



PADS-ASCII Format Specification PowerPCB 5.0

Version 5.0



Copyright Page

Copyright © 1995 - 2002 Innoveda, Inc.

293 Boston Post Road West
Marlboro, Massachusetts 01752-4615
All Rights Reserved.

This information is copyrighted; all rights are reserved by Innoveda, Inc. This information may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without the prior written consent of Innoveda, Inc.

The software programs described, or referenced, herein are copyrighted and all rights are reserved by Innoveda, Inc. The programs may not be copied or duplicated except as expressly permitted in the Master Software License Agreement. This documentation contains proprietary information that is not to be disclosed to persons outside the user's company without the prior written consent of Innoveda, Inc.

The copyright notice appearing above is included to provide statutory protection in the event of unauthorized or unintentional public disclosure.

All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

Introduction.....	6
Version 5.0 Format Changes	7
Header line	7
General system parameters	7
Text entry format	8
Clearance rules format.....	8
Routing rules format	8
Attributes.....	8
Format Conventions.....	11
File Structure	12
Control Statements	13
Header Line.....	14
End-of-File Marker.....	14
General System Parameters	15
General parameter control statement	15
Format for general parameter entries	15
General parameters.....	15
Reuse Definitions	23
Reuse control statement.....	23
Reuse entry format.....	23
Reuse type header format	23
Reuse part format.....	23
Reuse net format	24
Reuse instance header format.....	25
Text Definition Format	27
Text control statement	27
Text entry format	27
Lines Definition Format.....	28
Lines control statement.....	28
Lines entry format.....	28
Lines header definition	28
Piece definition	29
Text data entry format	31
Getting 2D items from the library	31
Part Decal Definition Format	32

Part decal control statement.....	32
Part decal format structure.....	32
Header line format	32
Piece definition format	33
Text data entry format.....	34
Label data entry format	34
Pad definition format	35
Pad stack definition format.....	35
Via Definition Format.....	38
Via entry format	38
Jumper Definition Format	40
Jumper header format.....	40
Part Type Definition Format	41
Part type control statement	41
Part type entry format	41
Part type header format.....	41
Gate format.....	42
Signal format	43
Alphanumeric pin format.....	43
Part List Definition Format.....	44
Part list control statement	44
Automatic standard signal netlist generation.....	44
Part list entry format	44
Shortcut part list entry format	46
Connection Definition Format	47
Connection list definition	47
Connection list control statement.....	47
Connection list entry format	47
Connection list header format.....	47
Pin pair line format.....	48
Netlist definition format	48
Netlist control statement	48
Netlist format	49
Shortcut netlist format.....	49
Route Definition Format.....	50
Route section control statement.....	50
Route entry format	50
Route header format.....	50
Pin pair line format.....	50
Route coordinates line.....	51

Copper Pour Definition Format	55
Pour control statement.....	55
Pour entry format.....	55
Pour header definition.....	55
Piece definition	56
Arc format.....	57
Miscellaneous Section	58
Board drilling rules	58
Hierarchical rules section	58
Net classes	59
Pin pair groups	60
Design rules.....	61
Rule set format	62
Differential pairs format	66
Attributes.....	66
Layer description format.....	71
Visibility description format	74
Router strategy attributes.....	74
DFT attributes.....	75
Cluster Definition Format	76
CAM Section.....	77
CAM document list.....	77
CAM drill speed feed table	81
CAM augment on the fly	82
CAM aperture table.....	82
CAM drill symbol table	83
Assembly Options Section.....	84
Test Point Definition Section.....	85
Index.....	86

Introduction

This specification defines the structure and syntax of the PADS-ASCII data format for PowerPCB 5.0. See the "Version 5.0 Format Changes" topic for information on format changes from version 4.0 of PowerPCB.

The PADS ASCII file format provides a way for other CAD or CAE systems to interface with Innoveda printed circuit design products without access to the internal databases or source code. PADS-ASCII is also the format used to input a connection and parts list from a schematic capture system into PADS design environments.

You can convert an entire PowerPCB database to an ASCII file. This includes system parameters, status parameters, part type descriptions, part decal descriptions, parts lists, connectivity data (both logical and physical), and free text and graphics. The only information in a PowerPCB design that will not convert to the ASCII file is the design rule checking error markers and OLE objects.

The ASCII format file uses specific calls to retrieve library information during file importing. Part decal and part type information do not need to be defined within the file. Control statements that access the library are:

- *GET* Explicitly calls named items from the Lines library database. If not found, an error message appears.
- *PARTTYPE* Implicitly accesses part decals. The software first looks for the description of the part decal in the ASCII file, then looks in the Part Decal library. If the part decal is still not found, an error message appears.
- *PART* Explicitly accesses part type and part decal items from the Part Type library database and Part Decal library database, respectively. The software looks first for the description of the part type, then for the description of the decal in the ASCII file. If neither is found, an error message appears.

Version 5.0 Format Changes

This section identifies changes in the structure and syntax of the ASCII data format for PowerPCB 5.0. Refer to the specific sections for a complete description of changed or added information.

Header line

Version number V5.0 is used in the header line.

Example

```
!PADS-POWERPCB-V5.0-BASIC! DESIGN DATABASE ASCII FILE 1.0
```

General system parameters

New fabrication parameters are added:

FABCHKFLAGS_N	Fabrication check flags
ATMAXSIZE_N	Acid trap maximum size
ATMAXANGLE_N	Acid trap maximum angle
SLMINCOPPER_N	Sliver minimum copper
SLMINMASK_N	Sliver minimum mask
STMINCLEAR_N	Starved thermal minimum clearance
STMINSPOKES_N	Starved thermal minimum spokes
TPMINWIDTH_N	Minimum trace width
TPMINSIZE_N	Minimum pad size
SSMINGAP_N	Silk screen over pads minimum gap
SBMINGAP_N	Solder bridges minimum gap
SBLAYER_N	Solder bridges layer
ARPTOM_N	Pad-to-mask annular ring
ARPTOMLAYER_N	Pad-to-mask annular ring layer
ARDTOM_N	Drill-to-mask annular ring
ARDTOMLAYER_N	Drill-to-mask annular ring layer
ARDTOP_N	Drill-to-pad annular ring
ARDTOPLAYER_N	Drill-to-pad annular ring layer
VERIFYFLAGS_N	New flags are added to VERIFYFLAGS
Bits	Values
12	Clearance Checking Setup: Via at SMD
13	Clearance Checking Setup: Differential Pairs
14	Clearance Checking Setup: Trace Length

Text entry format

The first line of the text entry format is updated as follows:

```
X_Y_ORI_LEVEL_HEIGHT_WIDTH_M_HJUST_VJUST_ [NDIM]_[.REUSE._INSTANCE]
```

NDIM This new parameter is the sequential number of autodimensioning items on the drawing to which the text belongs. Specify if text is a member of some drawing item.

Clearance rules format

In the Miscellaneous section, under the RULES_SECTION header, the following keywords are added:

```
DRILL_TO_COPPER_N
```

N Value of the Drill to Copper clearance

For more information see "Hierarchical rules section" in "Miscellaneous Section."

Routing rules format

In the Miscellaneous section, under the RULES_SECTION header, the COPPER_SHARE parameter is replaced by the following keywords:

PIN_SHARE_N N is a through pin share flag.

SMD_SHARE_N N is an SMD pin share flag.

VIA_SHARE_N N is a via share flag.

TRACE_SHARE_N N is a trace share flag.

For more information see "Hierarchical Rules Section" in "Miscellaneous Section."

Attributes

The following system attributes are eliminated:

- Strategy.SplitPairs.Intensity
- Strategy.SplitPairs.Pass
- Strategy.SplitPairs.Pause
- Strategy.SplitPairs.PlanePriority
- Strategy.SplitPairs.Priority
- Strategy.SplitPairs.Protect

The following system attributes are added:

- Accordion.Amplitude.Min
- Accordion.Gap.Min
- Placement.Grid.Use
- Placement.Grid.X
- Placement.Grid.Y
- Rules.Clearance.Board.Pad
- Rules.Clearance.Board.SMD
- Rules.Clearance.Board.Trace
- Rules.Clearance.Board.Via
- Rules.Clearance.Copper.Pad
- Rules.Clearance.Copper.SMD
- Rules.Clearance.Copper.Trace

Rules.Clearance.Copper.Via
Rules.Clearance.Drill.Pad
Rules.Clearance.Drill.SMD
Rules.Clearance.Drill.Trace
Rules.Clearance.Drill.Via
Rules.Clearance.Pad.Pad
Rules.Clearance.Pad.Trace
Rules.Clearance.Pad.Via
Rules.Clearance.SMD.Pad
Rules.Clearance.SMD.SMD
Rules.Clearance.SMD.Trace
Rules.Clearance.SMD.Via
Rules.Clearance.Text.Pad
Rules.Clearance.Text.SMD
Rules.Clearance.Text.Trace
Rules.Clearance.Text.Via
Rules.Clearance.Trace.Trace
Rules.Clearance.Via.Trace
Rules.Clearance.Via.Via
Rules.Fanout.Alignment
Rules.Fanout.Alignment.Multi-Row
Rules.Fanout.Direction
Rules.Fanout.Length.Maximum
Rules.Fanout.Length.Unlimited
Rules.Fanout.Nets.Plane
Rules.Fanout.Nets.Signal
Rules.Fanout.Nets.UnusedPins
Rules.Fanout.Sharing.Pin
Rules.Fanout.Sharing.SMD
Rules.Fanout.Sharing.Trace
Rules.Fanout.Sharing.Via
Rules.Fanout.ViaSpacing
Rules.Pad.Entry.Corner
Rules.PadEntry.AnyAngle
Rules.PadEntry.MinWidth
Rules.PadEntry.Side
Rules.PadEntry.Soft
Rules.Routing.RestrictedVias
Rules.SameNet.Pad.Crn
Rules.SameNet.SMD.Crn
Rules.SameNet.SMD.Via
Rules.SameNet.Via.Via
Rules.ViaAtSMD
Rules.ViaAtSMD.Center
Rules.ViaAtSMD.Ends
Rules.ViaAtSMD.FitInside
Rules.Width.Maximum
Rules.Width.Minimum
Rules.Width.Recommended
Strategy.Fanout.DiffPairPriority

Strategy.Fanout.MLGPriority
Strategy.Miters.DiffPairPriority
Strategy.Miters.MLGPriority
Strategy.Optimize.DiffPairPriority
Strategy.Optimize.MLGPriority
Strategy.Patterns.DiffPairPriority
Strategy.Patterns.MLGPriority
Strategy.Route.DiffPairPriority
Strategy.Route.MLGPriority
Strategy.TestPoint.DiffPairPriority
Strategy.TestPoint.MLGPriority
Strategy.Tune.DiffPairPriority
Strategy.Tune.Intensity
Strategy.Tune.MLGPriority
Strategy.Tune.Pass
Strategy.Tune.Pause
Strategy.Tune.PlanePriority
Strategy.Tune.Priority
Strategy.Tune.Protect

Format Conventions

The format of ASCII documents is important for correct interpretation by Innoveda software. Use the following conventions when creating or modifying the ASCII file:

- Use all uppercase characters for control statements and arguments.
- Allowable characters for names include all uppercase and lowercase letters, digits 0 through 9, and the special characters: '~@#\$%^&()_+ =:;''[]?/<>!
- Illegal characters for net names are spaces, commas (,), braces ({}), asterisks (*), and question marks (?).
- Illegal characters for part names are spaces, periods (.), commas (,), braces ({}), and question marks (?).
- All line entries require a final carriage return/line feed (Enter key). These are not explicitly specified in format definitions.

Tip: Optional data is enclosed in brackets []. Including some options may make other optional data required.

File Structure

The ASCII file consists of a number of sections, each of which may be independently entered. Each section has a unique control statement and format. Certain sections must be present before entering other data. These dependencies are provided in this document.

Sections of the ASCII file include:

- Header
- Attributes
- CAM
- Cluster placement groups
- Connection list
- Copper pour items
- Jumpers description
- Part decal items
- Part type items
- Parts list
- Reuse groups
- Route list
- Rules
- System parameters
- Test points
- Text and lines data
- Via type items
- End of file marker

Control Statements

Control statements denote a specific section of the PADS ASCII file. All control statements begin and end with an asterisk (*):

CLUSTER	Cluster placement data
CONN	Connection information
END	End-of-file marker
GET	Call items from 2D lines library into design
JUMPER	Jumper information
LINES	Board outline, copper, and 2D item
MISC	Miscellaneous data, including CAM and rules
NET	Net information
PART	Part list information
PARTDECAL	Part decal information
PARTTYPE	Part type information
PCB	System parameters
POUR	Copper pour information
REMARK	Comment line
REUSE	Reuse information
ROUTE	Route information
SIGNAL	Signal information
STANDARD	Generate standard signal list directive
TESTPOINT	Define test points
TEXT	Free text data
VIA	Via information

Header Line

The header line identifies the data that follows it as a PADS ASCII file. It must be included at the beginning of each file.

!PADS-PRODUCT-VERSION-UNITS[-MODE]!

PRODUCT	POWERPCB
VERSION	V5.0
UNIT	MILS Defined database units are in mils. Data is expressed to the nearest hundredth mil. Valid values range from -56000 to +56000.
	INCHES Defined database units are in inches. Data is expressed to the nearest one hundred thousandth of an inch. Valid values range from -56.000 to +56.000.
	METRIC Defined database units are millimeters. Data is expressed to the nearest ten thousandth of one millimeter. Valid values range from -1422.4 to +1422.4.
	BASIC Defined database units are in database units, 1 database unit = $2/3 \times 10^{-9}$ meters. Valid values range from -2133600000 to 2133600000.
MODE	250L Defined design in 250-layer mode. In other case design, it is in 30 layers mode.

Note: Within this specification all measurements are specified as MILS.
Allowed ranges in degrees are always 0.0 to 359.9 or 359.999 degrees, depending on the object type.

End-of-File Marker

The end-of-file control statement is required at the end of the ASCII file.

The control statement for the end-of-file is:

END

General System Parameters

This section defines the general parameters of the design, including the snap grid, maximum number of layers, and current display color selections. The general parameters section is not required to define a circuit board because this information may be supplied interactively.

Each of the entries in the general parameters subsection consists of a parameter name, followed by a space and values. The parameter name must be properly spelled, including spaces, and may be in either uppercase or lowercase letters. The allowable range of values for each parameter is defined in parentheses. Illegal or invalid entries are ignored.

The order of parameters is not fixed, and parameters may be left out of an input file.

General parameter control statement

PCB

Format for general parameter entries

PARAMETER_N

PARAMETER	Legal parameter name as listed in "General Parameters."
N	Numeric value in the allowable range for the parameter.

General parameters

UNITS_N	N is Unit of measure. Valid values are: 0 – Mils 1 – Metric 2 – Inches
USERGRID_N	N is User-defined snap grid in units defined upon original input. Valid values: Inches range from .00001 to 2.000 Metric range from 0.000254 to 5.080 Mils range from 0.01 to 2000
MAXIMUMLAYER_N	N is the maximum number of routing layers. Valid values range from 1 to 64.
WORKLEVEL_N	N is the current layer on which data is entered. Valid values range from 0 (all layers) to 250 (maximum layer).
DISPLAYLEVEL_N	N – 1 means display current layer on top. N – 0 means display "Top" layer on top.
LAYERPAIR_N1_N2	Two values are required, defining the layer pair for swapping during routing. N1, N2 are in range of 1 to 64. N2 should be greater than N1.

VIAMODE_N	N defines the current type of via to use when routing between layers. Values can be: T – Through via B – Buried via M – Micro via
LINEWIDTH_N	N is the width with which items will be created. Valid values range from 0 to 250.
TEXTSIZE_HEIGHT_WIDTH	HEIGHT is the text height. Valid height values range from 1 to 1000. WIDTH is the line width in which text will be created. Valid width values range from 1 to 50. Values are dependent upon units of measure.
JOBTIME_N	N is total time in minutes recorded for design in system memory.
DOTGRID_N	N is the space between graphic dots. Valid values range from 1 to 1000.
SCALE_N	N is the scale of window expansion. Valid values range from 1.0 to 7400.000++.
ORIGIN_X_Y	X and Y are coordinates of the current design origin relative to the system origin. Valid values range from -28000 to 28000.
WINDOWCENTER_X_Y	X and Y are coordinates defining the center of the window.
BACKUPTIME_N	N is the number of minutes between automatic backup of database to file. Valid values range from 1 to 30.
REAL WIDTH_N	N is the minimum track size to begin to display real track size. Tracks narrower than N appear with a center line only. Valid values range from 1 to 250.
ALLSIGONOFF_N	N is the value of the default signal view nets.

Net display type	Bit 1	Bit 9	Bit 18
No traces or unrouted	0	0	0
Traces and all unrouted	0	1	1
Traces and unrouted, except connected by plane	1	0	0
Traces and unrouted, except unrouted pin pairs	0	1	0
Traces and no unrouted	0	0	1

All other bits are ignored.

REFNAMESIZE_ HEIGHT_WIDTH	Height is the default height of the reference name and part name. Valid values range from 0.01 to 1000. Width is the default line width to draw reference name and part names. Valid values range from 0.01 to 50.
HIGHLIGHT_N	Highlight current net parameter
JOBNAME_N	User-defined job name. Up to 20 characters (prefix), 3 characters (suffix).
CONCOL_N	N is the color code for connections in the design. Valid values range from 0 to 31.
FBGCOL_N1_N2	N1 is the color code for foreground color in the design and ranges from 0 to 31. N2 is the color code for background color in the design and ranges from 0 to 31.
HATCHGRID_N	N is the value of the default copper pour hatching grid. Valid values range from 0.01 to 250.
TEARDROP_N	N is the value of teardrop: 0 – Off 1 – On
THERLINEWID_N	N is the default copper pour thermal line width for through hole pad stacks.
PADFILLWD_N	N is the CAM finger pad fill line width. Valid values range from 1 to 250. Default is 10.
THERSMDWID_N	N is the copper pour thermal line width for SMD pad stacks.
MINHATAREA_N	N is the minimum area of copper that will be hatched. Valid values range from 0 to 9999999. Default is 0.
HATCHMODE	N is the hatch generation mode for copper flood: 0 – Normal. This is the default. 1 – No hatch 2 – See through
HATCHDISP_N	N is the hatch display flag for copper flood 0 – Pour Outlines 1 – Hatch Outlines Default is 0.
MITRETYPE_N	N is the miter type: 2 – Line 1 – Arc. This is the default.
HATCHRAD_N	N is the ratio for creating smooth hatch outlines. Valid values range from 0 to 9999999. Default is 0.5.
HATCHANG_N	N is the hatch angle:

	0 – Horizontal and vertical hatch. This is the default.
	45 – Diagonal hatch
THERFLAGS_N	N is the reserved fixed flags used in thermal generation. Thermal flags definitions: Bits 0 – 7: Thermal directions (0 – diagonal, 1 – perpendicular) 0x00000001 – For through hole RND pad 0x00000002 – For through hole SQR pad 0x00000004 – For through hole RF pad 0x00000008 – For through hole OF pad 0x00000010 – For SMD RND pad 0x00000020 – For SMD SQR pad 0x00000040 – For SMD RF pad 0x00000080 – For SMD OF pad Bits 8 – 9: Thermal generation (0 – generate, 1 – don't generate) 0x00000100 – For through hole pads 0x00000200 – For SMD pads
DRLOVERSIZE_N	N is the drill oversize for plated holes.
PLANERAD REL_RAD	REL_RAD is the plane smoothing radius represented by floating number and setup mixed plane smoothing radius, relative to outline width.
PLANETPFLAGS SAVE_MODE DISPL_MODE REMOVE_ISOLATED_COPPER REMOVE_STUB_VIOL SHOW_TP LOCK_TP HIDE_TACK AUTO_THERMAL_UPDATE AUTO_LINK_UPDATE FLOOD_CONFIRM RELATIVE_SIZE SHOW_PROTECTION HATCH_REVERSE REMOVE_UNUSED_PADS	
SAVE_MODE	Data which will be saved in .pcb file: ALL or OUTLINE.
DISPLAY_MODE	Description of plane data displayed: ALL, THERMALS, or OUTLINE.
REMOVE_ISOLATED_COPPER	Automatic removal of isolated copper islands after flood or plane connect operations Y – Yes N – No
REMOVE_STUB_VIOL	Automatic removal of thermal stub violations after flood or plane connect operations Y – Yes N – No

SHOW_TP	Test points display: Y – Yes N – No.
LOCK_TP	Controls operations on test points: Y – Yes N – No
HIDE_TACK	Tack visibility: Y – Yes N – No
AUTO_THERMAL_UPDATE	Automatic update of pad thermal status: Y – Yes N – No
AUTO_LINK_UPDATE	Automatic update of link visibility: Y – Yes N – No
FLOOD_CONFIRM	Flood confirmation: Y – Yes N – No
RELATIVE_SIZE	Controls size: Y – Yes N – No
SHOW_PROTECTION	Protection display: Y – Yes N – No
HATCH_REVERSE	Controls keepout display: Y – Yes N – No
REMOVE_UNUSED_PADS	Controls appearance of unused pads on Split/Mixed Plane layers: Y – Yes N – No

Example

PLANETPFLAGS ALL ALL N N Y Y Y Y Y Y Y Y Y Y

COMPHEIGHT N Board top component height restriction

KPTHATCHGRID N Copper pour hatching grid

BOTCOMPHEIGHT 0 Board bottom component height restriction

PLNSEPGAP GAP GAP is the distance that separates mixed plane areas from each other and from the board outline.

IDFSHAPELAY LAY LAY is the layer number that extracts component shapes for IDF export.

FANOUTGRID_N	N is the value of the fanout grid. Valid values range from 0.01 to 1999.99.																																				
FANOUTLENGTH_N	N is the maximum fanout length. Valid values range from 0.01 to 1999.99																																				
ROUTERFLAGS_N	N defines fanout parameters and pad entry quality. Values are stored in bit fields as follows: <table> <thead> <tr> <th>Bits</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>0-2</td> <td>Fanout Alignment: 0 – Regular 1 – Staggered</td> </tr> <tr> <td>3-5</td> <td>Fanout Direction: 0 – Inside 1 – Outside 2 – Both Sides</td> </tr> <tr> <td>6 – 8</td> <td>Fanout Spacing: 0 – Use Grid 1 – 1 Trace 2 – 2 Trace</td> </tr> <tr> <td>9</td> <td>Fanout Alignment: Multi-Row</td> </tr> <tr> <td>10</td> <td>Fanout Pins: Power</td> </tr> <tr> <td>11</td> <td>Fanout Pins: Signal</td> </tr> <tr> <td>12</td> <td>Fanout Pins: Unused</td> </tr> <tr> <td>13</td> <td>Fanout Sharing: Drilled pins</td> </tr> <tr> <td>14</td> <td>Fanout Sharing: SMD pins</td> </tr> <tr> <td>15</td> <td>Fanout Sharing: Vias</td> </tr> <tr> <td>16</td> <td>Routing Pad Entry Quality: Allow Side Exit</td> </tr> <tr> <td>17</td> <td>Routing Pad Entry Quality: Allow Corner Exit</td> </tr> <tr> <td>18</td> <td>Routing Pad Entry Quality: Allow Odd Exit</td> </tr> <tr> <td>19</td> <td>Reserved</td> </tr> <tr> <td>20</td> <td>Fanout Length: Unlimited</td> </tr> <tr> <td>21-23</td> <td>Reserved</td> </tr> <tr> <td>24-26</td> <td>Design Line/Trace Angle: 0 – Diagonal 1 – Orthogonal 2 – Any Angle</td> </tr> </tbody> </table>	Bits	Values	0-2	Fanout Alignment: 0 – Regular 1 – Staggered	3-5	Fanout Direction: 0 – Inside 1 – Outside 2 – Both Sides	6 – 8	Fanout Spacing: 0 – Use Grid 1 – 1 Trace 2 – 2 Trace	9	Fanout Alignment: Multi-Row	10	Fanout Pins: Power	11	Fanout Pins: Signal	12	Fanout Pins: Unused	13	Fanout Sharing: Drilled pins	14	Fanout Sharing: SMD pins	15	Fanout Sharing: Vias	16	Routing Pad Entry Quality: Allow Side Exit	17	Routing Pad Entry Quality: Allow Corner Exit	18	Routing Pad Entry Quality: Allow Odd Exit	19	Reserved	20	Fanout Length: Unlimited	21-23	Reserved	24-26	Design Line/Trace Angle: 0 – Diagonal 1 – Orthogonal 2 – Any Angle
Bits	Values																																				
0-2	Fanout Alignment: 0 – Regular 1 – Staggered																																				
3-5	Fanout Direction: 0 – Inside 1 – Outside 2 – Both Sides																																				
6 – 8	Fanout Spacing: 0 – Use Grid 1 – 1 Trace 2 – 2 Trace																																				
9	Fanout Alignment: Multi-Row																																				
10	Fanout Pins: Power																																				
11	Fanout Pins: Signal																																				
12	Fanout Pins: Unused																																				
13	Fanout Sharing: Drilled pins																																				
14	Fanout Sharing: SMD pins																																				
15	Fanout Sharing: Vias																																				
16	Routing Pad Entry Quality: Allow Side Exit																																				
17	Routing Pad Entry Quality: Allow Corner Exit																																				
18	Routing Pad Entry Quality: Allow Odd Exit																																				
19	Reserved																																				
20	Fanout Length: Unlimited																																				
21-23	Reserved																																				
24-26	Design Line/Trace Angle: 0 – Diagonal 1 – Orthogonal 2 – Any Angle																																				

VERIFYFLAGS_N N defines verify design parameters. Values are stored in bit fields as follows:

Bits	Values
0	Clearance Checking Setup: Net to All
1	Clearance Checking Setup: Same Net
2	Clearance Checking Setup: Drill to Drill
3	Clearance Checking Setup: Trace Width
4	Clearance Checking Setup: Body to Body
5	Clearance Checking Setup: Nudge Outline
6	Mixed Plane Setup: Full Check
7	Reserved
8	Mixed Plane Setup: Same Layer Connectivity
9	Clearance Checking Setup: Board Outline
10	Clearance Checking Setup: Keepout
11	Clearance Checking Setup: Off Board Text
12	Clearance Checking Setup: Via at SMD
13	Clearance Checking Setup: Differential Pairs
14	Clearance Checking Setup: Trace Length

FABCHKFLAGS_N Fabrication checks flags. Values are stored in bit fields as follows:

Bits	Values
0	Run Fabrication Check
1	Check Acid Traps
2	Check Slivers
3	Check Starved Thermals
4	Check Trace Width and Pad Size
5	Check Silkscreen over Pads
6	Check Solder Bridges
7	Reserved
8	Check Annular Ring
9	Check Pad to Mask
10	Check Drill to Mask
11	Check Drill to Pad

ATMAXSIZE_N	Acid traps maximum size
ATMAXANGLE_N	Acid traps maximum angle
SLMINCOPPER_N	Slivers minimum copper
SLMINMASK_N	Slivers minimum mask
STMINCLEAR_N	Starved thermal minimum clearance
STMINSPOKES_N	Starved thermal minimum spokes
TPMINWIDTH_N	Minimum trace width
TPMINSIZE_N	Minimum pad size
SSMINGAP_N	Silk screen over pads minimum gap
SBMINGAP_N	Solder bridges minimum gap
SBLAYER_N	Solder bridges layer

ARPTOM_N	Pad-to-mask annular ring
ARPTOMLAYER_N	Pad-to-mask annular ring layer
ARDTOM_N	Drill-to-mask annular ring
ARDTOMLAYER_N	Drill-to-mask annular ring layer
ARDTOP_N	Drill-to-pad annular ring
ARDTOPLAYER_N	Drill-to-pad annular ring layer
TEARDROPDATA_WIDTH_LENGTH_FLAGS	
WIDTH	Relative width of teardrop, in percent. Valid values range from 1 to 100.
LENGTH	Relative length of teardrop, in percent. Valid values range from 1 to 1000.
FLAGS	Shape modifier (optional): L – lined C – curved Adjustable modifier (optional): A – add for adjustable teardrop

Reuse Definitions

Reuse sections include reuse type and reuse instance definitions.

Reuse control statement

The control statement for the reuse is:

REUSE

Reuse entry format

Each reuse entry consists of the following:

- Reuse type header line
- Reuse part information (optional)
- Reuse net information (optional)
- Reuse instances list

Reuse type header format

The Reuse Type header has two lines:

TYPE_T

TIMESTAMP_SECONDS

TYPE Keyword

T Reuse type name

TIMESTAMP Keyword

SECONDS Reuse creation timestamp, represented by the number of seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time, according to the system clock.

Reuse part format

Header line:

PART_NAMING_PARTNM

PARTNM Reuse parts naming preference. Possible variants are:
PREFIX
SUFFIX
NEXT
START
INCREMENT

Details for naming preferences are provided in the "Reuse Instance header format" section..

This section describes the original names of reuse parts. Typically, actual reference designators are generated applying naming style to the original reuse part names.

This section contains many single line entries:

PART_NAME

PART	Keyword
NAME	Original part name in the reuse definition

Reuse net format

Header line:

NET_NAMING_NETNM

NETNM	Reuse net naming preference. Possible variants are: PREFIX SUFFIX
-------	---

Details for naming preferences are provided below.

This section describes the original names of reuse nets. Typically, actual netnames are generated applying naming style to the original reuse netnames.

This section contains many single line entries:

NET_MERGE_NAME

NET	Keyword
MERGE	Merge flag for the reuse net. 1 to merge nets 0 to rename nets
NAME	Original netname from reuse definition.

Reuse instance header format

REUSE_INSTANCE_PARTNM_NETNM_X_Y_ORI_GLUED

REUSE	Keyword
INSTANCE	Reuse instance name
PARTNM	Reuse parts naming preference. Possible variants are: PREFIX PREF PREFIX is a keyword. PREF is a prefix string. SUFFIX SUF SUFFIX is a keyword. SUF is a suffix string. NEXT NEXT is a keyword. Next available number will be assigned the same Ref. Des. Prefix. START START is a keyword. NUM NUM is the starting numeric suffix for the part name. INCREMENT INCREMENT is a keyword. NUM NUM is the numeric increment for the part name suffix.
NETNM	Reuse net naming preference. Possible variants are: PREFIX PREF PREFIX is a keyword PREF is a prefix string SUFFIX SUF SUFFIX is a keyword SUF is a suffix string Note: An empty suffix or prefix is represented by quotation marks (" ").
X and Y	Coordinates of the reuse origin
ORI	Reuse orientation. Valid values for ORI are multiples of 90 degrees.
GLUED	Reuse glued status: Y – Glued N – Not glued

Example

REUSE
TYPE Amplifier Channel
SUFFIX PREFIX
PART U1
PART U2
PART C1
PART R1
NET 1 POWER
NET 1 GROUND
NET 0 INPUT
NET 0 OUTPUT
REUSE LeftChannel SUFFIX R PREFIX R~ Y
REUSE RightChannel SUFFIX L PREFIX L~ N

Text Definition Format

Text items are any free text string on the circuit board such as a non-reference designation or part type text.

Text control statement

The text control statement is:

TEXT

Text entry format

Each text entry consists of two lines as follows:

X_Y_ORI_LEVEL_HEIGHT_WIDTH_M_HJUST_VJUST_ [NDIM]_[.REUSE._INSTANCE]
TEXTSTRING

X and Y	Coordinates of the location of the text string relative to the origin of the schematic.
ORI	Orientation of the text string expressed in degrees. Valid values range from 0.0 to 359.999, in increments of 0.001.
LEVEL	Level or layer number on which the text has been defined. Valid values range from 0 to 250. 0 indicates all levels.
HEIGHT	Text height. Valid values range from 0.01 to 1000.
WIDTH	Text width. Valid values range from 0.01 to 250.
M	Value when text has been mirrored. If the text has not been mirrored, this field is omitted.
HJUST	Horizontal justification. Valid values are: LEFT CENTER RIGHT
VJUST	Vertical justification. Valid values are: UP CENTER DOWN
NDIM	Sequential number of auto-dimensioning in the drawing to which the text belongs. It should be specified if text is a member of some drawing item.
.REUSE.	Optional keyword. Should be specified if text is a member of some reuse instance.
INSTANCE	Reuse instance name
TEXTSTRING	Text string, up to 79 characters, including spaces.

Lines Definition Format

The Lines section defines board outline, copper shapes and cut outs, various keepouts, and 2D drawing items.

Lines control statement

The lines control statement is:

LINES

Lines entry format

A line definition consists of a header line, followed by two parts. If the line items have been combined with text in the schematic, the associated text appears following the line information:

- Header line
- Piece entry definition
- Coordinates
- Optional text information. For more information see the "Text" section.

Lines header definition

The header line consists of:

NAME_LINETYPE_X_Y_PIECES_[TEXT_[SIGSTR]]

NAME	User-defined name of the line item. If the user does not assign a name, the system assigns one. Values are up to 16 alphanumeric characters, no spaces.
LINETYPE	Item type. Valid values are: LINES – 2D lines BOARD – Board outline COPPER – Copper COPCUT – Poured copper cut outs KEEPOUT – Keepout area definition
X_Y	Coordinates of the origin of the line item
PIECES	Number of pieces that make up the line item. Valid values range from 1 to 32767.
TEXT (optional)	Number of text lines associated with the line item
SIGSTR (optional)	Used when LINETYPE is defined as copper. This indicates the netname with which the copper is to be associated.

Line objects that are members of reuse have one additional line:

```
.REUSE._NAME_[RSIGNAL]
.REUSE.           Keyword preceded and followed by dots.
NAME             Reuse instance name
RSIGNAL          Optional reuse signal name, or original signal name
                  from reuse definition. This parameter is specified
                  only for COPPER type.
```

Piece definition

Each piece definition consists of the line type, the number of corners to the piece, and the width of the piece. This line is followed by a line of coordinates for each corner.

```
TYPE_NUMCOORD_WIDTHHIGHT_LEVEL_[RESTRICTIONS]
```

TYPE	Line item type. The following combinations are allowed:		
	LINETYPE	Piece TYPE	Meaning
	LINES	OPEN	Polyline
	LINES	CLOSED	Unfilled closed polygon
	LINES	CIRCLE	Unfilled circle
	LINES	ARWLN1	First dimension arrow
	LINES	ARWLN2	Second dimension arrow
	LINES	ARWHD1	First dimension arrow head
	LINES	ARWHD2	Second dimension arrow head
	LINES	EXTLN1	First dimension extension line
	LINES	EXTLN2	Second dimension extension line
	LINES	BASPNT	Dimension base point
	BOARD	CLOSED	Boards as unfilled closed polygon
	BOARD	CIRCLE	Board as unfilled circle
	BOARD	BRDCLS	Polygon board cutout
	BOARD	BRDCIR	Circle board cutout
	COPPER	COPOP	Copper polyline
	COPPER	COPCLS	Filled copper polygon
	COPPER	COPCIR	Filled copper circle
	COPPER	COPCUT	Polygon void in copper
	COPPER	COPCCO	Circle void in copper

COPCUT	COPCUT	Polygon void in copper pour (unassociated)
COPCUT	CIRCUR	Circle void in copper pour (unassociated)
KEEPOUT	KPTCLS	Polygon keepout
KEEPOUT	KPTCIR	Circle keepout

Note: All polygons and polylines may include arcs.

NUMCOORD	Number of coordinate lines defining the item: For open items, this is the number of corners. Circles have two corners. For closed line items this is the number of corners plus one (to return to the starting corner). Valid values range from 2 to 32767.
WIDTHHGHT	Line width for all types of pieces except KTPCLS and KPTCIR. Valid values range from 0.01 to 250. For KTPCLS and KPTCIR, this parameter is a height restriction if the H flag is specified (see below). Valid values range from 0 to +28000. If H flag is not specified, the value is zero.
LEVEL	Level or layer number on which the piece is defined. Valid values range from 0 to 250. 0 indicates all levels. If the defined line type is KEEPOUT, the level optionally is defined using the code -2 – INNER LAYERS.
RESTRICTIONS	Specified only for KEEPOUT pieces. The following abbreviations are used for restrictions: P – Placement H – Component height R – Trace and copper C – Copper pour and plane area V – Via and jumper T – Test point The abbreviations can be used in sets, for example, HR would mean a keepout with component height and trace restrictions. PVT would mean placement, via, and test point restrictions.

All coordinates are relative to the origin of the line item.

Segment corner format:

X_Y

Arc corner format:

X1_Y1_AB_AA_AX1_AY1_AX2_AY2

X1_Y1	Beginning of the arc
AB	Beginning angle of the arc in tenths of a degree
AA	Number of degrees in the arc in tenths of a degree
AX1,AY1	Lower left point of the rectangle around the circle of the arc
AX2,AY2	Upper right point of the rectangle around the circle of the arc
$AX2 - AX1 = AY2 - AY1$	Diameter of the circle of the arc
$(AX1 + AX2)/2, (AY1 + AY2)/2$	Coordinates of the center of the arc circle

For closed polygons, the first corner will be repeated.

Text data entry format

Refer to the “Text Definition” section. Note reuse data should not be specified for text inside 2D item.

Getting 2D items from the library

Items may be called from the 2D lines library into the design using the GET command. The GET statement is valid only after the LINES statement. The format is as follows:

*GET*_NAME

GET	Control statement.
NAME	Name of the existing library line item in the lines library.

Part Decal Definition Format

The part decal section describes the physical shape of the electrical parts, including pad and drill sizes. Part decals for each of the part types in the design are required, and must be in either the ASCII data input or the parts decal library.

Part decal control statement

The part decal control statement is:

PARTDECAL

Part decal format structure

A part decal consists of the following:

- Header line
- Piece definitions
- Text definitions
- Label definitions
- Pad definitions
- Pad stack definitions

Note: All coordinates are relative.

Header line format

NAME_UNITS_X_Y_PIECES_TERMINALS_STACKS_[TEXT_[LABELS]]

NAME	User-defined decal name. Alphanumeric string can be up to 40 characters long.
UNITS	Units used for this part decal. Valid values are: I – Imperial units, given in mils M – Metric units, given in mm
X and Y	Coordinates where the symbol is placed when viewed in the PowerPCB Decal Editor. You can ignore this data when reading in ASCII files. If copying the data, simply preserve the values from the input field. Use the values 1000, 1000 if you are creating this data.
PIECES	Number of drawing outline items that make up the part decal. Valid values range from 0 to 32767. A definition follows each piece entry.
TERMINALS	Number of terminals that make up the part decal. Valid values range from 0 to 32767. A definition follows each entry.
STACKS	Number of unique pad stack descriptions that define the terminals within the decal. A definition follows each entry.

TEXT (optional)	Number of text strings associated with the decal item. Valid values range from 0 to 32767.
LABELS (optional)	Number of attribute labels associated with the decal item. Valid values range from 0 to 32767.

Piece definition format

Pieces may be 2D lines used to draw the component outline. Pieces may also be keepout, copper, or copper cutouts. Copper can be used as a special component pad or shape for shielding. Copper may have voids within it known as cutouts. Each PIECE contained within the decal, has an entry as follows:

TYPE_NUMCOORD_WIDTHHIGHT_LEVEL_[PINNUM/RESTRICTIONS]

TYPE	Line item type: OPEN CLOSED CIRCLE COPCLS COPOP COPCUT COPCCO COPCIR KPTCLS KPTCIR
LEVEL	Level or layer number on which the piece is defined. Valid values range from 0 to 250; 0 indicates all levels. If the defined piece type is KPTCLS or KPTCIR, the level is optionally defined using the codes: 1 – Opposite side 2 – Inner layers If the defined piece type is COPCLS, COPOP, or COPCIR, the level is optionally defined using the code 1 Opposite side.
PINNUM	Used when the type is COPCLS, COPOP, or COPCIR. It indicates the pin number (minus one) that the copper area is to be associated with. If the copper is not associated with a pin, the field is omitted
RESTRICTIONS	Keyword specified only for KEEPOUT pieces. Use the following abbreviations for a set of restrictions: R – Trace and copper C – Copper pour and plane area V – Via and jumper T – Test point.

All coordinates are relative to the origin of the line item.

For more information, see "Piece definition" in the "Line Definition Format" section of this document.

Text data entry format

Refer to the “Text Definition Format” section.

Restriction: Reuse data cannot be specified in this section.

Label data entry format

Label entry has two lines:

VISIBLE_LX_LY_LORI_LHEIGHT_LWIDTH_LLEVEL_MIRRORED_HJUST_V
JUST_[RIGHT_READING]

VISIBLE	Label visibility type. Valid values are: VALUE – 100 mil FULL_NAME – Geometry.Height NAME – Height FULL_BOTH – Geometry.Height = 100 mil BOTH – Height = 100 mil NONE – Nothing displayed
LX and LY	Coordinates of label origin
LORI	Relative label orientation. Precision is three digits after decimal point.
LHEIGHT	Height of label text.
LWIDTH	Pin width for label text.
LLEVEL	Layer on which the label is located. Valid values range from 0 to 250. 0 means all layers.
MIRRORED	Flag indicating when the symbol has been mirrored to the opposite side of the board. Valid values are: 0 – mirrored (on bottom layer) 1 – not mirrored (on top layer)
HJUST	Horizontal justification. Valid values are: LEFT CENTER RIGHT
VJUST	Vertical justification. Valid values are: UP CENTER DOWN
RIGHT_READING	Right reading status. Valid values are: ORTHO – Orthogonal ANGLED NONE. NONE is the default value.

The second line is the fully structured attribute name, for example “Ref. Des.” or “Part Type.”

Example

```
VALUE 100 100 90.000 100 10 1 0 LEFT DOWN ORTHO
Geometry.Height
```

This example describes a label that shows the value for the Geometry.Height attribute at coordinate 100,100 relative to part origin, and rotated 90 degrees counterclockwise. The label text height is 100 mil, and the pen width is 10 mil. The label appears on layer 1, not mirrored. Justification is left down, and right reading is in orthogonal mode.

Pad definition format

Each line of the pad definition begins with the letter T. For each pad terminal in the part decal, there is an entry as follows:

X_Y_NMX_NMY

X and Y Coordinates of the pin relative to the decal origin. Valid values range from -56000 to 56000.

NMX and NMY Coordinates of the terminal number relative to the pin. Valid values range from -56000 to 56000.

Restriction: Pin numbers are assigned by the order of entries. For example, pin 1 is defined first, pin 2 is defined second, and so on.

Pad stack definition format

The pad stack definition section contains one entry for each unique pad stack within the part decal. Each pad should have a pad stack assigned either explicitly or implicitly. The format for each pad stack entry is defined as follows:

PAD_PINNO_STACKLINES

LEVEL_SIZE_SHAPE_IDIA_FINORI_FINLENGTH_FINOFFSET_DRILL_[PLATED]_[SLOTORI_SLOTLENGTH_SLOTOFFSET]

LEVEL_SIZE_SHAPE_IDIA_DRILL_[PLATED]

PAD Keyword

PINNO Terminal number to which the pad stack applies. A value of 0 indicates all pads except those explicitly listed later.

STACKLINES Number of lines of information pertaining to the pin numbers. The minimum number of entries is 3: top layer, inner layer, and bottom layer.

LEVEL Layer number being defined. Valid values are:
-2 – Top layer
-1 – Inner layer
0 – Bottom layer
1 to 250 – Specific layer number

SIZE Diameter of the pad for round, odd, and annular pads. Length of a side for square pads. Width of oval or rectangular pads.

SHAPE	Pad shape. Valid values are: R – Round S – Square A – Annular O – Odd OF – Oval finger RF – Rectangular finger
IDIA	Inner diameter of an annular pad. If the pad is not annular, this field is omitted.
FINORI	Orientation, in degrees, of oval or rectangular pads. Precision is three digits after the decimal point. Valid values range from 0 to 179.999. If the pad is not oval or rectangular, this field is omitted.
FINLENGTH	Finger length of oval or rectangular pads. Valid values range from +1 to 1000. If the pad is not oval or rectangular, this field is omitted.
FINOFFSET	Offset of the finger of oval or rectangular pads from the electrical center (drill). For fingers with a 0 degree orientation, a positive offset shifts the pad from the electrical center to the right. For fingers with a 90 degree orientation, a positive offset shifts the pad from the electrical center up. Offset values must lie within the pad, and may not be greater than 500. If the pad is not oval or rectangular, this field is omitted.
DRILL	Drill size. Valid values range from 0 to 1000. DRILL only appears on the side where parts are mounted.
PLATED (optional)	If this is N the drill hole is unplated. If this is P or empty, the drill is plated. PLATED only appears on the side where parts are mounted.
SLOTORI	Orientation, in degrees, of the slotted drill. Precision is three digits after the decimal point. Valid values range from 0 to 179.999.
SLOTLENGTH	Length of the slotted drill. Valid values range from +1 to 1000.
SLOTOFFSET	Offset of slotted drill from the electrical center. For drills with a 0 degree orientation, a positive offset shifts the drill from the electrical center to the right. For a drill with a 90 degree orientation, a positive offset shifts the drill from the electrical center up. Offset values must lie within the drill, and may not be greater than 500.

New pad shapes were added in PowerPCB version 2.0 format to describe thermal pads and antipads.

Thermal pads have a shape code of RT or ST:

LEVEL SIZE SHAPE SPOKEORI OUTSIZE SPOKEWIDTH SPOKENUM

SHAPE	RT for a round pad shape ST for square pad shape
SPOKEORI	Orientation of the spoke measured in degrees from the right direction counterclockwise. Precision is to one degree.
OUTSIZE	The outer diameter of the thermal pad or the diameter of the void in plane copper for round pads. The length of the square side for the outer square or the square void in plane copper for square pads.
SPOKEWIDTH	Width of a spoke
SPOKENUM	Number of spokes

Example

-2 60 RT 45 84 15 4

This example describes a round thermal pad on a mounted component side (code -2). The first spoke orientation is 45.000 degrees. The outer diameter of the thermal pad, or diameter of the void in plane copper, is 84 mil. The width of each spoke is 15 mil. The number of spokes is 4. The angular step for spokes is $360/4 = 90$ degrees.

Antipads have a shape code of RA or SA.

LEVEL SIZE SHAPE

SIZE	For a round pad, SIZE is the diameter of the antipad or of the void in plane copper. For a square pad, SIZE is the side length of the square antipad or of the square void in plane copper.
------	--

Via Definition Format

The Via section defines the pad stacks for all the vias defined in the design. It uses the same description fields as the Part Decal section with the addition of the drill start and drill end values used in the definition of any partial vias.

Via entry format

NAME_DRILL_STACKLINES_[DRILL START]_[DRILL END]
LEVEL_SIZE_SHAPE_[INNER DIAMETER]

DRILL START (optional) Starting level for the pads of this via
DRILL END Ending level for the pads of this via

The widths of the pads in a pad stack using the DRILL START and DRILL END values are determined slightly differently from the STANDARDVIA or MICROVIA. In this case, the DRILL START level number is considered the top level in the pad stack. The DRILL END level number is considered the bottom level in the pad stack. Any level number between DRILL START and DRILL END is considered an interior level.

Example

```
STANDARDVIA 37 3  
-2 55 R  
-1 72 A 20  
0 55 R  
BURIEDVIA- 2-3 15 3 2 3  
-2 25 S  
-1 65 A 50  
30 R
```

In this example, DRILL START level 2 is the top level. Its width is 25 square. The DRILL END level of 3 is the bottom level. The pad diameter is 30 round. There are no interior levels defined for this pad stack.

New pad shapes were added in PowerPCB version 2.0 format to describe thermal pads and antipads.

Thermal pads have a shape code of RT or ST.

LEVEL SIZE SHAPE SPOKEORI OUTSIZE SPOKEWIDTH SPOKENUM

SHAPE	RT for a round pad shape. ST for square pad shape
SPOKEORI	Orientation of the spoke measured in degrees. Precision is to one degree.
OUTSIZE	For a round pad, OUTSIZE is the outer diameter of the thermal pad or the diameter of the void in plane copper. For a square pad, OUTSIZE is the length of the square side for the outer square or the square void in plane copper.
SPOKEWIDTH	Width of the spoke
SPOKENUM	Number of spokes

Example

-2 60 RT 45 84 15 4

This example describes a round thermal pad on a mounted component side (code -2). The first spoke orientation is 45.000 degrees. The outer diameter of the thermal pad, or diameter of void in plane copper, is 84 mil. The width of each spoke is 15 mil. The number of spokes is 4. The angular step for spokes is $360/4 = 90$ degrees.

Antipads have a shape code of RA or SA.

LEVEL SIZE SHAPE

SIZE

For round pads, SIZE is the diameter of the antipad or the diameter of the void in plane copper.

For square pads SIZE is the side length of the square antipad or the square void in plane copper.

Jumper Definition Format

The Jumper section defines all jumpers in the design.

Jumper header format

The Jumper header line consists of:

NAME_FLAGS_MINLEN_MAXLEN_LENINCR_LCOUNT_PADSTACK [END-PADSTACK]

NAME	Name or reference designator of the jumper
FLAGS:	Via flag is mandatory: V – Via enabled N – No via flag Other flags are optional: W – Wirebond jumper D – Display special silk G – Glued
MINLEN	Minimum possible length of the jumper
MAXLEN	Maximum possible length of the jumper
LENINCR	Length increment
LCOUNT	Number of reference designator labels associated with the jumper
PADSTACK	Pad stack used for the start jumper pin, or start and end jumper pins if END-PADSTACK is absent
END-PADSTACK	Pad stack used for end jumper pin

Each associated reference designator is specified on one line. The line uses the same format as the first line in the label definition. For more information see the “Label Data Entry Format” section.

Example

```
STDJMP VG 50 200 25 1 JMPSTACK  
VALUE 4500 3200 45.000 100 10 1 1 LEFT DOWN
```

In this example, the jumper named STDJMP has a label positioned at 4500 3200, lower left justified, rotated 45 degrees counterclockwise, with text height of 100, text pen width of 10, located on layer 1 and mirrored. The minimal length of this jumper can be 50, maximal length can be 200, and the increment 25 (only the following lengths would be valid: 50, 75, 100, 125, 150, 175, and 200). This jumper is glued and it uses the pad stack JMPSTACK at both ends.

Part Type Definition Format

The Part Type section defines parts used in the design. For every part referenced in the Part section, an entry must appear in the library or in the Part Type section. All part decals referenced in the Part Type section must be listed in the Part Decal section or exist in the library.

Part type control statement

The part type control statement is:

```
*PARTT[YPE*]
```

Part type entry format

Each part type entry consists of the following:

- Part type header line
- Gate information
- Signal information (optional)
- Alphanumeric pins (optional)

Part type header format

```
PTYPENM_DCALNM[:ALTDICAL...]_UNITS_TYPE_GATES_SIGNALS_ALPIN  
M_FLAG_[ECO]
```

PTYPENM	Part type name and optional value, tolerance, and PCB decal information. The part type name can be up to 40 alphanumeric characters long.
DCALNM[:ALTDICAL]	PCB decal name (and alternate PCB decal name). A PCB decal can be up to 40 alphanumeric characters long. These decals represent the physical elements of the part on the printed circuit board. The maximum number of decals per part type is 16. All decals should have the same number of terminals.
UNITS	The units in use for this part type. Valid values are: I – Imperial units, given in mils M – Metric units, given in mm
TYPE	The logic type. Valid values are any three alphanumeric characters. Default is UND for undefined.
GATES	Number of gates in the part. Valid values range from 0 to 32767.
SIGNALS	Number of standard signals predefined in the part, usually power and ground. Valid values range from 0 to 50. The default value is 0.
ALPINM	Number of alphanumeric pin numbers defined in the part. Valid values range from 0 to 32787. The default

	is 0. If a non-zero value is specified, it should be equal to the number of terminals in each part decal.
FLAG	Number representing the part type: 0 – Normal part 1 – Connector 2 – Off page reference 128 – Error in the part
ECO	Optional. ECO registration status for the part type: Y means part add and delete operations are registered in the ECO file. N means add and delete operations are not registered in the ECO file.

Gate format

The gate format consists of two parts: the header line which describes the gate type and the amount of pins in the gate and the second line which describes the actual pins within the gate.

G_GATESWAP_PINS

PINNUMBER.SWPTYP.PINTYP[.FUNCNAME]

G	Keyword
GATESWAP	Gate swap type: Gates with the same swap type are assumed to be electrically equivalent. A gate with a swap type of 0 is not swappable.
PINS	Number of pins in the gate
PINNUMBER	Electrical pin number of the pin in the gate. Pin numbers may not be duplicated. Valid values range from 1 to 32767.
SWPTYP	Swap type of the pin: Pins with the same swap type are assumed to be electrically equivalent. The scope of the pin swapping is within a gate, so pins must be in the same gate and have the swap type in order to be swappable. A pin with a swap type of 0 is not swappable.
PINTYP	Electrical type of the pin. Values are: S Source pin B Bidirectional pin C Open collector pin or-tieable source pin T Tri-state pin L Load pin Z Terminator pin P Power pin G Ground pin U Undefined pin

FUNCNAME Optional
Pin functional name, can be up to 14 alphanumeric characters.

Signal format

This section describes standard signals for parts. Typically, standard signals are power or ground, but any signal name may be used.

SIGPIN_PINNO_WIDTH_SIGNM

SIGPIN	Keyword
PINNO	Pin number of the signal pin. Valid values range from 1 to 32767.
WIDTH	Width of the track for the connection in the PCB design. Valid values range from 1 to 250.
SIGNM	Name of the standard signal. Signal names may be up to 47 alphanumeric characters long.

Alphanumeric pin format

This section defines a pin name for every pin in the part. Pin names may be up to seven alphanumeric characters long. There is a one-to-one correspondence between pin numbers and pin names. The order is from pin 1 to the last pin. If pin names are used, *all* pins must be defined.

The format is:

N1 _N2 _N3 _...NX

Part List Definition Format

The Part List section provides the list of components by reference designator and part type in the design. The Part List section is unique for each design. Part types referenced in the Part List section must be defined in either the Part Type section of the ASCII file or in the system library.

Part list control statement

The part list control statement is:

PART

Automatic standard signal netlist generation

A second control statement may, optionally, be added. This control statement automatically generates a netlist by extracting all standard signals found in the part type file which was referenced in the Part List section. The format for the automatic standard signal netlist generation control statement is:

STANDARD

Part list entry format

The format for each entry in the Part List section is as follows:

REFNM_PTYENM_X_Y_ORI_GLUE_MIRROR_ALT [CLSTID CLSTATTR
BROTHERID LABELS]

REFNM Unique reference designator name. The reference designator may be up to 15 alphanumeric characters long.

PTYENM Part type name and optional PCB decal information. The part type can be up to 40 alphanumeric characters long.

The PCB decal will be appended at the end of the part type separated by an @. The PCB decal will be appended to the part type in the ASCII file only if a part decal has been modified using Setup/Pad Stacks. Value and tolerance are not specified in this field.

X and Y Coordinates of the part's placement origin in the design. Default is 0.

ORI Orientation in degrees of the part in the design. Valid values range from 0.0 to 359.999, in increments of 0.001 of a degree. Default is 0.0.

GLUE System flag indicating when the symbol is glued down. Valid values are:

G – Glued

U – Unglued. U is the default.

MIRROR Flag indicating when the symbol has been mirrored to the opposite side of the board. Valid values are:

M – Mirrored (on bottom layer)

N – Not mirrored (on top layer)

ALT	Alternate decal number. This sequential number follows the sequence as defined in the part type file, that is, DIP14:DIP14\SO:DIP14\SOL are 0, 1, 2, respectively. Valid values range from 0 to 15. Default is 0.
CLSTID	ID number of the placement cluster to which this component belongs. Cluster information is found in the Cluster section.
CLSTATTR	Integer value representing the following set of bit definitions: 00 – Component intersects board 01 – Component outside board 02 – Glue flag 03 – SMD flag 04 – Component is a cluster set 05 – Brother is a cluster 06 – Mirror flag 07 – Pins mirror flag 08,09 – Unused 10 – Mark uncrossed component 11 – Don't draw connection 12 – Mark failure of push 13 – Hidden for other component 14 – Temporary highlight flag 15 – Temporary draw flag
BROTHERID	ID number of the placement cluster, union, or array to which this component belongs. Cluster information is found in the Cluster section.
LABELS	Number of part labels. Each label is a two-line entry as described in the " Label data entry format" section.

If the part is a member of a reuse instance, the following line is added just after the part header line:

```
.REUSE._INSTANCE_PART
.REUSE.      Keyword. Preceded and followed by dots.
INSTANCE    Name of the reuse instance
PART        Part reference designator inside the reuse.
```

Example

```
U1L MY_AMP
.REUSE. LeftChannel U1
U1R MY_AMP
.REUSE. RightChannel U1
```

Shortcut part list entry format

A shortcut for entering multiple parts in one line is available. The parts must be sequential, and must be of the same part type.

PRE{N1-N2}_PTYPENM_X_Y_ORI_GLUE_MIRROR_ALT

PRE Reference designator name prefix shared by all parts, up to 5 alphanumeric characters long*.

N1 First reference designator suffix in the sequence. *

N2 Last reference designator suffix in the sequence. *

* Prefix plus suffix may not exceed 15 characters.

Example

C{2-20}_CKO5...

This example creates 19 capacitors, labeled C2 through C20, with a part type CKO. All other values are identical to the normal part list definition.

Connection Definition Format

There are three ways to define connectivity in the PowerPCB ASCII format. These are the CONN connection list format, the NET netlist format, and the ROUTE route format. The ASCII OUT function produces output in the CONN format when CONN is selected, and produces output in the ROUTE format when ROUTE is selected. For user-defined connectivity input, using the NET format is recommended.

The NET format is the most flexible, and offers a shortcut. You can mix the NETLIST and CONN formats when inputting data, however, do not repeat a previously named signal.

All reference designators listed in the Net, Conn, and Route sections must have a reference designator and part listed in the Part List section. The format for each of the lists is defined below:

Connection list definition

The connection list has a series of entries for each signal. These entries consist of the signal name and all the connections in the signal. The format is an ordered set of connections, each made up of a pair of pins. After the first connection entry, one of the pins in each subsequent pair must already have been listed in a previous entry. The ordered pairs of points that make up the connection list explicitly define the connections created in PowerPCB.

Connection list control statement

The connection list control statement is:

```
*CONN[ECTION*]
```

Connection list entry format

The connection list format consists of the following:

- Header line
- Pin pair lines (one or more)

Connection list header format

The header line format is as follows:

```
*SIG[NAL]*_SIGNAME_[SIGFLAG_[COLOR]]
```

SIGNAL Keyword

SIGNAME Name of the signal. A signal may be up to 47 alphanumeric characters long.

SIGFLAG Signal visibility flag. Bit 10 should be set to specify visibility for the signal. Default is 0.

See PCB section ALLSIGFLAG for details.

COLOR Color by net value. Valid values can range from -2 to 15. Default is -2, meaning no specific color set for this net.

Example

```
*SIGNAL* $$$8878-0 -2
```

Pin pair line format

The Pin Pair Line format is as follows:

```
REFDES.PIN_REFDES.PIN
```

REFDES.PIN Reference designator and pin number. There are always two refdes.pin listed to each signal. Reference designators may be up to 15 alphanumeric characters long. Pins may be up to seven characters long and may be numeric, alphanumeric, or alphabetic.

If any pins are members of reuse, they are followed by reuse data in the same line:

```
REFDES.PIN_[.REUSE._INSTANCE_SIGNAL]_REFDES.PIN_[.REUSE._INST  
ANCE_SIGNAL]
```

.REUSE. Keyword. Preceded and followed by dots.

INSTANCE Reuse instance name

RSIGNAL Netname inside the reuse

Example

```
U1L.7 .REUSE. LeftChannel GROUND U1R.7 .REUSE. RightChannel  
GROUND
```

Netlist definition format

An alternative, and preferred, way to enter connection data is to use the Netlist format because of its flexibility. The number of pin entries may be 1 or more. For multiline entries, you don't need to repeat the pin from a previous entry. PowerPCB will tie together multiple SIGNAL entries into single nets based upon the signal name.

Netlist control statement

The netlist control statement is:

```
*NET[LIST*]
```

Netlist format

The Netlist format consists of one header line, and one line with the pins list.

The Header line starting with SIGNAL has exactly the same format as for the Connection section.

The next line is:

```
REFDES.PIN_REFDES.PIN_REFDES.PIN_REFDES.PIN...
```

REFDES.PIN Reference designator and pin number. At least two REFDES.PIN items should be listed for each signal. See the "Connection" section for details.

If any of the pins are members of the reuse, the pin should be followed by reuse data in the same line:

```
REFDES.PIN[_REUSE._INSTANCE_RSIGNAL]_REFDES.PIN[_REUSE._INST  
TANCE_RSIGNAL]...
```

Example

```
U1L.1 .REUSE. A1 CLC U2L.2 .REUSE. A1 CLC1 U1R.1 .REUSE. A2 CLC  
U2R.2 .REUSE. A2 CLC1
```

Shortcut netlist format

A shortcut is available for entering multiple pins as one item. All pins must have the same reference designator or pin number, with sequential names.

```
PRE{N1-N2}.{PIN1-PIN2}_PRE{N1-N2}.{PIN1-PIN2}
```

PRE Reference designator name prefix shared by all parts. This can be up to five alphanumeric characters long. *

N1 First reference designator suffix in the sequence. *

N2 Last reference designator suffix in the sequence. *

PIN1 First pin in the sequence. Only pins may be used. Valid values range from 1 to 32767.

PIN2 Last pin in the sequence. Only pins may be used. Valid values range from 1 to 32767.

* Prefix plus suffix may not exceed six characters.

Example

```
*SIG GND  
U{4-8}.{7-8}
```

This example creates a connection list for the signal ground, consisting of the following nodes:

```
U4.7 U5.7 U6.7 U7.7 U8.7 U4.8 U5.8 U6.8 U7.8 U8.8
```

Route Definition Format

The Route section defines both the logical and physical connectivity for a circuit. When present, the Route section replaces the Conn and Net sections of the ASCII file. The Route section allows for the specification of each segment of the route, including layer, via type, width, and routability. The route data is connection oriented; every connection begins and ends at a component pin. When shared copper is present, each connection that is part of the shared segments must be defined individually. Partial routes are supported in the format.

Each signal is separately identified, and consists of the signal header line and the pin pair line, followed by the coordinates which make up the route. As in the Connection List section, each connection pin pair in a signal entry must include one pin previously entered in the signal entry. The signal net must be listed in its entirety. All width statements in the format are modal, and will affect the global width, and subsequent route widths. It is necessary to reset to the correct width with a width statement in the next signal.

Route section control statement

The route control statement is:

```
*ROUTE*
```

Route entry format

The route entry format consists of the following:

- Header line
- Pin pair line
- Coordinates
-
- Coordinates

Route header format

The Route header line is the SIGNAL line as specified in the Connection section. For more information see "Connection list header" in "Connection Definition Format."

Pin pair line format

```
REFDES.PIN_REFDES.PIN
```

Reuse information should not be included in the pin pair line in this section. For more information see "Pin pair line format" in "Connection Definition Format."

Route coordinates line

Base part

The base part is mandatory. Other parts are optional and should be on the same line as the base part.

X_Y_LAYER_SEGWIDTH_FLAG_[ARCDIR/VIANAME][THERMAL]

X and Y	Route coordinates
LAYER	Layer on which the route starts, ends, or changes layer. For partial routes, the unrouted portion takes on a layer number 0. A layer value of 31 indicates the end of the route/connection at a component pin.
SEGWIDTH	Route segment width
FLAG	Router conditions flag. Integer field with valid values ranging from 0 to 32767, These values represent router conditions that exist at the X and Y coordinate location. Flag bit 6, if set together with a via name in the track corner <i>or</i> assigned to the first or last corner of a track, defines the coordinate of a test point. 0x1000 – Arc center flag 0x0200 – Teardrop prohibit flag (Left) 0x0400 – Teardrop prohibit flag (Right)
ARCDIR	Specified for arc centers only when the arc begins on the previous corner and ends on the following corner. Valid values for ARCDIR: CW -- Clockwise CCW -- Counterclockwise
VIANAME	Name of the via pad stack used at the coordinate location. The pad stack description for these vias should be found in the Via section, for example, BURIEDVIA1-2.
THERMAL	Optional keyword can be specified for component pins and vias. If THERMAL is omitted, PowerPCB will not generate thermal spokes for CAM planes and mixed-plane areas. Since PowerPCB version 2.0, this keyword is specified unless the user disables thermal generation to this pin or via. It is possible that no thermal will be actually generated because of improper pad, nonplane net, and so on.

Teardrop part

The base part is mandatory. Other parts are optional and should be on the same line as the base part.

TEARDROP_[P_WIDTH_LENGTH_[FLAGS]]_[N_WIDTH_LENGTH_[FLAGS]]

TEARDROP describes the actual parameters of the teardrop:

WIDTH Relative width of teardrop, expressed in percentage.

LENGTH Relative length of teardrop, expressed in percentage.

FLAGS: Position modifier (mandatory):

P – Teardrop on previous trace

N – Teardrop on next trace

Shape modifier(optional):

L – Lined

C – Curved

Adjustable modifier(optional):

A – Add for adjustable teardrop

Teardrops can be defined only for component pins and vias.

Position modifier P is illegal in the first line and N is illegal in the last line of the Pin Pair section.

Jumper part

The base part is mandatory. Other parts are optional and should be on the same line as the base part.

[JUMPER_NAME JUMPER_FLAG]

JUMPER_NAME NAME is the name of the jumper, starting at the current point. NAME should be previously declared in the Jumper section.

JUMPER_FLAG S – Route pin is the starting jumper pin

E – Route pin is the ending jumper pin

Note: The above Route section describes the ASCII OUT file which is generated when either ALL or TRACKS are selected during the creation of the ASCII OUT file.

Reuse part

The base part is mandatory. Other parts are optional and should be on the same line as the base part.

REUSE INSTANCE RSIGNAL

Or

R

The short form is used to avoid duplication. Data from the previous line is assumed. The short form cannot be the first reuse description in a pin pair.

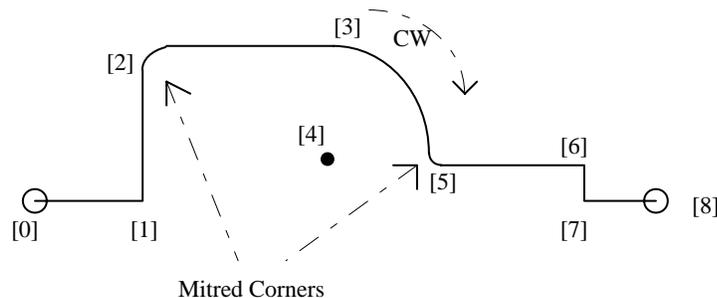
Route example

```

*ROUTE*
*SIGNAL* 5 495300 0 0 0 -2
  U18.14  U28.14  R 0 20
[0] 49530000 118110000 1 4953000 0 REUSE LeftChannel INPUT
[1] 55740300 118110000 2 4953000 0 STANDARDVIA R
[2] 55740300 135674100 1 4953000 0 BURIEDVIA1-2 R
[3] 84410000 135674100 31 4953000 0 R
*ROUTE* (ARC Example)
*SIGNAL* 5 495300 0 0 0 -2
  U18.14  U19.14  R 0 20
[0] 49530000 118110000 2 4953000 0
[1] 55740300 118110000 2 4953000 0 (No Miter)
[2] 55740300 135674100 2 4953000 24576 (Miter index 3)
[3] 72390000 135674100 2 4953000 24576
[4] 55740300 126530100 2 4953000 28672 CW (Arc Center + Miter
index 3)
[5] 81534000 126530100 2 4953000 24576
[6] 92583000 126530100 2 4953000 0
[7] 92583000 118110000 2 4953000 0
[8] 100965000 118110000 31 4953000 24576

```

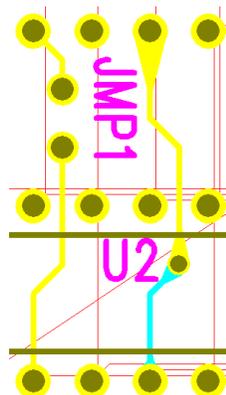
Explanation of miter description in the above example



FLAG value 24576 = 'x'0110 0000 0000 0'00'miter index = 3

FLAG value 28672 = 'x'0111 0000 0000 0'00'miter index = 3 PLUS arc bit ON

Jumper and Teardrop Example



```

*SIGNAL* NET2 10 0 0-0 -2
U2.2  U1.2  P 0 36
8500 6900 1 10 1536
8500 7050 1 10 1536
8550 7100 1 10 1536
8550 7300 0 10 1536 JMPVIA_AAAAA JUMPER JMP1 E
8550 7400 1 10 1536 JMPVIA_AAAAA
8550 7450 1 10 1536
8500 7500 31 10 1536
U3.2  U2.2  U 0 4
9400 6900 0 10 1536
8500 6900 31 10 1536
U4.2  U3.2  U 0 4
9400 7500 0 10 1536
9400 6900 31 10 1536

*SIGNAL* NET4 10 0 0-0 -2
U2.4  U1.4  R 0 20
8700 6900 4 10 1024 TEARDROP N 60 50 C
8700 7050 4 10 1536
8750 7100 1 10 0 SMALL TEARDROP P 90 90 N 90 200 L
8750 7300 1 10 1536
8700 7350 1 10 1536
8700 7500 31 10 512 TEARDROP P 100 200 L
U3.4  U2.4  U 0 4
9600 6900 0 10 1536
8700 6900 31 10 1536
U4.4  U3.4  U 0 4
9600 7500 0 10 1536
9600 6900 31 10 1536

```

Copper Pour Definition Format

The Copper Pour section is used to define copper flood items.

Pour control statement

The copper pour control statement is:

POUR

Pour entry format

A pour definition consists of the following:

- Header line
- Piece entry definition
- Coordinates

Pour header definition

The header pour item definition consists of:

Hatch, pour, and void outlines

NAME_TYPE_XLOC_YLOC_PIECES_FLAGS_[OWNERNAME
SIGNAME[_HATCHGRID_HATCHRAD]]

NAME	User-defined name of the pour item. If the user does not assign a name, the system assigns one. Values can be up to 16 alphanumeric characters long, no spaces.
TYPE	Type of item. Allowed values are: HATOUT – Outline for hatched flood. Created by the flood process. HATOUT is the outer outline of a hatched area. PADTHERM – Thermal relief for pads. Created by the flood process. POUROUT – Outline for solid flood. POUROUT is the outline created by the user. VIATHERM – Thermal relief on via. Created by the flood process. VOIDOUT – Outline for void cutout. Created by the flood process. VOIDOUT is the inner outline of a hatched area.
XLOC_YLOC	Coordinates of the origin of the pour item relative to the system origin; expressed in mils.
PIECES	Number of pieces that make up the pour item. Valid values range from 1 to 65535.
FLAGS	Pour flag bit mask associated with the line item.
OWNERNAME	(optional) Used if the pour item is owned by a previously defined pour item. Pour outlines always have themselves as an owner. Hatch outlines, via thermals, and pad thermals, are owned by the pour outline they are physically in. Void outlines are owned by the containing hatch outline. *

SIGNALNAME	(optional) Used if the pour item is associated with a signal. *
HATCHGRID	(optional) Copper pour hatch grid. Valid values range from 0.001 to 250
HATCHRAD	(optional) Ratio for creating smooth hatch outlines. Valid values range from 0 to 9999999.
FLAGS.	If bit 4 is set, a diagonal hatch is generated; otherwise an orthogonal hatch is generated, If bit 5 is set, vias are flooded over.

*If independent pour attributes are specified, OWNERNAME and SIGNAME are also specified.

Pour outline objects that are members of a reuse have one additional line:

.REUSE._ INSTANCE _[RSIGNAL]

.REUSE. Keyword. Preceded and followed by dots.

INSTANCE Reuse instance name

RSIGNAL Optional reuse signal name, or the original signal name from the reuse definition.

Pad and via thermals

NAME_TYPE_XLOC_YLOC_PIECES_FLAGS_OWNERNAME_INT1_INT2

INT1 and INT2 Indices used in PowerPCB. Other routines reading the ASCII file may safely ignore them. Routines writing the ASCII file should simply pass the values along unchanged.

Piece definition

Each piece definition consists of the type of pour, the number of corners to the piece, the number of arcs to the piece, its width, and its level. This line is followed by a line of coordinates for each corner.

TYPE_CORNERS_ARCS_WIDTH_LEVEL

XLOC_YLOC (format for circles and segments)

XLOC_YLOC_BEGINANGLE_ENDANGLE (format for arcs)

TYPE Type of pour item:
SEG – Segments
POLY – Polygons
CIRCLE – Circle, except for combined keepouts, where type is CUTOUT for polygons, CIRCUT for circle. Via thermals and pad thermals are described as segments only. All other pour shapes are described as polygons or circles.

CORNERS Number of coordinates defining the item. All coordinates are with respect to the pour item origin (XLOC and YLOC above). This value is set to the number of corners for open items. Closed polygons have this value set to the number of corners plus one (the first and last corners are repeated). Segments have two corners defining the endpoints. Circles have two points on a

horizontal diameter.

ARCS	Number of arcs defining the item
WIDTH	Line width in mils. Valid values range from 0.01 to 250.
LEVEL	Level or layer number on which the piece is defined. Valid values range from 1 to 250.

All coordinates are relative to the origin of the line item. For pieces made up of lines, X and Y are the coordinates, in mils (or respective units) of each successive corner of the line item relative to the first point in the line item. For closed polygons, the first corner will be repeated.

For pieces made up of arcs, X and Y are the coordinates of the center of the arc. Additionally, the following information is required:

Arc format

XLOC_YLOC_BEGINANGLE_ROTATION

XLOC_YLOC Center of the arc

BEGINANGLE Beginning angle of the arc in tenths of a degree

ROTATION Angle of the arc, in tenths of a degree. Counterclockwise for a positive angle, clockwise for a negative angle.

Pieces of type CIRCLE are described by two corners, representing ends of horizontal diameter. In other words, if XC, and YC are coordinates of the circle center and R is the circle radius, the following two lines will be generated:

(XC-R) YC

(XC+R) YC

Miscellaneous Section

Net_Restriction Vias section is removed.

Power Router Strategy section is removed.

This section is used to store CAM settings, design rules, attributes, layer parameters, and assembly variance data.

Board drilling rules

The board drilling rules can be created within PowerPCB. The MISC header ASSIGN_PAURED_LAYERS represents the top and bottom layer numbers for drilling.

Example

```
ASSIGN_PAURED_LAYERS
{
  PAURED_LAYERS 1 4
  PAURED_LAYERS 1 2
  PAURED_LAYERS 3 4
}
```

This set of drilling rules implies that the board will be drilled three times: layers 1 and 2 *and* layers 3 and 4 will be drilled individually, then all layers (1,2,3, and 4) will be drilled.

Hierarchical rules section

This section contains the textual representation of the Hierarchical Rules which the user sets using PowerPCB Design Rules.

A MISC header RULES_SECTION PARENT identifies the Hierarchical Rules section. The section consists of three subsections:

Net Classes subsection (header NET_CLASS DATA)

Pin Pair Groups subsection (header GROUP DATA)

Design Rules subsection (header DESIGN RULES)

The format is as follows:

RULES_UNIT	Hierarchical rules header for basic units, Use MIL., INCHES, or METRIC to set other units for the Layer section. It is recommended that you use the same units as in the ASCII file header.
{	
NET_CLASS DATA	Beginning of the Net Classes subsection. If there are no net classes defined, this section can be omitted.
{	
. . .	
}	End of the Net Classes subsection.
GROUP DATA	Beginning of the Groups subsection. If there are no groups defined, this section can be omitted.

```

{
. . .
}
DESIGN RULES      Beginning of the Design Rules subsection.
{
. . .
}
}
}
End of Hierarchical Rules section.

```

Net classes

```

NET_CLASS_DATA    Beginning of the Net Classes subsection. If there are no net
                  classes defined, this section can be omitted.
{
NET_CLASS class_name1  class_name1 is the name of the first net class that belongs to
the hierarchical rules.
{
NET net_name1         net_name1 is the name of the first net that belongs to class
class_name1.
. . .
NET net_nameN        net_nameN is the name of the last net that belongs to class
class_name1.
}
. . .
NET_CLASS class_nameN  class_nameN is the name of the last net class that belongs to
hierarchy rules.
{
NET net_name1         net_name1 is the name of the first net that belongs to class
class_nameN.
. . .
NET net_nameN        net_nameN is the name of the last net that belongs to class
class_nameN.
}
}
}
End of the Net Classes subsection.

```

Pin pair groups

GROUP DATA	Beginning of the Groups subsection. If there are no groups defined, this section can be omitted.
{	
GROUP <i>group_name1</i>	<i>group_name1</i> is the name of the first pin pair group that belongs to hierarchy rules.
{	
CONNECTION <i>1refdes.pin,2refdes.pin</i>	<i>refdes.pin</i> is a part reference designator and pin number, separated by a period. Separate the <i>refdes.pin</i> fields with a comma.
. . .	Other CONNECTION records.
CONNECTION <i>Nrefdes.pin,Xrefdes.pin</i>	
}	
. . .	Other GROUP records.
GROUP <i>group_nameN</i>	
{	
. . .	
}	
}	End of the Groups subsection.

Design rules

DESIGN RULES	Beginning of the Design Rules subsection.
{	
RULE_SET (1)	First rule set.
{	Beginning of rule set.
. . .	
}	End of rule set.
. . .	Other RULE_SET records.
RULE_SET (N)	Last rule set.
{	
. . .	
}	
DIF_PAIR D1	First differential pair.
{	
. . .	
}	
. . .	Other differential pair records.
DIF_PAIR DN	Last differential pair.
{	
. . .	
}	
}	End of the Design Rules subsection.

Rule set format

Rule Set may represent clearance rules (CLEARANCE_RULE keyword), high speed rules (HIGH_SPEED_RULE keyword), or routing rules (ROUTE_RULE keyword) FOR a particular object or group of objects from the same hierarchy level. Rule Set may also include conditional rules for particular objects AGAINST other objects on a particular LAYER.

RULE_SET (*N*) Beginning of the rule set, where *N* is a unique ID of the rule set.

{

FOR : Beginning of the object list.

{

DEFAULT : OR

Objects from the same hierarchy level can be grouped together.

NET_CLASS *name1*

. . .

NET_CLASS *nameN* OR

NET *name1*

. . .

NET *nameN* OR

GROUP *name1*

. . .

GROUP *nameN* OR

CONNECTION
name1

. . .

CONNECTION
nameN

} End of object list.

AGAINST : Beginning of conditional list.

{

DEFAULT : OR

DEFAULT lets you define groups of objects in the For section. Specific objects, like NET_CLASS, NET, GROUP, or CONNECTION, should be paired with a single object in the For section. If an object other than DEFAULT is defined in the Against section, the rule set represents conditional rules.

NET_CLASS *name* OR

NET *name* OR

GROUP *name* OR
 CONNECTION *name*
 } End of conditional list.
 LAYER *N* N is the layer number. Valid values range from 0 to 64. LAYER 0 means ALL layers. Any non-zero layer number means that conditional rules are defined for an object in the For section against this layer.

CLEARANCE_RULE
 :

Note: Only one rule section is allowed per rule set (CLEARANCE_RULE, HIGH_SPEED_RULE, or ROUTE_RULE, for example).

{
 TRACK_TO_TRACK 457200 All values are in PowerPCB basic units in this example.
 VIA_TO_TRACK 457200 Units are specified at the beginning of the rules section.
 VIA_TO_VIA 457200
 PAD_TO_TRACK 457200
 PAD_TO_VIA 457200
 PAD_TO_PAD 457200
 SMD_TO_TRACK 457200
 SMD_TO_VIA 457200
 SMD_TO_PAD 457200
 SMD_TO_SMD 457200
 COPPER_TO_TRACK
 457200
 COPPER_TO_VIA 457200
 COPPER_TO_PAD 457200
 COPPER_TO_SMD 457200
 TEXT_TO_TRACK 457200
 TEXT_TO_VIA 457200
 TEXT_TO_PAD 457200
 TEXT_TO_SMD 457200
 OUTLINE_TO_TRACK
 457200
 OUTLINE_TO_VIA 457200
 OUTLINE_TO_PAD 457200
 OUTLINE_TO_SMD 457200

```

DRILL_TO_TRACK 457200
DRILL_TO_VIA 457200
DRILL_TO_PAD 457200
DRILL_TO_SMD 457200
DRILL_TO_COPPER
457200
SAME_NET_SMD_TO_VIA
457200
SAME_NET_SMD_TO_CRN
457200
SAME_NET_VIA_TO_VIA
457200
SAME_NET_PAD_TO_CRN
457200
MIN_TRACK_WIDTH
381000
REC_TRACK_WIDTH
381000
MAX_TRACK_WIDTH
457200
DRILL_TO_DRILL 457200
BODY_TO_BODY 457200
SAME_NET_TRACK_TO_C
RN 457200
} OR
HIGH_SPEED_RULE :
{
MIN_LENGTH 0
MAX_LENGTH 1905000000
STUB_LENGTH 38100000
PARALLEL_LENGTH
38100000
PARALLEL_GAP 7620000
TANDEM_LENGTH
38100000
TANDEM_GAP 7620000
MIN_DELAY 0.000000 Delay in ns.
MAX_DELAY 10.000000

```

```

MIN_CAPACITANCE           Capacitance in pF.
0.000000

MAX_CAPACITANCE
10.000000

MIN_IMPEDANCE             Impedance in Ohm.
50.000000

MAX_IMPEDANCE
150.000000

SHIELD_NET *
SHIELD_GAP 7620000
MATCH_LENGTH_TOLER
AN-E -7620000
}                               OR
ROUTE_RULE :
{
LENGTH_MINIMIZATION_     Minimization types:
TYPE 1                    0 – None
                           1 – Total length
                           2 – Horizontal
                           3 – Vertical
                           4 – Serial source
                           5 – Parallel source
                           6 – Mid driven

PIN_SHARE Y
SMD_SHARE Y
VIA_SHARE Y
TRACE_SHARE Y
AUTO_ROUTE Y
RIPUP Y
SHOVE Y
SHOVE_PROTECTED Y
ROUTE_PRIORITY 3
VALID_LAYER 1
VALID_LAYER 2
VALID_VIA_TYPE
STANDARDVIA
}
}                               End of the rule set.

```

Differential pairs format

DIF_PAIR D1

Beginning of the differential pair set. This is an example of the differential pair defined for two nets, named GND and VCC.

```
{  
NET GND  
NET VCC  
MIN_LENGTH 3810000  
MAX_LENGTH 152400000  
GAP 7620000  
}
```

End of the differential pair set.

DIF_PAIR D2

Beginning of the differential pair set. This is an example of the differential pair defined for two connections (pin pairs).

```
{  
CONNECTION U2.7,U1.1  
CONNECTION U1.4,U2.4  
MIN_LENGTH 3810000  
MAX_LENGTH 152400000  
GAP 7620000  
}
```

Attributes

The Attribute section consists of two optional parts, the attributes dictionary and the attribute values.

Attributes dictionary format

The Attributes dictionary is optional. This part describes value types for attributes.

ATTRIBUTES DICTIONARY Attributes dictionary header.

```
{
```

ATTRIBUTE PowerGround Full structured name of the attribute is PowerGround.

```
{
```

TYPE BOOLEAN Attribute type is Boolean:

Yes and No are possible values.

```
INHERITANCE NET  
NETCLASS
```

Attribute can be defined for nets. If no value is specified for a net, and the net is a member of a net class, a value for the net class is assumed.

ECO_REGISTRATION Y	Register attribute changes in the ECO file.
READONLY N	Attributes can be modified
SYSTEM Y	Attribute is used by PowerPCB
HIDDEN N	Attribute is visible in dialog boxes.
}	End of attribute Power or Ground declaration.
ATTRIBUTE Voltage	Full structured name of attribute is Voltage.
{	
TYPE QUANTITY	Attribute type is quantity. Units of measurement are supported for this attribute.
QUANTITY Voltage	Quantity name.
ABBR V	Abbreviated unit name is V.
UNIT volt	Full unit name is volt.
MIN -100kV	Minimum allowed value is -100kV, or -100000 volt.
MAX 100kV	Maximum allowed value is 100kV, or 100000 volt.
INHERITANCE NET	Attribute can be defined for nets. If no value is specified for a net and the net is a member of a net class, a value for the net class is assumed.
NETCLASS	
ECO_REGISTRATION Y	Register attribute changes in the ECO file.
READONLY N	Attributes can be modified.
SYSTEM Y	Attribute is used by PowerPCB
HIDDEN N	Attribute is visible in dialog boxes.
}	End of attribute Voltage declaration.
ATTRIBUTE	Full structured name of attribute is Geometry.Height.
Geometry.Height	
{	
TYPE QUANTITY	Attribute type is quantity. Units of measurement are supported for this attribute.
QUANTITY Size/Dimension	This is a special quantity name: Current units will be used for input and output.
MIN 0.00 mil	Minimum allowed value is 0 mil.
MAX 25000.00 mil	Maximum allowed value is 25,000 mil or 25".
INHERITANCE PART	Allowed types and search order.
PARTTYPE DECAL PCB	
ECO_REGISTRATION N	Do not register attribute changes in the ECO file.
READONLY N	Attributes can be modified.
SYSTEM Y	Attribute is used by PowerPCB.

HIDDEN N	Attribute is visible in dialog boxes.
}	
ATTRIBUTE DFT.Nail Count Per Net	Full structured name of the attribute DFT.Nail Count Per Net.
{	
TYPE INTEGER	Attribute type is integer.
MIN 0	Minimum allowed value is 0.
MAX 2000	Maximum allowed value is 2000.
INHERITANCE NET NETCLASS	Attribute can be defined for nets. If no value is specified for a net and the net is a member of a net class, a value for the net class is assumed.
ECO_REGISTRATION Y	Register attribute changes in the ECO file.
READONLY N	Attributes can be modified.
SYSTEM Y	Attribute is used by PowerPCB.
HIDDEN N	Attribute is visible in dialog boxes.
}	End of attribute Nail Count Per Net declaration.
ATTRIBUTE Some Ratio	Sample for type float.
{	
TYPE FLOAT	Type is float or decimal number.
MIN 0	
MAX 1	
INHERITANCE VIA	Can be defined for vias or component pins.
INHERITANCE PIN	Can be defined for vias or component pins.
ECO_REGISTRATION N	Do not register attribute changes in the ECO file.
READONLY N	Attributes can be modified.
SYSTEM N	Attribute isn't used by PowerPCB.
HIDDEN N	Attribute is visible in dialog boxes.
}	
ATTRIBUTE Manufacturer	Declaring attribute named Manufacturer.
{	
TYPE LIST N	Attribute type is list. Values are strings selected from the set defined below. N means values are not case sensitive.
{	Possible values for the attribute:
	Intel
	Motorola

Texas Instruments

```
}  
  
INHERITANCE PART PARTTYPE Attribute can be defined for components. If no value  
is specified for a component, a value for the  
component part type is assumed.  
  
ECO_REGISTRATION Y Register attribute changes in the ECO file.  
READONLY N Attributes can be modified.  
SYSTEM N Attribute isn't used by PowerPCB.  
HIDDEN N Attribute is visible in dialog boxes.  
}  
End of attribute Manufacturer declaration.  
ATTRIBUTE User Note  
{  
TYPE FREETEXT N Free text attribute. N means this values are not case  
sensitive.  
  
INHERITANCE NET NETCLASS  
INHERITANCE PART DECAL  
PACKAGE  
ECO_REGISTRATION N Do not register attribute changes in the ECO file.  
READONLY N Attributes can be modified.  
SYSTEM N Attribute isn't used by PowerPCB.  
HIDDEN N Attribute is visible in dialog boxes.  
}  
}
```

Attributes values format

Attribute values is optional. This part assigns attribute values to a database object.

```
ATTRIBUTE VALUES This line is the top-level section header. An opening  
bracket should follow the section header. In this  
section, there are subsections for all objects with  
attributes in the database. The subsection header  
includes an object type and object name or other data  
to identify the object. The following object types are  
supported:  
  
PCB DEFAULT Design as a whole. Attributes in this subsection may  
be used as default attributes.  
  
PARTTYPE NAME PARTTYPE is a keyword.
```

	NAME is a part type (package) name.
DECAL NAME	DECAL is a keyword. NAME is a decal (symbol) name.
PART REFDES	PART is a keyword. REFDES is a part (component) reference designator (or name).
NETCLASS NAME	NETCLASS is a keyword. NAME is a net class name.
NET NAME	NET is a keyword. NAME is a net (signal) name.
PIN REFDES.NUMBER	PIN is a keyword. REFDES is a part (component) reference designator (or name). NUMBER is alphanumeric pin number. Note: REFDES.PIN together form a pin designation used elsewhere in the ASCII file to specify a component pin in the design. Attributes are assigned to a specific pin instance, not to a decal terminal or a package pin.
VIA VIATYPE NETNAME X Y	VIA is a keyword. VIATYPE is via type name. NETNAME is a net (signal) name for a net that includes the via. X and Y are the via coordinates in current units as specified in ASCII file header. Note: Attributes are assigned to the specific via instance, not to a via type. Jumper pins are processed as vias.
JUMPER REFDES	JUMPER is a keyword. REFDES is jumper reference designator (name). Note: Jumper attributes are internal and are not accessible through the user interface. Currently jumper attributes are only used to hold assembly options data.
After the subsection header and the opening bracket, one or more lines describing the attributes follow:	
NAME1.NAME2...NAMEN VALUE	
NAME1, NAME2	NAMEN are the names of the attributes group or attribute. If name includes spaces, it should be enclosed in double quotes. Together NAME1.NAME2...NAMEN form a structured attribute name.
VALUE	Attribute value. Value should not be enclosed in quotation marks even if it includes spaces.

Example

```
ATTRIBUTE VALUES
{
PCB DEFAULT
{
PowerGround N
}
NETCLASS PWRGND
{
PowerGround Y
}
NET SIGNAL_NET
{
"DFT.Nail Count Per Net" 1
"DFT.Nail Diameter" 40
PowerGround N
}
NET VCC
{
PowerGround Y
Voltage 3.3 V
}
PART R1
{
Geometry.Height 100 mil
}
DECAL DIP14
{
Geometry.Height 200 mil
}
PARTTYPE 7400
{
"Manufacturer #1" Texas Instruments
}
VIA TESTVIA NET1 8750 7700
{
DFT."Nail Number" ICT345
DFT."Nail Diameter" 25
}
PIN U1.7
{
"DFT.Nail Number" ICT346
"DFT.Nail Diameter" 50
}
}
```

Layer description format

LAYER DATA

Layer section header for basic units. Use MIL., INCHES, or METRIC instead of DATA to set other units for the Layer section. It is recommended to use the same units as in the ASCII file header.

{	
LAYER 1	Layer number. Valid values range from 1 to 250.
{	
LAYER_NAME TOP	Layer name; can include spaces.
LAYER_TYPE ROUTING	Layer type, can be: UNASSIGNED ROUTING ASSEMBLY SOLDER_MASK PASTE_MASK SILK_SCREEN
PLANE NONE	Plane status of layer can be: NONE CAM MIXED
ROUTING_DIRECTION HORIZONTAL	Preferable routing direction, can be: HORIZONTAL VERTICAL 45 -45 NO_PREFERENCE

The following four parameters are optional:

ASSOCIATED_SILK_SCREEN SilkscreenTop	Associated silk screen layer. This parameter is valid only for the COMPONENT layer type. The associated layer should be of SILK_SCREEN type.
ASSOCIATED_PASTE_MASK Paste Mask Top	Associated paste mask layer. This parameter is valid only for the COMPONENT layer type. The associated layer should be of PASTE_MASK type.
ASSOCIATED_SOLDER_MASK Solder Mask Top	Associated solder mask layer. This parameter is valid only for the COMPONENT layer type. The associated layer should be of SOLDER_MASK type.
ASSOCIATED_ASSEMBLY Assy. Drawing Top	Associated assembly drawing layer. This parameter is valid only for the COMPONENT layer type. The associated layer should be of ASSEMBLY type.
COMPONENT Y	Specify if layer can be used to place components.
VISIBLE Y	Layer is visible.
ENABLED Y	Layer is enabled.
SELECTABLE Y	Layer is selectable.
COLORS :	
{	
ROUTE 5	Trace color.

VIA 5	Via color.
PAD 5	Pad color.
COPPER 5	Copper color.
2DLINE 5	Line color.
TEXT 5	Text color.
ERROR 5	DRC error color.
TOPCOMPONENT 5	Top component color.
BOTTOMCOMPONENT 0	Bottom component color.
REFDES 13	Reference designator color.
PARTTYPE 0	Part type color.
ATTRIBUTE 0	Label color.
KEEPOUT 15	Keepout color
}	
PREPREG Y	Prepreg/Substrate status.
LAYER_THICKNESS 304800	Dielectric thickness.
COPPER_THICKNESS 38100	Copper thickness.
DIELECTRIC 3.800000	Relative dielectric constant.
COST 0	
}	
LAYER 2	
{	
LAYER_NAME Ground	
LAYER_TYPE ROUTE	
PLANE CAM	
ROUTING_DIRECTION	
VERTICAL	
LAYER_THICKNESS 228600	
COPPER_THICKNESS 38100	
DIELECTRIC 3.800000	
NET GND	One or more nets can be assigned to CAM or MIXED layer.
}	
}	

Visibility description format

```
VISIBILITY DATA
{
You can control visibility of various kinds of objects in PCB:
PADS Y
TRACKS Y
VIAS Y
COPPER Y
LINES Y
TEXT Y
ERRORS Y
CLUSTER Y Cluster view mode required at least one cluster.
TOP_COMPONENT Y
BOTTOM_COMPONENT Y
SELECTION_COLOR 15 You can also set some additional colors.
Valid values range from 0 to 15.
HIGHLIGHT_COLOR 14
FIXED_COLOR 3
PLANE_THERMAL_COLOR 15
}
```

Note: Not specifying any binary (Y/N) parameters is equivalent to N (no) value.

Router strategy attributes

New System, Hidden, and Read-Only attributes describe auto router strategy. Attribute names have the following format:

Strategy.PASS.PARAMETER

PASS Name of the auto router pass. Available passes are:

- SplitPairs
- Fanout
- Patterns
- Route
- Optimize
- Miters
- TestPoint

PARAMETER Pass parameter name. Available parameters are:

Pass is PCB (default) attribute of type list. Possible values are:
Done. Pass is already executed by the auto router.
Yes. The pass is selected for execution.
No. Skip the pass.

Protect is PCB (default) attribute of type Boolean (Yes/No).

Pause is PCB (default) attribute of type Boolean (Yes/No).

Priority is an attribute of type Integer (Number). Could be assigned to nets, net classes, parts (components), or PCB (default). Defines routing order for the object.

Intensity is PCB (default) attribute of type list. Possible values are:

Low
Medium
High

PlanePriority is PCB (default) attribute of type integer (number). Defines routing order for plane nets (nets assigned to plane layers).

DFT attributes

New System, Hidden, and Read-Only attributes describe DFT parameters. All attributes are defined on PCB (default) level only. Attribute type is Boolean (Yes/No) for flags and Measure (quantity name Size/Dimension) for all other DFT attributes.

DFT.Generate Test Points is the flag that controls test point generation. Valid values are Yes and No.

DFT.Probe to Trace Clearance is the minimal probe-to-trace clearance. Valid values range from 0 to 1000.

DFT.Probe to Pad Clearance is the minimal probe-to-pad clearance. Valid values range from 0 to 1000.

DFT.Allow Stabs is the flag that controls stab creation. Valid values are Yes and No.

DFT.Stub Length is the maximum stab length. Valid values range from 0 to 2000.

DFT.Use Via Grid is the flag to use via grid for test point generation. Valid values are Yes and No.

DFT.Grid X-coordinate is X (horizontal) grid size. Valid values range from 0 to 2000.

DFT.Grid Y-coordinate is Y (vertical) grid size. Valid values range from 0 to 2000.

Cluster Definition Format

The Cluster section defines areas (clusters) for the PowerPCB cluster placement algorithm.

The section heading is:

CLUSTER.

Each cluster has a name, location, cluster ID, parent, child number, attributes, and brother ID defined in the following format :

NAME_XLOC_YLOC_PARENT_CLUSTERID_CHILDNUM_ATTRIBUTE_ATT2_BROID

Example

```
CLU11 1300 3675 0 11 4 24 0 6
```

CAM Section

This section contains the Computer Automated Manufacturing (CAM) information for PowerPCB. For additional information about the various options and controls refer to the "CAM Outputs" section of the *PowerPCB Help*.

CAM_SECTION PARENT	Begin the CAM section.
{	
CAM_VERSION <i>version</i>	Version of CAM. Current version is 3.0.
CAM_DOC_LIST PARENT	Begin the CAM document list.
{	
. . .	
}	End the CAM document list.
CAM_DRILL_SPEED_FEED_TABLE PARENT	
{	Begin the drill speed feed table.
. . .	
}	End the drill speed feed table.
CAM_AUGMENT_ON_THE_FLY [YES NO]	Add new apertures to the aperture table as they are encountered.
CAM_APERTURE_TABLE PARENT	Begin the aperture table.
{	
. . .	
}	End the aperture table.
CAM_DRILL_SYMBOL_TABLE_PARENT	Begin the drill symbol table.
{	
. . .	
}	End the drill symbol table.
}	End of the CAM section.

CAM document list

This section contains the list of CAM documents defined for this board. Each CAM document describes a plot type (drill, silk-screen, routing), the associated output device (laser, photoplotter), and plot options (orientation, scaling, mirroring). Additionally the colors of the various items to plot are listed.

CAM_DOC_LIST PARENT	Begin the CAM document list.
{	
CAM_DOC_DIRECTORY <i>directory_name</i>	Subdirectory (under PADS\CAM) where CAM documents are stored. Defaults to

CAM_DOC_NAME <i>name1</i>	PADS\CAM\default. Any descriptive character string for the following CAM document.
{	
DOC_PLOT_TYPE	Available plot types: Custom Plane Routing Paste_Mask Solder_Mask Assembly Drill_Drawing NC_Drill Verify_Photo
DOC_DEVICE_TYPE	Output device for plot: Printer Pen Photo Drill
DOC_OUTPUT_FILE <i>file_name</i>	File name in CAM_DOC_DIRECTORY. Any legal file name for your system.
DOC_LAYER_NAMES layer1 ... layer30	List of up to thirty layer names that make up this CAM document. Separate by spaces.
DOC_LAYER_NUMBERS <i>n1 ... n30</i>	List of up to thirty layer numbers corresponding to DOC_LAYER_NAMES. Separate by spaces.
DOC_LAYER_TYPES <i>type1 ... type30</i>	List of up to thirty layer types corresponding to DOC_LAYER_NAMES. Separate by spaces. Legal layer types are: Unassigned Component Routing Plane Drill SilkScreen PasteMask SolderMask Assembly General
DOC_FABRICATION_LAYER_NUMBER <i>n</i>	Number of Fabrication layer if such a layer is associated with this document, or 0 otherwise
DOC_SCALE multiplier divisor	Scaling factors for the CAM output plots. Both integers.
DOC_ORIENTATION [0 90 180 270]	Orientation of CAM output plots.

DOC_MIRROR [Y N]		Mirroring of CAM output plots.
DOC_JUSTIFICATION	[OO CM LB LT RB RT]	Justification of CAM output plots. Horizontal and vertical offset to origin. Horizontal center, vertical middle. Horizontal left, vertical bottom. Horizontal left, vertical top. Horizontal right, vertical bottom. Horizontal right, vertical top.
DOC_OFFSET <i>x y</i>		Offset of CAM output plots. Both integers in basic units.

For all documents other than N/C drill type, specify the following:

The following sections select colors for the various items in a CAM output plot. The color choices for all items are None, Black, Green, Blue, Aqua, Red, Violet, Brown, Gray, White, Lblue, Lgreen, Laqua, Lred, Lviolet, Yellow, Lwhite.

You are allowed to enter only colors supported by the DOC_DEVICE_TYPE specified above. Currently, the only color available for Printer and Photo plots is Black. Pen plots have all the mentioned colors available. Color is not defined for NC Drill documents.

DOC_CLIP <i>boolean</i>		<i>boolean</i> is 1 or 0 and indicates whether output is clipped based on PowerPCB graphics view port.
DOC_FIT_OUTPUT <i>boolean</i>		<i>boolean</i> is either 1 or 0 and indicates whether the data is scaled to fit in the output x and y dimensions.
DOC_NO_PLANE_THERMALS <i>boolean</i>		<i>boolean</i> is either 1 or 0 and determines whether thermal shapes are generated while connecting vias to CAM planes.
DOC_BOARD <i>color</i>		Board outline color.
DOC_CONNECTIONS <i>color</i>		Connection color.
DOC_SLOTS <i>color_pl color_unpl</i>		Slot colors for plated and non-plated
DOC_OUTLINES <i>color_top color_bottom</i>		Component outline colors for top and bottom
DOC_OUTLINE_LAYERS [Y N] . . .		Y to plot component outlines on layer, N to not plot component outlines on layer. One value for each of the layers in DOC_LAYER_NAMES (up to 30).

The following sections select colors for the various items in a CAM output plot , one value for each layer in DOC_LAYER_NAMES.

DOC_REFDES color1 color2 color3 ... color30	Reference designator colors.
DOC_PARTTYPES color1 color2 color3 ... color30	Part type colors.
DOC_ATTRIBUTES color1 color2 color3 ... color30	Attribute colors.
DOC_TESTPOINT color1 color2 color3 ... color30	Test point colors.
DOC_PADS color1 color2 color3 ... color30	Pad color.
DOC_VIAS color1 color2 color3 ... color30	Via color.
DOC_ROUTES color1 color2 color3 ... color30	Route color.
DOC_COPPER color1 color2 color3 ... color30	Copper color.
DOC_LINES color1 color2 color3 ... color30	Lines color.
DOC_TEXT color1 color2 color3 ... color30	Text color.
DOC_KEEPOUT color1 color2 color3 ... color30	Keepout color.
DOC_PAD_OVERSIZE <i>value</i>	Positive value to increase plotted pad size, negative value to decrease plotted pad size. In basic units.
DOC_PLOT_JOBNAME [Y N]	Adds job name, date, and time to the plot.
DOC_PLOT_OLE [Y N]	Controls output of OLE documents included in the design.
DOC_MIRROR_REFS [Y N]	Mirrors the reference designators.
DOC_SUPPRESS_PREFIXES <i>string1, string2,</i>	Suppresses plotting of reference designators starting with indicated character strings. Separate each string with a comma.
DOC_PLOT_QUALITY [Filled Hollow]	Fills solids or plot outlines only.
DOC_DRILL_CHART [Y N]	Includes a drill chart, sometimes called a drill symbol table, in the plot. The drill chart is a table of drill sizes showing the symbol, plating, and number of times each drill size is used in the job.
DOC_CHART_OFFSET <i>x y</i>	Offset of drill chart.
DOC_SUPPRESS_EXCLUDE_PADS [Y N]	Y to suppress pads only for components referenced in DOC_SUPPRESS_PREFIXES.

N to suppress the whole components.
Effective for mask layers only.

For N/C drill type documents only, specify the following:

DOC_DRILL_PLATED [Y N]	Drill the plated holes on the board.
DOC_DRILL_NONPLATED [Y N]	Drill the nonplated holes on the board.
DOC_DRILL_THRU_VIA [Y N]	Drill the through vias on the board.
DOC_DRILL_BURIED_VIA [Y N]	Drill the buried vias on the board.
DOC_STEP_REPEAT [Y N]	Use step and repeat information if you have lay-ups of multiple boards on the same panel. Specify the counts below.
DOC_REPEAT_COUNTS horizontal_count vertical_count	Specify the step sizes to use for the multiple boards.
DOC_STEP_SIZES horizontal_size vertical_size	Specify the step sizes to use for the multiple boards.
}	End CAM document, up to 999 entries.
}	End CAM document list parent.

CAM drill speed feed table

This section contains the drill speed feed table defined for this board. Drill speeds and feed rates are assigned to ranges of drill sizes. Assuming mils as the design unit, the following statement assigns a feed rate of 10 mils/minute and a drill speed of 20 mils/minute to all the drill sizes from 15 mils to 25 mils.

Note: DRILL_RANGE_n 15 25 10 20.

CAM_DRILL_SPEED_FEED_TABLE	Up to 999 entries.
{	
DRILL_RANGE_1	<i>min_drill_size</i> Minimum drill size for this feed and speed in basic units.
	<i>max_drill_size</i> Maximum drill size for this feed and speed in basic units.
	<i>feed</i> Feed speed per minute in design units.
	<i>speed</i> Drill speed per minute, in design units.
. . .	
DRILL_RANGE_N	<i>min_drill_size</i> <i>max_drill_size</i> <i>feed</i> <i>speed</i> , up to 999 entries.
}	

For assembly drawing documents only, specify the following:

DOC_ASSEMBLY_OPTION <i>string</i>	Specifies the name of the assembly option used when plotting this assembly drawing
-----------------------------------	--

CAM augment on the fly

PowerPCB can create the aperture table below automatically. If you make changes to your design, PowerPCB will add to the table any new apertures needed if CAM_AUGMENT_ON_THE_FLY is set to YES. This feature is disabled if this value is NO.

CAM_AUGMENT_ON_THE_FLY [YES | NO]

Add new apertures to the aperture table as they are encountered.

CAM aperture table

This section specifies the association of D-codes with an aperture of a certain height, width, and shape. An aperture is a shape used in board manufacturing to expose a small area of the final board film. The D-code is an index to the mechanical aperture wheel or, for laser photoplotters, instructions on how to direct the laser.

```
CAM_APERTURE_TABLE PARENT
{
  APERTURE_1 dCode width height shape
                                dCode – D-code
                                width –In basic units
                                height –In basic units
                                Values for shape are:
                                RND – Round
                                SQR – Square
                                OVAL – Oval
                                RECT – Rectangular
                                ANNL – Annular
                                THER – Thermal
                                ODD – Odd
  ....
  APERTURE_999 dCode width height shape
                                Up to 999 entries
}
} End aperture table
```

CAM drill symbol table

This section contains the drill symbol table, also known as a drill chart. The drill chart is a table of drill sizes showing the symbol, plating, and number of times each drill size is used in the job. It is used by all documents of type Drill_Drawing.

CAM_DRILL_SYMBOL_TABLE PARENT

{

MARKER_SIZE height Height of the symbols below, in basic units.

MARKER_LINE width Width of the symbols below, in basic units.

MARKER_CHAR_HEIGHT height Height of any characters used for symbols, in basic units.

CHART_TEXT_HEIGHT height Height of legend in the Drill Symbol table, in basic units.

CHART_LINE_WIDTH width Width of lines and legend in the Drill Symbol table, in basic units.

SYMBOL_1 *drillSize* [P | N]*symbol* drillsize in basic units.

P – Plated

N – Nonplated

Symbols are:

+

X

Rectangle

Diamond

Hour_Glass

Bow_Tie

+A

+B

. . .

+Y

+Z

. . .

SYMBOL_32 *drillSize* [P | N]
symbol Up to 32 entries.

}

End of CAM section

Assembly Options Section

Assembly options data are stored as values for ASSEMBLY_OPTIONS attribute for PCB (default), Part (component), and Jumper. The ASSEMBLY_OPTIONS attribute values have the following structure:

ASSEMBLY_OPTION_NAME,OPTION_STATUS[,SUB_PART_TYPE][;...]

ASSEMBLY_OPTION_NAME Name of the assembly variance (option)

OPTION_STATUS

Part or jumper option status:

I – Part installed for the variance

U – Part uninstalled for the variance

S – Part substituted

SUB_PART_TYPE

Substituted part type. It is specified only if
OPTION_STATUS is S.

Example

```
PART U1
```

```
{
```

```
ASSEMBLY_OPTIONS PACKED_1,S,AM26SL30-423; CUT_1,U
```

```
}
```

In this example part (component) U1 is substituted for AM26SL30-423 in assembly option PACKED_1 and uninstalled in assembly option CUT_1.

Test Point Definition Section

The Test Point section is a new section that defines all test points and their attributes in the design.

The section heading is:

TESTPOINT.

Depending on the test point carrier (via, component pin) each test point may be in one of two allowed formats, either test point on vias or test point on components:

VIA_XLOC_YLOC_SIDE_SIGNAME_VIANAME

PIN_XLOC_YLOC_SIDE_SIGNAME_REFDES.PIN

VIA and PIN	Keywords defining the type of test point carrier
XLOC_YLOC	Coordinates of the test point relative to the system origin
SIDE	Board side on which the test point should be probed. Valid values: 0 – Bottom 1 – Top
SIGNAME	Name of the signal to which the test point belongs. A signal may have up to 47 characters. If a component pin is unused, then SIGNAME should be .NONE. (this name is forbidden for general nets).
VIANAME	Name of the via pad stack used at this coordinate. The pad stack description for these vias should be found in the Via section. (for example, BURIEDVIA1-2).
REFDES.PIN	Reference designator and pin number. Reference designators may be up to six alphanumeric characters long. Pins may be numeric, alphanumeric, or alphabetic, and may be up to four characters long.

Example

```
*TESTPOINT*
VIA  -3000 1000 0      GND   STANDARDVIA
VIA  -200  2000 1      $1234 MICROVIA
PIN  -300  390  0      +5     U1.5
PIN  -100  100  1      .NONE. U1.8
```

Index

2

2D items, 26

A

alphanumeric pin format, 38

arc format, 52

assembly options, 79

attributes, 61

attributes dictionary format, 61

attributes values format, 64

B

board drilling rules, 53

C

CAM, 72

CAM aperture table, 77

CAM augment on the fly, 77

CAM document list, 72

CAM drill speed feed table, 76

CAM drill symbol table, 78

clearance rules format, 3

cluster definition format, 71

connection definition format, 42

connection list control statement, 42

connection list definition, 42

connection list entry format, 42

connection list header format, 42

control statements, 8

copper pour definition format, 50

D

design rules, 56

DFT attributes, 70

differential pairs format, 61

E

end-of-file control statement, 9

F

fabrication parameters, 2

file structure, 7

format changes, 2

format conventions, 6

G

gate format, 37

general parameter control
statement, 10

general parameters, 10

H

hatch outline, 50

header line, 2, 9, 18, 19, 23, 27, 35,
36, 37, 40, 42, 45

header line format, 27

hierarchical rules section, 53

J

jumper, 47, 48

jumper definition format, 35

jumper header format, 35

L

label data entry format, 29

layer description format, 66

lines control statement, 23

lines definition format, 23

lines entry format, 23

lines header definition, 23

N

net classes, 54
netlist control statement, 43
netlist definition format, 43
netlist format, 44

P

pad and via thermals, 51
pad definition format, 30
pad stack definition format, 30
part decal control statement, 27
part decal definition format, 27
part decal format structure, 27
part list control statement, 39
part list definition format, 39
part list entry format, 39, 41
part type control statement, 36
part type definition format, 36
part type entry format, 36
part type header format, 36
piece definition, 24, 51
piece definition format, 28
pin pair groups, 55
pin pair line format, 43, 45
pour control statement, 50
pour entry format, 50
pour header definition, 50
pour outline, 50

R

reuse, 47

reuse control statement, 18
reuse definitions, 18
reuse entry format, 18
reuse instance header format, 20
reuse net format, 19
reuse part format, 18
reuse type header format, 18
route coordinates line, 46
route definition format, 45
route entry format, 45
route header format, 45
route section control statement, 45
router strategy attributes, 69
routing rules format, 3
rule set format, 57

S

signal format, 38
system attributes, 3

T

teardrop, 47, 48
test point definition, 80
text control statement, 22
text definition format, 22
text entry format, 3, 22

V

via definition format, 33
via entry format, 33
visibility description format, 69
void outline, 50

