A Methodical Approach to Library Work

Regardless of whether you are creating a component for a library or have taken on the arduous task of scrubbing a library, there needs to be a methodical approach. Both of these endeavors share a similar list of actions.

Library editors provide so many options for component creation that no two components will have the same look and feel if guidelines are not given and followed.

Even if by IPC or IEEE standards, there isn't much guidance when it comes to symbols other than declaring a symbol to represent the actual component. Though footprints must be drawn to accommodate the manufacturing process, each company may have different mechanical layers dedicated for different aspects of the design.

Therefore, the key to this effort is consistency.

Below are several tables provided to guide the user in designing robust and consistent components.

- In creating a component, all of the actions must be addressed in the tables to ensure consistency.

- In scrubbing a component, all of the components are being scrutinized against each action in the tables to ensure consistency.

Scrubbing one component at a time is tedious. One loses a synergy that could be obtained by looking at one aspect of all components at the same time. To give a practical example, if all of the discrete components need to be horizontally aligned with pin 1's hot spot at the origin, this can be checked methodically by looking at every component one at a time, even if it is a manual procedure. Depending on what needs to be addressed, some of the fixes can be remedied by global edits.

<u>It All Starts with Names</u>

The proper naming of the symbol and footprint is the first place to start. Yet this is one of the most neglected aspects of the component. The failure to properly name a symbol or footprint will have serious repercussions in the future if not addressed up front. With regards to names of components in the libraries, there are two concerns one must keep in mind:

1. Changing the symbol name will break the link between the part in the library and the part in the schematic editor. Note that if the library has already been used for other projects, those links within those projects will be broken. It may be a matter of retaining the old library to support prior projects while copying the library to make changes to support new projects.

2. Changing the footprint name will break both the link to the symbol and the link to the PCB editor.

The second situation is worse, since a careful and methodical approach must be taken when changing an existing footprint name. If the user simply goes into the footprint library and changes the names without keeping an *is/was list*, the relinking of the symbol to the footprint will be difficult at best.

Note - the terms *attribute* and *parameter* mean the same thing in this table. Both are defined as simply a name linked to a value.

Component Fundamentals		
	Task	**Guidelines / considerations**
1	Establish / repair symbol name	Be specific and consistent. For example, if the name needs to be capitalized, then all names need to be as such. If there needs to be a prefix for the name, there needs to be consistency in the prefixes used. If an underscore is being used to concatenate information, white spaces and other delimiters should not be used.
2	Establish / repair the footprint name	Consider using the *IPC-7351B Naming convention for Standard SMT Land Patterns* as the basis for the name. Decide if the company will use the M, L, N footprints, a footprint specific to that component, or all four Be sure to create an is/was list if the footprint is already linked to a symbol. Similar to the symbol name, be specific and consistent with the names.
3	Link / relink the footprint to the symbol	Refer to the is/was list if the link is not obvious.
4	Obtain the datasheet	If the library component represents a purchasable component, the datasheet needs to be obtained and downloaded to a company directory agreed upon by all. Do not rely on links provided by the manufacturer. Provide a name for the datasheet that describes the component. It should contain at least the manufacturer's name and manufacturer's Part number. If scrubbing a library, any purchasable component that does not have any reference to a company part number or a manufacturer's part number can be purged. One should take care not to remove templates or embedded components such as test points or fiducials.
5	Link the datasheet as a parameter	At a minimum, the file name should be captured as a parameter for the component if the directory for datasheets is well known by all members of the company; otherwise, provide another parameter for the directory path.

6	Obtain the component status	Regardless of whether the part is new or the library is being scrubbed, one should not remove an obsolete component. The symbol graphic should contain a marking to indicate it as such. A parameter called status could be used, though it will require manual updating. The reason why the obsolete component should not be purged is due to the fact that the library provides information to the designer. In this case, the information is "do not use this part." Without this information, the designer may attempt to add the part back into the library. This is especially true for components that designers have leaned on in the past.
7	Review the descriptions parameter for consistency	The description is a VITAL aspect of the component because it summarizes key aspects the component. Granted, there may be other parameters that provided characteristics in the library for the component; however, this field will find its way into a bill of materials, which in turn, will be uploaded to a purchasing database. In the purchasing database, this will be the most human readable summary of the component. This field can be extremely useful for component searches in large libraries, therefore consistency is key. The delimiters need to be consistent. If one was to import the descriptions to Excel, one should not have to repair the results due to inconsistent delimiters.
8	Link / copy description to PCB footprint	This refers to copying of the symbol's description field into the corresponding footprint when the netlist is being imported or ECO'd (Engineering Change Order) into the PCB. This information summarizes the component for which footprint represents. This is useful information that saves the designer the hassle of having to refer back to the schematic to obtain this information. This situation also depends upon the EDA tool's handling of the parameters. In some tools, all parameters may be allowed to propagate to the PCB editor; other tools may only allow a single parameter to propagate. Regardless, the component's description is very useful in the PCB editor.

9	Type review	The vast majority of components will need a footprint on the PCB, a reference in the bill of materials, and a disallowance of any copper shorts. However, there are exceptions. Depending on the complexity of the EDA tool, there may be ways of disabling some of these aspects of the component. The following are some know exceptions 1. A heat sink for an FPGA has no footprint which needs to find its way onto the bill of materials. 2. A shorted jumper that has been built into the board will connect two different nets. By default, the rule checkers will consider this a short. In addition, since this component is embedded into the copper, there is no need to list it in the bill of materials.

The following table outlines the tasks for the symbol graphic:

Symbol		
	Task	**Guidelines / considerations**
1	Ensure that the grid is consistent on all components	Since symbols are dimensionless by their nature, it is highly recommended that the default grid established by the EDA vendor be adhered to. This will allow for the copying of components from the EDA tool library or by those who may make their components available online.
2	Ensure that the designator prefix is correct	One can refer to IEEE 315, especially if the design needs to be seen by individuals outside of the company. A search on the web will provide prefix lists as well.
3	Ensure that the reference to the origin is consistent	Given that ICs and discrete components have a different graphical structure, there may be different provisions for handing each one.
4	Ensure that the pin hot spots are on the grid	Failure to do so may compromise the netlist in the design
5	Ensure reference designators appear in the same location	Similarly, if other values need to be displayed (i.e., value, tolerance, etc.), they should also be consistent
6	Ensure pin length is consistent	Pin lengths should be kept to a minimal to allow optimal wire routing. Given that ICs and discrete components have a different graphical structure, there may be different provisions for handing each one.

7	Ensure that any parameters that are displayed (i.e., designator, value, tolerance, etc.) appear in the same locations	The failure to do this will result in a great deal of editing by the designer in the schematic.
8	Ensure that colors used for the graphics are consistent.	
9	Ensure that fonts used for any text displayed is consistent	A commonly used font is highly recommended
10	Ensure that font sizes used for any text displayed is consistent	
11	Ensure that font color used for any text displayed is consistent	
12	Ensure that pin numbering is consistent for discrete components	For example, for 2 pin discretes, pin 1 is always consistent with the left pin (if horizontal) or with the top (if vertical).
13	Ensure the pin orientation is consistent for ICs and other non-passive components	The general rules of thumb: 1. Inputs on the left 2. Outputs on the right 3. Power and ground positioned to allow bypass capacitors in close proximity to the component Note that the pin number order is not as critical as the grouping of similar functions It is not recommended that pins be placed on the top or bottom of the IC symbol.
14	Check for hidden pins	No pin should be hidden due to the lack of assurance that the pin has the right net name associated to it.
15	Check for missing pins	Even if a pin is not used, it should still be present so that there is a complete 1:1 correspondence with the footprint pads.
16	Ensure that the pin numbers are visible	This includes “no connect” pins as well. This is for readability in the schematic.
17	Ensure that the pin names are visible	This includes “no connect” pins as well. This is for readability in the schematic.
18	Location of the pin numbers is consistent	
19	Location of the pin names is consistent	It is okay to use abbreviations. It is also okay to have wide IC components if the pin names are lengthy
20	For discretes, the overall length should match other discretes	

Intelligent Data

The parametric information that is provided with a component serves two functions:

1. To provide information within the component to assist the designer in making an intelligent decision during component section for the design.
2. To allow for the automated creation of a bill of materials.

It has to be remembered that in the end, the parameter information in each component REGARDLESS of library structure being used, is generally organized by the EDA tool in a database fashion for the purposes of sorting and filtering. Each row of this database represents a component, each column represents a parameter.

Some of the more commonly used parameters

1. Manufacturer's part name. If using abbreviations, must be consistent
2. Manufacturer's part number
3. Company part number
4. Datasheet link
5. Vendor name
6. Vendor part number
7. Vendor 2 Name
8. Vendor 2 Part number
9. Component Value

Intelligent Information		
	Task	**Guidelines / considerations**
1	Ensure that the name of the value is consistent for all components	Failure to do so will result in multiple columns of the same information in a spreadsheet, especially a bill of materials. Otherwise, the data will not be aligned in a single column.
2	Ensure that the format of the value conforms to other values for the parameter	The value for each parameter needs to conform to the format of other values of the same parameter. This allows for sorting and filtering. Even if the parameter does not require a value, there should be either: 1. Phrase, such as "Not Applicable" 2. Abbreviation, such as "NA" or "N/A" 3. Marking, such as a dash "-" Failure to add something gives two false impressions: 1. That it was skipped over. 2. That it is okay to skip it.
3	If scrubbing, evaluate extraneous parameters	Even if there are 10,000 components in a library, a new column of information will be generated if only one of the components has a parameter that is not found in the other components. Therefore, a decision needs to be made as to keeping it and filling in the rest of the 9999 components, or removing it. If push comes to shove, it is okay to have a note parameter to capture information that cannot be readily sorted into crisp categories. However, the note parameter should not be used as a catch all for sloppy or rushed component creation, nor should it have information that could be represented by another parameter.

The following table outlines the tasks for the footprint graphic:

Footprint		
	Task	**Guidelines / considerations**
1	Review the mechanical layers to ensure consistency	Failure to use consistent layers will result in graphical information being scattered on different layer in the PCB, making it difficult (if not impossible) to generate documentation without heavy editing.
2	Review footprint pads against the datasheet	Even if the component claims to conform to a standard package definition, this is no guarantee.
3	Ensure that the footprint is centered on the origin	Most EDA tools can generate pick and place data automatically, based on the center points of the component.
4	Ensure that the courtyard has been designed to consider spacing requirements of the component	The courtyard is still a necessary visual for the designer. The courtyard should take into consideration special aspects which are not obvious by looking at the footprint
5	Ensure that the component is properly represented on the assembly layer	The assembly layer is still used as a guide for hand soldering, in addition to providing a graphical layout for the operator setting up the pick and place equipment for assembly.
6	Ensure that the silk layer marking are consistent for each component	This may include font type and size. It also includes trace and arc width, locations, and when they are to be used.
7	Ensure the solder paste of the component conforms to the guidelines of the component manufacturer	By default, the solder paste layer is usually auto generated by the EDA tool; however, thermal pads or the use of copper primitives not associated to the pads may result in solder paste not being applied.
8	Ensure that the solder mask of the component conforms to the guidelines of the component manufacturer	By default, the solder mask layer is usually auto generated by the EDA tool; however, thermal pads or the use of copper primitives not associated to the pads may result in solder mask being applied in areas that should not have it.
9	Ensure that the 3D component is orientated correctly	If the EDA tool provides for it, the 3D component should be added to the footprint. It is a good sanity check to see if the landings of the footprint conform to the 3D representation of the component. It is generally necessary to adjust the orientation of the component. Unfortunately, there is no universal definition of origin between MCAD tools, therefore, adjustments need to be made.