KiCad Library Formats

The main purpose of this document is to cover the format of the schematic and footprint libraries of the KiCad EDA suite. KiCad itself also comes with documentation on the file formats. The purpose of this document is to *update* that original documentation, but not to replace it.

Symbol Library Format

The symbol library uses two files per library: a ".lib" file with the definition and drawing of the symbol and a ".dcm" file with descriptions and keywords. According to comments in the KiCad source code, it is planned to merge these files in a future version.

All dimensions are in mil, and angles in 0.1th of a degree. The positive vertical axis is pointing up (so the inverse of native display coordinates).

Lines in the library that start with a "#" are comments.

Header and trailer

`EESchema-LIBRARY Version 2.3  Date:` *`timestamp`*	
	Identifies the file as a symbol library. The time-stamp is in the form of "29/09/2012 15:26:33".
`#encoding utf-8`	Indicates that the file uses UTF-8 encoding; if absent, the encoding is Latin-1.

The library file ends with "`#End Library`" on a single line. Note that this is a comment, so libraries do not have a true trailer.

Symbols

Each symbol starts with a "DEF" line, with a name and a series of parameters. The symbol definition ends with "ENDDEF" (on a line of its own). In between, are field definitions and a section with the drawings and pin definitions.

DEF *name ref* 0 *offs* Y Y 1 F N	If *name* starts with a "~", the symbol name is set to be invisible. The second parameter is the symbolic prefix, so "U" for an integrated circuit, "R" for a resistor, etc. It may be set to "~" if no symbolic prefix exists. The third parameter is always zero. The fourth parameter (*offs*) is the pin *name* offset, in mil from the end-point of a pin; when set to zero, the pin name is set outside the shape. The firth parameter is either "Y" or "N" to indicate whether pin numbers are shown, and the sixth parameter does the same for pin names. The seventh parameter gives the number of "parts" in the symbol. After the seventh parameter, the others are optional. The eighth parameter is an "L" if the parts are locked or an "F" otherwise. The ninth parameter is "P" for a power symbol or "N" otherwise.
F0 "*ref*" *X Y Size* H V C CNN	The symbolic prefix text, a mandatory field. The first parameter, *ref*, should be the same is in the "DEF" line; it may be prefixed with a "#" to indicate a virtual component (e.g. a power symbol). The next parameters give the position and size of the text field, in mil. The fifth field is "H" for horizontal orientation, or "V" for vertical. The sixth field is "I" for invisible text, or "V" otherwise. The seventh and eighth parameters give the horizontal and vertical alignment: C(entred), L(eft), R(ight), T(op) or B(ottom). The ninth parameter is "I" for italic text, or "N" otherwise. The

	tenth parameter is "B" for bold text or "N" otherwise. There are no spaces between the eighth, ninth and tenth parameters.
F1 "*name*" *X Y Size* H V C CNN	The symbol name text, a mandatory field. The *name* parameter must be the same as in the "DEF" line. See the "F0" definition for the other parameters.
F2	Footprint name.
F3	Relative path to a data-sheet.
F*n* "*text*" *X Y Size H V C CNN*	User fields, with *n* starting at 4. The user fields are written to the output file only when the text of the field is not empty. User fields may have an eleventh parameter at the end: a name for the field.
ALIAS	A list of alternative names for the same symbol. The names are separated with spaces.
$FPLIST … $ENDFPLIST	A section that contains list of footprints that are valid for the symbol. This section is described separately, below.
DRAW … ENDDRAW	A section with the drawing and the pin definitions for the symbol. This section is described separately, below.

Footprint filter section

$FPLIST	Starts a section with footprint patterns. CvPcb can use these patterns to present a list with only matching footprints.
pattern	Each line in the section is a pattern, meaning normal text with wild-cards. A "?" matches any single character and a "*" matches zero or more characters. Leading white-space must be trimmed from the pattern (the patterns are typically indented with a single space character.
$ENDFPLIST	Ends the section

Drawing section

A few parameters are common to all shapes. The *part* parameter indicates which of the parts the shape applies to, or zero if it applies to all shapes (this relates to symbols that have multiple parts). The *dmg* parameter stands for a "De Morgan alternate shape", and gives the unit number to which this shape applies, or zero if it applies to both units. The *pen* parameter is the thickness of the pen; when zero, the default pen width is used. The *fill* parameter is "f" for a filled shape in the background colour, "F" for a filled shape in the pen colour, or "N" for an unfilled shape.

A *X Y radius start end part dmg pen fill Xstart Ystart Xend Yend*	
	Arc. The *start* and *end* parameters are angles in 0.1 degrees. The *Xstart* and *Ystart* parameters give the coordinate of the start point; it can be calculated from the radius and the *start* angle. Similarly, the *Xend* and *Yend* parameters give the coordinate of the end point. The arc is drawn in counter-clockwise direction, but the angles are swapped if there (normalized) difference exceeds 180 degrees.
C *X Y radius part dmg pen fill*	Circle.
P *count part dmg pen X Y … fill*	Polygon with *count* vertices, and an *X,Y* position for each vertex. A filled polygon is implicitly closed, other polygons are open.
S *X1 Y1 X2 Y2 part dmg pen fill*	Rectangle, from *X1,Y1* to *X2,Y2*.
T *angle X Y size hidden part dmg text italic bold Halign Valign*	

	Text (which is not in a field). Parameter *angle* is in 0.1 degrees. Parameter *hidden* is 0 for visible text and 1 for hidden text. The *text* can be in double quotes, or it can be unquoted, but with the ~ character replacing spaces. Parameter *italic* is the word "Italic" for italic text, or "Normal" for upright text. Parameter *bold* is 1 for bold and 0 for normal. Parameters *Halign* and *Valign* are for the text alignment: C(entred), L(eft), R(ight), T(op) or B(ottom).
B *count part dmg pen X Y … fill*	Bezier curves with *count* vertices, and an *X,Y* position for each vertex.
X *name pin X Y length orientation sizenum sizename part dmg type shape*	
	Pin description. The pin name is not in double quotes (and therefore a pin name may not contain a space). When a pin has no name, parameter *name* is a "~", but when a "~" is followed by a name, the name has an overbar. The *pin* parameter is the pin number (it need not be numeric and may be a "~"). Parameter *orientation* is a single letter, U(p), D(own), L(eft) or R(ight). The *sizenum* and *sizename* parameters give the text sizes for the pin number and the pin name respectively. The *type* is a single letter: I(nput), O(utout), B(idirectional), T(ristate), P(assive), (open) C(ollector), (open) E(mitter), N(on-connected), U(nspecified), or W for power input or w of power output. If the *shape* is absent, the shape is a line, otherwise it is one of the letters I(nverted), C(lock), L for input-low, V for output-low (there are more shapes…). If the *shape* is prefixed with an "N", the pin is invisible.

Documentation file

The description for a schematic symbol is in a separate file, with the same name as the library, but the extension ".dcm". If a symbol lacks the description, list of keywords and documentation file, no entry for that component is present in the documentation file.

$CMP *name*	Starts a section for the component.
D *description*	A general description.
K *keywords*	A list of keywords, separated by spaces.
F *filename*	A relative path to a data-sheet file (relative to the "doc" directory).
$ENDCMP	Ends the section.

Footprint Library Format

The "legacy" file format is a file with the extension ".mod" and which contains a series of footprint definitions, plus an index. The original version of the format specified dimensions in *decimil* ($1/10^{th}$ of a mil). This was also the internal unit of KiCad. Later versions of KiCad used nanometres as the internal unit, and in the file format, dimensions were specified in millimetres. The footprint library format still identifies itself as "version 1", but there appears the setting "Units mm" in the file before the index. In both legacy file formats, angles are in $1/10^{th}$ of a degree; so a value of 900 specifies an angle of 90 degrees.

The new library format is based on the file system and data in "s-expressions". The library is a directory (not a file); to get an index of the library, you browse the directory, and filenames are the symbol names, The s-expressions are a format with sections enclosed in parentheses. Dimensions in the s-expressions format is in millimetres and angles are in degrees.

In both the legacy and s-expression formats, the positive vertical axis points down.

Legacy footprint library format

The legacy library format starts with a header. This is followed by an index, and then a list of “modules”.

Header & trailer

The first line in a library identifies the file as a legacy footprint library and indicates the date and time of the last modification. The line may be followed by lines containing additional settings.

`PCBNEW-LibModule-V1` *timestamp*	The time-stamp is in the form “Wed Jun 12 15:26:33 2013”.
`# encoding utf-8`	The text is in UTF-8 encoding. If absent, the text is in Latin-1.
`Units mm`	Dimensions are in millimetres and may contain a fractional part. If absent, dimensions are in decimil, and specified as integers (*whole numbers*).

The library ends with “`$EndLIBRARY`” on a separate line.

Index

The index starts with a line “`$INDEX`” and ends with “`$EndINDEX`”. Between the start and the end is a list of symbol names, one per line. KiCad sorts the index with a case-sensitive alphabetical sorting order, but sorting is probably not required.

Modules

The list of modules starts below the index. Each module starts with:

```
$MODULE symbolname
```

and ends with

```
$EndMODULE symbolname
```

The “symbolname” must be the name of the symbol, and it must appear in the index.

Po *X Y A Layer Tedit Tstamp Attributes*	
	X and *Y* are the module position; these should be 0 in a module library. The rotation angle *A* is in 0.1 degree, and also should be 0 in a library. The layer is typically 15 (“front”). The *Tedit* field is a hexadecimal value with the time-stamp of the last edit; the *Tstamp* field is another time-stamp, but apparently always zero for libraries. The *Attributes* field consists of two characters. The first is an “F” if the component is “fixed” (locked) for auto-placement and “~” if it is movable. The second character indicates whether the component was auto-placed (in which case it is a “P”); for libraries, it should always be a “~”.
`Li` *symbolname*	The symbol name should be the same as the one on the $MODULE line.
`Cd` *description*	Description of the symbol.
`Kw` *keywords*	A list of keywords separated with space characters.
`Sc` T*stamp*	A time-stamp code, which apparently is always zero on libraries. Not to be confused with the “Sc” command in the $SHAPE section.
`At` *Type*	The component type, STD or SMD. Each pad may also have a type, see the “At” keyword in the $PAD section.
`AR` *Name*	Alternate reference, which links the module to a schematic symbol; it is not set for libraries.
`Op` *penalty90 penalty180* `0`	Penalties for rotation for auto-placement. The first field is the penalty for

	rotation by 90 degrees, the second for rotation by 180 degrees. These two values must be between 0 (no penalty) and 10. The last field is always zero.
`T0` *X Y H W A Pen Mirror Visible Layer Italic* "*Text*"	
	Reference text, at position *X, Y*. The *H* and *W* fields specify the size of a character (note that the *height* field is given first). Field *A* is the rotation of the text; the current KiCad release restricts it to the values 0 or 900. The *Pen* field is the pen width. The *Mirror* field is an "M" or an "N" (for "not" mirrored). The field *Visible* is either a "V" or an "I" (for "invisible"). The *Layer* is typically 21 (silk screen of the top layer). The field *Italic* is either an "I" or an "N". The *Text* is always between double quotes; the text is centred on the *X,Y* position. In the case of a library, the text is usually the symbol name. The space between the *Italic* and the *Text* fields may be missing and the *Italic* field may be absent (in old libraries).
`T1` X *Y H W A Pen Mirror Visible Layer Italic* "*Text*"	
	Value text. See T0 for the fields. The text itself is irrelevant; it is usually "VAL**".
`T2` X *Y H W A Pen Mirror Visible Layer Italic* "*Text*"	
	User text. See T0 for the fields.
`DS` *X1 Y1 X2 Y2 Pen Layer*	Draw line.
`DC` *X Y Xp Yp Pen Layer*	Draw circle. *X, Y* is the centre of the circle. *Xp, Yp* gives a point on the circle; you can calculate the circle radius from it.
`DA` *X Y Xp Yp Angle Pen Layer*	Draw arc. *X, Y* is the centre of the arc circle. *Xp, Yp* is the start point. The *Angle* is in 0.1 degrees.
`DP 0 0 0 0` *Count Pen Layer*	Draw polygon. The *Count* is the number of vertices in the polygon; this number of "`Dl`" commands should follow.
`Dl` *X Y*	Vertex in a polygon. This field must follow a DP command.
`.SolderPaste` *dimension*	General aperture reduction (for all pads, unless overruled).
`.SolderMask` *dimension*	General aperture enlargement (for all pads, unless overruled).
`.SolderPasteRatio` *value*	General aperture reduction (for all pads, unless overruled), expressed as a ratio of the pad size. The value is between -0.5 (no solder paste) and 0 (full pad). A value greater than zero even enlarges the solder paste aperture.
`.LocalClearance` *dimension*	Module-specific clearance.
`.ZoneConnection` *value*	0=pads are not covered; 1=thermal relief; 2=pad fully inside the zone; 3=thermal relief only for TH pads.
.ThermalWidth *dimension*	Module-specific thermal relief width.
`.ThermalGap` *dimension*	Module-specific thermal relief gap.

Pad descriptions

There is a pad description section for each pad. The section starts with the keyword "$PAD" on a line of its own, and ends with the keyword "$EndPAD" on a line of its own. Between these two, the following definitions appear.

`Po` *X Y*	The pad position, relative to the origin of the module.
`Sh` *"Name" Shape Width Height Ydelta Xdelta Orientation*	
	The pad name is between double quotes. It should be the pin number (but

	KiCad does not enforce this). The *Shape* is a single letter: "C" for circle, "R" for rectangle, "O" for obround (oval), "T" for trapezoid. The *Orientation* is in 0.1 degrees. The *Xdelta* and *Ydelta* are for trapezoid pads and specify the difference between the short edge and the long edge. One of these two values is always zero, the other can be positive (for increasing the left/bottom edge and reducing the right/top edge) or negative (for increasing the right/top edge and reducing the left/bottom edge).
`Dr` *Size X Y*	Round hole of *Size*. The *X,Y* position is relative to the position of the pad (see Po).
`Dr` *Size X Y* O *W H*	Slotted hole (obround shape). The *Size* field can be ignored, as it is overruled by the *W* and *H* fields.
`At` *Type* N *Mask*	The *Type* is one of "STD" (plated-through hole), "SMD", "CONN" (test pin or card-edge connector) and "HOLE" (non-plated hole). The *Mask* field is a hexadecimal number that has a bit set for every layer that the pad affects. For through-hole pads/holes, this must include all copper layers, so its value is typically 00E0FFFF; for an SMD pad, it is typically 00888000.
Le *dimension*	Pad-to-die length. This pad-specific value is relevant for high-frequency designs (when the trace lengths must be taken into account, and the distance of the pad to the die of the chip is part of the transmission line).
`Ne` *Number "Name"*	The net number and name. In the case of libraries, the number is 0 and the name an empty string.
.SolderMask *dimension*	Pad-specific aperture enlargement. This value should be positive.
.LocalClearance *dimension*	Pad-specific clearance. This value should be positive.
.SolderPaste *dimension*	Pad-specific aperture reduction. This value should be negative (or zero).
.SolderPasteRatio *value*	The pad-specific aperture reduction on the solder paste stencil as a ratio of the pad with/height. This value should be negative (or zero). When set to -0.5, there will be no solder paste on the pad at all.
.ZoneConnection *value*	0=pads are not covered; 1=thermal relief; 2=pad solid inside the zone; 3=thermal relief only for TH pads; if absent, the module-specific field settings are used.
.ThermalWidth *dimension*	Pad-specific thermal relief width.
.ThermalGap *dimension*	Pad-specific thermal relief gap.

3-D shape

A 3-D shape is optional. It starts with "$SHAPE3D" and ends with "$EndSHAPE3D". Between these, the following commands should be present.

Na *path*	The relative path to the VRML file.
Sc *X Y Z*	Scale factors for X, Y and Z.
Of *X Y Z*	Offset.
Ro *X Y Z*	Rotation in degrees (not 0.1 degrees).

S-Expression libraries

A library is s-expression format is a directory. Each footprint definition is a file; the footprint name is the filename. As a result, footprint names may not contain characters as a slash or the backslash (as these cannot appear in a valid filename).

The s-expression consists of "(keyword value-list)" sections, where the "value list" is a list of items separated by white-space. Each item may be a single word or quoted string, or it may be a nested s-expression (i.e. another "keyword value-list" tuple between parentheses). The nesting may be arbitrarily deep. The s-expressions format is free-format (as opposed to the legacy format, which is line-based), but there should not appear white-space immediately after an opening parenthesis.

The keywords are always lower case. Texts in the file use UTF-8 encoding (but keywords use only the ASCII subset).

When items in a value list are a single word or number, the item need not be between double quotes. Empty strings must be appear like "". Items must be between double quotes if they contain white-space (including line breaks), the characters "(", ")", "{", "}", "%", "#" or "-" (with the exception that a "-" at the start of the string does not require the string to be quoted). Line breaks and special characters need to be escaped, so for a newline, you would write "\n". A double quote inside a quoted string may either be doubled or escaped.

The "#" character starts a comment that runs up to the end of the line.

All dimensions are in millimetres; angles are in degrees.

(module *name*	Start of a module definition
locked	If present, the component is locked for auto-placement.
placed	The component has been auto-placed (should never appear in libraries).
(layer *name*)	The layer on which the component is placed, typically "F.Cu".
(tedit *hex*)	Time-stamp of the last edit operation, as a hexadecimal number.
(tstamp *hex*)	Time-stamp from the schematic, as a hexadecimal number. This field is usually absent in library files.
(at *x y a*)	Position of the module. The "*a*" field (angle) is optional; it is typically absent when it would be zero. In libraries, the position should be 0,0.
(attr *type*)	The *type* field can be either "smd" or "virtual"; when this section is absent, the component is through-hole.
(descr *text*)	Description of the component.
(tags *keywords*)	A list of keywords separated by spaces.
(path *path*)	A pseudo-path that links the module to a symbol in the schematic. This section should not appear in libraries.
(autoplace_cost90 *val*)	An integer value between 0 and 10, that represents the "cost" of rotating the component 90 (or 270) degrees in auto-placement.
(autoplace_cost180 *val*)	An integer value between 0 and 10, that represents the "cost" of rotating the component 180 degrees in auto-placement.
(solder_mask_margin *dim*)	The component-specific solder mask margin (in mm). This should be zero or positive.
(solder_paste_margin *dim*)	The component-specific solder paste margin (in mm). This should be zero or negative.
(solder_paste_margin_ratio *val*)	The general aperture reduction for the solder paste. A rational value between -0.5 (no solder paste) and 0 (full pad). A value greater than zero even enlarges the solder paste aperture.
(clearance *dim*)	The component-specific clearance dimension (in mm).
(zone_connect *val*)	0=pads are not covered; 1=thermal relief; 2=pad solid inside the zone; 3=thermal relief only for TH pads; if absent, the global settings are used.
(thermal_width *w*)	The width of the traces for thermal relief pads. If absent, the global settings are used.

(thermal_gap *w*)	The width of the ring for the thermal relief pads. If absent, the global settings are used.
(fp_text *type text*	Footprint text. The *type* field can be “reference”, “value” or “user”. For a library the text for the reference and value are place-holders; they will be replaced by the actual reference and value texts. The “user” text is entirely optional.
(at *x y a*)	The position and angle of the text. The angle is usually omitted if it is zero.
(layer *name*)	The name of the layer for the text, usually “F.SilkS”.
hide	If present, the text is hidden.
(effects	Text attributes.
(font	The font size.
(size *w h*)	The character size.
(thickness *t*)	The weight of the text (line thickness)
)	
)	
)	
(fp_line	A line segment.
(start *x y*)	The start position, relative to the origin of the component.
(end *x y*)	The end position.
(layer *name*)	The name of the layer for the line, usually “F.SilkS”.
(width *pen*)	The pen width.
)	
(fp_circle	A circle.
(center *x y*)	The position of the centre, relative to the origin of the component.
(end *x y*)	The position of a point on the circle. The radius is calculated as the distance of this point from the centre.
(layer *name*)	The name of the layer for the line, usually “F.SilkS”.
(width *pen*)	The pen width.
)	
(fp_arc	An arc.
(start *x y*)	The position of the *centre* of the circle that is the basis of the arc.
(end *x y*)	The *starting* point of the arc. Both the radius of the arc and the start angle are calculated from this point.
(angle *a*)	The angular span that the arc covers, from the start angle, in *clock-wise* direction. It is added to the start angle to find the end angle.
(layer *name*)	The name of the layer for the line, usually “F.SilkS”.
(width *pen*)	The pen width.
)	
(fp_poly	Polygon.
(pts	A list of coordinates.
(xy *x y*)	The coordinates of a point in the polygon. There may be an arbitrarily

	number of points in the “pts” list.
)	
(layer *name*)	The name of the layer for the line, usually “F.SilkS”.
(width *pen*)	The pen width.
)	
(fp_curve	A Bezier curve.
(pts	A list with four coordinates.
(xy *x y*)	
(xy *x y*)	
(xy *x y*)	
(xy *x y*)	
)	
(layer *name*)	The name of the layer for the line, usually “F.SilkS”.
(width *pen*)	The pen width.
)	
(pad *name type shape*	A pad. The *name* is the pin number (or name). The *type* is one of “thru_hole” (plated), “smd”, “connect” (for a test pin or card-edge connector) or “np_thru_hole” (non-plated). The *shape* is one of “circle”, “rect”, “oval” (obround) or “trapezoid”.
(at *x y a*)	Position of the pad, relative to the origin of the component. The *a* field is the rotation of the pad; it is usually absent when it would be zero.
(size *w h*)	The size of the pad.
(rect_delta *dy dx*)	For trapezoid pads. The values specify difference between the short edge and the long edge. One of these two values is always zero, the other can be positive (for increasing the left/bottom edge and reducing the right/top edge) or negative (for increasing the right/top edge and reducing the left/bottom edge).
(drill *size*	Round hole of the given size (this section is only present for through-hole pads).
(offset *x y*)	The offset of the hole relative to the centre of the pad. If this section is absent, the hole is centred in the pad.
)	
(drill oval *w h*	Slotted hole of the given width and height (this section is only present for through-hole pads). Note that if the pad is rotated, the hole shape rotates too.
(offset *x y*)	The offset of the hole relative to the centre of the pad. If this section is absent, the hole is centred in the pad.
)	
(layers *name* …)	A list of layers affected by this pad. For through-hole pads, it should include all copper layers; the *name* list typically is “*.Cu *.Mask F.SilkS”. For SMD pads, the *name* list typically is “F.Cu F.Paste F.Mask”.
(net *number name*)	The net to which this pad is connected. This field is typically absent in library files.
(die_length *dim*)	Pad-to-die length. This pad-specific value is relevant for high-frequency

	designs (when the trace lengths must be taken into account, and the distance of the pad to the die of the chip is part of the transmission line).
(solder_mask_margin *dim*)	Pad-specific solder mask margin (in mm). This value should be positive.
(clearance *dim*)	Pad-specific clearance. This value should be positive.
(solder_paste_margin *dim*)	Pad-specific solder paste margin (in mm). This value should be negative (or zero), for aperture reduction on large pads.
(solder_paste_margin_ratio *val*)	The pad-specific aperture reduction on the solder paste stencil as a ratio of the pad with/height. This value should be negative (or zero). When set to -0.5, there will be no solder paste on the pad at all.
(zone_connect *val*)	0=pads are not covered; 1=thermal relief; 2=pad solid inside the zone; 3=thermal relief only for TH pads; if absent, the module-specific field settings are used.
(thermal_width *dim*)	Pad-specific thermal relief width.
(thermal_gap *dim*)	Pad-specific thermal relief gap.
)	
(model *path*	The optional 3D model is in a separate file. The *path* parameter gives the relative path to that file.
(at (xyz *x y z*))	
(scale (xyz *x y z*))	
(rotate (xyz *x y z*))	
)	
)	

Index