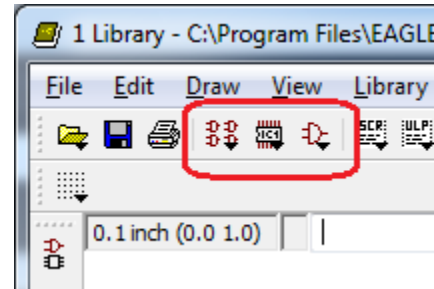


## Creating Eagle Parts Libraries

The Eagle library interface is composed of three main sections: A **symbol** editor where a part schematic symbol is drawn, a **package** editor where the part layout is created, and a **device** editor where the symbol and package drawings are linked. The feel of the library symbol and package editors are very similar to the schematic and layout editors, with many of the same commands acting in the same way. You can assume that the many of the basic commands described in document 3.1.Eagle Commands apply.

To select between the different editors, click on one of the highlighted buttons shown in the figure. These buttons are visible in any of three screens of the library editor. The left most button opens the device editor, the middle button opens the package editor, and the right most button opens the symbol editor.



### Symbol View

Clicking on the symbol button will allow you to create a new symbol by typing in a new symbol name, or edit an existing symbol by selecting the name of that symbol. A symbol is composed of pins (to which signals connects), a drawing of the symbol (for example, parallel lines for a capacitor, a jagged line for a resistor, or a box for a microcontroller), and designators showing the names of the pins and a place for the name of the part. Relevant commands that are necessary to accomplish this are **pin**, **wire**, **name**, **text**, and **change**.

**pin:** Pin allows the user to place pins. Pins have a number of properties, and calling the pin command will display these properties in a ribbon of buttons at the top of the editor.



The first four buttons control pin *orientation*. Pin orientation can also be cycled through by right clicking.

The second four buttons show the style of the pin – typically we only use the first style, but these pin styles can be used to indicate symbolically some property of the signal that will go on the pin. Of the four buttons, the first is normal, the second shows that signal is logically inverted, the third shows that it is a clock signal, and the fourth shows that it is an inverse clock signal.

The third set of four buttons just controls the length of the pin.

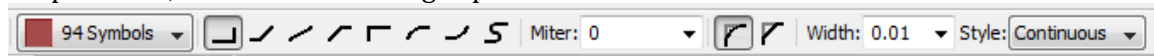
The fourth set of buttons controls how the pin is labeled – the first button turns labels off (good for passive components like resistors), the second shows the number of the pin on the physical package, the third shows the name of the signal, and the fourth shows both (recommended for any chip or IC).

Direction defines the way that data or power flow through the pin. I/O is the most general purpose of these options, and we recommend that you use it even for VCC or GND signals to limit silly ERC warnings.

Swaplevel should remain at 0 (the default).

**change:** Note that you can change the pin properties after they have been placed using the change command. When you type change you will be presented with a menu; you can use this to change direction, function, length, and visibility.

**wire:** The wire mode is different in the symbol editor – it is used to draw the symbol. Like the pin mode, the wire mode brings up a ribbon:



Of the options presented in this ribbon, only the first is important – typically we want the symbol drawing to be in layer 94 (symbols).

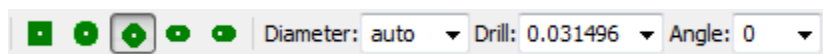
**name:** It is important to name the pins so that they match up roughly with the names of those pins in the datasheet. For example, the ATtiny had 8 named pins, including VCC, GND, and PB0-PB5. PB0-PB5 had descriptive names that described some of the special functions on each pin (for example, PB5/RESET). Note that when you are linking the pins of the symbol to the pins of your package drawing, the names you give those pins will be the only clue you have as to which pins link together. Thus, it is important to give descriptive names to your pins *even if you do not have the pin names displayed*.

**text:** The process of giving your part a name is a bit idiosyncratic. Essentially, you must create placeholder name and value text in your symbol. To do this, type 'text', enter '>NAME' for the placeholder name, and select layer 95. Place the resulting text where you would like your part's name to appear in your schematic. To add a value, enter '>VALUE' and select layer 96 and place it where the value should appear.

## Package View

The package view allows you to create a circuit board layout representation of the part. The part is typically composed of a number of holes or pads, a drawing or outline of the part, and designators showing the name of the part and giving information in how it should be oriented (if orientation is important). Important commands to know to draw the package are **pad/smd**, **name**, **wire**, and **text**.

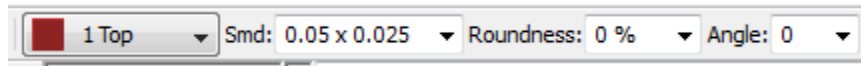
**pad/smd:** These two commands allow you to place either through hole pads (with 'pad') or surface mount pads (with 'smd'). Pads are where the electrical signals meet the component and are analogous to pins in the symbol view. If you type pad, you will see the following ribbon:



There are five options for selecting the shape of the pad – square, round, octagonal, narrow, and narrow with an offset. The diameter is the diameter of the full pad including the green area. Auto means that it will use design rules in the layout to select the diameter based on

the drill size. The drill is the diameter of the hole to be drilled – this is a parameter that comes directly from a datasheet. Angle is the orientation of the pad – right clicking changes the orientation by 90 degrees.

Smd also creates a ribbon:



Since surface mount pads only exist on one side of the board, you have the choice of placing them on the top or bottom. It is customary to use the top layer for this if your part has pads on only one side. The Smd dimensions are very important – you can enter your own or select from a wide range. These dimensions are typically provided as part of the description of the package in the datasheet. Note that by default, they are shown in inches, but you can change the unit of measurement with the grid command. Roundness is the % you wish the edge of the pad to be rounded; typically this is not desired. Angle is the orientation of the pad, right clicking changes this by 90 degrees.

**name:** All of the pads you create have names. Typically they are named P\$1, P\$2,...,P\$N in the order you placed them. You should rename them, even if it is only to 1, 2, ..., N. Make sure to pay attention to how the datasheet names the pads and to follow that order.

**wire:** Wire and some of the other tools for drawing (circle, arc, etc) allow you to draw the part outline. Regardless of what you draw, make sure you are drawing it in layer 21. If your part has directionality, make sure to indicate that either in the drawing (maybe by putting a circle next to pin 1) or with text.

**text:** As with the schematic, you will probably want to have your part designators appear on the layout. To do this, use text to add >NAME and >VALUE. >NAME should be added to layer 25, and >VALUE to layer 27.

## Device View

In order to complete a library part, you must use the device editor to link the schematic symbol and layout package you just created. To do this, you will need to do two things:

- Add a symbol
- Link the symbol to one or more packages. Many symbols have multiple packages, known as **variants** – for example, a resistor symbol may have through hole and surface mount packages associated with it.

To add a symbol, type add and select the name of the symbol in the list. Place it in the left hand view, preferably centered on the cross hairs.

To add package variants, click on the “New” button in the lower right corner. This gives you a list of all available packages in your library – only packages that have at least as many pads as the symbol has pins are selectable. This is a logical requirement of Eagle – a package can have more pads than a symbol has pins, but not fewer.

Select the package you would like to use, and give it a descriptive **variant name** by typing in the lower left text box of the package selection window.

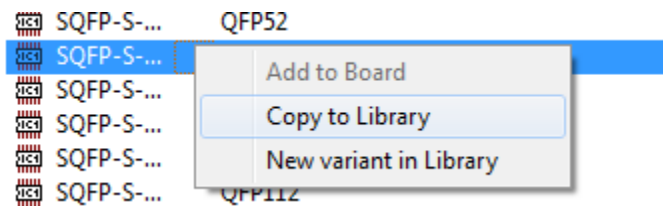
Now, to connect a symbol with a package, click on the “Connect” button in the lower hand corner. You will see a list of pin names and pads names – if you never bothered to name your pins or pads, you’ll see P\$1 through P\$n in both cases; if this is the case, go back to the symbol and package views and make sure to name your pins and pads. Select a pin to connect to pad and hit connect; do this until there are no pins left. **It is vitally important that you get this step right; make sure to refer to the datasheet frequently to check that you are connecting the correct schematic pins to the right physical pads.** With larger parts, it is often helpful to do this step while the datasheet is open and visible in an adjacent window.

Once you have connected the part properly, a green check mark will appear next to the variant name; this indicates that the part is ready to use!

## Copying Parts

It is possible to copy whole devices as well as layout packages from other libraries into your library. This is especially useful if you’d like a completed version of a complicated package to use yourself.

To copy a part or package have the library that you wish to copy the part into open in the library editor. Now in the Eagle Control Panel, expand the libraries tab. Look through these libraries for a part you’d like to copy. To copy the part or package, right click on it and select “copy to library”, as shown below:



Once copied, the part or package will show up in your library, in the device editor or package editor respectively.

Eagle comes with a large selection of libraries, but you may find a library online that you wish to copy parts from. To have such a library show up in your control panel view, you can either stick it in the Eagle lbr directory, or you can add the path of another folder. To add an additional path for library folders, go to Options->Directories, and add another folder path. Make sure you do not delete the existing path, just append to it and separate the two paths with a “;”. The following is an example of an edited path:

