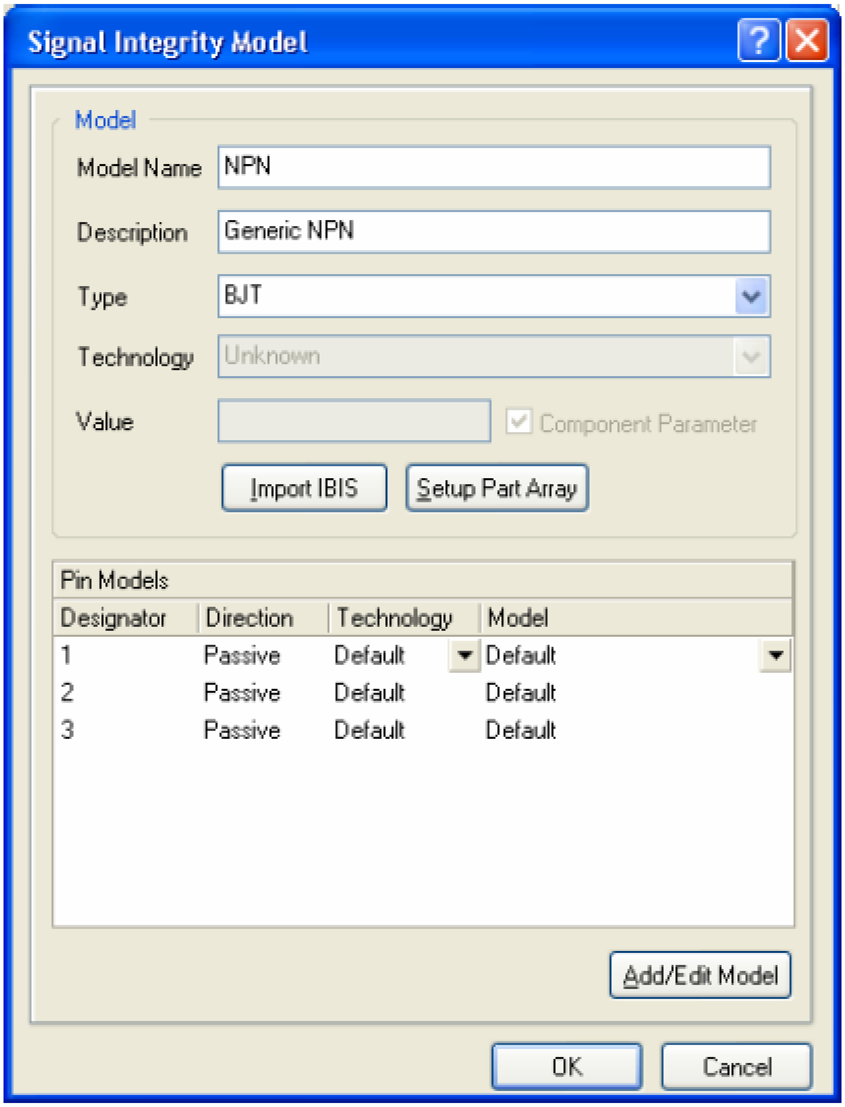
Figure 21. NPN with_ _the footprint and simulation models listed in the Models view.

Adding Signal Integrity Models

The Signal Integrity Simulator uses pin models rather than component models. To configure a component for signal integrity simulation, you either set the Type and Technology options, which will use the built-in default pin models, or you can import an IBIS model (which is essentially a set of pin models).

1. To add a signal integrity model, repeat the process you followed to add a footprint, except this time choose **Signal Integrity** . The *Signal Integrity Model* dialog displays.

2.If you wish to import an IBIS file, click on the **Import IBIS** button and navigate to the required .ibs file. For this tutorial we will use built-in default pin models. Set the **Type** to **BJT** and type in a suitable entries for the **Model Name** and **Description** fields (eg. NPN), as shown in Figure 22.



Model Name: NPN

Description: Generic NPN

Type: BJT

Technology: Unknown

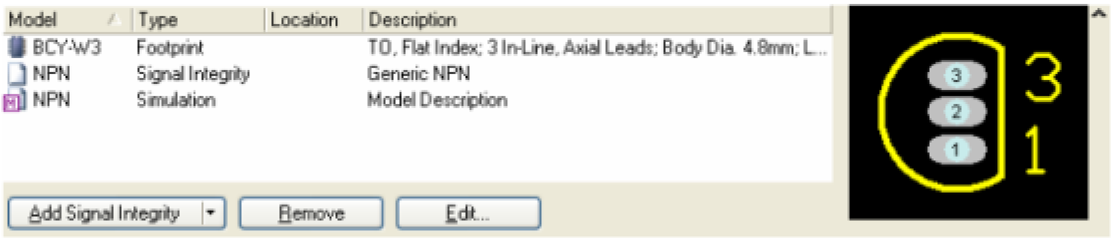
Value: ☑ Component Parameter

Pin Models

Designator	Direction	Technology	Model
1	Passive	Default	Default
2	Passive	Default	Default
3	Passive	Default	Default

Figure 22. Signal Integrity model editor, configured for the NPN transistor.

3.Click **OK** to return to the *Library Component Properties* dialog where the model is added to the Models list, as shown in Figure 23.



Model	Type	Location	Description
BCY-W3	Footprint		TO, Flat Index; 3 In-Line, Axial Leads; Body Dia 4.8mm; L...
NPN	Signal Integrity		Generic NPN
NPN	Simulation		Model Description

Figure 23. The simulation and signal integrity models have been added to the transistor.

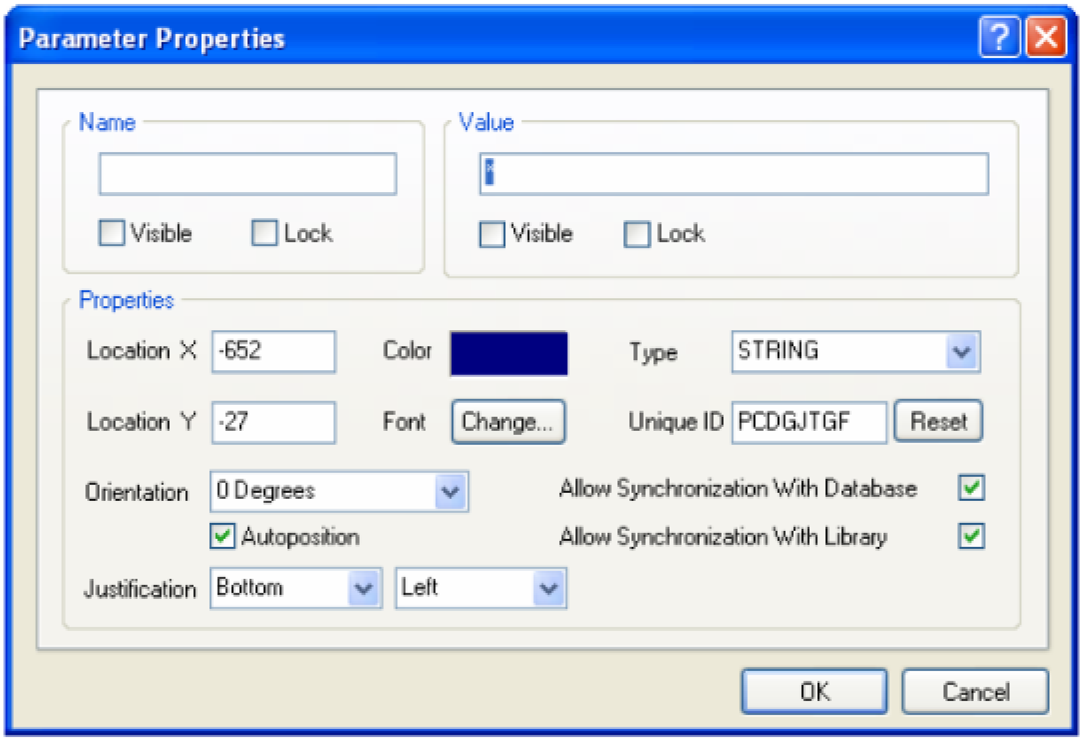
For more information about adding and editing Signal Integrity models, refer to the Performing Signal Integrity Analyses tutorial.

To edit and manage parameters across all components in a library use the Parameter Manager (Tools menu).

Adding Component Parameters

> To edit and manage parameters across all components in a library use the Parameter Manager (Tools menu).

Component parameters are a means of defining additional information about the component. This could include data your company needs in the BOM, manufacturers data, a reference to the component datasheet, design instruction information such as design rules or assignment to a PCB class, Spice simulation parameters, and so on - parameters can be used to add any useful information that you might need for a component.



Name:
□ Visible □ Lock

Value:
□ Visible □ Lock

Location X: -652
Color: ■
Type: STRING
Location Y: -27
Font: Change...
Unique ID: PCDGJTGF
Orientation: 0 Degrees
☑ Autoposition
Allow Synchronization With Database ☑
Allow Synchronization With Library ☑
Justification: Bottom Left

Figure 24. Parameters are configured in the Parameter Properties dialog.

To add a parameter to a schematic component:
1.Parameters are added to the component in the *Library Component Properties* dialog, double click on the component name in the list in the **Sch Library** panel to open the dialog.
2. To add a new parameter, click the **Add** button in the **Parameters for** ... section of the *Library Component Properties* dialog to display the *Parameter Properties* dialog (Figure 24).
3. Enter the name of the parameter and a value. Make sure **String** is selected as the parameter **Type** if you require a text string and the value's **Visible** option is enabled if you want the value to display when the component is placed on a schematic sheet. Click **OK** . The parameter is added to the Parameters list in the *Library Component Properties* dialog.

Parameters for Component-to-datasheet Linking

Parameters can be used to create links from the component to reference material, such as datasheets. Linkage is established by adding specific component parameters. One approach allows us to use the **F1** key to access a referenced document. The other, which caters for multiple references, uses the right-click context menu.

HelpURL

If a component includes a parameter with the reserved name *HelpURL* , then the URL will be resolved when the **F1** key is pressed while the cursor is hovering over the component. The URL can actually be a web address, a text file, or a PDF file.

Component Links

The second technique supports multiple links, and naming of each link. Here you add a pair of parameters, one that points to the linked document or URL, the second defines a label (or description) for this link. The parameter pairs are defined as follows.

	Parameter Name	**Example Parameter Value**
1 st parameter	*ComponentLink1URL*	*C:_ _My Datasheets\XYZDatasheet .pdf*
2 nd parameter	*ComponentLink1Description*	*Datasheet for XYZ*
1 st parameter	*ComponentLink2URL*	*C:_ _My Datasheets_ _Alternate XYZDatasheet.pdf*
2 nd parameter	*ComponentLink2Description*	*Datasheet for Alternate XYZ*

Any number of links can be defined using the same parameter pair, except with the number incremented. When you right-click on a component that uses datasheet linking, a **Reference** menu entry will appear in the **Context** menu, in it you will find an entry for each component link, as shown in Figure 25.

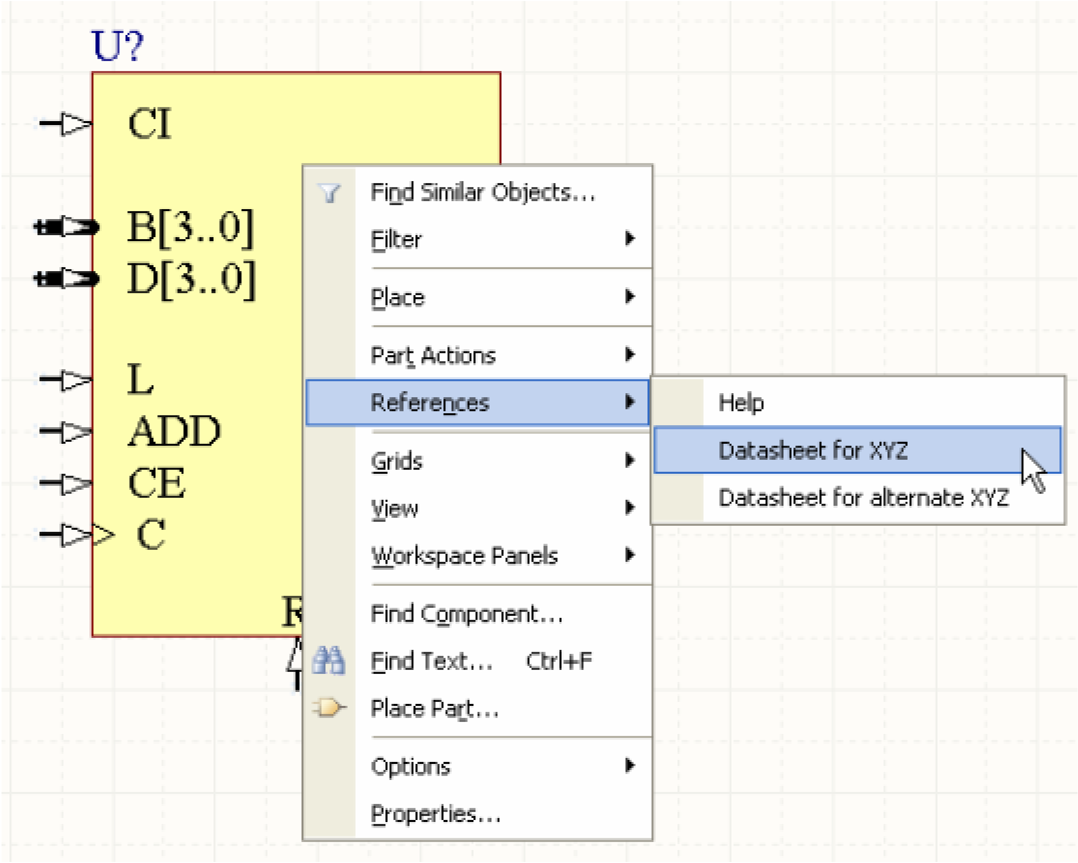


Figure 25. Right-click to access the datasheet links.

Component-to-datasheet linking linkage can be used when you are browsing components in the **Libraries** panel - press **F1** or right-click on the component name in the panel to access the linked documents/URLs.

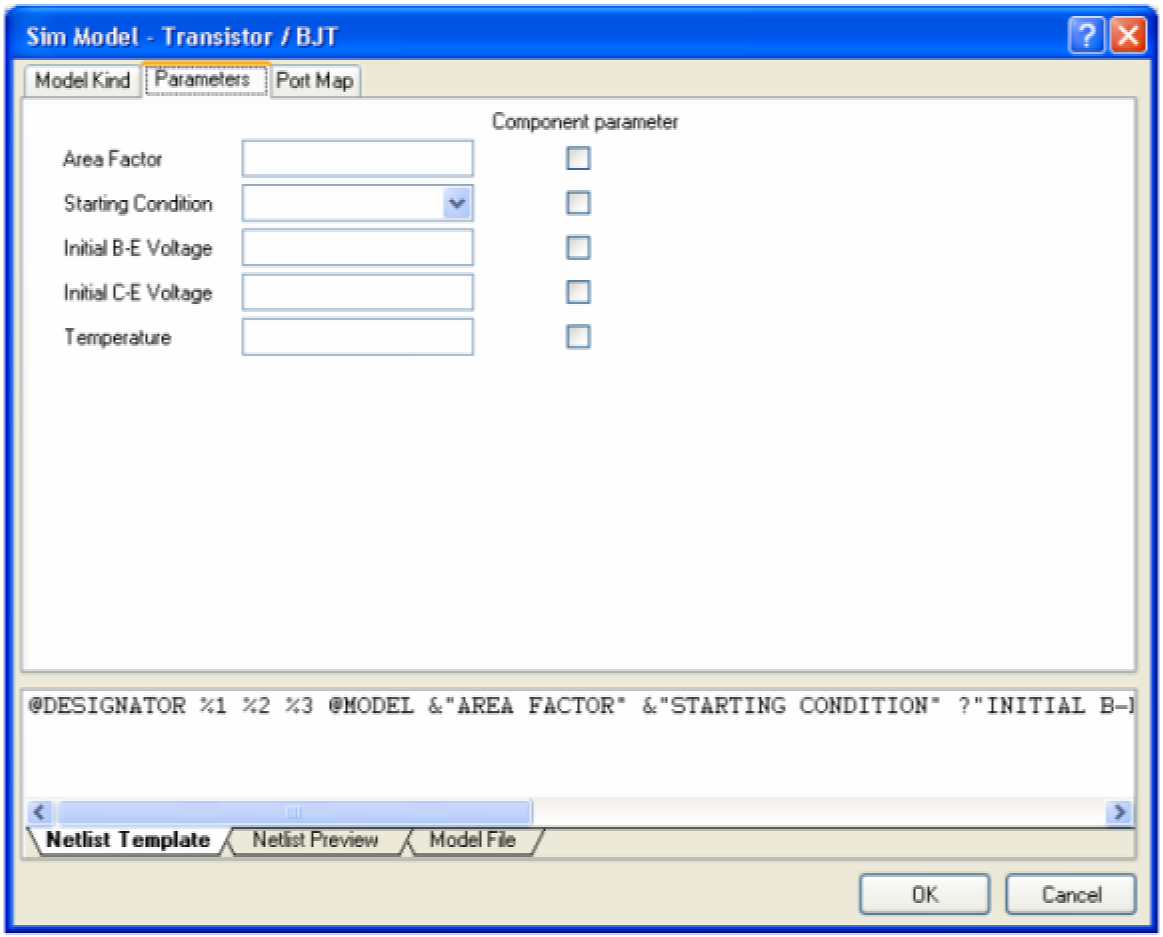
String Indirection

Often there are situations where you need to define a placeholder that is populated with some text later. For example, you might want a parameter called *DesignedBy* on a schematic template, whose value is defined when the template is used for a new schematic. Altium Designer uses a technique known as string indirection to support this requirement. At the schematic sheet level, you can add a document parameter, for example *DesignedBy* , whose value is left blank. You then place a standard string on the document, whose value is the text =DesignedBy. The equals sign sets this string to be an indirection string, instead of displaying the text it will display the current value of the document parameter DesignedBy. **Note** : The default behavior is to not analyze indirection strings and display the final value, to do this you must enable the **Convert Special Strings** option in the **Schematic - Graphical Editing** page of the *Preferences* dialog. Be aware that if the value of the parameter is blank then nothing will be displayed, which is why the **Convert Special Strings** option is disabled by default.
String indirection can also be used with components too - as well as displaying any parameter that has been added to the component in its own right by enabling its own **Visible** option, you can also indirect the string to the component's **Comment** field.
One situation where string indirection is useful is for a component that is used for both PCB design and circuit simulation. During schematic-to-PCB design transfer the schematic **Comment** field is mapped to the **Comment** field of the PCB component. But for circuit simulation the **Comment** field is not used, since the simulator can require many properties for a component - a BJT has five simulation properties for example. These five properties are defined as parameters instead. In this case any of the circuit simulation parameters can be mapped to the **Commen**

t field using string indirection, by entering the name of the parameter preceded by an equals sign. For example, a resistor has one simulation parameter, called *Value* . If the resistor's **Comment** field is set to =*Value* , then the contents of the *Value* parameter will be displayed as the **Comment** . If you are tuning the resistance value during simulation the correct resistance will be used when you transfer the design to PCB layout.

Simulation Parameters

As mentioned above, the string indirection feature can be used to map any parameter to the component's **Comment** field. **Note** : You do not need to add the simulation parameters to the component manually, they are built into the simulation model. If you edit the simulation model for the transistor you are creating you will see that the BJT model supports five simulation parameters, as shown in Figure 26.



```
@DESIGNATOR %1 %2 %3 @MODEL &"AREA FACTOR" &"STARTING CONDITION" ?"INITIAL B-I
```

Figure 26. Simulation parameters are defined in the Sim Model dialog.

If you want easier access to the simulation parameters, or want to display them on the schematic, or if you want to include them in output documentation, you can promote them to be component parameters by enabling the **Component parameter** option next to each parameter.

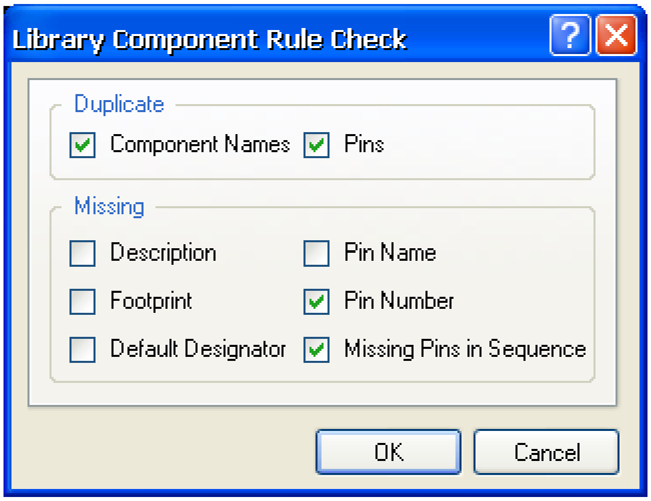
Checking the Component and Generating Reports

To check that the new components have been created correctly there are three reports that can be generated. Make sure the library file is saved before the reports are generated. Close the report file to return to the Schematic Library Editor.

> Linkage from the component pins to the model is not checked by the Component Rule Checker.
>
> This level of linkage is checked, however, when you compile a library package into an Integrated library. Even if you do not intend to use the compiled integrated libraries it is beneficial to create and manage your libraries using library packages, and compile them to perform a more comprehensive verification of your components.

Component Rule Checker

The Component Rule Checker tests for errors such as duplicates and missing pins.
1. Select **Reports » Component Rule Check** [shortcut: **R** , **R**]. The *Library Component Rule Check* dialog displays (Figure 27).



Library Component Rule Check

Duplicate

☑ Component Names ☑ Pins

Missing

□ Description □ Pin Name

□ Footprint ☑ Pin Number

□ Default Designator ☑ Missing Pins in Sequence

Figure 27. Configure the Component Rule Check dialog to test the current component.

2.Set the attributes you wish to check. Click **OK** . A report named libraryname.err displays in the Text Editor, listing any components that violate the rule check.
3. Make any adjustments necessary to the library and rerun the report.
4.Save the schematic library.

Component Report

To create a report that lists all the information available for the active component:
1. Select **Reports » Component** [shortcut: **R** , **C**].
2. The report 'libraryname.cmp' is displayed and includes the number of parts with the pin details for each part in the component.

Library Report

To create a extensive report of each component in the library:
1. Select **Reports » Library Report** [shortcut: **R** , **L**].
2. Configure the report settings in the *Library Report Settings* dialog.

The report will be opened in Microsoft Word, or your web browser, depending on the style chosen.

Copying Components from Other Libraries

You can also copy components to your schematic library from other open schematic libraries and then edit their properties as required. If the component is part of an integrated library, you will have to open the .IntLib file (**File » Open**) and choose **Yes** to extract the source libraries. Then open the generated source library (.SchLib) from the **Pr ojects** panel.
1. Select the component that you wish to copy in the **Components** list of the **SCH Library** panel so it displays in the design window.
2. Select **Tools » Copy Component** to copy a component from the current library document to any other open library document. The *Destination Library* dialog displays listing all currently open schematic library documents.
3. Select the document to which you want to copy the component. Click **OK** and a copy of the component will be placed in the destination library where you can edit it, if necessary.

Copying Multiple Components

You can also copy single or multiple components via the **SCH Library** panel. Select the component(s) in the list of names in the panel using the standard **CTRL** + Click or **SHIFT** + Click features, then right-click on one of the selected component and choose **Copy** from the pop-up menu, as shown in Figure 28.

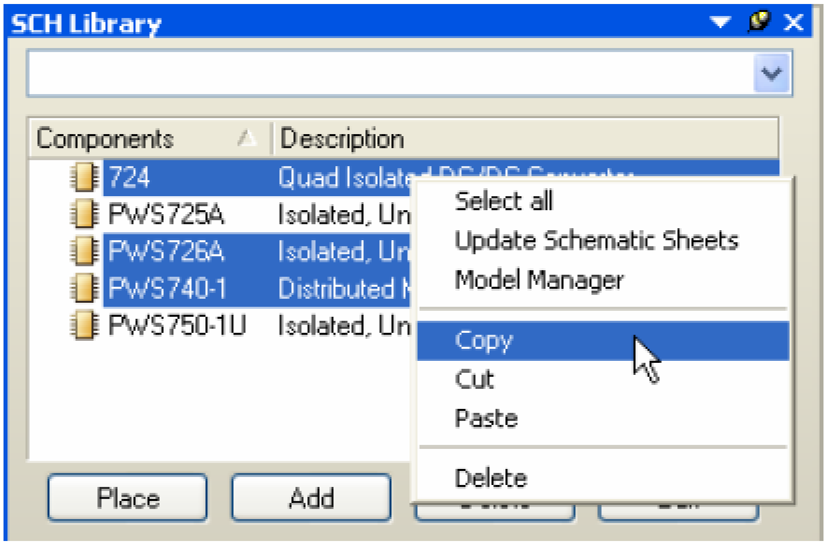


Figure 28. Copying the selected components from the current library.

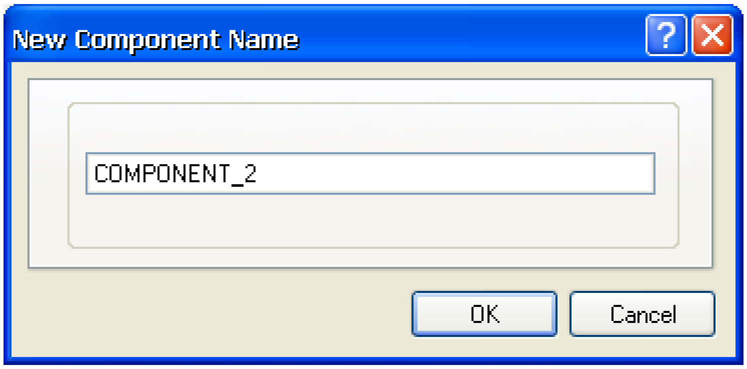
You can then right-click in the **Components** list and:

- paste the component(s) back into the same library
- paste the component(s) into another open library
- you can also copy and paste components from a schematic into an open library using the same technique.

Creating a New Schematic Component with Multiple Parts

The transistor symbol that you have created represents the entire component, that is the one symbol represents what is supplied in the physical package delivered by the device manufacturer.
There are situations where the one physical component is better represented as a collection of parts. For example

there are resistor networks that contain 8 individual resistors, and each one can be used independently of the others. Another example would be a 74F08 quad 2 input AND gate - in this device there are four independent 2 input AND gates. While the component could be drawn as a single symbol showing all four gates, it could be more useful if it is drawn as four separate gates, where each gate can be placed independently of the others, anywhere on the schematic. This approach of drawing a component as a set of separate parts is referred to as a multi-part component.



COMPONENT_2

Figure 29. Enter the name of the new component.

This section of the tutorial will show the steps you take to create a 74F08SJX Quad 2-IN AND gate. We will also create an alternate view mode for the component, an IEEE representation of the device.

1. Select **Tools » New Component** [shortcut: **T** , **C**] when in the Schematic Library Editor. The *New Component Name* dialog displays (Figure 29).

2. Type in the name of the new component, e.g. 74F08SJX, and click **OK** . The new component name displays in the **Components** list in the **SCH Library** panel and an empty component sheet displays with a crosshair through the center (origin) of the sheet.

3. Now we will create the first part of the new component as shown above, including its pins, as detailed in the following sections. The first part will then be used as the basis for the other parts, as only the pin numbers need to change between the parts.

Creating the Body of the Component

The body of this component is constructed from a multi-segment line and a circular arc. Make sure the component sheet origin is in the center of the workspace by selecting **Edit » Jump » Origin** [shortcut: **J** , **O**]. Also, make sure the grid is visible [shortcut: **PAGE UP**].

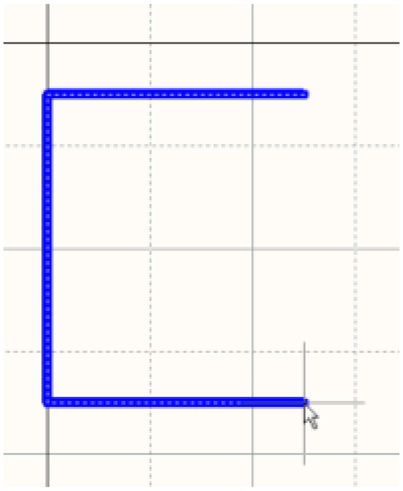
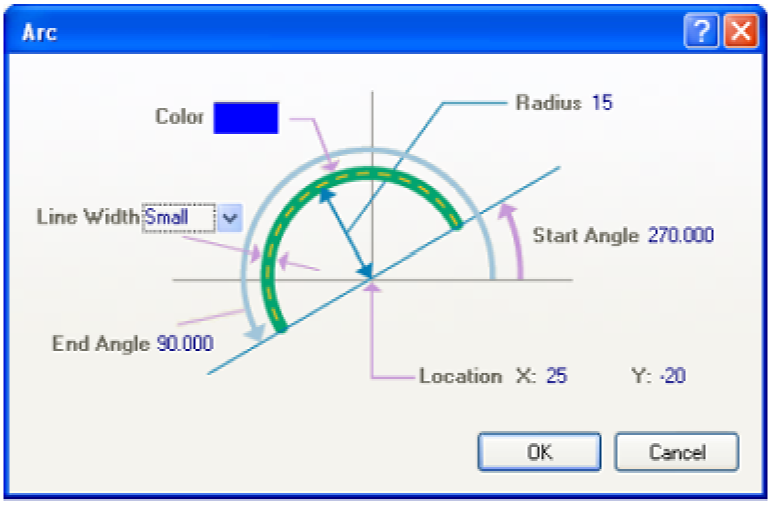


Figure 30. Place the Polyline to define the body of the first part

Placing Lines

1. Note the current grid setting displayed on the Altium Designer Status bar (bottom left). You can cycle through the three definable grid settings at any time by pressing the **G** key - set the grid to 5.
2. Select **Place » Line** [shortcut: **P** , **L**] or click on the toolbar button. The cursor changes to a crosshair and you are now in multi-segment line placement mode.
3. Press the **TAB** key to set the line's properties. Set the line width to Small in the *Polyline* dialog.
4.Referring to the X, Y co-ordinates at the left end of the Status bar, position the cursor at 25, -5, then click or press **ENTER** to anchor the starting point of the line. Then position the mouse and click to anchor a series of vertex points that define the segments of the line (at 0,-5; 0,-35 and 25,-35).
5. When you have finished drawing the line, right-click or press **ESC** to exit line placement mode.
6. The completed polyline is shown in Figure 30. Save the component.



Color
Radius 15
Line Width Small
Start Angle 270.000
End Angle 90.000
Location X: 25 Y: -20

Figure 31. The properties of the arc can be defined in the dialog, or using the mouse.

Drawing an Arc

Placing an arc is a four-step process that sets the center point, radius, start angle and end angle of the arc. **Note** : You can press **ENTER** instead of click to place the arc.
1. Select **Place » Arc (Center)** [shortcut: **P** , **A**]. The last arc drawn appears on the cursor and you are now in arc placement mode.
2. Press the **TAB** key to set the arc's properties. The *Arc* dialog displays. Set the radius to 15, start angle to 270, end angle to 90, and the line width to **Small** , as shown in Figure 31.
3. Position the cursor at 25, -20 and press **ENTER** or click to anchor the center point of the arc. There is no need to move the mouse, the cursor will jump to the correct location to define the arc radius of 15, as set in the *Arc* dialog. Press **ENTER** to accept the radius setting.
4. The cursor will then jump to the start point of the arc, as set in the dialog. Without moving the mouse press **ENTER** to accept the arc start angle, and when the cursor jumps again press **ENTER** again to define the arc end angle.
5. Right-click, or press **ESC** , to exit arc placement mode.

Adding Signal Pins

Add the pins to the first part, using the same technique described in the section Adding pins to a schematic component earlier in this tutorial. Pins 1 and 2 have an electrical type of Input and pin 3 is Output. Set the pin length to 20. The completed part is shown in Figure 34.

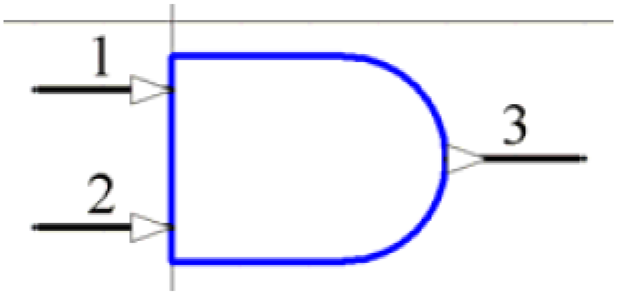
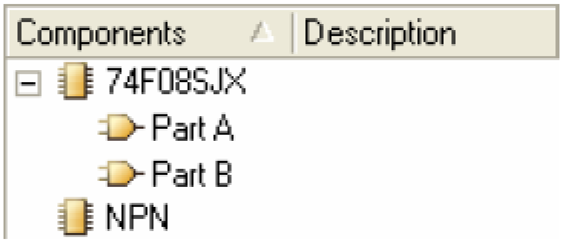


Figure 32. Part A of component 74F08SJX. The input/output indicator triangles are a display feature, controlled by the Pin Direction option in the Schematic - General page of the Preferences dialog.



Components	Description
⊟ 74F08SJX	
Part A	
Part B	
NPN	

Figure 33. Part B has been added.

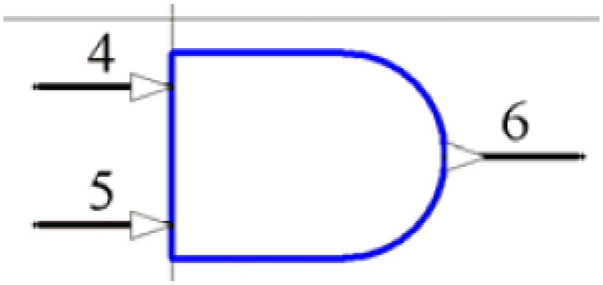


Figure 34. Part B of the 74F08SJX.

Creating Parts 2, 3 and 4

1. Select the component using **Edit » Select » All** [shortcut: **CTRL** + **A**].
2. Select **Edit » Copy** [shortcut: **CTRL** + **C**] to copy this part to the clipboard.
3. Select **Tools » New Part** . A blank component sheet displays. The Part counter in the **SCH Library** panel will be updated to include Part A and Part B if you click on the **+** to the right of the component name in the Components list in the **SCH Library** panel, as shown in Figure 34.
4. Select **Edit » Paste** [shortcut: **C TRL** + **V**]. The outline of the component part will appear on the cursor, place it at the same relative location to the sheet origin as Part A (the black crosshair in the center of the sheet indicates the origin). If necessary select and move the copied part until it is positioned the same as the original part.
5. Update the pin information in the new part, Part B, by double-clicking on each pin and changing the pin name and

number in the *Pin Properties* dialog. Once complete, Part B will look like Figure 34.
6. Repeat steps 3 to 5 above to create the remaining two parts, Part C and D, as shown in Figure 35. Save the library.

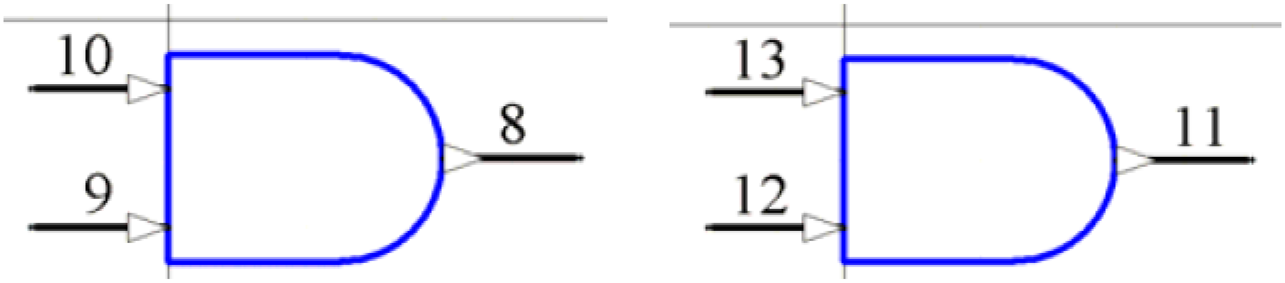


Figure 35. Part C and D of the 74F08SJX

Adding Power Pins

There are two approaches to defining the power pins. You can either create a 5th part for the component and place the VCC and GND pins on that part. If you use this approach remember to enable the **Locked** option in the *Component Properties* dialog for this part to ensure that it cannot be swapped with any of the gates during re-annotation.
The second approach is to define the power pins as hidden pins, in this case the software automatically connects them to the specified nets.
Hidden power pins do not belong to any specific part in a multi-part component, they actually belong to all parts (ensuring they exist on the schematic regardless of which part has been placed). To support this requirement pins can be assigned to Part zero - a special part that is used to store pins that you want added to every part in the component.
1.Add a GND (pin number 7) and VCC (pin number 14) pins to the component, setting their **Part Number** property to 0, **Electrical Type** to Power, **Hide** status to hidden, and **Connect to** net property to GND and VCC respectively.

2.To enable the display of hidden objects, select **View » Show Hidden Pins** from the menus. The completed part should look like the one shown in Figure 36. Check that the power pins appear on each part.

Setting the Component's Properties

1. Set the component's properties by clicking on the **Edit** button in the **SCH Library** panel when the component is selected in the **Components** list. Fill in the *Library C omponent Properties* dialog specifying the **Default Designator** as U?, the **Description** as Quad 2-Input AND Gate and add the footprint name DIP14 to the **Models** list. We will create a DIP14 footprint using the **PCB Component Wizard** later in this tutorial.
2. Save the component in the library by selecting **File » Save**.

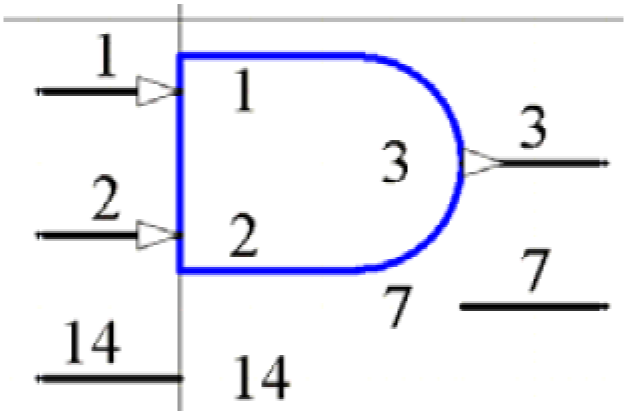


Figure 36. Part A with the hidden power pins displayed.

Creating an Alternate View Mode for a Part

You can add up to 255 alternate view modes to a component part. These view modes can contain any different graphical representation of the component, such as a DeMorgan or an IEEE representation. A selection of IEEE symbols are available from the Sch Lib IEEE toolbar (**View » Toolbars » Utilities**), or select **Place » IEEE Symbols** . Each alternate view mode should always have the same set of pins as the Normal mode.
If an alternate view of a part has been added it is displayed for editing in the Schematic Library Editor by selecting the alternate mode from the **Mode** tool in the **Mode** toolbar, as shown in Figure 37.

To add an alternate view mode, with the component part displayed in the design window of the Schematic Library Editor:
1. Select **Tools » Mode » Add** or click on the button. A blank sheet for *Alternate 1* displays.
2.Typically you would copy the part you created in the Normal mode and paste it into the new Alternate mode. This gives you the correct set of pins, you can then modify the graphical elements and position the pins as required.
3.Save the library.
Once the component has been placed on a schematic sheet the view mode can be selected from the **Mode** list in the **Graphical** region of the *Component Properties* dialog.

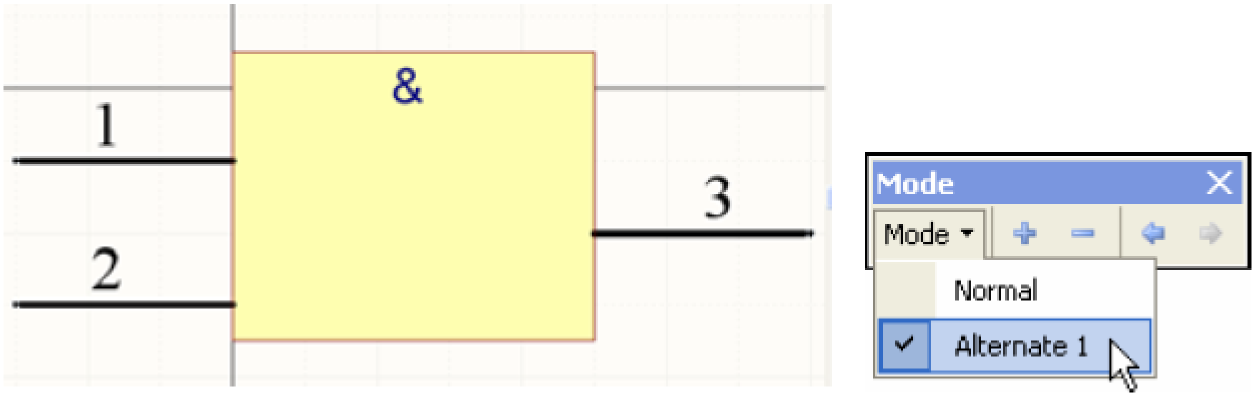


Figure 37. View mode Alternate 1 used to draw an IEEE representation of an AND gate

Creating PCB Component Footprints

> Footprints are always built on the top side, regardless of the final side of board you place them on. Layer-specific attributes, such as surface mount pads and solder mask definitions are automatically transferred to appropriate bottom side layers when you flip the footprint to the other side of the board during component placement.

This section of the tutorial covers the following topics:

- creating a new PCB library
- using the **PCB Component Wizard** to create a footprint for a schematic component
- manually creating a footprint
- other special footprint requirements, including irregular pad shapes
- including three-dimensional component body detail (3D bodies).

Footprints can be copied from the PCB Editor into a PCB library, copied between PCB libraries, or created from scratch using the PCB Library Editor's **PCB Component Wizard** or drawing tools. If you had a PCB design with all the footprints already placed, you could use the **Design » Make PCB Library** command in the PCB Editor to generate a PCB library that includes those footprints only.
Altium Designer also includes comprehensive libraries of predefined through-hole and SMD component footprints for use in designing PCBs. The footprint libraries (.PcbLib files) supplied are stored in the *Library\Pcb* folder in your Altium Designer installation.
The footprints that are created manually in this part of the tutorial are only to illustrate the procedures required, they are not dimensionally accurate. Always check the specifications of a new footprint against the manufacturer's datasheet.

Creating a New PCB Library

To create a new PCB library:
1.Select **File » New » Library » PCB Library** . A new PCB library document, called PcbLib1.PcbLib, is created and an empty component sheet called PCBComponent_1 displays.
2. Rename the new PCB library document to PCB Footprints.PcbLib, for example, by selecting **File » Save As** . The new PCB footprint library should be part of your library package, as shown in Figure 38.

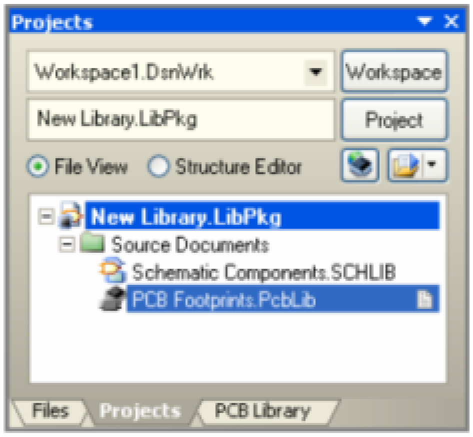
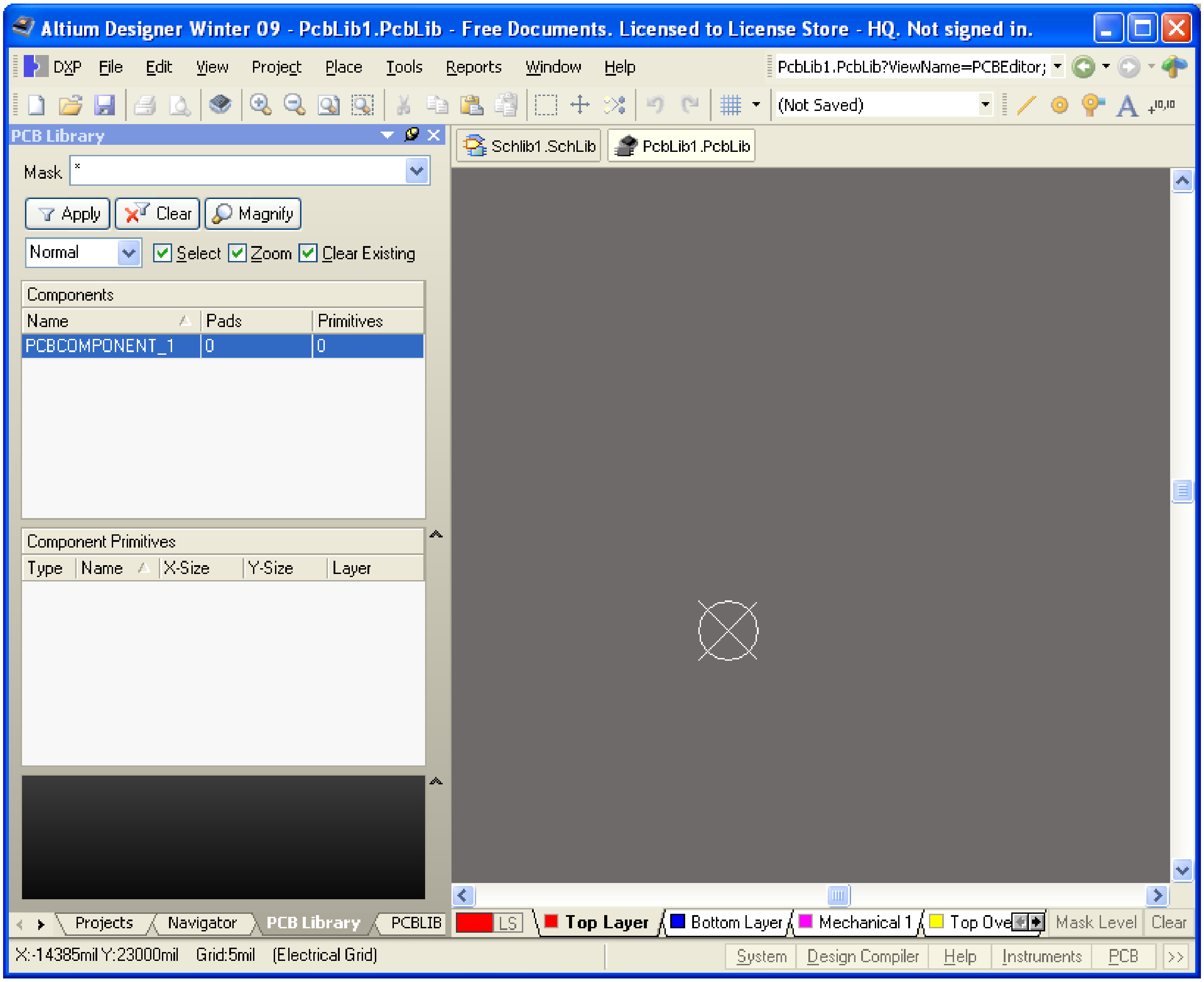


Figure 38. Library package after_ _adding the footprint library

3. Open the **PCB Library** panel by clicking on the **PCB Library** tab.
4. Click once in the grey area of the PCB Library Editor workspace and press the **PAGE UP** key a few times, until you can see the grid, as shown in Figure 39.

You are now ready to add, remove or edit the footprint components in the new PCB library using the PCB Library Editor commands.


Altium Designer Winter 09 - PcbLib1.PcbLib - Free Documents. Licensed to License Store - HQ. Not signed in.

Mask: *

Normal ☑ Select ☑ Zoom ☑ Clear Existing

Components

Name	Pads	Primitives
PCBCOMPONENT_1	0	0

Component Primitives

Type	Name	X-Size	Y-Size	Layer

X:-14385mil Y:23000mil Grid:5mil (Electrical Grid)

Figure 39. New PCB library, ready to create a footprint.

Using the PCB Component Wizard

The PCB Library Editor includes a **PCB Component Wizard** that will build a component footprint based on your input to a series of questions. We will use the wizard to create a footprint for a DIP14.

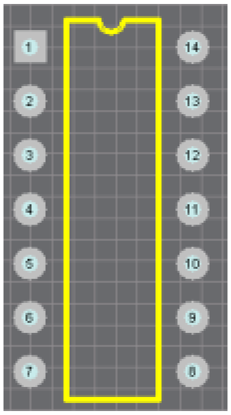
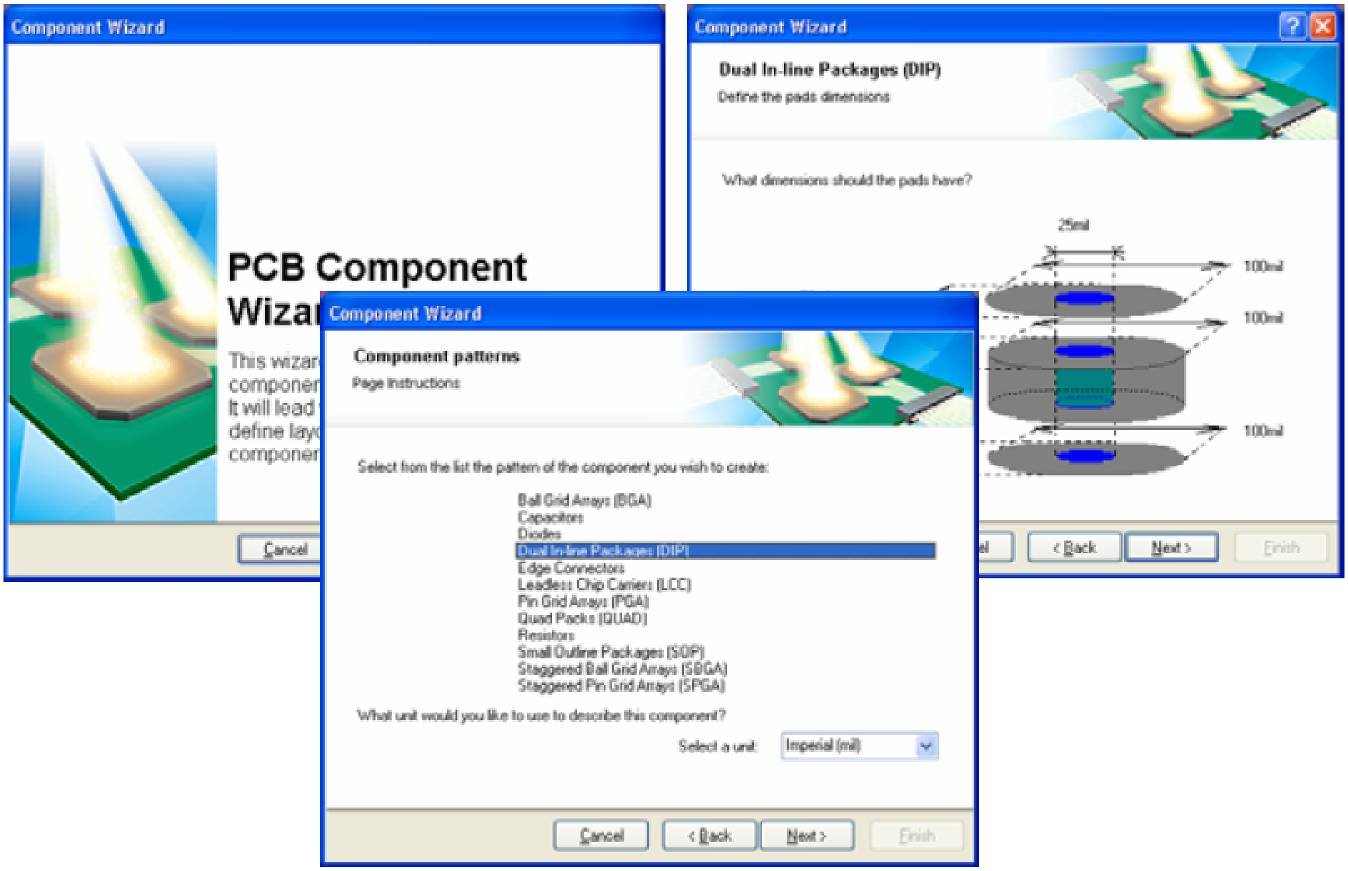


Figure 40. DIP14 footprint created by the PCB Component Wizard.



This wizar
componen
It will lead
define lay
componen

Dual In-line Packages (DIP)

Define the pads dimensions

What dimensions should the pads have?

Component patterns

Page instructions

Select from the list the pattern of the component you wish to create:

Ball Grid Arrays (BGA)
Capacitors
Diodes
Dual In-line Packages (DIP)
Edge Connectors
Leadless Chip Carriers (LCC)
Pin Grid Arrays (PGA)
Quad Packs (QUAD)
Resistors
Small Outline Packages (SOP)
Staggered Ball Grid Arrays (SBGA)
Staggered Pin Grid Arrays (SPGA)

What unit would you like to use to describe this component?

Select a unit: Imperial (mil)

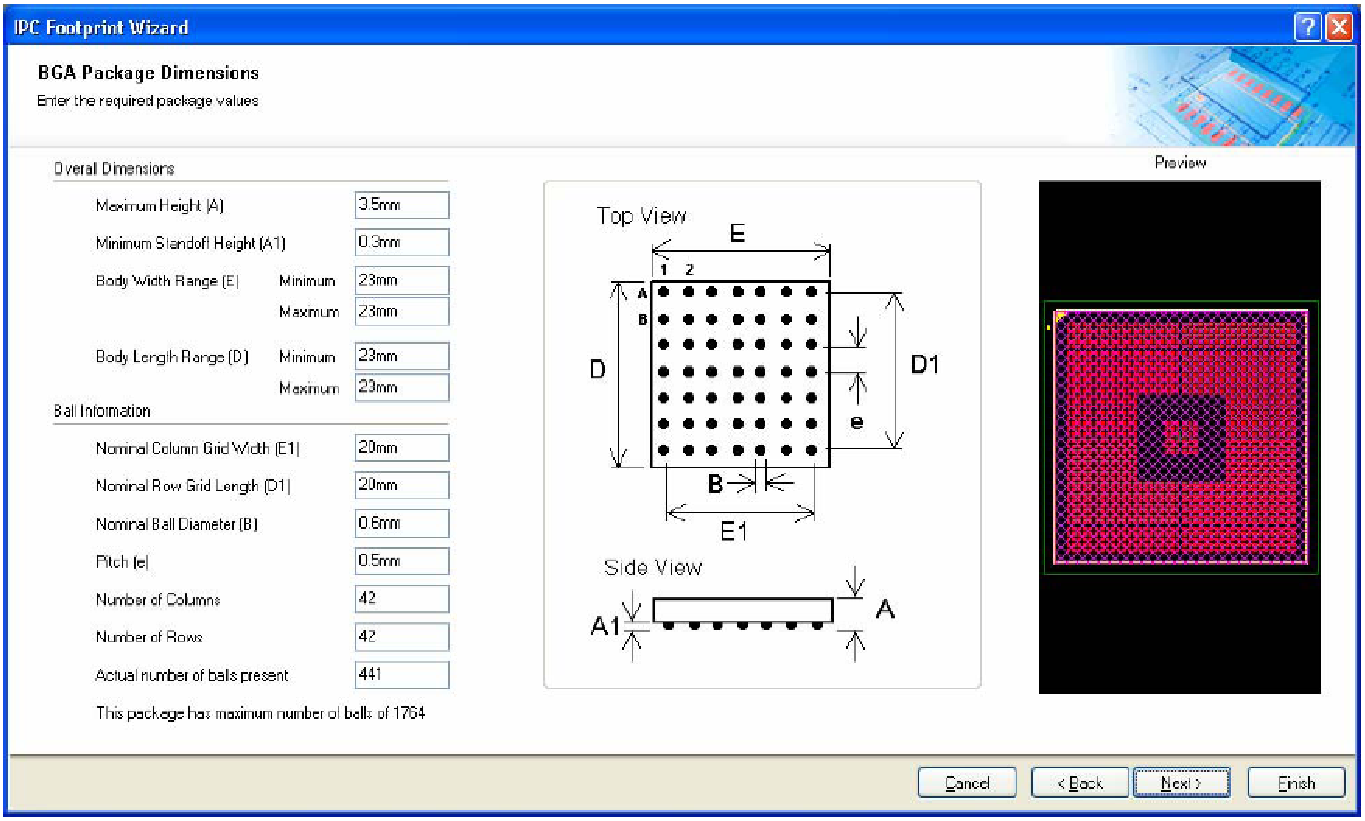
Figure 41. Work through the wizard to build a DIP14 footprint.

To create our new component footprint, DIP14, using the **Component Wizard** :
1. Select the **Tools » Component Wizard** [shortcut: T*, **C**]. The **PCB Component Wizard** will automatically start. Click **Next >** to progress through the wizard.
2. Answer the questions asked by selecting from the options available. To create our DIP14, select Dual in-line Package (DIP) as the pattern, Imperial units, 60mil round pads with 32mil hole (select and type over the dimensions), a distance between pads of 300mil (horizontal) and 100mil (vertical), then accept the rest of the defaults until you need to specify the number of pads required. Type in 14 for the number of pads required.
3.Click **Next >** until you come to the final page of the wizard and click **Finish** . The filename of new footprint, DIP14,

will appear in the **Components** list in the **PCB Library** panel and the new footprint will display in the design window, as shown in Figure 40. You can then further modify the component to suit requirements.
4. Save the library with its new footprint component by selecting **File » Save** [shortcut: **CTRL** + **S**].

Using the IPC® Compliant Footprint Wizard

As well as the **PCB Component Wizard** , you can also create component footprints with the **IPC® Compliant Footprint Wizard** . Rather than requiring you to enter the properties of the pads and tracks that are used to define the footprint, the **IPC® Compliant Footprint Wizard** takes the actual component dimensions as its inputs. Based on the formulae developed for the IPC-7351 standard, the wizard then generates the footprint using standard Altium Designer objects, such as pads and tracks. The wizard is launched from the PCB Library Editor **Tools** menu.



BGA Package Dimensions
Enter the required package values

Overal Dimensions

Maximum Height (A)		3.5mm
Minimum Standoff Height (A1)		0.3mm
Body Width Range (E)	Minimum	23mm
	Maximum	23mm
Body Length Range (D)	Minimum	23mm
	Maximum	23mm

Ball Information

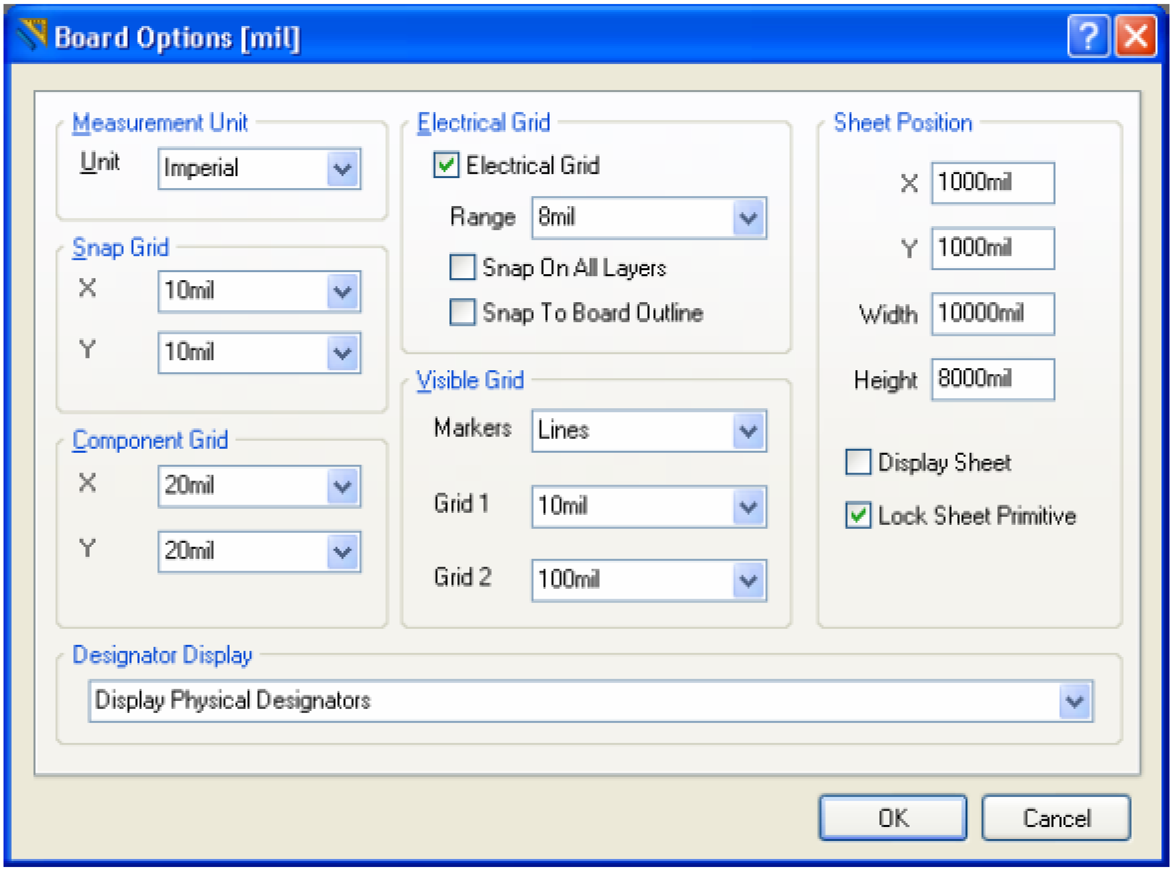
Nominal Column Grid Width (E1)	20mm
Nominal Row Grid Length (D1)	20mm
Nominal Ball Diameter (B)	0.6mm
Pitch (e)	0.5mm
Number of Columns	42
Number of Rows	42
Actual number of balls present	441

This package has maximum number of balls of 1764

Figure 42. The IPC® Compliant Footprint Wizard builds the footprint based on the component dimensions.

Manually Creating a Footprint

Footprints are created and modified in the PCB Library Editor using the same set of tools and design objects available in the PCB Editor. Anything can be saved as a PCB footprint, including corner markers, phototool targets and mechanical definitions. **Note** : Once a footprint has been placed onto a PCB you can set the **Type** property, defining it as Graphical or Mechanical if required. Use the What's This help in the *Component* dialog for more information on these settings.



Board Options [mil]

Measurement Unit
Unit: Imperial

Snap Grid
X: 10mil
Y: 10mil

Component Grid
X: 20mil
Y: 20mil

Electrical Grid
☑ Electrical Grid
Range: 8mil
□ Snap On All Layers
□ Snap To Board Outline

Visible Grid
Markers: Lines
Grid 1: 10mil
Grid 2: 100mil

Sheet Position
X: 1000mil
Y: 1000mil
Width: 10000mil
Height: 8000mil
□ Display Sheet
☑ Lock Sheet Primitive

Designator Display
Display Physical Designators

Figure 43. Set the units and grids_ _in the Board Options dialog.

To create the component footprint, we will place pads to form the component pin connections, and then place tracks and arcs for the outline. Design objects can be placed on any layer, however the outline is normally created on the Top Overlay (silkscreen) layer and the pads on the multilayer (for thru hole component pins) or the top signal layer (for a surface mount component pins). When you place the footprint on a PCB, all objects that make up the footprint will be assigned to their defined layers.

To manually create a footprint suitable for the NPN transistor:

1. Before creating the footprint, check that the units and grids are suitable. Select **Tools » Library Options** [shortcut : **D** , **O**] to display the *Board Options* dialog, and confirm that **Units** are **Imperial** , and the **Snap Grid** is set to 10mil in the X and Y directions. You will need to set the Grid to suit the spacing required by the pads in the footprint you are creating. Set **Visible Grid 1** to 10mil and **Visible Grid 2** to 100mil.

2.An empty component footprint workspace is created when you select the **Tools » New Blank Component** [shortcut: **T** , **W**], however, the new library already has a blank footprint, so we will use that one.

> To change the snap grid while you are working press CTRL + G.
> To display or hide the visible grids press the L key to display the *View Configurations* dialog.
> If the origin marker is not displayed, open the *View Configurations* dialog and enable the Origin Marker option in the View Options page.

3. To rename this default blank footprint, double click on its name in the list in the **PCB Library** panel (it will be called something like PCBComponent_1). Let's name the footprint the same as the one used earlier in the tutorial, BCY-W3, type in the new footprint name in the *PCB Library Component* dialog.

4. It is recommended that you build the footprint around the workspace 0, 0 reference point, indicated by the origin marker. Use the [shortcut **J** , **R**] to jump the cursor to the origin at any stage while you are working.

The reference point is the point you will be 'holding' the component by when you place it. Typically, the reference point is either the center of pad 1 or the geometric center of the component. The reference point can be set to either of these at any time using the **Edit » Set Reference** submenu options.

Placing Pads on a New Footprint

The *Pad* properties dialog has a viewer which allows you to inspect the pad shapes on the defined layers. You will be able to define normal circular, oval (slotted) or square holes in pads and toggle their plated property (plated or unplated) and all the work needed to support thermal reliefs generation, clearances calculation, output to Gerber, ODB++ and NC Drill for example will be automatically handled. The NC Drill Output (NC Drill Excellon format 2) will generate up to six different NC files for three different hole kinds and whether or not they are plated or non-plated. One of the most important procedures in creating a new component footprint is placing the pads that will be used to solder the component to the PCB. These must be placed in exactly the right positions to correspond to the pins on the physical device.

To place the pads:
1. Select **Place » Pad** [shortcut: **P** , **P**] or click the button on the toolbar. A pad will appear floating on the cursor. Before placing the first pad, press the **TAB** key to define the pad properties. The *Pad* dialog displays (Figure 45).
2. Edit the various regions of the dialog as shown in Figure 45. This creates a stretched round pad.
3. Using the coordinates displayed in the Status bar, position the first pad at X:0, Y:-50, and click (or press **ENTER**) to place.
4. After placing the first pad, another will appear on the cursor. Position the cursor at X:0, Y:0, then click to place the second pad. **Note** : The pad designator is automatically incremented.
5. Position the cursor at X:0, Y:50, then click to place the third pad.
6. Right-click, or press **ESC** , to exit pad placement mode. The three pads should look like Figure 44.
7. Save the footprint by selecting **File » Save** [shortcut: **CTRL** + **S**].

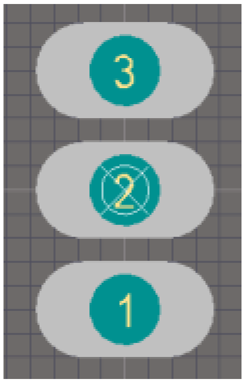
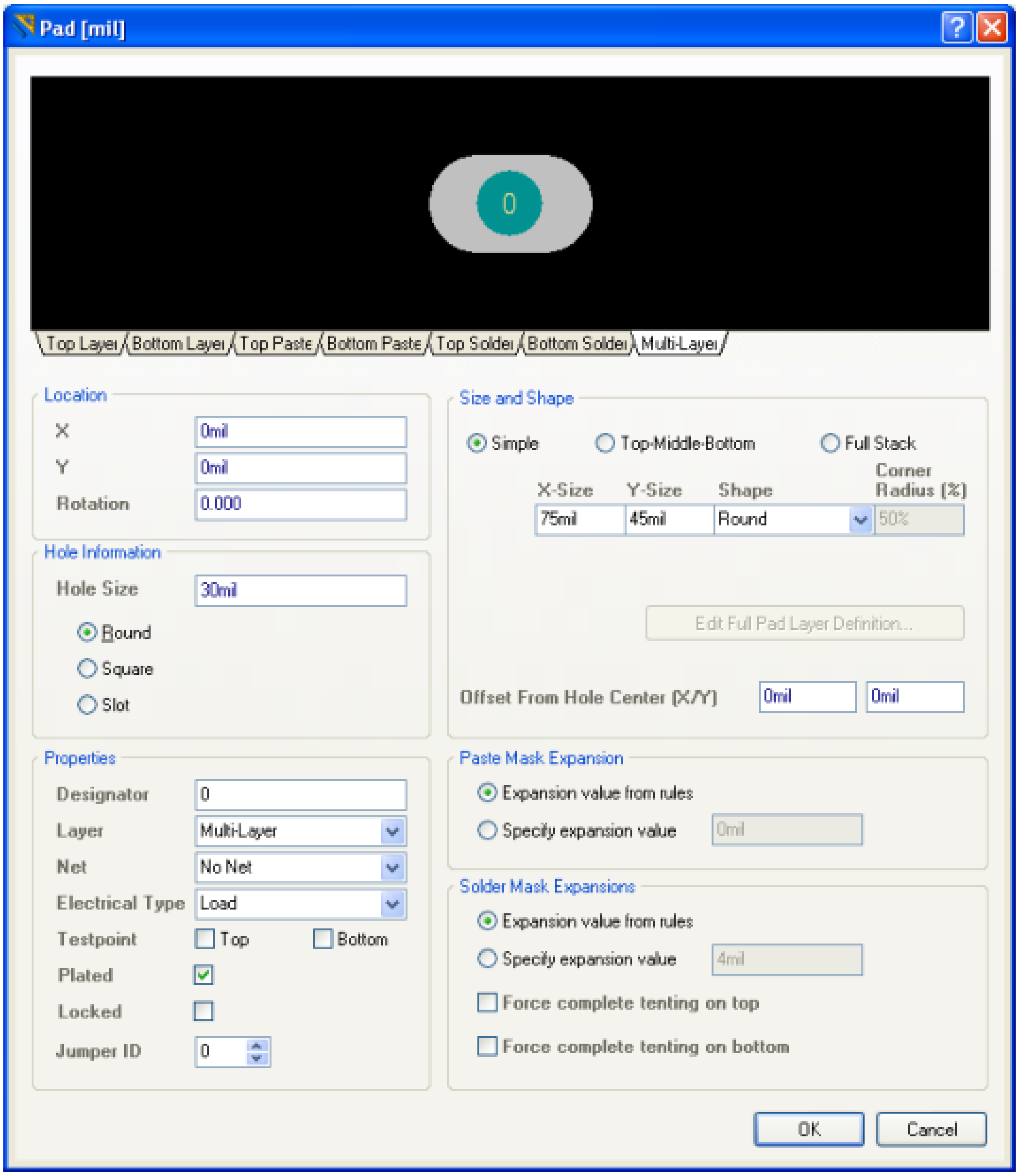


Figure 44. Stage 1, the pads placed.

> For a surface mount pad set the Layer property to Top Layer.
>
> For a thru-hole pad that has different size requirements for each layer, use the Size and Shape properties.



Location
X 0mil
Y 0mil
Rotation 0.000

Hole Information
Hole Size 30mil
● Round
○ Square
○ Slot

Properties
Designator 0
Layer Multi-Layer
Net No Net
Electrical Type Load
Testpoint □ Top □ Bottom
Plated ☑
Locked □
Jumper ID 0

Size and Shape
● Simple ○ Top-Middle-Bottom ○ Full Stack

X-Size	Y-Size	Shape	Corner Radius (%)
75mil	45mil	Round	50%

Offset From Hole Center (X/Y) 0mil 0mil

Paste Mask Expansion
● Expansion value from rules
○ Specify expansion value 0mil

Solder Mask Expansions
● Expansion value from rules
○ Specify expansion value 4mil
□ Force complete tenting on top
□ Force complete tenting on bottom

Figure 45. Set the properties of the pads before placing the first pad.

To position the pad that is floating on the cursor without the mouse, use the J, L shortcut to display the *Jump to Location* dialog. Press TAB to move between the X and Y fields, press ENTER to accept the changes, and press ENTER again to place the pad in the workspace.

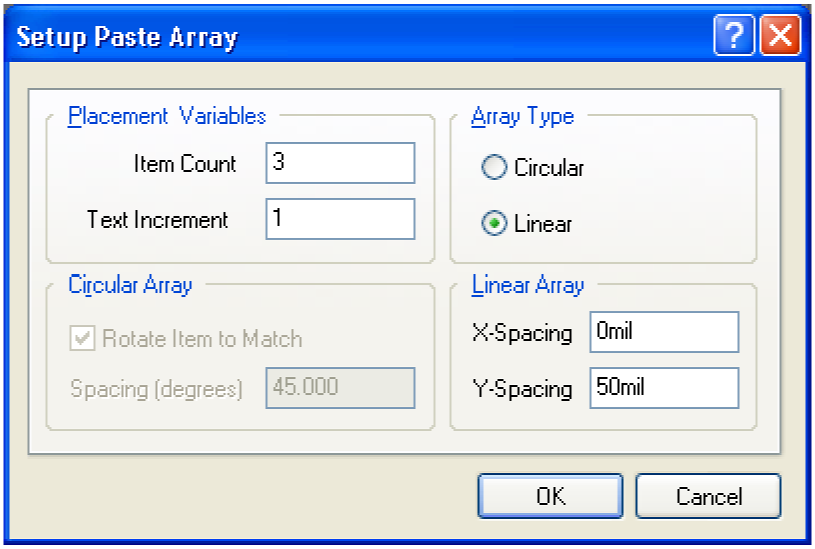
Pad Designators

Pads can be labeled with a designator (usually representing the component pin number) of up to 20 alphanumeric characters. The designator can be left blank if desired.
If the designator begins or ends with a number, the number will auto-increment when placing a series of pads sequentially. To achieve alpha increments, eg. 1A, 1B, or numeric increments other than 1, use the Paste Array feature.

Paste Array Feature

By setting the designator of the pad prior to copying it to the clipboard you can use the Paste Array feature to automatically apply a designation sequence whilst pasting the pads. By using the **Text Increment** field in the *Paste Array* dialog, the following pad designator sequences can be placed:

- Numeric (1, 3, 5)
- Alphabetic (A, B, C)
- Combination of alpha-numeric (A1 A2, or 1A 1B, or A1 B1 or 1A 2A, etc).



Setup Paste Array

Placement Variables
Item Count: 3
Text Increment: 1

Array Type
○ Circular
● Linear

Circular Array
☑ Rotate Item to Match
Spacing (degrees): 45.000

Linear Array
X-Spacing: 0mil
Y-Spacing: 50mil

Figure 46. Pasting muliple pads at once.

- To increment numerically, set the **Text Increment** field to the amount you wish to increment by.
- To increment alphabetically, set the **Text Increment** field to the letter in the alphabet that represents the number of letters you wish to skip. For example, if the initial pad had a designator of 1A, set the field to A, (first letter of the alphabet), to increment designators by 1. Set the field to C (third letter of the alphabet) and the designators will become 1A, 1D (three letters after A), 1G etc.

To use the Paste Array feature:

- Create the initial pad with the designator required, eg. 1A. Copy this pad to the clipboard using **Edit » Copy** [shortcut: CTRL + C]. Click on the pad center to define the copy reference point.
- Select **Edit » Paste Special** [shortcut: E, A]. The *Paste Special* dialog (Figure 46) displays.
- Click the **Paste Array** button to display the *Setup Paste Array* dialog, and configure as required.

Drawing the Outline on the Component Overlay

Press Q to toggle the coordinates from imperial (mil) to metric (mm).

The outline that appears on the PCB silkscreen is defined on the Top Overlay layer. If the component is flipped to the bottom of the board during placement, the overlay is automatically transferred to the Bottom Overlay layer.
1. Click on the **Top Overlay** layer tab at the bottom of the main editing window before placing overlay objects such as arcs or lines (tracks).
2. First, we will place the arc, as shown in Figure 47. To place the arc, select **Place » Arc (Center)** from the menus. Position the cursor at X:0, Y:0 and click to define the arc center. If you know the arc radius and start and end angles it is actually easier to complete the arc placement without trying to interactively define these settings, then edit the

placed arc through the settings in the *Arc* dialog.
3. Click somewhere to approximately define the arc radius, then click to define the arc start angle. If required you can press the **SPACEBAR** to toggle the direction the arc is rendered before defining the end angle, set the render direction as shown in Figure 47 then click again to define the arc end angle. Right-click to exit arc placement mode.

> If you make a mistake during Line placement, press BACKSPACE to remove the last track segment.

Now double-click on the placed arc to display the *Arc* dialog, and set the properties as follows: **Width** =6mil, **Radius** =105mil, **Start Angle** =55, **End Angle** =305.
4.Next is the line. Select **Place » Line** [shortcut: **P** , **L**], or click on the button. Position the cursor near the end of the arc and press **PAGE UP** to zoom in, as shown in Figure 47. As you move the cursor close to the end of the arc it will snap to it, this is the electrical grid pulling the cursor to the end of the existing object. Click to start the line segment.
5. Press **TAB** to define the line width (6mil) and check the layer in the *Line Constraints* dialog.
6.Move the mouse until it is over the other end of the arc, then click again to define the other end of the line. **Note** : During line placement you can cycle through the different line corner modes by pressing the **SHIFT** + **SPACEBAR** key combination.
7. To exit line placement mode, right-click or press **ESC** .

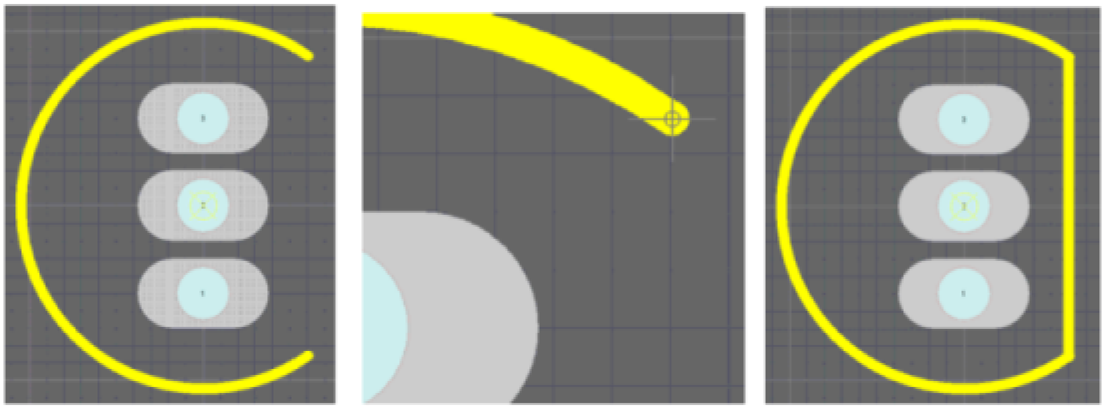


Figure 47. Placed arc, using the electrical grid to snap to the arc end when starting the line, the completed overlay.

Creating Footprints with an Irregular Pad Shape

There will be situations where you need to create a footprint with pads that have an irregular shape. This can be done using any of the design objects available in the PCB Llibrary Editor, but there is an important factor that you must keep in mind.

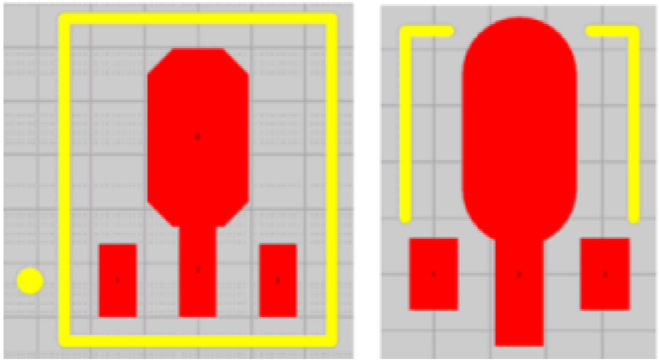


Figure 48. Create irregular shaped pads by placing multiple objects.

The software automatically creates solder and paste masks based on the shape of pad objects, if you use pad objects to build up an irregular shape then the matching irregular mask shape will be generated correctly. If you build the irregular shape from other objects, such as lines (tracks), fills, regions or arcs, then you will also need to define any required solder or paste masks by placing suitably enlarged or contracted objects on the solder mask and paste mask layers.

Figure 48 shows two versions of an SOT-89 footprint created by different designers. The one on the left uses two pads to create the large irregular shaped pad in the center, the one on the right uses a pad and a line (track). This one would need to have the solder and paste masks defined manually.

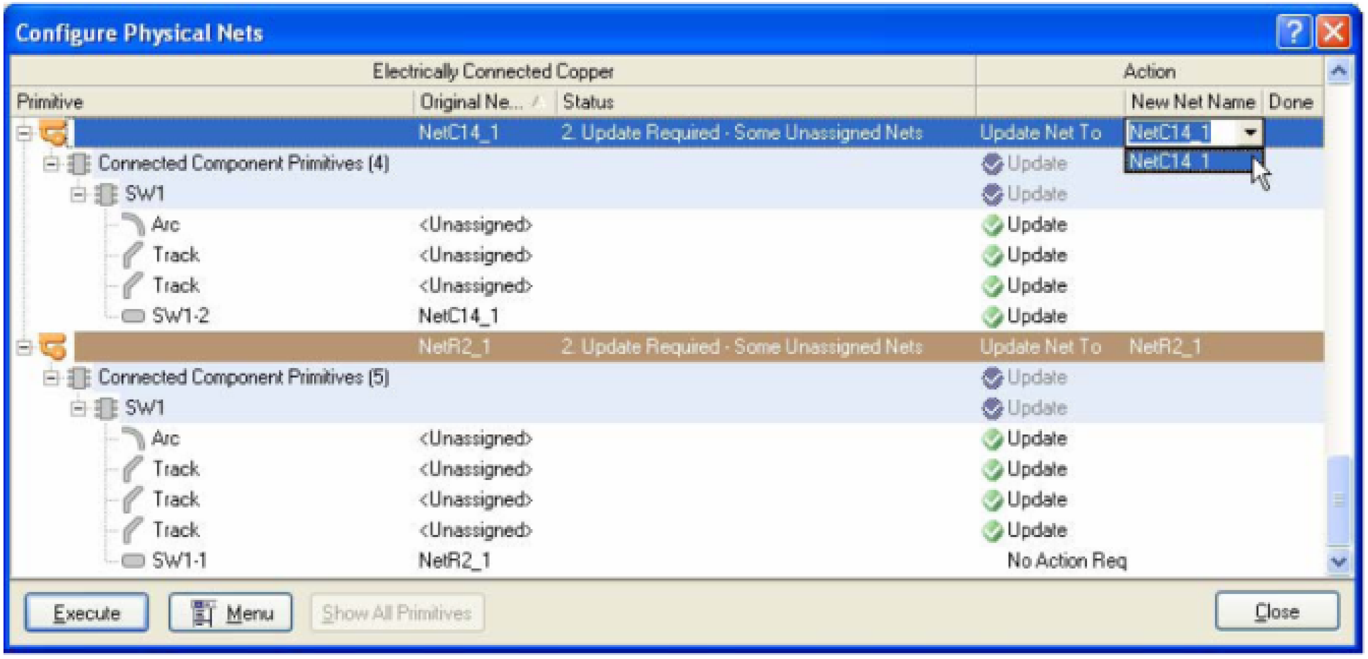
Info

Other examples can be found at the Wiki article Creating a Custom Pad Shape

Managing Components that Include Routing Primitives in their Footprint

When you transfer a design, the footprint specified in each component is extracted from the available libraries and placed on the board. Then each pad in the footprint has its net property set to the name of the net connected to that component pin in the schematic. If the footprint includes copper primitives touching the pads, these primitives will not be assigned the net name automatically, and will create a design rule violation. In this case, you will need to perform an update process to assign the net name.

The PCB Editor includes a comprehensive net management tool, to launch it select **Design » Net list » Configure Physical Nets** from the menus. Figure 49 shows the *Configure Physical Nets* dialog being used to update the extra primitives detected in the switch footprint shown in Figure 51. Click the **Menu** button for a menu of options, and click the **New Net Name** region to select the net to assign to the unassigned primitives.



Electrically Connected Copper			Action		
Primitive	Original Ne...	Status		New Net Name	Done
	NetC14_1	2. Update Required - Some Unassigned Nets	Update Net To	NetC14_1	
Connected Component Primitives (4)			Update	NetC14_1	
SW1			Update		
Arc	<Unassigned>		Update		
Track	<Unassigned>		Update		
Track	<Unassigned>		Update		
SW1-2	NetC14_1		Update		
	NetR2_1	2. Update Required - Some Unassigned Nets	Update Net To	NetR2_1	
Connected Component Primitives (5)			Update		
SW1			Update		
Arc	<Unassigned>		Update		
Track	<Unassigned>		Update		
Track	<Unassigned>		Update		
Track	<Unassigned>		Update		
SW1-1	NetR2_1		No Action Req		

Figure 49. Update the net name on unnamed footprint primitives in the Configure Physical Nets dialog.

Footprints with Multiple Pads Connected to the Same Pin

The footprint shown in Figure 50, a TO-3 transistor, has multiple pads that are connected to the same logical schematic component pin. For this component both of the two mounting hole pads have the same designator of '3'. When the **Design » Update PCB** command is used in the Schematic Editor to transfer design information to the PCB, the resulting synchronization will show the connection lines going to both pads in the PCB Editor, ie. they are on the same net, as shown in Figure 50. Both of these can be routed.

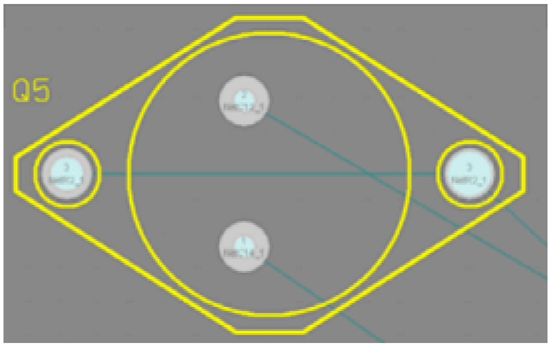


Figure 50. TO-3 footprint showing two pads with a designator of 3, on the same net

Handling Special Solder Mask Requirements

The footprint shown in Figure 51 is the contact set for a push button switch, which is implemented directly in the copper on the surface layer of the PCB.

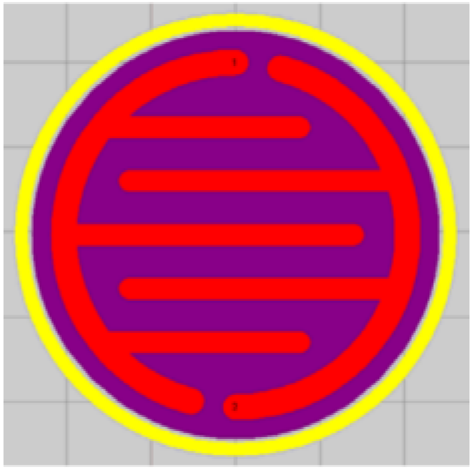


Figure 51. Printed push button footprint, designed by placing pads, lines and arcs.

A rubber switchpad overlay is placed on top of the PCB, with a small captive carbon button that contacts both sets of fingers in the footprint when the button is pressed, creating the electrical connectivity.

For this to happen, both sets of fingers must not be covered by the solder mask. The circular solder mask opening has been achieved by placing an arc whose width is equal to or greater than the arc radius, resulting in the solid circle shown behind the two sets of fingers.

Each set of copper fingers has been defined by an arc, horizontal lines, and a pad. The pads are required to define the points of connectivity.

Note : Manually placed solder mask definitions are automatically transferred when the component is placed on the bottom of the board.

Other Footprint Attributes

Solder and Paste Masks

Solder and paste masks are created automatically at each pad site on the Solder Mask and Paste Mask layers respectively. The shape that is created on the mask layer is the pad shape, expanded or contracted by the amount specified by the Solder Mask and Paste Mask design rules set in the PCB Editor, or specified in the *Pad* dialog.

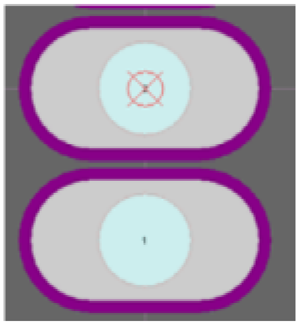


Figure 52. Pads with the solder mask displayed

When you edit a pad you will see the settings for the solder mask and paste mask expansions. While these settings are included to give you localized control of the expansion requirements of a pad, you will not normally need them. Generally it is easier to control the paste mask and solder mask requirements by defining the appropriate design rules in the PCB Editor. Using rules you define one rule to set the expansion for all components on the board, then if required you can add other rules that target any specific situations - such as all instances of a specific footprint type used on the board, or a specific pad on a specific component, and so on.

Displaying the Masks

To check the solder and/or paste masks have been automatically defined in the PCB Library Editor, we will turn on the Top Solder layer and examine the contents.
1. To make the mask layers visible, open the *View Configurations* dialog (**Tools » Layers & Colors** [shortcut: **L**]) and enable the **Show** option for each mask layer then click **OK** .
2.Now click on the layer tab, eg. **Top Solder** , at the bottom of the design window to view the top solder mask, as shown in Figure 52. **Note** : The ring that appears around the edge of each pad in the color of the Top Solder Mask layer represents the edge of the solder mask shape protruding by the expansion amount from under the multilayer pad (because *multilayer* is at the top of the layer drawing order, it is drawn on top. The layer drawing order is set through the **PCB Editor - Display** page of the *Preferences* dialog).

Setting Mask Expansions by Design Rules

To set the mask expansions in the design rules:
1. Confirm that the **Expansion value from rules** option is enabled in the **Paste Mask Expansion** and/or **Solder Mask Expansion** sections of the *Pad* dialog.
2. With a PCB open (you can simply create a temporary new PCB file if you do not have a PCB open) select **Design » Rules** from the PCB Editor menus and examine the Mask category design rules in the *PCB Rules and Constraints Editor* dialog. These rules will be obeyed when the footprint is placed in the PCB. **Note** : The rule system is hierarchical, you can define a higher priority rule to selectively override the general rule that applies to the entire board, if required.

Manually Specifying Mask Expansions

To override the expansion design rules and specify a mask expansion as a pad attribute:

1. Select **Specify expansion value** in the **Paste Mask Expansion** and/or **Solder Mask Expansion** sections of the *Pad* dialog.
2. Type the required value(s) and click **OK** . Save the footprint.

Designator and Comment Strings

Default Designator and Comment Strings

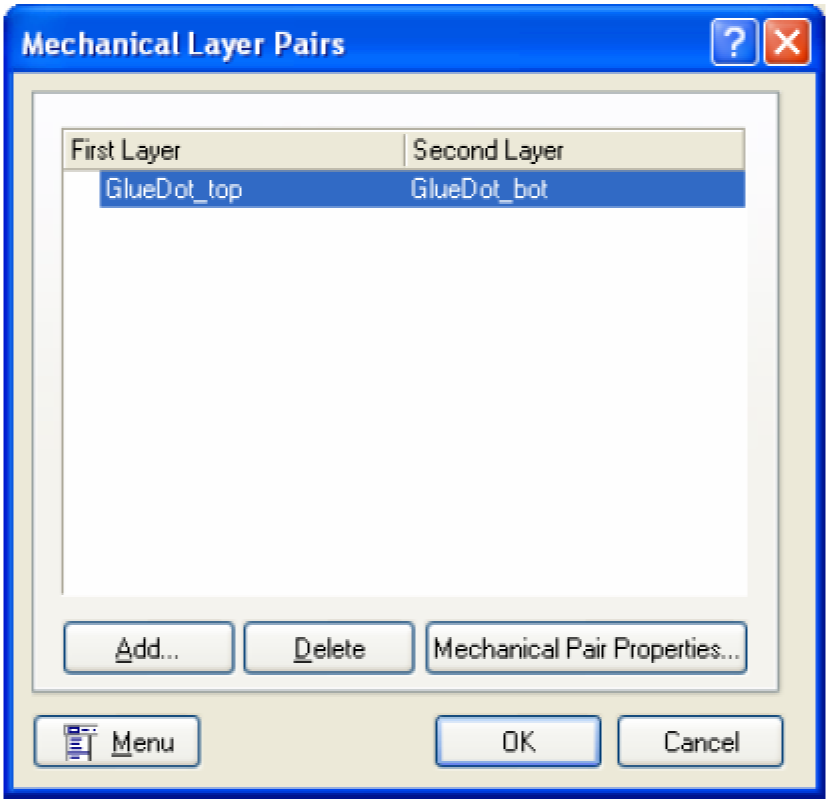
What you are building in the library is a footprint. When that footprint is placed on a board it is given a designator and comment - then it is referred to as a component. You do not need to manually define placeholders for the designator and comment strings when you build the footprint, these are added automatically when the footprint is placed on a board. The locations of these strings is determined by the designator and comment string **Autoposition** options in the *Component* dialog. You can pre-define the required string position (and size) in the **PCB Editor - Defaults** page of the *Preferences* dialog.

Additional Designator and Comment Strings

There may be situations where you would like additional copies of the designator or comment strings - for example your assembly house wants a detailed assembly drawing with the designator shown within each component outline, while your company requires the designator to be located just above the component on the component overlay on the final PCB. This requirement for an additional designator can be achieved by including the .Designator special string in the footprint (there is also a .Comment special string). To cater for your assembly house you would place the .Designator string on a mechanical layer in the library editor, and then generate a printout that included this layer.
If you need this feature, you would:
1.Display the required mechanical layer by enabling the **Show** and **Enable** options for each mechanical layer in the *View Configurations* dialog (**Tools » Layers & Colors**).
2. Click on the *Mechanical* layer tab at the bottom of the design window to activate this layer. The tab will be highlighted and all new text will be placed on this layer.
3. Select **Place » String** [shortcut: **P** , **S**] or click on the **Place String** button .
4. Press the **TAB** key to display the *String* dialog, where you can type in the text string and define its properties, eg. font, size and layer etc, before you place the text. Select .Designator from the **Text** list. Set **H eight** to 40mil and **Width** to 6mil and click **OK** . The bottom left corner of the actual designator will locate where the dot in the *.Designator* string is located.
5. Now we can place the text string. Press the **S PACEBAR** to rotate the text string, position it in the required location, and click to place it. Right-click or press **ESC** to exit string placement mode.
6.If required. place the *.Comment* special string using the same procedure.
7.To test the special strings, place the footprint on a PCB. You can place the footprint by right-clicking on its name in the **PCB Library** panel and selecting **Place** (assuming there is a PCB open). If the designator is not displaying when the footprint is placed in the PCB document, make sure that the **Convert Special Strings** option is selected in the **View Options** page of the *View Configurations* dialog in the PCB Editor.

Handling Special Layer-specific Requirements such as Glue Dots

There are a number of special requirements a PCB component can have, such as needing a glue dot, or a peelable solder mask definition. Many of these special requirements will be tied to the side of the board that the component is mounted on, and must flip to the other side of the board when the component is flipped.
Rather than including a large number of special purpose layers that may rarely be used, Altium Designer's PCB editor supports this requirement through a feature called *layer pairs* . A layer pair is simply two mechanical layers that have been defined as a pair, whenever you flip a component from one side of the board to the other, any objects on a paired mechanical layer are flipped to the other mechanical layer in that pair.



First Layer	Second Layer
GlueDot_top	GlueDot_bot

Figure 53. Layer pairs are defined in the PCB Editor

Using this approach you select a suitable mechanical layer to include the glue dot (or other special requirement), and define its shape using the available objects. When you place that footprint onto a board you must set up the layer pairing, this instructs the software which layer it must transfer objects to when this component is flipped to the other side of the board. **Note** : You cannot define layer pairs in the PCB Library Editor, this is done in the PCB Editor.

Including Three-Dimensional Component Detail

Given the density and complexity of today's electronic products, today's PCB designer must consider more than the horizontal component clearance requirements, they must also consider height restrictions and component-under-component placement options. There is also the need to transfer the final PCB to a mechanical CAD tool, where a virtual product assembly can verify the complete packaging of the product being developed. Altium Designer includes a number of features, including realistic 3D visualization, to cater for these different situations.

Adding Height to a PCB Footprint

At the simplest level you can add a height attribute to your footprint. To do this, double-click on the footprint in the **C omponents** list in the **PCB Library** panel to display the *PCB Library Components* dialog and enter the recommended height for the component in the **Height** field.
Height design rules can the be defined during board design (select **Design » Rules** in the PCB Editor), typically testing for maximum component height in a class of components, or within a room definition.

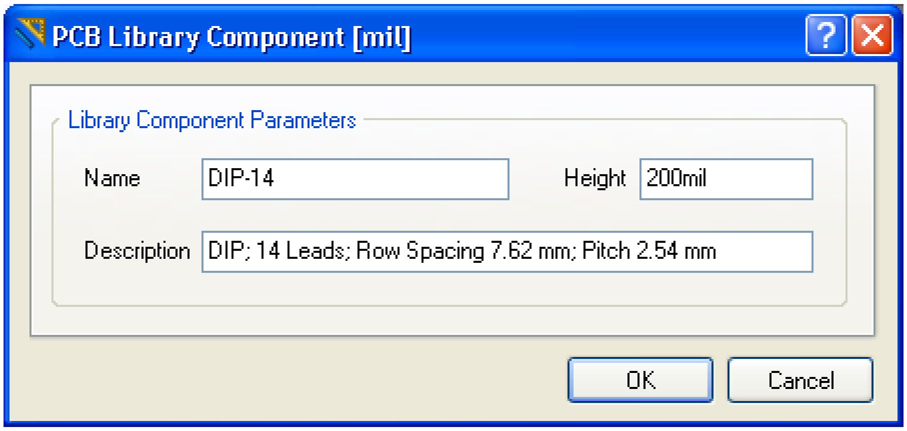
Adding a 3D Body to a Footprint

For more realistic component rendering in 3D view mode [shortcut: **2** (2D), **3** (3D)] in the PCB Library Editor), you can add 3D body objects to the footprint. 3D bodies can be added to a footprint on enabled mechanical layers only. An extruded (simple) 3D body is a 2D polygon-type object that has surface color and a height attribute to pull or extrude the shape when rendered in 3D. 3D bodies can also be created as spheres or cylinders.

One or more 3D bodies can be combined to define the physical size and shape of a component in all directions and are used by the Component Clearance design rule. Using high accuracy 3D models improves component clearance checking accuracy and generally improves the visual appeal and realism of the finished PCB assembly.

Altium Designer supports directly importing 3D STEP models (*.step or *.stp files) into PCB footprints to render the 3D model. This functionality extends to having STEP models either embedded or linked to Altium Designer PCB documents, however, linked STEP models are not available in the PCB Library Editor.

Note : 3D bodies will flip to the other side of the board when the component is flipped. However, if you want the 3D body data (which is stored on a mechanical layer) to also be flipped on to another mechanical layer, you will need to define a layer pair in the PCB document, as discussed in the section Handling special layer-specific requirements such as glue dots .



Name: DIP-14

Height: 200mil

Description: DIP; 14 Leads; Row Spacing 7.62 mm; Pitch 2.54 mm

Figure 54. DIP-14 component details

Manually Placing 3D Bodies

3D bodies can be placed manually in the PCB Library Editor (**Place » 3D Body**). They can also be added automatically to footprints in the PCB Library Editor (and to placed footprints in the PCB Editor) using the 3D Body Manager dialog (**Tools » Manage 3D Bodies for Library/Current Component**).

Note : You can place 3D bodies in either 2D or 3D modes.

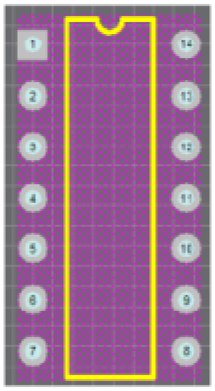
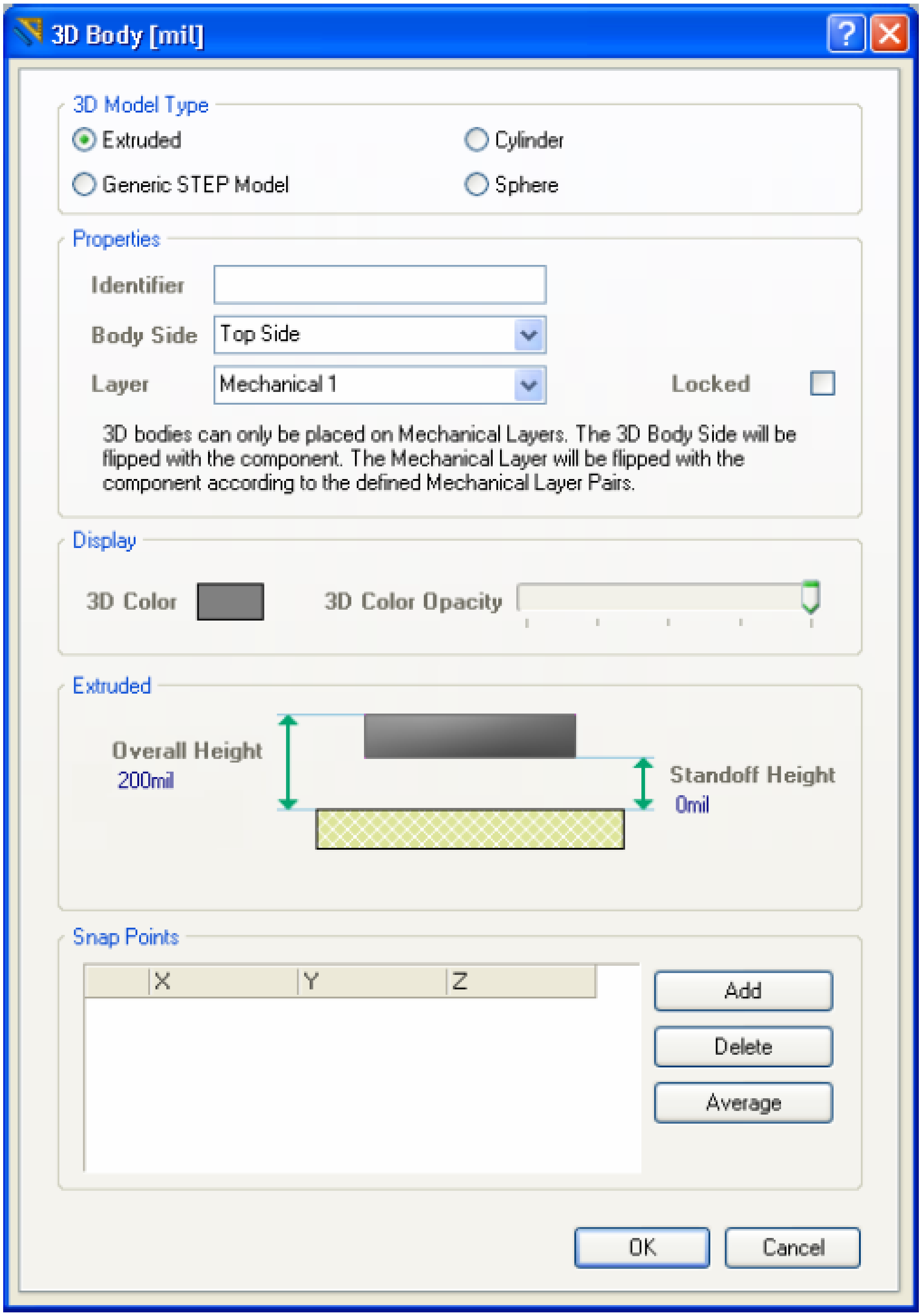


Figure 55. DIP-14 footprint with 3D body.

We will now add a 3D body to footprint DIP-14, which we created previously in this tutorial. To manually place a 3D body in the PCB Library Editor:

1.In the **PCB Library** panel, double-click the **DIP-14** entry to open the *PCB Library Component* dialog (Figure 54), which details the name, height and description. The height of the component is important here as we need to make the 3D body representative of the actual height.

Note : In cases where the manufacturer's data for component dimensions is available, use that.



3D Body [mil]

3D Model Type

● Extruded ○ Cylinder

○ Generic STEP Model ○ Sphere

Properties

Identifier:

Body Side: Top Side

Layer: Mechanical 1

Locked □

3D bodies can only be placed on Mechanical Layers. The 3D Body Side will be flipped with the component. The Mechanical Layer will be flipped with the component according to the defined Mechanical Layer Pairs.

Display

3D Color

3D Color Opacity

Extruded

Overall Height 200mil

Standoff Height 0mil

Snap Points

X	Y	Z

Figure 56. Define 3D body properties in the 3D Body dialog.

2. Select **Place » 3D Body** . After launching the command, the *3D Body* dialog (Figure 56) appears. Select the **Extruded** option in the **3D Model Type** region.
3. Use the controls in the **Properties** region to give the 3D body object an identifying name (**Identifier**), Leave the **Body Side** (which side of the board the 3D body should project vertically) as **Top Side** .
Note : You can enter a negative standoff height for components that protrude *through* the PCB, such as pins. Standoff heights are not checked by the Design Rules Checker.
4. Define the **Overall Height** as 200mil and **Standoff Height** (distance from the board to the underside of the 3D body) to 0mil; set the **3D Color** to a suitable color
5. Click **OK** to close the *3D Body* dialog and enter placement mode. The cursor will change to a crosshair in 2D, or a blue-cone cursor in 3D.
6. Position the cursor and click to anchor the starting point for the body, then continue to anchor a series of vertex points that define the polygonal shape of the body.
7. After placing the final vertex point, right-click or press **ESC** to complete placement of the body. There is no need to "close" the polygon as the software will automatically complete the shape by connecting the start point to the final point placed.
While defining the shape, use **SHIFT** + **SPACEBAR** to cycle through various corner modes. Modes available are: any angle, 45°, 45° with arc, 90° and 90° with arc. Arcs can be increased or decreased in radius using **SHIFT** + **.** (period or full stop) or **SHIFT** + **,** (comma) respectively. Use **SPACEBAR** to toggle the direction of the corner.
When an extruded 3D body object is selected, editing handles are displayed at each vertex. When the cursor changes to ⤢ over a handle, click and drag to move the vertex. When this cursor appears over the middle of an edge, click & drag to add a vertex to that edge and move it.

When the cursor changes to ✥ over an object edge, click & drag to move that edge.

When the cursor changes to ✥ over the object, click and drag to move the 3D body. The 3D body can be rotated or flipped while dragging. Use the editing controls to adjust the shape of the 3D body.

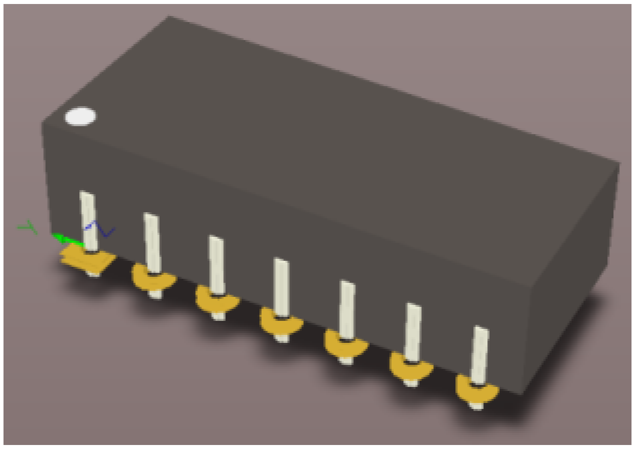


Figure 57. Example of a DIP14 3D representation. This model uses 16 3D bodies - extrusions for the main body and legs and a cylinder for the pin 1 reference marker.

Use the **BACKSPACE** key while in placement mode, to remove the last placed vertex point. Repeatedly use this key to 'unwind' the outline for the polygon, right back to the initial starting point.

This shape would be suitable the Component Clearance design rule, but may not be precise enough for 3D visualization. You can design and add more 3D bodies for further component detail.

After finishing the 3D body, the *3D Body* dialog will appear. Continue placing further 3D bodies, or click **Cancel** or press **ESC** to close the dialog. Figure 57 shows a DIP14 3D created in Altium Designer.

To view the 3D body in 3D at any time, press **3** to enter 3D viewing mode. If you cannot see the 3D body, press **L** to open the *View Configurations* dialog and enable the **Show Simple 3D Bodies** option on the **Physical Materials** pa ge or use the **3D Bodeis Display Options** controls on the **PCB** panel, when its in 3D Models mode. To return to 2D mode, press **2** .

Save the PCB library.

Interactively Creating 3D Bodies

Interactively creating 3D body objects from a footprint is very similar to the manual method. The basic difference is using Altium Designer to detect closed shapes that can be used to "extrude" into 3D bodies from the existing objects that comprise the footprint details. This is accomplished through the *3D Body Manager* dialog.

Note : Only *closed* polygons will create 3D body objects.

We will use the *3D Body Manager* dialog to define a 3D body for the transistor package, TO-39. Using this approach is easier than attempting to define the shape manually, because of the curved shape and orientation tab of the package body.

To use the *3D Body Manager* dialog:

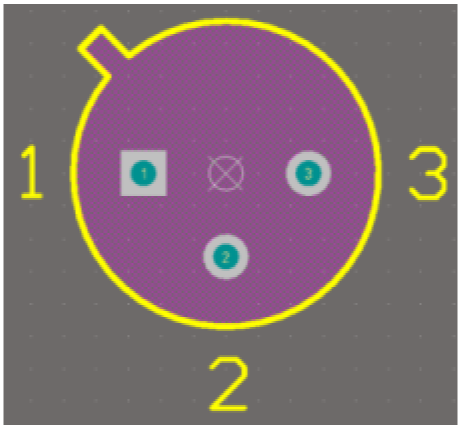


Figure 58. TO-39 2D footprint with 3D body added.

1. Make the TO-39 footprint the active footprint in the library.
2. Select **Tools » Manage 3D Bodies for Current Component** . The *3D Body Manager* dialog appears.

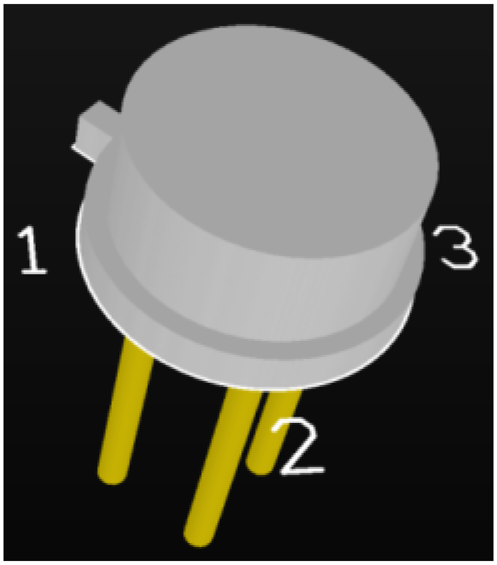
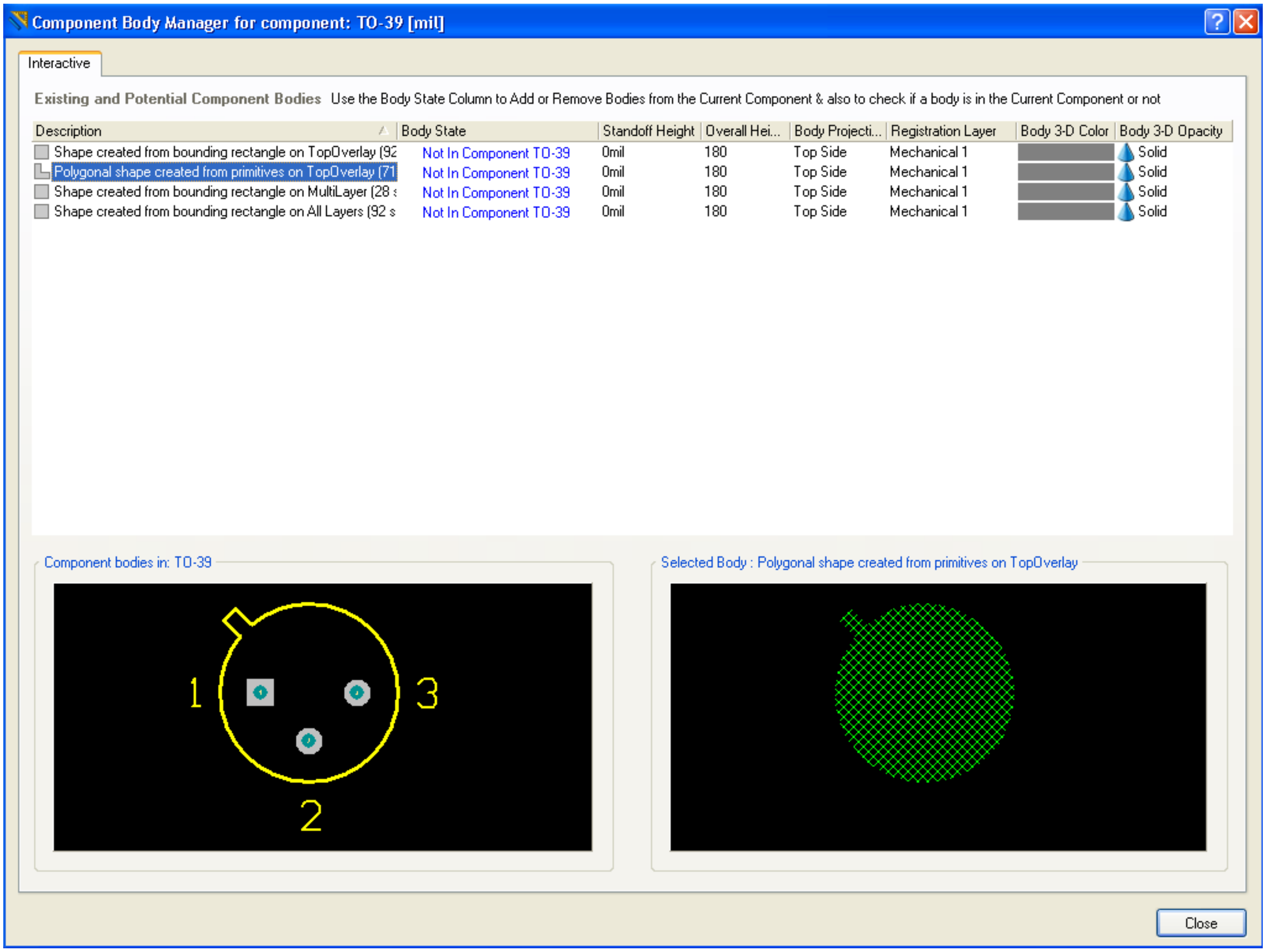


Figure 59. TO-39 3D model.



Existing and Potential Component Bodies Use the Body State Column to Add or Remove Bodies from the Current Component & also to check if a body is in the Current Component or not

Description	Body State	Standoff Height	Overall Hei...	Body Projecti...	Registration Layer	Body 3-D Color	Body 3-D Opacity
☐ Shape created from bounding rectangle on TopOverlay (92	Not In Component TO-39	0mil	180	Top Side	Mechanical 1	■	Solid
☐ Polygonal shape created from primitives on TopOverlay (71	Not In Component TO-39	0mil	180	Top Side	Mechanical 1	■	Solid
☐ Shape created from bounding rectangle on MultiLayer (28 s	Not In Component TO-39	0mil	180	Top Side	Mechanical 1	■	Solid
☐ Shape created from bounding rectangle on All Layers (92 s	Not In Component TO-39	0mil	180	Top Side	Mechanical 1	■	Solid

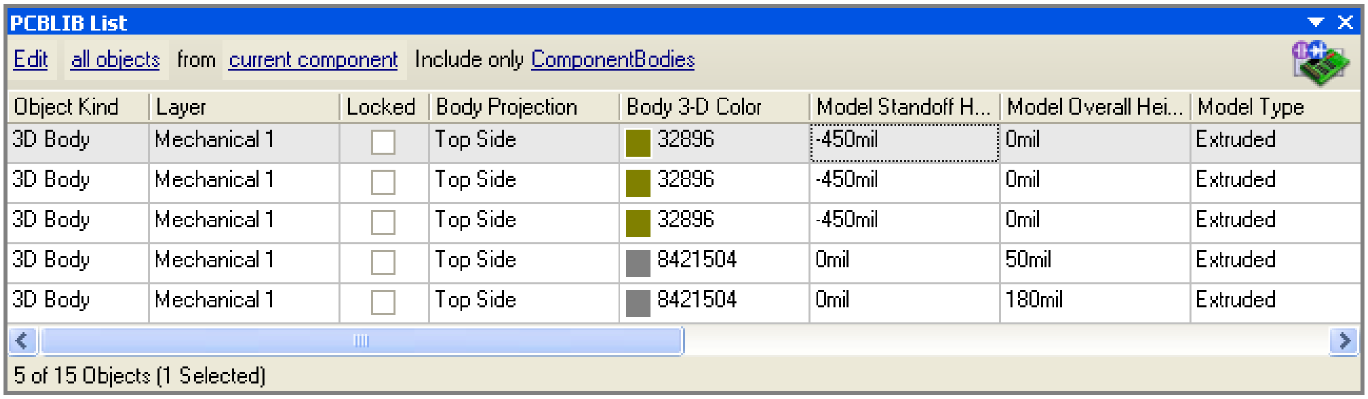
Component bodies in: TO-39

Selected Body : Polygonal shape created from primitives on TopOverlay

Figure 60. Use the 3D Body Manager dialog to quickly create 3D body objects based on existing primitives.

3.To create a shape that follows the outline defined on the component overlay we will use the second option that appears in the list, **Polygonal shape created from primitives on TopOverlay** . For this row in the dialog, click **Add to** (component_name) in the **Action** column, set the **Registration Layer** to the mechanical layer that the body object should be placed on (Mechanical1 in this case), set the **Overall Height** to a suitable value, eg. 180mil, and set the **Body 3D Color** to a suitable color, as shown in Figure 60. Scroll through the list to select any closed polygons that you want to use to define the component model.
4.Click **Close** and the 3D body shape will appear on the component, as shown in Figure 58. Save the library.
Figure 59 shows a completed 3D model for TO-39. This model consists of five 3D body objects:

- one for the base, which was created from the footprint outline (overall height 50mil, standoff height 0mil, Body 3D color gray)



Object Kind	Layer	Locked	Body Projection	Body 3-D Color	Model Standoff H...	Model Overall Hei...	Model Type
3D Body	Mechanical 1	□	Top Side	■ 32896	-450mil	0mil	Extruded
3D Body	Mechanical 1	□	Top Side	■ 32896	-450mil	0mil	Extruded
3D Body	Mechanical 1	□	Top Side	■ 32896	-450mil	0mil	Extruded
3D Body	Mechanical 1	□	Top Side	■ 8421504	0mil	50mil	Extruded
3D Body	Mechanical 1	□	Top Side	■ 8421504	0mil	180mil	Extruded

5 of 15 Objects (1 Selected)

Figure 61. PCBLib List panel showing 3D body details.

- one for the body of the casing, which was created by placing a circle and then taking the closed polygon from it as detected by the *3D Body Manager* dialog and given the following properties - overall height 180mil, standoff height 0mil, color gray
- one for the pins, which was created also by placing a circle (overall height 0mil, standoff height -450mil, color gold). This was then copied, pasted and positioned twice to make up the remaining pins. **Note: Copy and Paste of 3D bodies occurs in 2D mode, the copy command in 3D mode is for taking a snapshot for the clipboard.

To edit any 3D body, right-click it and select **Properties** from the pop-up menu to open the *3D Body* dialog (Figure 56). You can also use the **PCBLib List** panel (Figure 61) to list 3D bodies and edit them directly.
See the 3D Body section in the PCB Editor and Object Reference for further details on 3D bodies.

Importing a STEP Model as a 3D Body

Many component vendors supply detailed 3D models for use in popular mechanical CAD packages. Altium Designer can import 3D STEP models (**.step or**.stp) directly into a component footprint. This saves time in creating the model yourself and may provide a more sophisticated model too.
STEP files in the AP214 and AP203 formats are supported. The AP203 format does not support coloration - the entire imported model will have a generic shading.

Linked STEP Models

Linked STEP models are not supported in the PCB Library Editor. Embedded STEP models are supported.

Importing STEP Models

To import a STEP model, do the following:
1. Select **Place » 3D Body** [shortcut: **P** , **B**]. The *3D Body* dialog appears.
2. Select the **Generic STEP Model** option in the **3D Model Type** region.
3. Click the **Embed STEP Model** button. The *Choose Model* dialog appears, where you can browse for the *.step or *.stp file.
4. Locate the desired STEP file, select it, then click the **O pen** button to close the *Choose Model* dialog.
5. Back in the *3D Body* dialog, click **OK** to close it. The 3D body appears floating on the cursor.
6. Click in the workspace to place the 3D body object with the selected model loaded into it.

Positioning and Orienting STEP Models

When a STEP model has been imported, the placeholder 3D body re-sizes to house the model. The STEP model may not be oriented correctly in relation to the axes of the PCB document due to the origin used in the originating application. There are several methods for graphically positioning STEP models, using reference points (known as *s nap points*) placed on the model to manipulate it, and using faces or surfaces on the model in relation to the board. Non-graphical positioning can be carried out through the settings in the **Generic STEP Model** region of the *3D Body* dialog.

> **Refer to the Integrating MCAD Objects and PCB Designs d ocument for details on positioning and orienting STEP models.**

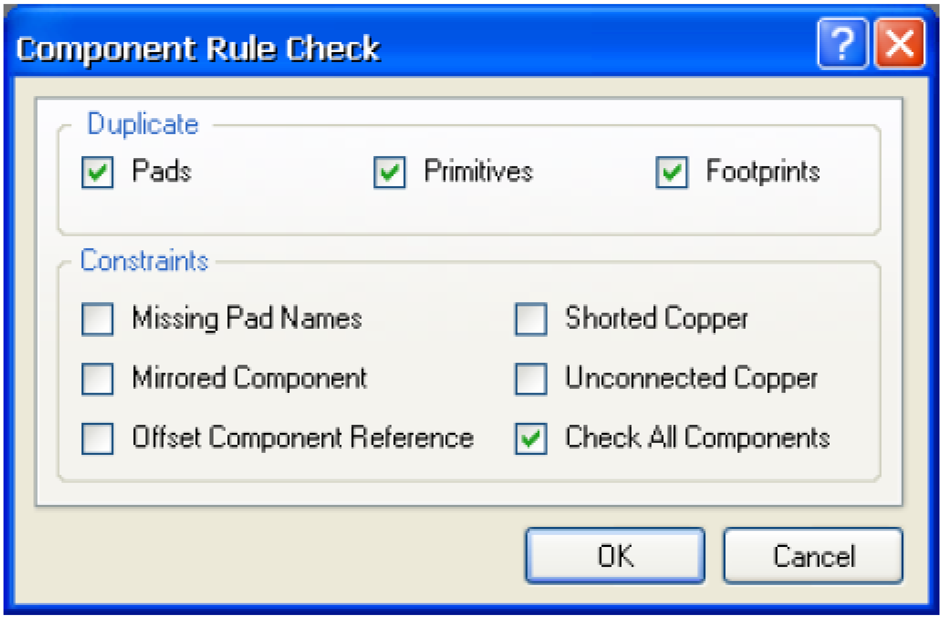
Adding Footprints from Other Sources

You can copy existing footprints into your PCB library. The copied footprint can then be renamed and modified to match the specifications required.
If you want to copy existing footprints to your PCB library, you can:

- select placed footprint(s) in a PCB document and copy (**Edit » Copy**) and paste them into an open PCB library using **Edit » Paste Component** , or
- Select **Edit » Copy Component** when the footprint to be copied is active in the PCB Library Editor, change to the open PCB destination library and select **Edit » Paste Component** , or
- select one or more footprints in the list in the PCB Library panel using the standard **SHIFT** + Click or **CTRL** + Click, right-click and choose **Copy** , switch to the target library, right-click in the list of footprint names and choose **Paste** .

Validating Component Footprints

As in the Schematic Library Editor, there are a series of reports that you can run to check the footprints have been created correctly and identify which components are in the current PCB library. To validate all components in the current PCB library, we will run the Component Rule Check report. The Component Rule Checker tests for duplicate primitives, missing pad designators, floating copper and inappropriate component reference.
1. Save your library file before running any of these reports.
2. Select **Reports » Component Rule Check** [shortcut: **R** , **R**] to open the *Component Rule Check* dialog.
3. Check all the boxes available and click **OK** . A report, named PCBlibraryfilename.err, is generated and opens in the Text Editor. Any errors will be noted.
4. Close the report to return to the PCB Library Editor.



Duplicate

☑ Pads ☑ Primitives ☑ Footprints

Constraints

□ Missing Pad Names □ Shorted Copper
□ Mirrored Component □ Unconnected Copper
□ Offset Component Reference ☑ Check All Components

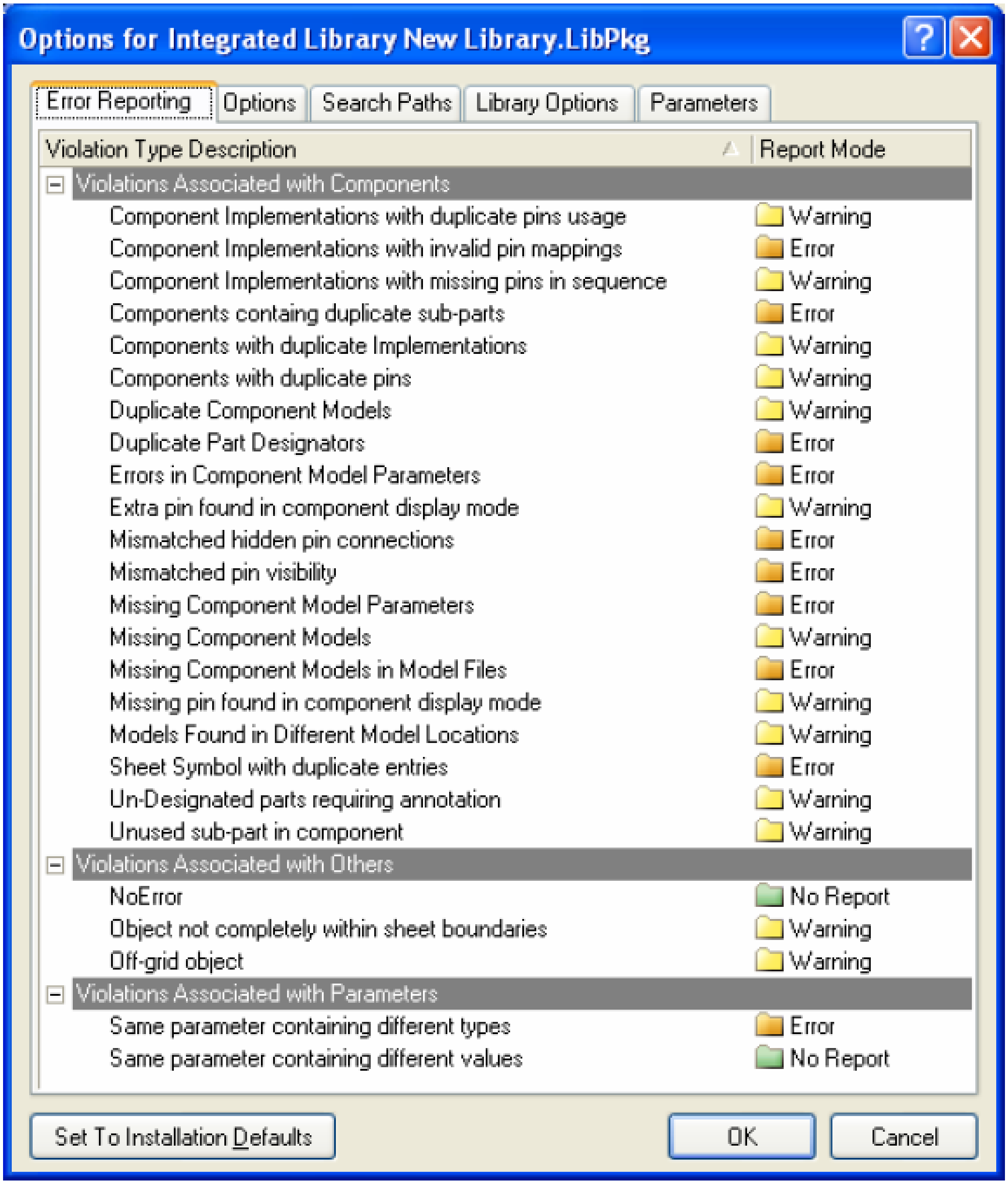
Figure 62. Verify the footprints in the library before using them in a design.

Creating an Integrated Library

So far we have:

- created an Integrated Library Package, the source project for a compiled integrated library,
- added a new schematic library to the Library Package, and created schematic components in it,
- specified various models to be used when the component is used in other domains, such as board design or circuit simulation,
- added a new PCB footprint library to the Library Package, and created footprints in it and created a 3D model
- learned how to handle a number of special footprint requirements.

For the last task in this tutorial we will now compile the Library Package to create an integrated library, creating a single file that includes the components and all their referenced models. Even if we did not want to use the integrated library, but preferred to work directly from the source library and model files, there is a strong incentive to compiling the Library Package. Doing this will perform an extensive set of checks on the components and the component-to-model relationships, as shown in Figure 63.


Options for Integrated Library New Library.LibPkg
Error Reporting
Options
Search Paths
Library Options
Parameters
Set To Installation Defaults
OK
Cancel

Violation Type Description	Report Mode
Violations Associated with Components	
Component Implementations with duplicate pins usage	Warning
Component Implementations with invalid pin mappings	Error
Component Implementations with missing pins in sequence	Warning
Components containg duplicate sub-parts	Error
Components with duplicate Implementations	Warning
Components with duplicate pins	Warning
Duplicate Component Models	Warning
Duplicate Part Designators	Error
Errors in Component Model Parameters	Error
Extra pin found in component display mode	Warning
Mismatched hidden pin connections	Error
Mismatched pin visibility	Error
Missing Component Model Parameters	Error
Missing Component Models	Warning
Missing Component Models in Model Files	Error
Missing pin found in component display mode	Warning
Models Found in Different Model Locations	Warning
Sheet Symbol with duplicate entries	Error
Un-Designated parts requiring annotation	Warning
Unused sub-part in component	Warning
Violations Associated with Others	
NoError	No Report
Object not completely within sheet boundaries	Warning
Off-grid object	Warning
Violations Associated with Parameters	
Same parameter containing different types	Error
Same parameter containing different values	No Report

Figure 63. Error checks performed during the compile.

To compile the Library Package:

1. Compile the source libraries and model files in the library package into an integrated library by selecting **Project » Compile Integrated Library** . Any errors or warnings found during compilation are displayed in the **Messages** panel (**View » Workspace Panels » System » Messages**). Double-click on an error in the **Messages** panel to view more information and jump to the component. Fix any inconsistencies in the individual source libraries at this point and recompile the integrated library.

2. A new *Integrated Libraryname.INTLIB* is generated, saved in the output folder nominated in the **Options** tab of the *Project Options* dialog. The new integrated library is automatically added to the installed libraries list and displays in the **Libraries** panel, ready to use.

Note, that you can also create an integrated library from a completed project using the **Design » Make Integrated Library**command - this will first create the source libraries and then the integrated library.

For more detailed information about integrated libraries, refer to the Building an Integrated Library tutorial.

Glossary

The following definitions are used in this tutorial.

Component	A component is a physical device that is placed on the board, e.g. the integrated circuit or resistor. Within these components, there may be either a single part or a set of parts that are packaged together.
3D body	A 3D body is a polygonal shaped object that can be added to a footprint, on any enabled mechanical layer. It can be used to define the physical size and shape of a component in the horizontal and vertical planes, enabling more controlled component clearance checking, and better 3D visualization. 3D body objects also act as placeholders for imported STEP models in the component footprint or as non-PCB mounted, free-floating objects, such as housings and assemblies.
Designators	Unique identifiers that are used to tell one component from another in a PCB. They can alpha, numeric, or a combination of both. Pads also have unique designators that correspond to the component pin numbers.
Footprint	A footprint defines (or models) the space required by the component to mount it on the PCB. The footprint model of a component is stored in a PCB library. A footprint may contain pads for connecting to the pins of a device and a physical outline of the package created from track and/or arc segments on the silkscreen (overlay) layer. Device mounting features may also be included. Footprints in the PCB library have no designator or comment. They become components when placed on a PCB sheet where the designators and comments are allocated.
Hidden pins	These are pins that exist on the component but do not need to be displayed. Typically, they are power pins and can automatically be connected to a specified net.

Library	A Schematic Library is a set of components and its parts stored on individual sheets. A PCB Library contains the component footprints. Each library type has its own Editor. Integrated libraries combine schematic libraries with their related models and cannot be edited directly by the Library Editors.
Object	Any individual item that can be placed in the Library Editor workspace.
Pads	Pad objects are normally used in a footprint to create connection pads for component pins.
Part	A collection of graphical objects represents one part of a multi-device component. Parts are stored in separate sheets within components in the schematic component libraries.
Pins	Component pins give a component its electrical properties and define connection points on the component.