PADS Layout ASCII Format Specification

PADS 9.4

Contractor/manufacturer is:
Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/
Send Feedback on Documentation: supportnet.mentor.com/doc_feedback_form

Table of Contents

Chapter 1
ASCII Format Specification

Introduction

This specification defines the structure and syntax of the PADS ASCII data format for PADS Layout 9.4. See the Version 9.4 Format Changes topic for information on format changes from version 2007.

The PADS ASCII file format provides a way for other CAD or CAE systems to interface with PADS 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 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 design information 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.

ASCII Version 9.4 Format Changes

This section identifies changes in the structure and syntax of the ASCII data format for ASCII version 9.4. Refer to the specific sections for a complete description of changed or added information.

Header line

Version number V9.4 is used in the header line.

Example

```
!PADS-POWERPCB-V9.4-BASIC! DESIGN DATABASE ASCII FILE 1.0
```

General System Parameters Section

New keywords have been added to the *PCB* section to store information about Object Snap settings used in the design:

Format	Description
OSNAP *flags*	*flags* is set of flags used in Object Snap options. Snap flags definitions: 0x00000001 – Snap to Objects switch on/off 0x00000002 – Show Snap to Objects Marker 0x00000004 – Snap to midpoint 0x00000008 – Snap to intersection 0x00000010 – Snap to center 0x00000020 – Snap to corner 0x00000040 – Snap to quadrant 0x00000080 – Snap to component origin 0x00000100 – Snap to pin/via origin
OSNAPRAD *value*	value is user-defined snap radius

New Associated Nets dialog parameters have been added to the *PCB* section:

Statement	Description
ASSOCIATEDNETNETCOUNT	*value* is the maximum number of nets in associated nets
ASSOCIATEDNETPLANEPINCOUNT	value is the maximum pin count in nets, which can belong to associated nets
ASSOCIATEDNETPREFIX0	value is the sting of capacitor discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX1	value is the sting of connector discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX2	value is the sting of diode discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX3	value is the sting of inductor discrete component prefixes for associated nets

ASSOCIATEDNETPREFIX4	value is the sting of resistor discrete component prefixes for associated nets

A new Associated Nets subsection (header ASSOCIATED_NET DATA) has been added to the Hierarchical Rules Section.

A new Diff Pair format has been added for diff pairs of associated nets.

High Speed rules (minimum/maximum length and matched length groups) can be assigned to associated nets.

Lines Definition Format

Piece definition in Line entry has been extended to handle 2D Line styles.

Format Conventions

The format of ASCII documents is important for correct interpretation by Mentor Graphics products. 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: ‘~@#$%^&()_-+=:;”’[]?/<>!
- Lowercase italic text represents variables. Definitions for variables and allowed ranges of values are given.
- 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 are 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 (*):

Statement	Description
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*][-*encoding*]!

Format	**Description**
product	POWERPCB
version	V9.2

units	MILS Defined database units are in mils. Data are expressed to the nearest hundredth mil. Valid values range from –56000 to +56000. INCHES Defined database units are in inches. Data are 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 are 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 x 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.
encoding	CP*n*, where *n* is the code page number: 1252 – Western (Latin I) (default), 932 – Japanese Other code pages (not supported yet or supported partly): 936 – Simplified Chinese 1250 – Central Europe 1251 – Cyrillic 1253 – Greek

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 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

The 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. Parameters may be left out of an input file.

General Parameter Control Statement

PCB

Format for General Parameter Entries

Parameter value

Format	Description
parameter	Legal parameter name as listed in General Parameters.
value	Numeric value in the allowable range for the parameter.

General Parameters

Format	Description
UNITS value	value is Unit of measure. Valid values are: 0 - Mils 1 - Metric 2 - Inches
USERGRID value	value is a 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 value	value is the maximum number of routing layers. Valid values range from 1 to 64.
WORKLEVEL value	value is the current layer on which data are entered. Valid values range from 0 (all layers) to 250 (maximum layer).
DISPLAYLEVEL value	value - 1 means display current layer on top. value - 0 means display "Top" layer on top.
LAYERPAIR value1 value2	Two values are required, defining the layer pair for swapping during routing. value1, value2 are in the range of 1 to 64. value2 should be greater than value1.
VIAMODE value	value 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 value	value 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 value	value is the total time in minutes recorded for the design in system memory.
DOTGRID value	value is the space between graphic dots. Valid values range from 1 to 1000.
SCALE value	value is the scale of window expansion. Valid values range from 1.0 to 7400.000++.

<table>
<tr><td>ORIGIN x y</td><td colspan="4">x and y are coordinates of the current design origin relative to the system origin. Valid values range from -28000 to 28000.</td></tr>
<tr><td>WINDOWCENTER x y</td><td colspan="4">x and y are coordinates defining the center of the window.</td></tr>
<tr><td>BACKUPTIME value</td><td colspan="4">value is the number of minutes between automatic backup of a database to a file. Valid values range from 1 to 30.</td></tr>
<tr><td>REAL WIDTH value</td><td colspan="4">value is the minimum track size to begin to display real track size. Tracks narrower than value appear with a center line only. Valid values range from 1 to 250.</td></tr>
<tr><td rowspan="8">ALLSIGONOFF value</td><td colspan="4">value is the value of the default signal view nets.</td></tr>
<tr><td>Net Display Type</td><td>Bit 1</td><td>Bit 9</td><td>Bit 18</td></tr>
<tr><td>No traces or unrouted</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Traces and all unrouted</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>Traces and unrouted, except connected by plane</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Traces and unrouted, except unrouted pin pairs</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Traces and no unrouted</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="4">All other bits are ignored.</td></tr>
<tr><td>REFNAMESIZE height width</td><td colspan="4">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.</td></tr>
<tr><td>HIGHLIGHT value</td><td colspan="4">Highlight the current net parameter</td></tr>
<tr><td>JOBNAME value</td><td colspan="4">The user-defined job name. Up to 20 characters (prefix), 3 characters (suffix).</td></tr>
<tr><td>CONCOL value</td><td colspan="4">value is the color code for connections in the design. Valid values range from 0 to 31.</td></tr>
<tr><td>FBGCOL value1 value2</td><td colspan="4">value1 is the color code for the foreground color in the design. It ranges from 0 to 31. value2 is the color code for the background color in the design. It ranges from 0 to 31.</td></tr>
<tr><td>HATCHGRID value</td><td colspan="4">value is the value of the default copper pour hatching grid. Valid values range from 0.01 to 250.</td></tr>
</table>

TEARDROP value	value is the value of teardrop: 0 - Off 1 - On
THERLINEWID value	value is the default copper pour thermal line width for through hole pad stacks.
PADFILLWD value	value is the CAM finger pad fill line width. Valid values range from 1 to 250. The default is 10.
THERSMDWID value	value is the copper pour thermal line width for SMD pad stacks.
MINHATAREA value	value is the minimum area of copper that will be hatched. Valid values range from 0 to 9999999. The default is 0.
HATCHMODE value	value is the hatch generation mode for copper flood: 0 - Normal. The default. 1 - No hatch 2 - See through
HATCHDISP value	value is the hatch display flag for copper flood 0 - Pour Outlines. The default. 1 - Hatch Outlines
MITRETYPE value	value is the miter type: 2 - Line 1 - Arc. The default.
HATCHRAD value	value is the ratio for creating smooth hatch outlines. Valid values range from 0 to 9999999. The default is 0.5.
HATCHANG value	value is the hatch angle: 0 - Horizontal and vertical hatch. The default. 45 - Diagonal hatch
VIAPSPACING value	value is the via patterns via spacing
VIAPSHAPE value	value is the via patterns via to shape spacing
VIAPTOTRACE value	value is the via patterns via to edge spacing
VIAPFILL value	value is the via patterns fill type 1 - Fill aligned 2 - Fill staggered 16 - Perimeter
VIAPSHSIG name	name is the via patterns shielding via type
VIAPSHVIA name	name is the via patterns shielding net

VIAPFLAG flags	flags is via patterns flags. Via patterns flags definitions: 0x001 - Glue vias 0x002 - Fill selected hatch outlines 0x004 - Ignore via grid 0x008 - Via to ground edge spacing
THERFLAGS flags	flags 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 value	value 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.

PLANEFLAGS *save_mode display_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*

Format	**Description**
save_mode	Data which will be saved in the .pcb file ALL or OUTLINE.
display_mode	The 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

PLANEFLAGS ALL ALL N N Y Y Y Y Y Y Y Y Y Y

Format	**Description**

COMPHEIGHT *value*	*value* is the board top component height restriction
KPTHATCHGRID *value*	*value* is the copper pour hatching grid
BOTCMPHEIGHT 0	The 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 *value*	*value* is the value of the fanout grid. Valid values range from 0.01 to 1999.99.
FANOUTLENGTH *value*	*value* is the maximum fanout length. Valid values range from 0.01 to 1999.99

<table>
<tr><td>ROUTERFLAGS flags</td><td>flags defines the fanout parameters and pad entry quality. Values are stored in bit fields as follows:

<table><tr><th>Bits</th><th>Values</th></tr><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>20 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></table></td></tr>
</table>

<table>
<tr><td>VERIFYFLAGS flags</td><td>flags defines the verify design parameters. Values are stored in bit fields as follows:<table>
<tr><th>Bits</th><th>Values</th></tr>
<tr><td>0</td><td>Clearance Checking Setup: Net to All</td></tr>
<tr><td>1</td><td>Clearance Checking Setup: Same Net</td></tr>
<tr><td>2</td><td>Clearance Checking Setup: Drill to Drill</td></tr>
<tr><td>3</td><td>Clearance Checking Setup: Trace Width</td></tr>
<tr><td>4</td><td>Clearance Checking Setup: Body to Body</td></tr>
<tr><td>5</td><td>Clearance Checking Setup: Nudge Outline</td></tr>
<tr><td>6</td><td>Mixed Plane Setup: Full Check</td></tr>
<tr><td>7</td><td>Reserved</td></tr>
<tr><td>8</td><td>Mixed Plane Setup: Same Layer Connectivity</td></tr>
<tr><td>9</td><td>Clearance Checking Setup: Board Outline</td></tr>
<tr><td>10</td><td>Clearance Checking Setup: Keepout</td></tr>
<tr><td>11</td><td>Clearance Checking Setup: Off Board Text</td></tr>
<tr><td>12</td><td>Clearance Checking Setup: Via at SMD</td></tr>
<tr><td>13</td><td>Clearance Checking Setup: Differential Pairs</td></tr>
<tr><td>14</td><td>Clearance Checking Setup: Trace Length</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>FABCHKFLAGS flags</td><td>The fabrication checks flags. Values are stored in bit fields as follows:<table><tr><th>Bits</th><th>Values</th></tr><tr><td>0</td><td>Run Fabrication Check</td></tr><tr><td>1</td><td>Check Acid Traps</td></tr><tr><td>2</td><td>Check Slivers</td></tr><tr><td>3</td><td>Check Starved Thermals</td></tr><tr><td>4</td><td>Check Trace Width and Pad Size</td></tr><tr><td>5</td><td>Check Silkscreen over Pads</td></tr><tr><td>6</td><td>Check Solder Bridges</td></tr><tr><td>7</td><td>Reserved</td></tr><tr><td>8</td><td>Check Annular Ring</td></tr><tr><td>9</td><td>Check Pad to Mask</td></tr><tr><td>10</td><td>Check Drill to Mask</td></tr><tr><td>11</td><td>Check Drill to Pad</td></tr></table></td></tr>
<tr><td>ATMAXSIZE value</td><td>value is the maximum size of the acid traps</td></tr>
<tr><td>ATMAXANGLE value</td><td>value is the maximum angle of the acid traps</td></tr>
<tr><td>SLMINCOPPER value</td><td>value is the slivers minimum copper</td></tr>
<tr><td>SLMINMASK value</td><td>value is the slivers minimum mask</td></tr>
<tr><td>STMINCLEAR value</td><td>value is the starved thermal minimum clearance</td></tr>
<tr><td>STMINSPOKES value</td><td>value is the starved thermal minimum spokes</td></tr>
<tr><td>TPMINWIDTH value</td><td>value is the minimum trace width</td></tr>
<tr><td>TPMINSIZE value</td><td>value is the minimum pad size</td></tr>
<tr><td>SSMINGAP value</td><td>value is the silk screen over pads minimum gap</td></tr>
<tr><td>SBMINGAP value</td><td>value is the solder bridges minimum gap</td></tr>
<tr><td>SBLAYER value</td><td>value is the solder bridges layer</td></tr>
<tr><td>ARPTOM value</td><td>value is the pad-to-mask annular ring</td></tr>
<tr><td>ARPTOMLAYER value</td><td>value is the pad-to-mask annular ring layer</td></tr>
<tr><td>ARDTOM value</td><td>value is the drill-to-mask annular ring</td></tr>
<tr><td>ARDTOMLAYER value</td><td>value is the drill-to-mask annular ring layer</td></tr>
<tr><td>ARDTOP value</td><td>value is the drill-to-pad annular ring</td></tr>
</table>

ARDTOPLAYER *value*	*value* is the drill-to-pad annular ring layer
OSNAP *flags*	*flags* is set of flags used in Object Snap options. Snap flags definitions: 0x00000001 – Snap to Objects switch on/off 0x00000002 – Show Snap to Objects Marker 0x00000004 – Snap to midpoint 0x00000008 – Snap to intersection 0x00000010 – Snap to center 0x00000020 – Snap to corner 0x00000040 – Snap to quadrant 0x00000080 – Snap to component origin 0x00000100 – Snap to pin/via origin
ASSOCIATEDNETNETCOUNT	value is the maximum number of nets in associated nets
ASSOCIATEDNETPLANEPINCOUNT	value is the maximum pin count in nets, which can belong to associated nets
ASSOCIATEDNETPREFIX0	value is the string of capacitor discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX1	value is the string of connector discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX2	value is the string of diode discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX3	value is the string of inductor discrete component prefixes for associated nets
ASSOCIATEDNETPREFIX4	value is the string of resistor discrete component prefixes for associated nets

TEARDROPDATA *width length flags*

Format	**Description**
width	Relative width of a teardrop, in percent. Valid values range from 1 to 100.
length	Relative length of a 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.

Control Statement

The control statement for the reuse is:

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

Type Header Format

The Reuse Type header has two lines:

TYPE *t*

TIMESTAMP *seconds*

Format	**Description**
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.

Part Format

Header line:

PART_NAMING *partnm*

Format	Description
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*

Format	Description
PART	Keyword
name	Original part name in the reuse definition

Net Format

Header line:

NET_NAMING *netnm*

Format	Description
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 the naming style to the original reuse netnames.

This section contains many single line entries:

NET *merge name*

Format	Description
NET	Keyword
merge	Merge flag for the reuse net. 1 to merge nets 0 to rename nets
name	Original netname from the reuse definition.

Instance Header Format

REUSE *instance partnm netnm x y ori glued*

Format	Description
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 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.

Control Statement

The text control statement is:

TEXT

Entry Format

Each text entry consists of three lines as follows:

x y ori level height width M *hjust vjust* [*ndim*] [.REUSE. *instance*]
fontstyle[:*fontheight*:*fontdescent*] *fontface*
textstring

Format	Description
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	The reuse instance name
fontstyle	The font style used for the text. Supported styles are Regular or any combination of Bold, Italic, and Underline style parameters, separated by commas.
fontheight	Font height. This parameter is used for text box calculation in cases in which the font is skipped in the design. It may be omitted.
fontdescent	Font descent. This parameter is used for text box calculation in cases in which the font is skipped in the design. It may be omitted.
fontface	The font face used for the text
textstring	Text string, up to 128 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 flags [text [sigstr]]

Format	Description
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, with 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.
flags	Line flags: 1 – display solid filled copper
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*]

Format	Description
.REUSE.	Keyword preceded and followed by dots.
name	Reuse instance name
rsignal	Optional reuse signal name, or original signal name from the reuse definition. This parameter is specified only for the 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 widthhght linestyle level [*restrictions*]

Format	**Description**		
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	COPOPN	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.
linestyle	Line style for 2D Lines: 0 - solid 1 - dashed 2 - dotted 3 - dash dotted 4 - dash double-dotted For pieces other than 2D Line this value is 0
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 A - Accordion 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

Format	Description
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 that reuse data should not be specified for text inside a 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*

Format	Description
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.

Control Statement

The part decal control statement is:

PARTDECAL

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*]]

Format	Description
name	User-defined decal name. The 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 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, copper cutout, or tag . Copper can be used as a special component pad or shape for shielding. Copper may have voids within it known as cutouts.

The tag piece is used to combine coppers and copper cutouts inside the part decal into one item. It does not contain any coordinates and is used as either opening or close bracket.

Each PIECE contained within the decal, has an entry as follows:

type numcoord widthhght level [*pinnum*/*restrictions*]

Format	Description
type	Line item type: OPEN CLOSED CIRCLE COPCLS COPOPN COPCUT COPCCO COPCIR KPTCLS KPTCIR TAG
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, COPOPN, or COPCIR, the level is optionally defined with the code -1 Opposite side. If the defined piece type is TAG, the level is used as a flag of opening/close tag and is equal to 1/0 correspondingly
pinnum	Used when the type is COPCLS, COPOPN, COPCIR, or TAG. 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 A - Accordion

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 three lines:

visible lx ly lori llevel lheight lwidth mirrored hjust vjust [*right_reading*]

fontstyle[:*fontheight*:*fontdescent*] *fontface*

attrname

Format	Description
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 the decimal point.
llevel	Layer on which the label is located. Valid values range from 0 to 250. 0 means all layers.
lheight	Height of label text.

lwidth	Pin width for label text.
mirrored	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)
hjust	Horizontal justification. Valid values are: LEFT CENTER RIGHT

Format	**Description**
vjust	Vertical justification. Valid values are: UP CENTER DOWN
right_reading	Right reading status. Valid values are: ORTHO – Orthogonal ANGLED NONE. The default value.
fontstyle	The font style used for the text. Supported styles are Regular or any combination of Bold, Italic, and Underline style parameters, separated by commas.
fontheight	Font height. This parameter is used for text box calculation in cases in which the font is skipped in the design. It may be omitted.
fontdescent	Font descent. This parameter is used for text box calculation in cases in which the font is skipped in the design. It may be omitted.
fontface	The font face used for the text

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

Example

```
VALUE 100 100 90.000 1 100 10 N LEFT DOWN ORTHO
Italic,Underline Courier New
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 appears on layer 1. The label text height is 100 mils, and the pen width is 10 mils, not mirrored.

Justification is left down, and right reading is in orthogonal mode. It is drawn by Courier New font using Italic and Underline styles.

Pin Definition Format

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

x y nmx nmy pinnumber

Format	Description
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.
pinnumber	Alphanumeric number of the pin up to 7 alphanumeric characters

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:

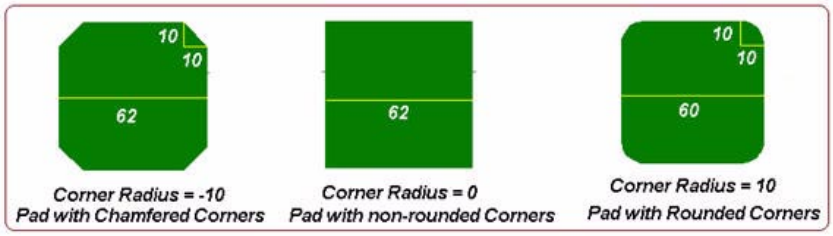
PAD *pinno stacklines*

level size shape idia finori finlength finoffset [*corner*] [*drill* [*plated*]] [*slotori slotlength slotoffset*]

level size shape idia [*corner*] [*drill* [*plated*]]

Format	Description
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.

Format	**Description**
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.
corner	This field stores the numerical “corner radius” value and is used to support pads with rounded and chamfered corners. It only exists for square and rectangular finger pad shapes. As shown in the figure, zero value is used for 90 degree (non-rounded) pad corners; positive value is used for pads with rounded corners; negative value is used for pads with chamfered corners. 
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.

Format	**Description**
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 the PowerPCB version 2.0 format to describe thermal pads and antipads.

Thermal pads have a shape code of RT or ST:

level sizeshape spokeori outsize spokewidth spokenum

Format	**Description**
PAD	Keyword
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 mils. The width of each spoke is 15 mils. 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 sizeshape

Format	Description
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.

Entry Format

name drill stacklines [*drill_start*] [*drill_end*]

level size shape [*inner_diameter*]

Format	Description
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. No interior levels are defined for this pad stack

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

Thermal pads have a shape code of RT or ST.

level sizeshape spokeori outsize spokewidth spokenum

Format	Description
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 mils. The width of each spoke is 15 mils. 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 sizeshape

Format	Description

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.

Header Format

The Jumper header line consists of:

name flags minlen maxlen lenincr lcount padstack [*end_padstack*]

Format	Description
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 two lines. The lines use the same format as the first and the second lines 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 1 100 10 1 LEFT DOWN
Regular <PADS Stroke Font>
```

In this example, the jumper named STDJMP has a label positioned at 4500 3200, lower left justified, rotated 45 degrees counterclockwise, located on layer 1, with text height of 100, text pen width of 10 and mirrored. The label is drawn by stroke font having regular style. 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.

Control Statement

The part type control statement is:

PARTT[YPE]

Entry Format

Each part type entry consists of the following:

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

Header Format

ptypenm dcalnm[:*altdcal*…] *type gates signals alpinm flag* [*eco*]

Format	Description
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[:*altdcal*]	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.

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 pin mapping of 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 part 32 – Flip Chip part (for advanced packaging) 64 – Die part (for advanced packaging) 128 – Error in the part
eco	Optional. ECO registration status for the part type: Y – part add and delete operations are registered in the ECO file. N – 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 number of pins in the gate, and the second line, which describes the actual pins within the gate.

G *gateswap pins*

pinnumber.*swptyp*.*pintyp*[.*funcname*]

Format	Description
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 are up to 7 alphanumeric characters..

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, 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*

Format	**Description**
SIGPIN	Keyword
pinno	Pin number of the signal pin. Valid values are up to 7 alphanumeric characters.
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 Mapping Format

This section defines a pin name for every pin in the part. All pins of the part must have numeric pin numbers from 1 to total number of pins in the part. Pin names may be up to 7 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 mapping is 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.

Control Statement

The part list control statement is:

PART

Optional Second Control Statement (Automatic Standard Signal Netlist Generation)

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

STANDARD

Entry Format

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

refnm ptypenm x y ori glued mirrored alt [*clstid clsattr brotherid labels*]

Format	Description
refnm	Unique reference designator name. The reference designator may be up to 15 alphanumeric characters long.
ptypenm	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. The 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. The default is 0.0.

glued	System flag indicating when the symbol is glued down. Valid values are: G – Glued U – Unglued. The default.
mirrored	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. The default is 0.
clstid	ID number of the placement cluster to which this component belongs. Cluster information is found in the Cluster section.
clsattr	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*

Format	**Description**
.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 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 glued mirrored alt*

Format	Description
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 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 or PADS Layout.

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* [*viatype*]]]

Format	Description
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. The default is –2, meaning no specific color is set for this net.
viatype	Via type for stitching shapes of the net

Example

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

Pin Pair Line Format

The Pin Pair Line format is as follows:

refdes.pin refdes.pin

Format	Description
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 rsignal*] *refdes.pin* [.REUSE. *instance rsignal*]

Format	Description
.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 need not repeat the pin from a previous entry. Multiple SIGNAL entries are tied 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...

Format	Description
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. *instance 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*}

Format	Description
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 are 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 that 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 the Connection List Header in the Connection Definition Format section.

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 the Connection Definition Format section.

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]

Format	Description
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 65 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 *or* assigned to the first or last corner of a track, defines the coordinate of a test point. 0x00080 – Route protected flag 0x00100 – Plane thermal 0x00200 – Teardrop prohibit flag (Left) 0x00400 – Teardrop prohibit flag (Right) 0x01000 – Arc center flag 0x00800 – Glued via flag 0x0e000 – Miter bits: 1 - arc miter, 2 - line miter 0x10000 – Stitching via flag
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, thermal spokes for CAM planes and mixed-plane areas are not generated. 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:

Format	**Description**
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*]

Format	**Description**
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.

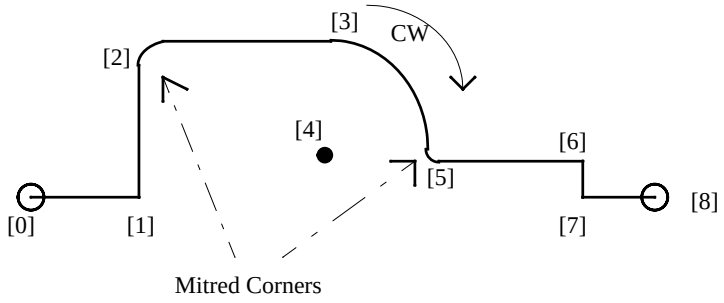
```
REUSE instance rsignal
Or
R
```

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
```

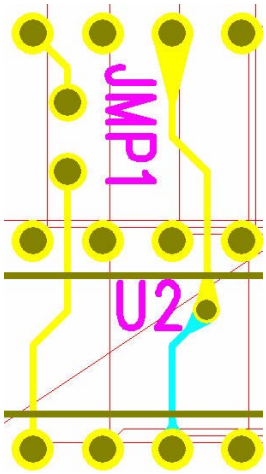
Explanation of miter description in the above example



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.

Control Statement

The copper pour control statement is:

POUR

Entry Format

A pour definition consists of the following:

- Header line
- Piece entry definition
- Coordinates

Header Definition

The header pour item definition consists of:

Hatch, pour, and void outlines

name type xloc yloc pieces flags [*ownername signame*[*hatchgrid hatchrad*[*priority*]]]

Format	Description
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 with 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.

Format	**Description**
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 in which they are physically located. Void outlines are owned by the containing hatch outline. *
signame	(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.
priority	(optional, used in pour outlines only) is the flooding priority value. Valid values range from 0 to 250
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*]

Format	Description
.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

Format	Description
int1 and *int2*	Indices. Other routines reading the ASCII file may safely ignore them. Routines writing the ASCII file should merely 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)
```

Format	Description
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 endangle

Format	Description
xloc yloc	Center of the arc
beginangle	Beginning angle of the arc in tenths of a degree
endangle	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

The Net_Restriction Vias section is removed.

The 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 the layout product. The MISC header ASSIGN_PAIRED_LAYERS represents the top and bottom layer numbers for drilling.

Example

```
ASSIGN_PAIRED_LAYERS
{
PAIRED_LAYERS 1  4
PAIRED_LAYERS 1  2
PAIRED_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 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)
- Associated Nets subsection (header ASSOCIATED NET DATA)
- Design Rules subsection (header DESIGN RULES)

The format is as follows:

Format	Description
RULES_UNIT	Hierarchical rules header for basic units. Use MIL, INCHES, or METRIC to set other units for the Layer section. We recommend 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.
{	
. . .	

}	End of the Groups subsection.
ASSOCIATED NET DATA	Beginning of the Associated Nets s subsection. If there are no associated nets defined, this section can be omitted.
{	
. . .	
}	End of the Associated Nets subsection.
DESIGN RULES	Beginning of the Design Rules subsection.
{	
. . .	
}	End of the Design Rules subsection.
}	End of Hierarchical Rules section.

Net Classes

Format	Description
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 c*lass_name1*.
. . .	Other NET records.
NET *net_nameN*	*net_nameN* is the name of the last net that belongs to class c*lass_name1*.
}	
. . .	Other NET_CLASS records.
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 c*lass_nameN*.
. . .	Other NET records.
NET *net_nameN*	*net_nameN* is the name of the last net that belongs to class c*lass_nameN*.
}	
}	End of the Net Classes subsection.

Pin Pair Groups

Format	Description
GROUP DATA	Beginning of the Groups subsection. If there are no groups defined, this section can be omitted.
{	
GROUP *group_name1*	group_name1 is the name of the first pin pair group that belongs to hierarchy rules.

{	
CONNECTION *1refdes.pin,2refdes.pin*	refdes.pin is a part reference designator and pin number, separated by a period. Separate the refdes.pin fields with a comma.
. . .	Other CONNECTION records.
CONNECTION *Nefdes.pin,Xefdes.pin*	
}	
. . .	Other GROUP records.
GROUP *group_nameN*	
{	
. . .	
}	
}	End of the Groups subsection.

Associated Nets

Format	Description
ASSOCIATED_NET DATA	Beginning of the Associated Nets subsection. If there are no associated nets defined, this section can be omitted.
{	
ASSOCIATED_NET associated_net_name1	associated_net_name1 is the name of the first associated net.
{	
NET net_name1 net_name1	NET net_name1 net_name1 is the name of the first net that belongs to associated net associated_net_name1
. . .	Other NET records.
NET net_nameN	net_nameN is the name of the last net that belongs to associated net associated_net_name1.
}	
. . .	Other ASSOCIATED_NET records.
ASSOCIATED_NET associated_net_nameN	associated_net_nameNassociated_net_nameN is the name of the last associated net.
{	
NET net_name1	net_name1 is the name of the first net that belongs to associated net associated_net_nameN.
. . .	Other NET records.
NET net_nameN	net_nameN is the name of the last net that belongs to associated net associated_net_nameN.
}	
}	End of the Associated Nets subsection.

Design Rules

Format	Description
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.

Format	**Description**
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. Several specific objects, like NET_CLASS, NET, GROUP, CONNECTION, or ASSOCIATED NET, can be listed in the For section. If an object other than DEFAULT is defined in the Against section, the rule set represents conditional rules. Only HIGH SPEED RULE rules can be defined for ASSOCIATED NET objects.

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).

Format	**Description**
{	
TRACK_TO_TRACK 457200	All values are in 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	
COPPER_TO_COPPER 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	
OUTLINE_TO_COPPER 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_CRN 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 0.000000	Capacitance in pF.
MAX_CAPACITANCE 10.000000	
MIN_IMPEDANCE 50.000000	Impedance in Ohm.
MAX_IMPEDANCE 150.000000	
SHIELD_NET *	
SHIELD_GAP 7620000	
MATCH_LENGTH_TOLERAN–E - 7620000	
}	OR
ROUTE_RULE :	
{	
LENGTH_MINIMIZATION_TYPE 1	Minimization types: 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	

MAX_NUMBER_OF_VIAS 500	
VALID_LAYER 1	
VALID_LAYER 2	
VALID_VIA_TYPE STANDARDVIA	
}	
}	End of the rule set.

Differential Pairs Format

There are three types of diff pairs:

- Diff pair of nets
- Diff pair of pin pairs
- Diff pair of associated nets

Diff pair of nets

Format	Description
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	Gap between the traces on <All Layers>
MAX_OBSTACLE_SIZE 952500	Maximum size of obstacle to route around
MAX_OBSTACLE_NUMBER 3	Maximum number of obstacles to route around
WIDTH 381000	Trace width value on <All Layers>
LAYER_NAME <Plane Layers>	Layer of the diff pair rules: <Outer Layers>, <Inner Layers>, <Plane Layers> or electrical layer number.
{	
GAP 11430000	Gap value on the layer (or group of layers)
WIDTH 381000	Trace width value on the layer (or group of layers)
}	
}	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	
}	

Diff pair of pin pairs

Format	Description
DIF_PAIR D2	Beginning of the differential pair set. This is an example of the differential pair defined for two pin pairs, named U1.15,U2.9 and U2.10,U1.14.
{	
CONNECTION U1.15,U2.9	
CONNECTION U2.10,U1.14	
MIN_LENGTH 3810000	
MAX_LENGTH 152400000	
GAP 7620000	Gap between the traces on <All Layers>
MAX_OBSTACLE_SIZE 952500	Maximum size of obstacle to route around
MAX_OBSTACLE_NUMBER 3	Maximum number of obstacles to route around
WIDTH 381000	Trace width value on <All Layers>
LAYER_NAME <Plane Layers>	Layer of the diff pair rules: <Outer Layers>, <Inner Layers>, <Plane Layers> or electrical layer number.
{	
GAP 11430000	Gap value on the layer (or group of layers)
WIDTH 381000	Trace width value on the layer (or group of layers)
}	
}	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	

}	

Diff pair of associated nets

Format	Description
DIF_PAIR D3	Beginning of the differential pair set. This is an example of the differential pair defined for two associated nets, named $$$1*$$$2 and $$$3*$$$4.
{	
ASSOCIATED NET $$$1*$$$2	
ASSOCIATED NET $$$3*$$$4	
MIN_LENGTH 3810000	
MAX_LENGTH 152400000	
GAP 7620000	Gap between the traces on <All Layers>
MAX_OBSTACLE_SIZE 952500	Maximum size of obstacle to route around
MAX_OBSTACLE_NUMBER 3	Maximum number of obstacles to route around
WIDTH 381000	Trace width value on <All Layers>
LAYER_NAME <Plane Layers>	Layer of the diff pair rules: <Outer Layers>, <Inner Layers>, <Plane Layers> or electrical layer number.
{	
GAP 11430000	Gap value on the layer (or group of layers)
WIDTH 381000	Trace width value on the layer (or group of layers)
}	
}	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	

}	

Diff pair of nets

Format	Description
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	Gap between the traces on <All Layers>
MAX_OBSTACLE_SIZE 952500	Maximum size of obstacle to route around
MAX_OBSTACLE_NUMBER 3	Maximum number of obstacles to route around
WIDTH 381000	Trace width value on <All Layers>
LAYER_NAME <Plane Layers>	Layer of the diff pair rules: <Outer Layers>, <Inner Layers>, <Plane Layers> or electrical layer number.
{	
GAP 11430000	Gap value on the layer (or group of layers)
WIDTH 381000	Trace width value on the layer (or group of layers)
}	
}	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.

Format	Description
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 used internally.
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 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 used internally.
HIDDEN N	Attribute is visible in dialog boxes.
}	End of attribute Voltage declaration.
ATTRIBUTE Geometry.Height	Full structured name of attribute is 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 PARTTYPE DECAL PCB	Allowed types and search order.
ECO_REGISTRATION N	Do not register attribute changes in the ECO file.
READONLY N	Attributes can be modified.
SYSTEM Y	Attribute used internally.
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 used internally.
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 is not used.
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 is not used.

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 value is 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 is not used.
HIDDEN N	Attribute is visible in dialog boxes.
}	
}	

Attributes Values Format

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

Format	**Description**
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.

Format	**Description**
VIA *viatype netname x y*	VIA is a keyword. *viatype* is the 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

Format	**Description**
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

Format	Description
LAYER DATA	Layer section header for basic units. Use MIL, INCHES, or METRIC instead of DATA to set other units for the Layer section. We recommend that you 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.
VISIBILE 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
PINNUMBER 1	Pin number color
NETNAME 3	Net name 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:

```
PADS Y
TRACKS Y
VIAS Y
COPPER Y
LINES Y
TEXT Y
ERRORS Y
CLUSTER YCluster view mode required at least one cluster.
TOP_COMPONENT Y
BOTTOM_COMPONENT Y
PLANE_THERMAL_COLOR 15
SELECTION_COLOR 15You can also set some additional colors. Valid values
range from 0 to 15.
HIGHLIGHT_COLOR 14
```

```
FIXED_COLOR 3
PINNUM Y
NET_NAME Y
NET_NAME_ON_TRACES Y
NET_NAME_ON_VIAS Y
NET_NAME_ON_PINS Y
FONT_FACE ArialDefault text font face and style are set
{
FONT_STYLE_BOLD Y
FONT_STYLE_ITALIC Y
FONT_STYLE_UNDERLINE Y
}
}
```

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*

Format	Description
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 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 additional information about the various options and controls, refer to the CAM Outputs section of the Help.

Format	Description
CAM_SECTION PARENT	Begin the CAM section.
{	
CAM_VERSION version	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.

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.

Format		**Document**
CAM_DOC_LIST PARENT		Begin the CAM document list.
{		
CAM_DOC_DIRECTORY *directory_name*		Subfolder(under \CAM) where CAM documents are stored. Defaults to \CAM\default.
CAM_DOC_NAME *name1*		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 *file_name*		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 *n1 ... n30*		List of up to thirty layer numbers corresponding to DOC_LAYER_NAMES. Separate by spaces.
DOC_LAYER_TYPES *type1 ... type30*		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 *value*		Number of the 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.
		Justification of CAM output plots.
DOC_JUSTIFICATION	[OO	Horizontal and vertical offset to origin.
	CM	Horizontal center, vertical middle.
	LB	Horizontal left, vertical bottom.
	LT	Horizontal left, vertical top.
	RB	Horizontal right, vertical bottom.
	RT]	Horizontal right, vertical top.

DOC_OFFSET *x y*	Offset of CAM output plots. Both integers in basic units.

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

Format	**Description**
	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, and Lwhite.
	You may 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 boolean	boolean is 1 or 0 and indicates whether output is clipped based on the graphics workspace.
DOC_FIT_OUTPUT boolean	boolean is either 1 or 0 and indicates whether the data are scaled to fit in the output x and y dimensions.
DOC_NO_PLANE_THERMALS boolean	boolean is either 1 or 0 and determines whether thermal shapes are generated while connecting vias to CAM planes.
DOC_BOARD color	Board outline color.
DOC_CONNECTIONS color	Connection color.
DOC_SLOTS color_pl color_unpl	Slot colors for plated and non-plated.
DOC_OUTLINES color_top color_bottom	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 … coloe30	Via color.
DOC_ROUTES color1 color2 color3 … coloe30	Route color.
DOC_COPPER color1 color2 color3 … color30	Copper color.
DOC_LINES color1 color2 color3 … coloer30	Lines color.
DOC_TEXT color1 color2 color3 … color30	Text color.
DOC_KEEPOUT color1 color2 color3 … color30	Keepout color.
DOC_PAD_OVERSIZE value	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 string1, string2,	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 x y	Offset of drill chart.
DOC_SUPRESS_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:

Format	**Description**
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.

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	min_drill_size	Minimum drill size for this feed and speed in basic units.
	max_drill_size	Maximum drill size for this feed and speed in basic units.
	feed	Feed speed per minute in design units.
	speed	Drill speed per minute, in design units.
. . .		
DRILL_RANGE *value*		min_drill_size max_drill_size feed speed, up to 999 entries.
}		

For assembly drawing documents only, specify the following:

DOC_ASSEMBLY_OPTION string	Specifies the name of the assembly option used when plotting this assembly drawing

Augment on the Fly

The aperture table below can be created automatically. If you make changes to your design, any new apertures needed are added to the table 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.

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_APERATURE_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

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 ASSEMLY_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*][;…]

Format	**Description**
ASSEMBLY_OPTION	Keyword
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*

Format	**Description**
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-300010000GNDSTANDARDVIA
VIA-20020001$1234MICROVIA
PIN-3003900 +5    U1.5
PIN-1001001 .NONE.U1.8
```

Index

End-User License Agreement

The latest version of the End-User License Agreement is available on-line at:
www.mentor.com/eula

IMPORTANT INFORMATION

USE OF ALL SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF SOFTWARE INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.

END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section 2) and hardware (collectively "Products") between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties' entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

1. **ORDERS, FEES AND PAYMENT.**

   1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer's appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement ("Order(s)"), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not these documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order will not be effective unless agreed in writing by an authorized representative of Customer and Mentor Graphics.

   1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice(s). Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer's sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer's behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.

   1.3. All Products are delivered FCA factory (Incoterms 2000), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics' delivery of Software by electronic means is subject to Customer's provision of both a primary and an alternate e-mail address.

2. **GRANT OF LICENSE.** The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation and design data ("Software") are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection 5.2); (b) for Customer's internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee's residence, an airport or hotel, provided that such employee's primary place of employment is the site where the Software is authorized for use. Mentor Graphics' standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer requests any change or enhancement to Software, whether in the course of receiving support or consulting services, evaluating Software, performing beta testing or otherwise, any inventions, product

improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics' sole discretion) will be the exclusive property of Mentor Graphics.

3. **ESC SOFTWARE.** If Customer purchases a license to use development or prototyping tools of Mentor Graphics' Embedded Software Channel ("ESC"), Mentor Graphics grants to Customer a nontransferable, nonexclusive license to reproduce and distribute executable files created using ESC compilers, including the ESC run-time libraries distributed with ESC C and C++ compiler Software that are linked into a composite program as an integral part of Customer's compiled computer program, provided that Customer distributes these files only in conjunction with Customer's compiled computer program. Mentor Graphics does NOT grant Customer any right to duplicate, incorporate or embed copies of Mentor Graphics' real-time operating systems or other embedded software products into Customer's products or applications without first signing or otherwise agreeing to a separate agreement with Mentor Graphics for such purpose.

4. **BETA CODE.**

4.1. Portions or all of certain Software may contain code for experimental testing and evaluation ("Beta Code"), which may not be used without Mentor Graphics' explicit authorization. Upon Mentor Graphics' authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. This grant and Customer's use of the Beta Code shall not be construed as marketing or offering to sell a license to the Beta Code, which Mentor Graphics may choose not to release commercially in any form.

4.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer's use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer's evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.

4.3. Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer's feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection 4.3 shall survive termination of this Agreement.

5. **RESTRICTIONS ON USE.**

5.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer's employees and on-site contractors, excluding Mentor Graphics competitors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use it except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer learns or becomes aware of such unauthorized disclosure or use. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse-assemble, reverse-compile, reverse-engineer or in any way derive any source code from Software. Log files, data files, rule files and script files generated by or for the Software (collectively "Files"), including without limitation files containing Standard Verification Rule Format ("SVRF") and Tcl Verification Format ("TVF") which are Mentor Graphics' proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Software or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Software, or disclose to any third party the results of, or information pertaining to, any benchmark.

5.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized use. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer's employees or contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.

5.3. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense or otherwise transfer the Products, whether by operation of law or otherwise ("Attempted Transfer"), without Mentor Graphics' prior written consent and payment of Mentor Graphics' then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics' prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics' option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer's permitted successors in interest and assigns.

5.4. The provisions of this Section 5 shall survive the termination of this Agreement.

6. **SUPPORT SERVICES.** To the extent Customer purchases support services, Mentor Graphics will provide Customer updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics' then current End-User Support Terms located at http://supportnet.mentor.com/about/legal/.

7. **AUTOMATIC CHECK FOR UPDATES; PRIVACY.** Technological measures in Software may communicate with servers of Mentor Graphics or its contractors for the purpose of checking for and notifying the user of updates and to ensure that the Software in use is licensed in compliance with this Agreement. Mentor Graphics will not collect any personally identifiable data in this process and will not disclose any data collected to any third party without the prior written consent of Customer, except to Mentor Graphics' outside attorneys or as may be required by a court of competent jurisdiction.

8. **LIMITED WARRANTY.**

8.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Products will meet Customer's requirements or that operation of Products will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates or (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification or improper installation. MENTOR GRAPHICS' ENTIRE LIABILITY AND CUSTOMER'S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY, PROVIDED CUSTOMER HAS OTHERWISE COMPLIED WITH THIS AGREEMENT. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."

8.2. THE WARRANTIES SET FORTH IN THIS SECTION 8 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

9. **LIMITATION OF LIABILITY.** EXCEPT WHERE THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 9 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

10. **HAZARDOUS APPLICATIONS.** CUSTOMER ACKNOWLEDGES IT IS SOLELY RESPONSIBLE FOR TESTING ITS PRODUCTS USED IN APPLICATIONS WHERE THE FAILURE OR INACCURACY OF ITS PRODUCTS MIGHT RESULT IN DEATH OR PERSONAL INJURY ("HAZARDOUS APPLICATIONS"). NEITHER MENTOR GRAPHICS NOR ITS LICENSORS SHALL BE LIABLE FOR ANY DAMAGES RESULTING FROM OR IN CONNECTION WITH THE USE OF MENTOR GRAPHICS PRODUCTS IN OR FOR HAZARDOUS APPLICATIONS. THE PROVISIONS OF THIS SECTION 10 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

11. **INDEMNIFICATION.** CUSTOMER AGREES TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE OR LIABILITY, INCLUDING ATTORNEYS' FEES, ARISING OUT OF OR IN CONNECTION WITH THE USE OF PRODUCTS AS DESCRIBED IN SECTION 10. THE PROVISIONS OF THIS SECTION 11 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

12. **INFRINGEMENT.**

12.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will pay costs and damages finally awarded against Customer that are attributable to the action. Customer understands and agrees that as conditions to Mentor Graphics' obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.

12.2. If a claim is made under Subsection 12.1 Mentor Graphics may, at its option and expense, (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.

12.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics' licensors who do not provide such indemnification to Mentor Graphics' customers; or (h) infringement by Customer that is deemed willful. In the case of (h), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.

12.4. THIS SECTION 12 IS SUBJECT TO SECTION 9 ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS FOR DEFENSE, SETTLEMENT AND DAMAGES, AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.

13. **TERMINATION AND EFFECT OF TERMINATION.** If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term.

13.1. Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer's obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.

13.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer's possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.

14. **EXPORT.** The Products provided hereunder are subject to regulation by local laws and United States government agencies, which prohibit export or diversion of certain products and information about the products to certain countries and certain persons. Customer agrees that it will not export Products in any manner without first obtaining all necessary approval from appropriate local and United States government agencies.

15. **U.S. GOVERNMENT LICENSE RIGHTS.** Software was developed entirely at private expense. All Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to US FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in this Agreement, except for provisions which are contrary to applicable mandatory federal laws.

16. **THIRD PARTY BENEFICIARY.** Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.

17. **REVIEW OF LICENSE USAGE.** Customer will monitor the access to and use of Software. With prior written notice and during Customer's normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer's software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer's compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FLEXlm or FLEXnet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics' request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section 17 shall survive the termination of this Agreement.

18. **CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION.** The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the United States. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, USA, if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply. Notwithstanding the foregoing, all disputes in Asia arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International Arbitration Centre ("SIAC") to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. This section shall not

restrict Mentor Graphics' right to bring an action against Customer in the jurisdiction where Customer's place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.

19. **SEVERABILITY.** If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.

20. **MISCELLANEOUS.** This Agreement contains the parties' entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements, including but not limited to any purchase order terms and conditions. Some Software may contain code distributed under a third party license agreement that may provide additional rights to Customer. Please see the applicable Software documentation for details. This Agreement may only be modified in writing by authorized representatives of the parties. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.

Rev. 100615, Part No. 246066