PADS® Layout Concepts Guide

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

List of Figures

List of Tables

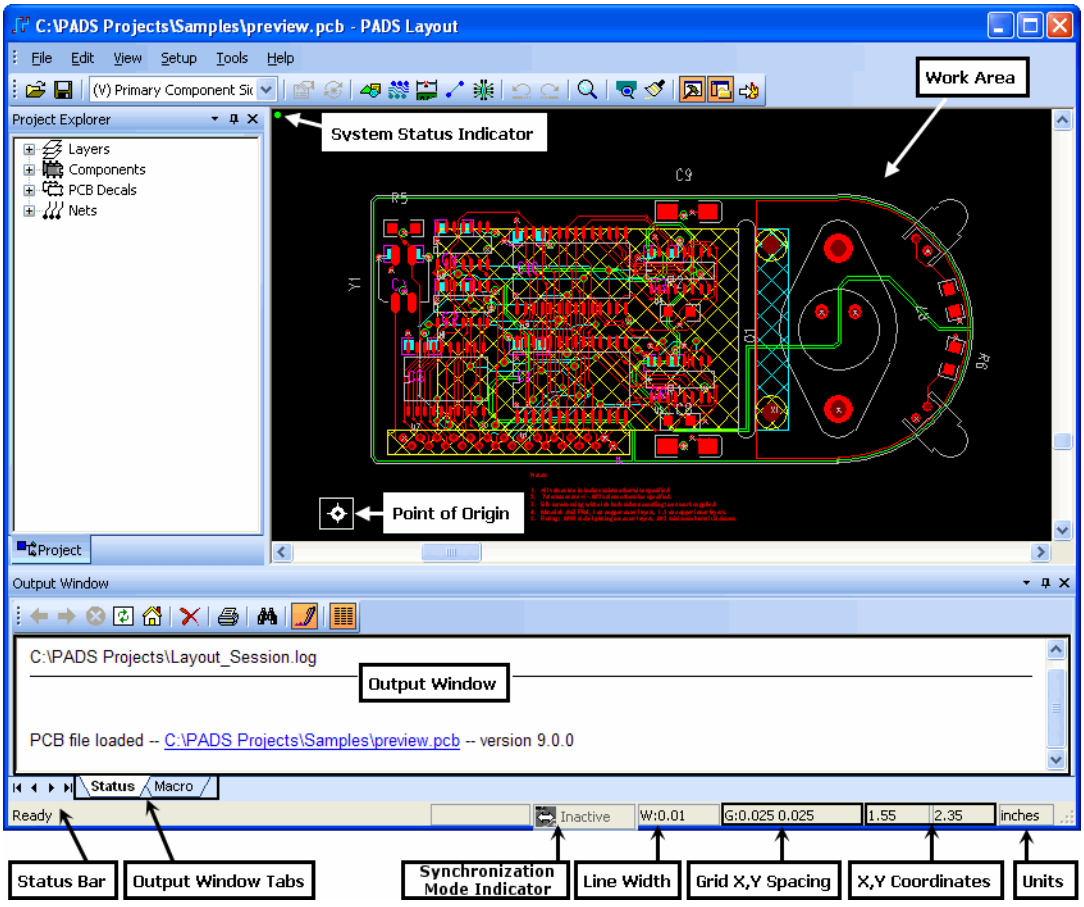
Chapter 1
Interface

This chapter introduces you to the PADS Layout interface, including information on screen elements, work areas, and view. It also discusses standard Windows menus and icons for accessing commands as well as common commands and settings.

User Interface Elements

The PADS Layout interface contains standard menus and buttons to access commands and settings. Figure 1-1 shows the various parts of the interface. The remainder of this section describes each screen element.

Figure 1-1. PADS Interface



Title Bar

The application icon, document name, and application name appear on the title bar. Click the application icon to open the Windows-standard control menu, which contains commands for working with the application window.

openObjectId gui.fm:2 34442

The document name changes to reflect the state of the current document. For example, when no design file is loaded "Untitled - PADS Layout" appears as the document name. When a design file is loaded, the path, the file name, and the file extension (.pcb) appear on the title bar.

Menu Bar

The menu bar lists PADS Layout commands. The menus also show the appropriate command icons, access keys, and shortcuts. When a command ends with ellipses (...), additional information is needed to complete the command. For more information, see *PADS Layout Help*.

Standard Toolbar

The Standard toolbar contains commands that open and save designs, change the view, redraw, and access the Drafting, Design, Dimensioning, ECO, and BGA toolbars.

Status Bar

The status bar displays a command name or information on a selected connection, route, or component. The layer on which a selected trace exists appears on the status bar.

System Status Indicator

The system status indicator shows the processing state of the application in the upper left corner of the screen. It appears green when the system is idle or ready for operation, and red when the system is unavailable either because the system is processing or the work space cannot receive user input, such as when in CAM.

Point of Origin

The 0,0 coordinate location. X,Y coordinates are calculated from this point.

Synchronization Mode Indicator

Displays whether PADS Layout and PADS Router are in Synchronization mode. Also indicates which program is the active one versus the inactive one or whether the inactive program is out of sync.

Line Width

Displays the current line width setting.

X,Y Coordinate

Displays the horizontal distance of the cursor from 0,0 as the x-coordinate. Also indicates the polar radius if you are using a polar grid.

Displays the vertical distance of the cursor from 0,0 as the y-coordinate. Also indicates the polar angle if you are using a polar grid.

Units

Displays the current units used in the design.

Work Area

The area in which you enter all design information is called the work area or the workspace. The work area contains two editors: the Layout editor and the PCB Decal editor. The Layout editor appears when you open PADS Layout. You can place, route, and otherwise modify your board in this editor. The PCB Decal Editor is an editor that you start (through the Tools menu) where you can create or edit a decal.

The maximum work area is a 56 by 56 inch square. You can measure it in inches, mils, or metric units. Set the units of measure in Global Options. The work area is divisible by an X,Y, or horizontal and vertical grid, which you can set to a minimum of .00001" between points. You can set the X and Y values separately. This is called the Design Grid on the Grids tab of the Options dialog box.

Setting Grids in the Work Area

The current grid settings also appear in the Design Grid area at the bottom of the work area. When you move an object or use a drafting command, the grid readout is replaced by a Delta X and Y reading, calculated from the cursor selection point when the command starts. Negative numbers indicate left and down.

Display Grid

The *display grid*, a field of white dots, is a valuable drafting aid. Set the display grid to either match the design grid or at larger multiples of the design grid.

Design Grid

The design grid is based on the origin. If you move a part across the board during design, the cursor may move smoothly but the part snaps from grid point to grid point. When Snap to Grid is on, you cannot place a part off the design grid.

Via Grid

The via grid controls the placement of vias.

Fanout Grid

The fanout grid sets the spacing of ball grid arrays and fanouts. This controls placement of substrate bond pads on a die and placement of fanout vias. This data is passed to PADS Router.

Hatch Grid

The hatch grid changes the distance between hatch lines.

Origin

When you create a new file, the default drawing format is centered at medium magnification in the work area with the *origin*, or the 0,0 point, in the lower left corner. The origin appears as a large white dot. As you move the cursor, its position relative to the origin appears in the X,Y Coordinates area. The numbers change in multiples of the design grid.

Controlling Views

Several methods control which portion of the design appears on the screen.

View Commands and Scroll Bars

The commands listed in Table 1-1 control the view in the workspace.

Table 1-1. View Commands

Graphic	Command	Description
	Zoom	Moves closer to or farther from the design.
	Board	Fits the board outline into the workspace.
	Redraw	Redraws the work area.
	Extents	Zooms the view to encompass all objects in the current design.
	Scroll Bars	Pans the view.

Mouse Operations

PADS Layout follows Microsoft Windows conventions for two-button mouse operations, but also supports using a three-button mouse. The middle button provides quick access to the pan and zoom commands. Table 1-2 shows the operations associated with mouse buttons.

Table 1-2. Mouse Button Operations

Operation	Mouse Action
Select objects.	Move the pointer to the item you want to select and click the left mouse button, this is called left-click.
Select menu commands, tabs, and dialog box options.	Move the pointer to the command, tab, or option you want and click the left mouse button, this is called left-click.

Table 1-2. Mouse Button Operations (cont.)

Operation	Mouse Action
Open a shortcut menu of edit commands related to a selected object.	Position the pointer on an item and press the right mouse button. This is called right-click. You don't have to hold the mouse button to keep the menu visible. The pointer is free for selecting from the menu. Once you select a command from the shortcut menu, the menu closes and the pointer returns to its original position.
Pan the design. This moves the view from side to side or up and down without changing the size.	With the pointer where you want to center the view, click the middle mouse button. The screen refreshes with the point you chose at the center.
Zoom out from the design. This decreases a specific area.	Press the middle mouse button and pull the box diagonally and down across the area you want included in the decreased new view. A solid box appears at the pointer. This represents the current view size. The thin line that expands from the solid box represents the new view size in proportion to the old. The zoom-out ratio appears with the pointer.
Zoom into the design. This enlarges a specific area.	Hold the middle mouse button down, and move the mouse diagonally and up across the area you want to enlarge. A rectangle starts and grows with the movement of the pointer. When you release the left mouse button, the view zooms to the rectangle.
Initiate an edit action or complete the current action.	Rapidly press and release the left mouse button twice, this is called double-click.
Highlight a design or object for the next command execution.	To position the pointer on a design or text object and click the left mouse button.
Select a menu command, button, or dialog box option, button, or tab.	Position the pointer on the menu item or dialog box item and click the left mouse button. This is usually called click, but may be called left-click.
Move an object.	Position the pointer at the starting point or on the object, press and hold the left mouse button while you move the pointer and object you are moving. Release the mouse button to complete the move. This is called drag and drop.

Keypad Operations

As shown in Table 1-3, the keypad can also control the view. NumLock can be on or off except where noted.

Table 1-3. Keypad Operations

Key	Description
Home	Fits the board to the view.
End	Redraws the current view.
Arrows	Pans the viewing window in the direction of the arrow. Moves one-half the screen width when NumLock is on. Moves by grid unit when NumLock is off.
5	Draws a zoom rectangle when NumLock is on.
Pg Up	Zooms in centered at the cursor location.
Pg Dn	Zooms out centered at the cursor location.
Ins	Centers the view at the current cursor location, without zooming.

View Modes

Outline View Mode

Use Outline mode to speed redraw time by displaying traces and pads as outline objects instead of as filled objects. Traces appear as two parallel lines, separated by the established trace width. Pads appear as outlines of their shapes.

Resolution Option

With resolution options you can change the display resolution for faster redraw of very large designs. High resolution, which is the default, displays objects in their true shape. Low resolution displays pads as square or rectangular objects. The last resolution setting you used is saved, for example, if you set low resolution, selecting outline mode toggles between normal view and low resolution mode.

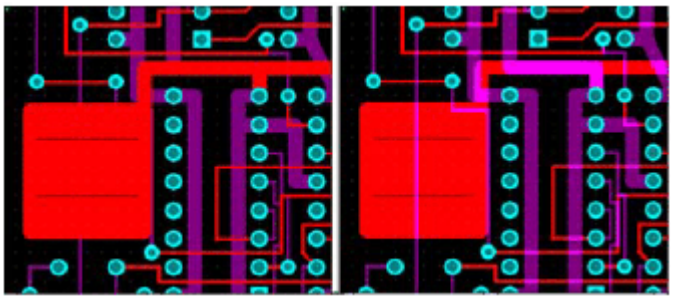
Viewing Protected Traces

If your design contains protected traces, you can set them to display with an outline pattern opposite that of other traces. In other words, if normal traces are solid, protected traces are outlined, and vice-versa. For more information, see “Route Protection” on page 227.

Transparent View Mode

With Transparent mode you can view traces on several layers at once. Transparent mode can show obstacles that may be hidden directly under the current active layer. When Transparent mode is on, you see trace-over-trace or trace-over-part outline overlaps drawn in a third, lighter color. The overlap color is determined by the colors you assign to your traces. With Transparent mode off, overlaps do not appear.

Figure 1-2. A Routed Board with Transparent Mode Off and On

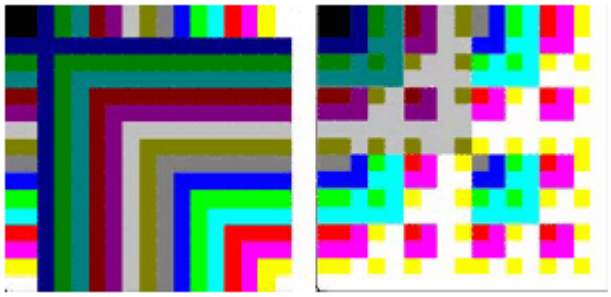


You should use transparent mode for inspecting dense route areas or when routing in areas with layer-specific obstructions.

Color Selection and Transparent Mode

The transparent effect is more visible if you use darker colors for traces. Bright colors like yellow and aqua wash out the effect.

Figure 1-3. A Sixteen Color Grid with Transparent Mode Off and On



Tip: Items that you select may not clear normally and may remain highlighted until you click Redraw on the toolbar.

View Nets

Use View Nets to hide or show routed or unrouted paths by netname. You can select one or more netnames and specify view details: all, none, traces but no connections, and so on. When View Nets hides unrouted connections, neither Verify Design nor Find can see them. Be sure View Nets does not disable nets or traces you want to search for or select.

Object Types in the Project Explorer

Objects in the Project Explorer are placed in object groups. As shown in Table 1-4, object groups can be either primary and secondary. You cannot remove or rename primary object groups.

Table 1-4. Object Groups

Primary Group	Secondary Group	Description
Layers	Electrical layers	Lists all electrical layers, including plane layers and routing layers.
	General layers	Lists all other layers except electrical.
Components		Lists all components and pin pairs.
Part decals		Lists all part decals in the design or all components that use the selected part decal.
Nets		Lists all nets.

Output Window

The Output window contains a session log, status reports, a macro editor, and Internet browser linking capabilities. Use the Output window to:

- Display reports and session logs.
- Create and debug macros.

The Output window contains two tabs and is located in the lower portion of the screen (see Figure 1-1 on page 18).

Table 1-5. Two Tabs of the Output window

Tab	Description
Status	Displays information on the current session.
Macro	Run, edit and debug macros.

Status Tab

The Status tab displays information on the current session. You can also use the Status tab to work with HTML documents. Although it is not a Web browser, you can open Internet Web pages referenced by links.

If the Status tab is closed, and you get an error while autorouting (or performing other tasks), the output window automatically opens with the Status tab active.

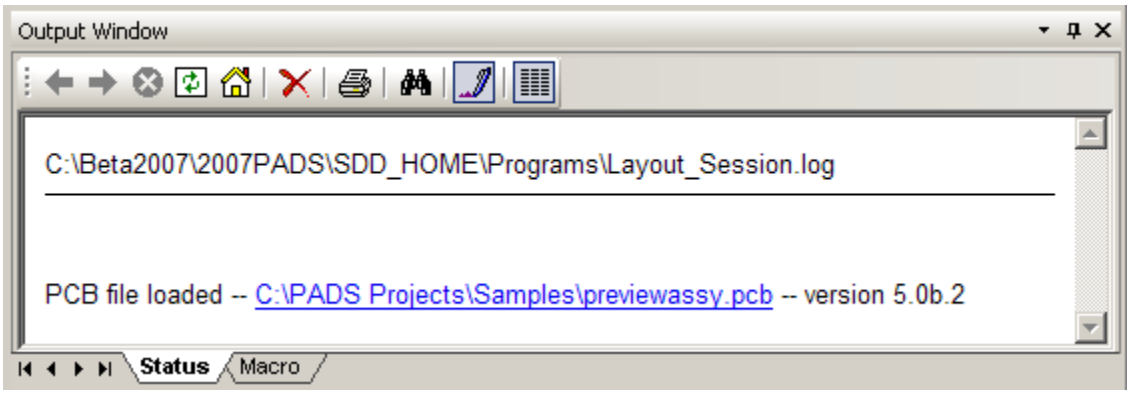
Use the Status tab to:

- Record, display, and print the session log.
- Print and display report files.
- Print and display web pages.
- Open program files.

Session Log

The contents of the session log, session.log, appear in the Status tab, and contain all program output for the current session, including names of open and saved files, integrity test results, and messages.

The following graphic illustrates the type of status information that appears when you open a design file. It specifies the file name of the open file.



The Status tab displays information using different colors. Underlined items are links.:

Table 1-6. Text Color Indicators

Color	Represents
Red	Errors

Table 1-6. Text Color Indicators

Color	Represents
Green	Warnings
Black	Messages
Blue	Links to files, Web pages, and database objects

Saving the Session Log

To save the session log, click Log to file on the Status tab toolbar. If a session log file already exists, new information is appended. If a session log file does not exist, clicking Log to file creates a new file.

The default path for the session log comes from the .ini file entry *FileDir=C:\PADS Projects*. When you first start the software, the location is set in the following registry key:

HKEY_CURRENT_USER\Software\Mentor Graphics\<version>\PADS Layout\Status Window\LastLogName

See also: Managing Session Logs

Shortcut Keys

Table 1-7 shows keyboard shortcut keys and key combinations. Use keyboard shortcuts to start commands and to change system settings.

Table 1-7. Keyboard Shortcuts

Command	Shortcut Key
Absolute/Relative coordinates toggle (macros)	F9
Add jumper	Ctrl+Alt+J
Adds route	F2
Align component	Ctrl+L
AutoRoute selected	F7
Cancels command	Esc
Copy	Ctrl+C
Create cluster	Ctrl+K
Create union	Ctrl+G
Cut	Ctrl+X
Cycle pick	Tab
Display Colors Setup	Ctrl+Alt+C
Dynamic route	F3
Ends recording (macros)	F10
Flip selected	Ctrl+F
Highlight	Ctrl+H
Left-click	Spacebar
Length minimization	Ctrl+M
Mouse move compression on/off (macros)	F8
Move	Ctrl+E
New file	Ctrl+N
Next View	Alt+N
Not in use	Ctrl+P
Not in use	Ctrl+T
Open file	Ctrl+O
Opens help	F1

Table 1-7. Keyboard Shortcuts (cont.)

Command	Shortcut Key
Opens shortcut menu for current mode. Same as right-click.	M
Options	Ctrl+Enter
Options > General tab	Ctrl+Alt+G
Options > Design tab	Ctrl+Alt+D
Paste	Ctrl+V
Properties	Alt+Enter
Previous View	Alt+P
Quick Measure • Works the same as the Q modeless command. Resets the Delta X,Y values to zero and begins measuring from the current location. This command works in any mode, with Delta X,Y on the status bar instead of the grid display.	Ctrl+PageDown
Record prompt windows on/off (macros)	Ctrl+Shift+P
Redo	Ctrl+Y
Redraw	Ctrl+D
Removes last route corner	Backspace
Reset delta coordinates to measure from current position	Ctrl+Page Down
Rotate selected	Ctrl+R
Route loop	Ctrl+J
Save	Ctrl+S
Scale	Ctrl+Alt+Z
Select all	Ctrl+A
Selection filter	Ctrl+Alt+F
Selects net	F6
Selects pin pair	F5
Spin selected	Ctrl+I
Stretch	Ctrl+Alt+Y
Toggle menu bar	Ctrl+Alt+M
Toggles layer pair	F4

Table 1-7. Keyboard Shortcuts (cont.)

Command	Shortcut Key
Undo	Ctrl+Z
Unhighlight	Ctrl+U
View board	Home
View extents	Ctrl+Alt+E
View nets	Ctrl+Alt+N
View zoom mode	Ctrl+W
Mouse Click Substitute Commands	
Activate shortcut menu for current mode (right mouse button).	M
Perform a left mouse button click (to add corners, select items, complete, etc.) at the current cursor location.	Spacebar

Related Topic

Modeless Commands

Chapter 2
File Operations

This chapter introduces the common file operations in PADS Layout. You can use these commands to open, save, import, and export designs. This chapter also covers creating reports and customizing reports.

Opening Files

PADS Layout can open these files:

- Native design files, *.pcb
- Physical design reuse files, *.reu, known as the *reuse definition*.
- Old file format .job files. (Only PADS Perform V6 and PADS Work V7 format .job files can be opened.)

When you open a file in PADS Layout, several conversions may take place to update your data to current data format.

File Open Conversions

Moving Copper

In some previous versions of PADS Layout you could create copper that appeared on all layers (Layer 0). This version of PADS Layout requires you to create copper on a specific layer. If you open a file containing copper created for all layers, the copper is placed on Layer 1 and a message appears indicating which items moved.

Converting Attributes

When you open a design, the Attribute Dictionary is loaded. You can change the default attributes that are imported by changing the default attribute dictionary. For more information, see “Default Attributes” on page 167.

If you open a design created with a version prior to PowerPCB version 3.0 any part type attributes found are converted to new attributes, added to the Attribute Dictionary (if necessary), and applied to the items on which they were found. For example, if you assigned value and tolerance in your existing design, they are converted to Value and Tolerance attributes and then added to the appropriate objects.

Table 2-1. Items Converted to Attributes in Version 3

Attribute Name	Created When
Value	A value is found appended to a part type name.
Tolerance	A tolerance is found appended to a part type name.
ASSEMBLY_OPTIONS	Assembly Variants exist in the design.
DFT.Nail Count Per Net	The nail count per net is greater than one, the default. The nail count may be greater than one if the Insert Multiple Test Points Per Net check box is selected in the Audit Rules tab of the DFT Audit dialog box.

Converting Labels

If you open a design created with a version prior to PowerPCB version 3.0, the reference designators and part types are converted to current labels. The location, orientation, and color of the labels are preserved if they were on the Top or Bottom layers. The default height and width are used, mirroring is turned off, and labels are placed on the Top layer. Also, visibility is set to Value. For information on changing these settings see the Modifying Part Label Properties topic.

Setting Differential Pair Trace Width and Gap When Opening Files

You can set the width and gap for differential pair traces for all layers or per layer. The width is set for all layers and is the larger of the recommended widths for each member of the differential pair. The gap assigned to the differential pair is set for all layers.

To set the width and gap per layer, see Creating Differential Pair Design Rules.

Creating Files

You can clear the current design from memory and create a new design.

Use the Set Start-up File command to change the start-up settings for new designs. This command will not change the settings in a current design, only for subsequent designs.

Start-up Files

Use start-up files to save color settings, the attribute dictionary, and other universal parameters and then apply them to every new .pcb file. To set and apply default settings use the System Default (default.asc) start-up file.

For information on creating start-up files, see the Specifying the Start-up File and Creating Start-up Files topics.

The start-up files provided with PADS Layout and listed in Table 2-2 show typical values for different technologies, and do not represent any specific technology.

Table 2-2. Start-up Files

	System Default	**Chip on Board**	**Low Temp Cofired Ceramic**	**MCM-L**
Filename	default.asc	cob-startup.stp	ltcc-startup.stp	mcml-startup.stp
Units	mils	mils	mils	metric (mm)
Design grid	100x100	5x5	1x1	0.005
Via grid	25x25	5x5	1x1	0.005
Layers	2	2	13	2
Trace Width	12	3	4	0.05
Vias (top/inner/bottom)				
Standard	55/55/55	30/30/30	55/55/55	0.2/0.2/0.2
Micro	None	None	6/4/6	None
Clearances				

Table 2-2. Start-up Files (cont.)

	System Default	Chip on Board	Low Temp Cofired Ceramic	MCM-L
Trace-Trace	6	3	5	0.35
Trace-Via	6	3	4	0.35
Via-Via	6	3	4	0.35

Unsupported Start-up Files

Warning: This method of creating a start-up file is not supported.

The start-up file is an ASCII file with an .stp file extension. By design, you are restricted with what you can include in the start-up file by the Start-up File Output dialog box.

But, you could export an unrestricted ASCII file using the ASCII Output dialog box. Then you could rename the extension from .asc to .stp and place the file in the C:\MentorGraphics\<version>\SDD_HOME\Settings folder where it would be found and used as a start-up file. For example, if you repeatedly design products using the same board outline, you could use this method to export your board outline into the start-up file. Then, when starting a new design, you could begin with your board outline.

Library Operations

Libraries store decal and part type attributes and attribute labels, but they do not store attribute values. Table 2-3 shows the four kinds of items that can exist in a library.

Table 2-3. Library Items

Item	Description
Decal	The graphical representation of the part when it is drawn. This is often referred to as the *footprint*.
Part	Data about a part, including logic family, attributes, pins, and gates. For example, as a 74LS02.
Lines	Graphical data you can store in the library to use in any design file. For example, a company logo.
CAE Decal	The graphical representation of a schematic part, such as a NOR gate. This section functions as a part list reader only. Use PADS Logic to create and modify CAE decals.

Library Search Order

The list of libraries in the Library Manager dialog box also displays the library search order. When you have multiple libraries available, they are processed in their order in the library list whenever the libraries are searched. The libraries are searched during the following procedures:

- Searching for library items using various dialogs
- Importing a netlist from the schematic
- Updating your design from the library
- Annotating your design with new components from the schematic

You can change the search order of libraries using the Library List dialog box.
See also: Setting the Library List Order.

Part Types

You cannot add a part to a PADS Layout design, or import the part as a component of a netlist, unless a part type and a PCB decal for the part are available in your library.

- The PCB decal represents the “footprint” of the part (including pad and drill sizes) on the board, and is required both to place the part on the board and to route the nets connecting to the part’s pins.
- The part type represents the electrical configuration of the part, including its logic family, pins, gates and gate pins, and mapping of gate pins to the part’s pins. It also includes other information, such as part attributes and the list of PCB decals assigned to the part. When you add a part to a design, it is added as an instance of the part type.

You create a new PADS part type, or modify an existing part type, by specifying the properties of the part in the tabs of the Part Information dialog box, as follows:

In this tab:	**Set these properties:**
General	Part statistics, logic family, pin number mapping status, connector status, ECO-registered status, list of reference designator prefixes to define the set of parts to which these properties apply, define part as special purpose (connector, die part, or flip chip).
PCB Decals	List of assigned PCB Decals **Requirement:** Assign a decal to the part before specifying gate information, signal pin names, and alphanumeric pin names.
Gates	Default and alternative gates, gate swapping eligibility, gate properties, gate pin-swapping eligibility
Pins	Signal names for pins, such as power and ground pins
Attributes	Attributes of the part
Connectors	Default and alternative CAE decals used as connectors in PADS Logic **Restriction:** This tab is unavailable when the Connector check box on the General tab is cleared, or when a gate has been assigned to the part on the Gates tab.
Pin Mapping	Optional pin mapping of logical to physical pin numbers **Restriction:** This tab is unavailable when you select the **Define Mapping Of Part Type Pin Numbers To PCB Decal** check box on the General tab, and assign one or more decals containing alphanumeric pin numbers. The decal determines the number of pins in the part.

Modifying Gates in Parts in the Library

You can modify parts in the library using the Library Manager. Information you can modify includes location, decal, connector, gates, alphanumerics, signal pins, and attributes. For more information, see the Creating and Modifying Part Types topic.

For each gate, you can type the CAE Decal name; the name of the logic symbol that is used to display the part in the schematic. Alternate decal assignments must have the same number of pins. You can define one primary and three alternate decals for each gate.

When at least one decal is assigned to a part, you can type or modify its gate information. This includes swap enabling or disabling for gates within a part or between similar parts. This information lets PADS Layout know which gates it can substitute for connection length minimization after placement.

You can also swap pins within gates to uncross connections and facilitate routing. In both gate and pin swapping, you assign a number to the gate or pin in the Gates tab. Pins with like numbers can swap within a gate. Gates with like numbers can swap within the part, or to other similar parts. A one (1) indicates the gate is swappable with gates of the same part type in the PCB design database. If a part contains more than one type of swappable gate, then identify the second type with the number 2, the third type with 3, and so on. Zero (0) indicates that no swapping can occur.

Library Conversion

To support new functionality, the library structure for PADS Layout and PADS Logic is updated from time to time. When you convert earlier versions to the latest version libraries, they change so they are compliant with the latest software. The conversion changes the file name extension.

For steps detailing how to convert libraries, see the Library Converter Help.

> **Tip**: If you have libraries from PowerPCB versions 1 or 2 you need to use the library converter in PowerPCB 3.x.

A report file is created listing which libraries converted along with their conversion status (fully converted, converted with *n* failures, or no conversion possible). For each library the report will list the items that converted with their status (converted OK or failed to convert).

The Decal Wizard Options Configuration File

The Default configuration of the Decal Wizard Options cannot be deleted or modified but you can save a new configuration file. This configuration file is saved in the *UserDir* folder. The *UserDir* environment variable is set in the /SDD_HOME/Programs/powerpcb.ini file, and by

default is set to the /SDD_HOME/Settings/ folder. The file has an extension of .dwc. These files are in text format and can be opened with your default text editor.

Custom Package Types and Pitch Ranges

You can use your default text editor to further customize the configuration file you saved from the Decal Wizard Options dialog box. You can edit the existing package types or add new package types. If you edit the text of your .dwc file, you must consider the following requirements:

- Only pre-defined component families can be used, and all the required parameters for the component family must exist (for example, Toe).
- The component_type section numbers must not skip a number; however they do not need to be in consecutive order in the file. For example, if numbers 4 and 6 exist, there must also be a number 5 somewhere in the file.
- Pitch ranges cannot overlap and, all together, must cover all values from 0 to infinity.
- The *DecalName* parameters are restricted to <Pitch>, <LeadSpan>, <LeadSpan1>, <LeadSpan2>, <Height>, <BodyWidth>, <BodyLength>, and <PinCount>.

If these requirements are not met and you attempt to use the configuration in the Decal Wizard Options dialog box, a warning appears to explain the problem and you cannot use the configuration file.

Layout-Driven Design

Most PCB designs are Schematic-Driven. They are created in schematic software and the components, connections, and constraints are passed to PADS Layout. Then according to those constraints, components are then placed, and their connections are made permanent with copper traces.

But if the design is simple, and you are able to skip creating the schematic(s), you can add components, create connections, and create constraints - all within PADS Layout. There is a set of tools on the ECO toolbar that allow you to create a layout-driven design. The ECO toolbar contains all the tools that are design-altering. Schematic-driven designs must remain synchronized/in parallel between the schematic and layout tool. Therefore, design modifications by tools on the ECO toolbar can be recorded in an ECO file in order for the schematic design to be updated to bring the two designs in sync again.

Using the tools on the ECO toolbar, you can:

- Add and delete components
- Add and name connections/nets
- Rename and delete nets

- Rename component reference designators
- Change components
- Swap pins and gates
- Assign constraints/design rules

Importing and Exporting Files

You can import design or setup data from variously formatted files into a .pcb file. You can export, or extract, design data from the open .pcb file into other formats. Table 2-4 shows available import and export formats.

Table 2-4. Import and Export Formats

Format	Description
ASCII (.asc)	Import and export. PADS-format ASCII. For more information, see “ASCII Format” on page 305.
Data eXchange Format, or DXF (.dxf)	Import and export. AutoCAD 14 format. For more information, see “DXF Format” on page 316.
Engineering Change Order (.eco)	Import only. Forward annotation information generated in a schematic capture application, contains logic changes to the design. For more information, see “ECO Process” on page 123.
Intermediate Data Format, or IDF (.emn and .emp)	Exporting IDF files. Import and export IDF 2.0 and 3.0 format. IDF is an industry standard for exchanging data between electrical and mechanical design systems. With IDF you can exchange data with applications such as Pro/ENGINEER. You need the IDF Interface option in PADS Layout to import IDF files. For more information, see “Intermediate Data Format” on page 319.
OLE (.ole)	Import and export. You can embed files from other applications as OLE objects in a design using Insert New Object on the Edit menu. Once you have an OLE object in your design, you can export the object as a singular item to an .ole file using Export. Then you can import the .ole file into other PADS Layout designs. For more information, see “Object Linking and Embedding” on page 282.

Table 2-4. Import and Export Formats (cont.)

Format	Description
HyperLynx (.hyp)	You can create HyperLynx files using BoardSim from the Tools menu. For more information, see the Creating HyperLynx BoardSim - HYP Files topic.
CAM350 (.cam)	You can create .cam, or CAM350, files using CAM350 on the Tools menu. For more information, see “CAM Document Creation Workflow” on page 248.

Creating Reports

The Reports command generates reports for the currently loaded design. Several report formats are provided, plus you can create reports formatted to match existing design standards. Each report pass creates a file called report.rep, which is stored in the \PADS Projects folder.

Report Types

PADS Layout includes two types of reports:

- Predefined Reports
- Customizable Reports

Predefined Reports

Table 2-5 shows predefined report types, which you cannot delete.

Table 2-5. Predefined Reports

Report	Description
Unused	Provides a listing of all unused pins for each package in a design.
Statistics	Provides a variety of statistical information in a design such as number of layers, drill locations, and routed connections.
Limits	Provides maximum numbers of the various design items, based on your program's package limits.

Customizable Reports

PADS Layout includes report format files that you can customize to fit specific output requirements. These files are located in C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Settings and were created using Report Generation Language (RGL). The file name extension used for these files is .fmt.

To help determine which report to select, report files are listed in the Reports dialog box by description of output, instead of by file name. Table 2-6 lists the formats, file names, and their descriptions.

Table 2-6. Customizable Reports

Report	Format File	Report Description
Net List w/o pin info	netlist.fmt	Signals by netname without pin information
Net List w/pin info	netlistp.fmt	Signals by netname with pin information

Table 2-6. Customizable Reports (cont.)

Report	Format File	Report Description
Parts List 1	parts1.fmt	Parts by reference designator
Parts List 2	parts2.fmt	Reference designator by part type
Test points report	testpnts.fmt	Test point locations and netname
Jumper List	jumpers.fmt	Jumper locations and netnames
PADS Format Netlist	padsnet.fmt	Netlist in current PADS Layout format
PADS Format Netlist	padsnetV2.fmt	Netlist in PowerPCB 2.0 format
DFT Extended Test Point	testpoint.fmt	Test points by nets, nets without test points, and number of test points per net **Note:** You must have the DFT Audit licensing option to select DFT Extended Test Point.

Jumper List Report

You can create a Jumper List Report that lists all jumpers and their characteristics. Table 2-7 shows a sample report file.

```
JUMPER LIST REPORT -- a b c d e f g h i j.pcb
TOTAL = 3 jumper(s)
```

Table 2-7. Jumper List Report

Ref.Nm	Angle	Length	X1	Y1	X2	Y2	Signal
JMP1	90.000000	350	1825	3200	1825	3550	GND
JMP2	0.000000	250	2600	3275	2850	3275	GND
JMP3	90.000000	525	2725	2400	2725	2925	DA01

Report Generation Language

You can create report formats using Report Generation Language (RGL). This language uses keywords for the information to extract, such as parts and pin numbers. Additional keywords control the appearance of the report file.

Use a text editor to create or modify format files. You must use the default .fmt extension and save the file to C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Settings. Before you can use a report format you created or modified, add the format file to the list in the Reports dialog box.

Searches and Loops

The FOR statement searches for information in a design. The FOR statement acts as a loop to continue searching until all of the specified items are found.

The FOR statement can search for specific information, such as components attached to VCC, or FOR can combine with the ALL, find all, statement. You can also embed wild card characters in the text string.

The following examples use FOR statements with a SIGNAL keyword:

```
FOR VCC SIGNALS
FOR ALL SIGNALS
FOR V* SIGNALS
```

The keyword that specifies what information to search for immediately follows the FOR statement. The first example searches for all components attached to signal VCC. This information is called the *body* of the statement and is enclosed by opening and closing braces ({ }). The format to use is explained below.

You can classify searches as either a single-level search or a multi-level search. Use a single-level search for a simple component count. You can only access certain types of information by first searching for the *parent*, or top level information. For example, you cannot obtain the pin numbers attached to a specific signal unless you first search the signal name. The FOR statement uses nesting to accomplish this. For more information, see “Nested Loops” on page 44.

Format File Structure

The basic file structure of a single-level search follows:

```
Menu name Keyword
# comments (This report file lists all components)
Header information to include in report
FOR statement for loop
{
body of statement
}
```

A pound sign (#) preceding any text denotes a comment string. The pound sign does not appear in the report but the comment will.

Place keywords in uppercase letters for recognition. Unrecognized text appears in the report file as is.

All text and database item fields placed in braces are considered part of the loop.

Spaces between text strings and data items in the format file appear in the report file.

Table 2-8 shows an example of format file structure that creates a report containing a list of all components along with the package type. Keywords appear in bold.

Table 2-8. Example Format File Structure

Format File Structure	Description (not part of format file)
MENUTXT Component List	Text for list
# Component report	User comment
Parts List Report	Header information
Ref Des(no space)Part Type	Header information
FOR ALL **COMPONENTS**	Search all components
{	Open statement
COLUMNS 0 10	Format in columns
COMPNAME PKGNAME	List component and package names
}	Close statement

The above format generates the following report:

```
Parts List Report
Ref Des      Part Type
C1           CL25,10UF
C10          CK05,.01UF
U1           74139
```

Nested Loops

The nested loop structure is the same as the single-level search loop structure, but nested loops are contained completely within the loop. This includes the FOR statement, as well as the body of the statement which is enclosed by its own braces to define the inner level search.

The following format is an example of nested loop:

```
MENUTXT Net List Output File
# Company XYZ required output format
Net List Report
FOR ALL SIGNALS
{
SIGNAME
FOR ALL PINS
   {
   MAXCOLS 5
   BETWEEN 8
   DELIMIT ,
   COMPNAME, PINNAME
   }
}
```

Tip: The inner loop and its associated braces are indented here only for ease of reading. For information, see "Field Keywords" on page 48.

Keywords

Table 2-9 shows keyword types to use to handle single-level and multiple-level searches. Tables 2-11, 2-10, and 2-12 list the actual keywords.

Table 2-9. Keyword Types for Single-Level and Multiple-Level Searching

Keyword	Description
Top level keywords	Access data items that exist as single entities. You can use top level keywords at any time.
Sublevel keywords	Search for items which exist as multiple entries and may need a nested FOR loop.
Field keywords	Produce a column-style format when used within the body of the loop. Each field keyword is followed by a space and the appropriate values.

Top Level Keywords

Table 2-10 lists top-level keywords to use when searching.

Table 2-10. Top-Level Search Keywords

Keyword	Returns
JOBNM	Job name
TIME	Current date and time
LAYERCNT	Total number of layers
PKGCNT	Total number of packages
SYMCNT	Total number of symbols
COMPCNT	Total number of components
EQUIV_IC	Equivalent IC count
BD_DENSITY	Board density in current units
SIGCNT	Total signal count
PSIGCNT	Power net signal count
SSIGCNT	Signal net count
BOARDSZ	Board size in current units
TOPCOMPCNT	Total number of components on top layer

Table 2-10. Top-Level Search Keywords (cont.)

Keyword	Returns
BOTCOMPCNT	Total number of components on bottom layer
PADCNT	Total number of pads
DRPADCNT	Number of drilled pads
NDPADCNT	Number of undrilled pads

Sublevel Keywords

Table 2-11 lists sublevel keywords to use when searching.

Table 2-11. Sublevel Search Keywords

Keyword	Returns
SIGNALS	Search within signals
SIGNAME	Name of signals
PINS	Search within pins
COMPNAME	Component name
PINNUM	Pin number
PINTYP	Pin type
TPASSIGNED*	Whether test points exist in the signal: YES or NO
TESTPOINTCNT*	Number of all test points in this signal
TESTPINCNT*	Number of test point pins in this signal
TESTVIACNT*	Number of test point vias in this signal
SIGWITHTP*	Search for signals with test points
SIGNOTP*	Search for signals without test points
SIGNAME	Name of signals
PINS	Search within pins
COMPNAME	Component name
PINNUM	Pin number
PINTYP	Pin type
CONNECTIONS	Search connections
SIGNAME	Name of signals
COMP1	Reference designator of connection end 1
PIN1	Pin number of connection end 1

Table 2-11. Sublevel Search Keywords (cont.)

Keyword	Returns
COMP2	Reference designator of connection end 2
PIN2	Pin number of connection end 2
ROUTSEGS	Search within routes
END1	First endpoint of segment
END2	Second endpoint of segment
WIDTH	Width of route segment
LAYER	Layer number of route segment
VIAS	Search within VIAs
LOCX	X-coordinate of VIA
LOCY	Y-coordinate of VIA
VIANAME	VIA name
PACKAGES	Search within packages
PKGNAME	Package name
PKGDSCR	Package description
COMPONENTS	Search within packages
COMPNAME	Component name
SYMNAME	Symbol name
PKGTYPE	Package name
ANGLE	Component placement angle
LOCX	X-coordinate of placement
LOCY	Y-coordinate of placement
TESTPOINTS	Search within test points
TPNAME	Test point name The test point name for component pins is the standard pin name, such as U1.2. The test point name for vias is the via type name, such as STANDARDVIA.
SIGNAME	Signal (net) name When the test point is on an unused component pin, *NONE* is returned.
LOCX	X coordinate for test point
LOCY	Y coordinate for test point

Table 2-11. Sublevel Search Keywords (cont.)

Keyword	Returns
TESTSIDE	Testing side for the test point: TOP or BOTTOM

* You must have DFT Audit to create a report using keywords marked with asterisks.

Field Keywords

You can produce columns by using the keywords shown in Table 2-12within the body of the loop.

Table 2-12. Field Keywords

Keyword	Action
MAXCOLS	Determines the number of columns to use in the current loop.
LEADING	Specifies the starting character position for the first field.
BETWEEN	Specifies the number of characters to use as a separator between the first text character of each column.
COLUMNS	Specifies the starting print position for each column of data.
DELIMIT	Distinguishes keywords as entities separate from other fields, by delimiters which are usually spaces, but can be a character other than a space, such as a period.

Chapter 3
Editing Basics

Most design work involves editing the database and adding, changing, and deleting items. This chapter introduces selecting objects, and basic editing commands.

Selecting Objects

There are two ways to edit: select mode, where you select the object to edit first and then select the command, and verb mode, where you select the edit command first and then select the object to edit.

There are several ways to select objects:

- One at a time
- Several objects at once
- All objects within an area
- All objects of the same type

To select one object, place the pointer over the object and left-click. The object you selected appears highlighted. Any previous selection of objects is cleared. If you click over empty space, all previously selected objects are cleared. To select an object in a dense or crowded area, use the Selection Filter to disable other items from selection.

To select several objects, press and hold Ctrl while you left-click at each item you want to select. Any object not previously selected is added to the set of selected objects. Any object that was previously selected is removed from the set of selected objects.

To select all objects in an area, hold the left mouse button down and drag a selection rectangle around one or more objects; start at one corner of the area and drag to the diagonally opposite corner. When you release the button, all objects within the rectangle are selected. You can add additional objects to the selection or remove objects from the selection using Ctrl+click. The Drag Moves option can affect your ability to select by area in dense designs. If an object is selected and starts to move when you select an area, right-click and click Cancel and try starting in a different area. To disable drag moves, set the Drag Moves area to No Drag Moves.

Table 3-1 shows how to use Shift or function key combinations to automatically select multiple items.

Table 3-1. Selecting Using Shift and Function Keys

Item to select	Key combination
Pin Pair (traces, connection, and both pins)	Shift+click trace or connection.
Whole Net (traces, connections, and pins)	F6+click a trace, pin, or connection.
Whole Net (pins only)	Shift+click a pin.
Multiple Trace Segments	Click the first segment, then Shift+click the last segment. All segments in between are selected.
Whole Drawing Shape	Shift+select an edge.

Table 3-2 shows how to extend the selection of currently selected objects.

Table 3-2. Extending the Selection of Selected Objects

Additional items to select	Command
Select Pin Pairs from selected pins	Right-click and click **Select Pin Pairs**, or press **F5**.
Select Pin Pairs from selected traces	Right-click and click **Select Pin Pairs**, or press **F5**.
Select Nets from selected pins	Right-click and click **Select Nets**, or press **F6**.

Table 3-2. Extending the Selection of Selected Objects (cont.)

Additional items to select	Command
Select Nets from selected pin pairs	Right-click and click **Select Nets**, or press **F6**.
Select Nets from selected traces	Right-click and click **Select Nets**, or press **F6**.
Select Associated Nets from selected nets	Right-click and click **Select Associated Net**.

Controlling Selections

Sometimes you cannot easily select the object you want because there are several objects at the same location. Use the Selection Filter or cycle picking to solve this problem.

Sometimes you want to find and select objects without using the pointer. Use the Find command to accomplish this. For more information, see "Finding Objects" on page 51.

Selection Filter

When you cannot easily select the object you want because there are several objects at the same location, you can use the Selection Filter (Filter on the Edit menu) to solve this problem. The Selection Filter has two tabs: the Object tab and the Layer tab. Use the Selection Filter Object tab to specify objects that you can or cannot select. Use the Selection Filter Layer tab to specify selecting from designated levels.

Cycle Picking

When you cannot select the object you want because several objects occupy the same location, use cycle picking (Cycle on the Edit menu).

Finding Objects

Use Find to find and select objects by attribute, keepouts, physical design reuse, test point types, and copper pour types.

If you click Test Point Types in the Find By list, you can search by Via, Component pin, or Net in the Test Point Types list box.

As shown in Table 3-3, the find command works two ways, depending on how you select objects:

Table 3-3. Find Modes

Find mode	Description
Select mode	Find ignores the Selection Filter settings and selects whatever you ask it to.
Verb mode	Find only looks for items that are logical for the command.

Cut, Copy, and Delete

Use the Cut command to remove selected items and place them on the Windows clipboard. Use the Copy command to place a copy of selected items on the Windows clipboard.

Use the Paste command to place the most recent contents of the clipboard anywhere on the design. You can paste the same copy repeatedly; it remains on the clipboard until you overwrite it with a new copy or cut action. You can also paste into a different design.

Cut, Copy, and Paste support attributes; meaning that when you cut, copy, or paste objects their attributes are cut, copied, and pasted with them. Attribute labels are also cut, copied, and pasted with the objects to which they are assigned.

Cut, Copy, and Paste also support physical design reuses; however, you can copy only one physical design reuse at a time. When you copy a physical design reuse and paste it into a different design, the reuse file is compared against the current design to detect possible reference designator, layer, decal, and netname conflicts. This comparison will also detect other errors and warnings, as described in “Adding a Physical Design Reuse” on page 154.

The Copy and Paste commands work in several different modes, depending on what is selected when you begin to copy and where you paste the copy.

Tip: When you paste a group, an error may appear in one of the Trace Copy dialog boxes. For more information, see the Trace Copy Dialog Box topic in the Reference Guide. You can only paste items in DRC Off mode.

Setting the Origin for Items in the Clipboard

The default origin for the Clipboard contents appears in the lower left corner of the area that encompasses all information in the clipboard. You can set the origin at a user-specified location. For more information, see the Setting the Pointer Location for Items in the Paste Buffer topic.

Selection Preferences for Copy

There are two options in the Design tab of the Options dialog box that handle specific selection and data importing situations.

- When items are copied to a new .pcb database you can set Keep Signal and Part Names to retain the netnames of traces and the reference designators. However, if you paste into an existing design and these netnames or reference designators already exist, incoming netnames or reference designators are sequentially updated. The default is to automatically generate new default reference designators and netnames when you paste to a new file.
- You can set Include Traces Not Attached to include all routing patterns within the selection rectangle. The default is to not include in the copy any routing in the rectangle that is not attached to a selected part.

Copy Traces, Traces and Vias, or Routed Pin Pairs Only

When you select and copy a trace or a trace and via combination, you start a specialized operation that immediately attaches a copy of the selected routing to the pointer and lets you repeatedly paste with a mouse click.

Capture Area

You can use the Capture Area command to define a rectangular area from which to copy graphics information to the Clipboard as a bitmap image, which you can use in other files and documents.

Copying and Pasting in ECO

Any addition to the design that would force updating the netlist or part list must take place in ECO mode. Use the ECO Options dialog box to target and direct the .eco file. If something is in the buffer that is not in the current part list or netlist, and you try to paste outside of ECO mode, PADS Layout prompts you to enter ECO mode to complete pasting.

Copy Multiple Selections

You can copy and paste multiple or mixed selections, or different item types on different layers using a selection rectangle. You can also build extended selections. For example, you can use a selection rectangle to copy an SMD part, its via fanouts, and an associated dimensioning item that was assigned to layer 25. You can only select items that appear on screen. You can copy the selected items to the Clipboard, retaining their layers and relative positions.

Paste Multiple Selections

When the Clipboard holds multiple selections comprised of some ECO and some non-ECO registered items, Paste operates differently depending on whether you are in ECO mode. If you paste in ECO mode, all items can be pasted. Part reference designators are added sequentially. If you are not in ECO mode, you can paste only non-ECO registered items.

Delete Command

Like the Copy and Paste operations, the Delete command is sensitive to whether you are in ECO mode. When the ECO Toolbar is not open, you can press Delete to unroute selected segments, pin pairs, or nets, leaving the connections intact.

Tip: You can't delete physical design reuses that contain glued components or protected routes.

Delete in ECO Mode

When the ECO Toolbar is open, you can use Delete to completely remove parts or nets from the design. You can remove any non-ECO items like copper, lines, or text normally. You can also unroute routed traces normally. If you delete a group of ECO and non-ECO items, or combined ECO items (parts and pin pairs or nets), confirmation prompts, one for each item, will appear sequentially.

Step and Repeat

The Step and Repeat tool defines complex, repetitive array patterns so that the fanout of traces from a component on a Device Under Test PCB is consistent, ensuring simulation and testing of a device under exacting conditions.

Step and Repeat arranges terminal, drawing, copper, cutout, or text items in a planar or polar array pattern. You can replicate multiple or single items.

Step and Repeat also automatically increments text. You can create an array using Step and Repeat in the PCB Decal Editor for terminals, drafting items, text items, or group selection.

Linear Step and Repeat

Figure 3-1 how to set up a linear step and repeat. Figure 3-2 show the results.

Figure 3-1. Initial Setup for a Linear Step and Repeat

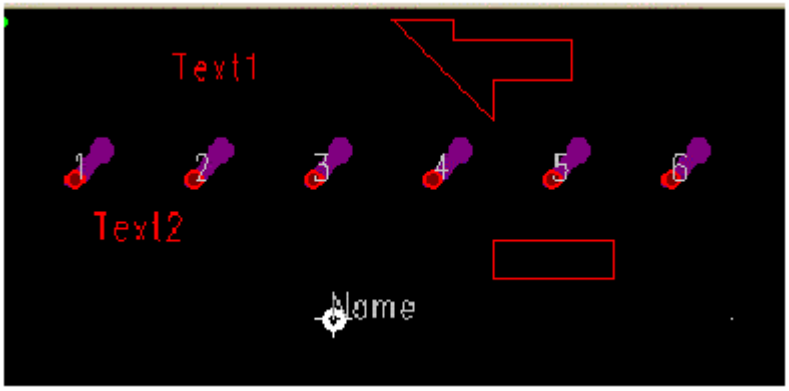
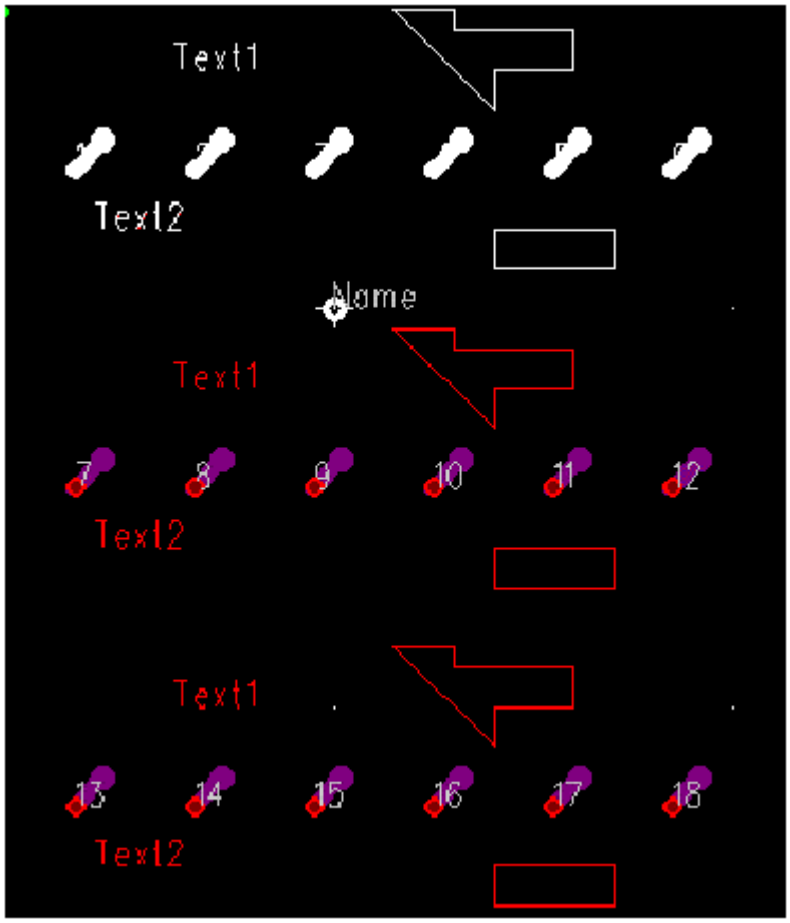


Figure 3-2. Results after Performing a Linear Step and Repeat



Polar Step and Repeat

Use the Polar tab in the Step and Repeat dialog box to create angular, or circular, arrays (Figures 3-3 and 3-4).

Figure 3-3. Initial Setup for a Polar Step and Repeat

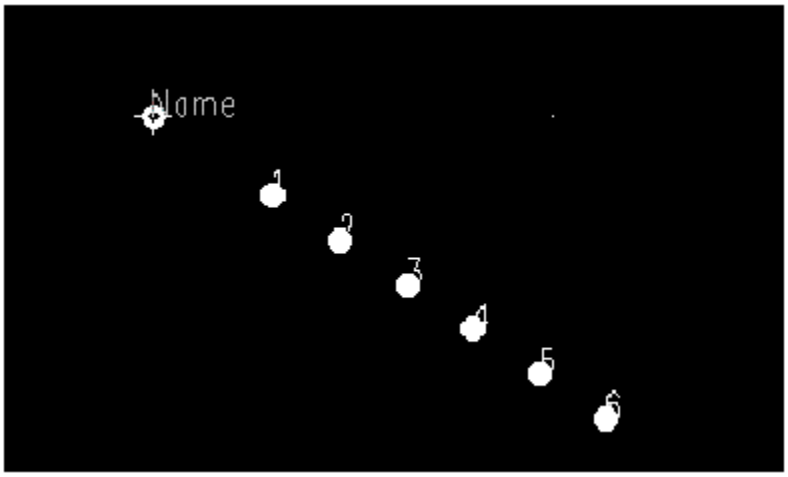
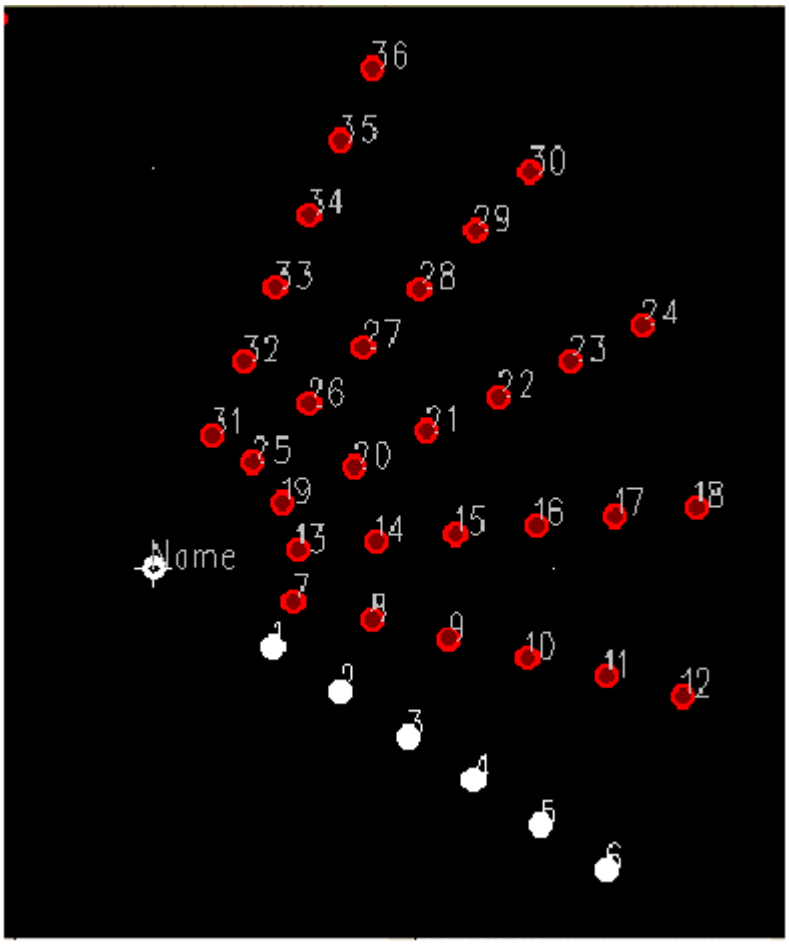


Figure 3-4. Results after Performing a Polar Step and Repeat

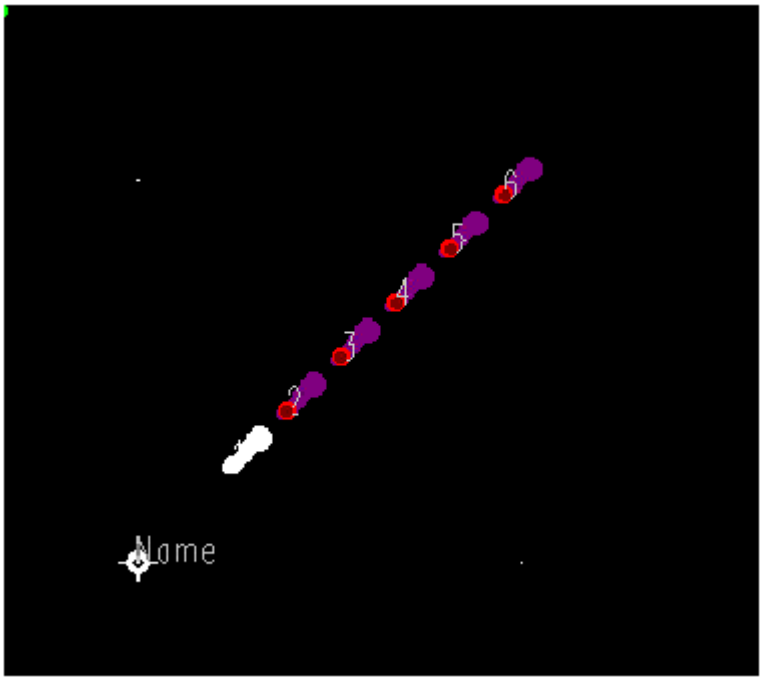


Radial Step and Repeat

Use the Radial tab in the Step and Repeat dialog box to create radial arrays (Figures 3-5 and 3-6).

Figure 3-5. Initial Step for a Radial Step and Repeat with Associated Copper

Figure 3-6. Results after Performing a Radial Step and Repeat with Associated Coppers



Editing in the PCB Decal Editor

Editing Decals

PADS Layout uses components from the parts libraries. Use the PCB Decal Editor to create and edit the decal associated with a part type in the parts library. Many PCB Decal Editor drafting operations are identical to Layout Editor drafting operations.

When the PCB Decal Editor starts, your current design is stored and PCB Decal Editor takes over. The colors used in the PCB Decal Editor come from the Layout Editor. When you exit the PCB Decal Editor, use File commands to save information, and exit as you would a stand-alone program. You will return to the current design in PADS Layout.

When editing decals, note the following:

- PADS Layout supports 16 alternate decals per part type.
- You can use Dimensioning within the PCB Decal Editor; however, dimensions are converted to 2D lines and text when you save the decal. For more information, see the Dimensioning Process topic.

PCB Decal Editor drafting works with the objects that make up a decal.

- Decal name
- Terminals
- 2D lines
- Text
- Copper
- Copper voids

Renumbering Terminals

When renumbering terminals, remember the following:

- To undo the renumbering of a terminal, right-click and click Back or press Backspace to restore the original number. If you selected a group of terminals to renumber, Back will undo the renumbering one terminal at a time.
- To cancel the terminal renumbering process at any time, right-click and click Cancel. PADS Layout restores the original terminal numbering.
- To undo completed terminal renumbering click Undo from the Edit menu.

To renumber multiple terminals, click and hold the left mouse button, then drag the pointer across the terminals while pressing Shift. When you release the mouse button, all of the selected terminal numbers update, excluding terminals that are already renumbered.

Creating Keepouts in the PCB Decal Editor

You can create keepouts in the PCB Decal Editor. Decal-level keepouts are similar in appearance to other keepouts. When you create a decal-level keepout, any part using the decal in the Layout Editor uses the specified keepout restrictions. You can create a decal-level keepout that restricts Trace and Copper, Copper Pour and Plane Area, Via and Jumper, and Test Points. You cannot create keepouts for placement, component height, or component drills. You create a keepout in the PCB Decal Editor the same way you create one in the Layout Editor.

You cannot move a decal-level keepout independently of the part to which it belongs. Once you create a decal-level keepout, you must enter the PCB Decal Editor to modify any properties of the keepout. You modify keepouts in the PCB Decal Editor exactly as you would in the Layout Editor.

For more information on creating and modifying keepouts, see the Using Keepouts topic.

Layers for Decal-Level Keepouts

To create keepouts assigned to Inner Layers (any layer other than top or bottom), you must increase the number of layers to three.

Decal-level keepouts use the same layer assignments as keepouts created in the Layout Editor, plus an additional option for Opposite Layer. Opposite Layer assigns restrictions to the side opposite the one on which you place the component. For example, while editing a decal, create a copper pour keepout and choose Opposite Side from the Layer list. In the Layout Editor, and with the component mounted on the top side, the keepout prevents copper pour on the bottom layer. If you click Flip Side to place the component on the bottom layer, the keepout prevents copper pour on the top layer.

Modifications to Components

When you add a component from the library to a design, the component is copied from the library into the design as a separate instance. There are then two versions of the component—the library version and the design version. You can modify either or both of these; but which one you modify makes a difference:

- If you want the changes to apply only to the Library version of the component , use the PCB Decal Editor to make the changes. If you modify the Library version of a component, only that version is changed; instances of the component existing in the design are not changed. (If you *want* to change the design components to match the

Library version, you can do that using **Tools > Update from Library**.)

- If you want the changes you make to apply only to the component(s) in the current design, use the Layout Editor to make the changes. If you modify the design version of a component, the change affects only the selected component(s) in the current design (or all instances of that component in the design); *the Library definition of the component is not changed* (unless you specifically change it using **Right-click > Save to Library** with the component selected in the design.).

Most of the items that make up a component definition can be edited only in the Layout editor, or only in the PCB Decal Editor, not in both places. But the following items can be edited in either editor, and you must decide which editor to use:

- In the component part type:
  - Attributes
- In the component's decals:
  - Attributes
  - Labels
  - Silkscreen
  - Keepouts
  - Copper
  - Solder mask
  - Paste mask
  - Part outline width
  - Decal rules
  - Pad stack

Chapter 4
Designing

This chapter covers how to set up PADS Layout to fit your work style and preferences. As part of customizing your application, you will also learn how to use macros and Sax Basic scripts to work faster.

Design Operations

When you start PADS Layout or use the New command, a drawing format is automatically added to the work area. Once you have a new design, you can set the colors to use, layer modes, add parts, set up pad stacks, and route design information.

To add new parts to the design, use the ECO Add Part Command. For more information, see Adding a Component in ECO Mode.

To create and edit libraries, use the Library Manager. For more information, see Creating a Library, or Setting Library Availability and Search Options.

To manipulate and route design information, use the Drafting Toolbar or the Design Toolbar.

- The Drafting Toolbar adds drafting items such as the board outline, copper, keepouts, and planes.
- The Design Toolbar allows easy access to routing and placement tools.

Database Limits

As of PowerPCB 4.0, the database limits were increased. Table 4-1 shows the old and new database limits:

Tip: If your design uses these new limits, you may not be able to export it into a PADS-format ASCII file compatible with a previous version of the program. For more information, see "ASCII Format" on page 305.

Table 4-1. Database Limits

Description	PowerPCB version 2 and 3	PowerPCB version 4 and PADS Layout
Drawing items per design	16,777,216	Same
Drawing pieces per design	16,777,216	Same
Corners per design	16,777,216	Same
Arcs per design	16,777,216	Same
Text strings per design	32,768	Same
Text length per design	16,777,216	Same
Pieces per drawing	32,768	Same
Corners per drawing	16,777,216	Same
Arcs per drawing	16,777,216	Same
Text strings per drawing	32,768	Same
Text length per drawing	16,777,216	Same
Corners per piece	32,768	Same
Arcs per piece	32,768	Same

Table 4-1. Database Limits

Description	PowerPCB version 2 and 3	PowerPCB version 4 and PADS Layout
Drawings per PCB decal	32,768	Same
Reference designator characters	15	Same
Components per design	32,768	16,777,216
Terminals, via types, and jumper types per design	32,768	16,777,216
Gates per part type	100	702
Gates per design	32,768	16,777,216
Pin pairs per design	32,768	16,777,216
Nets per design	32,768	16,777,216
Alphanumeric pin numbers per design	32,768	16,777,216
Alphanumeric pin number length	4	7
Number of layers	30	250
Number of electrical layers	30	64
Number of pins per component	2000	32,768
Decal name length	40	Same
Part type name length	40	Same
Net name length	47	Same
Layer name length	31	Same
Component rotation precision	0.001×	Same
Pad rotation precision	0.01×	Same
Generic polylines angle precision	0.1×	Same
ASCII file line length	2307	Same
Attribute name length	256	Same
Attribute value length	2048	Same

Tip: Consider all limits of 32,768 and 16,777,216 as formal. Actual limits may be smaller due to memory limitations.

Color Maintenance

To manage colors in a design, use the Display Colors Setup dialog box. Use the Assign All button, or row and column selections to make the following color changes in a design:

- Change the color of all objects on the same layer to a user-selected color.
- Change the color of the same object type on all layers to a user-selected color.
- Change the color of the same object type, when it is currently set to the background color, on all layers, to a system-assigned color.
- Change the color of all objects, when they are currently set to the background color, on the same layer, to a system-assigned color.
- Change the color of all objects, when they are currently set to the background color, on all layers.
- Change the color of all objects, on all layers, to the current background color.

Changing the Color of All Objects on the Same Layer

To change the color of all objects on the same layer, use Display Colors Setup dialog box. Click a layer number to select a the whole row of a layer. In the Selected Color area, select the color tile. The program assigns the color you selected in the color palette to all objects on the selected object's layer.

Changing the Color of an Object Type on All Layers

To change the color of the same object type on all layers, use the Display Colors Setup dialog box. Click and object type column to select the object color on all layers. In the Selected Color area, select the color tile. PADS Layout assigns the color you selected in the color palette to the same object type on all layers.

Making Objects Visible

To make objects visible, change the color of background-colored objects by using the Display Colors command on the Setup menu. Click Assign All in the Color by Layer area of the Display Colors Setup dialog box to open the Assign Color to All Layers dialog box.

To change the colors of objects that are currently set to the background color, use the Automatically Make Objects Visible options. To assign the current background color to all objects on all layers, use the Assign Background Color to All Objects option.

For more information on any of these color preference options, see the Making All Objects Visible topic.

Changing Object Type Color

One Color Per Object Type assigns color for a certain object type, when it is currently set to the background color, on all layers. PADS Layout assigns color according to the color set in the immediately adjacent tile, or, if no adjacent tile exists, according to color palette order.

Changing Layer Color

One Color for a Layer assigns color for all objects, when they are currently set to the background color, on the same layer. PADS Layout assigns color according to the color set in the immediately adjacent tile, or, if no adjacent tile exists, according to color palette order.

Changing to the Selected Color

Selected Color assigns the color you select from the Display Colors Setup dialog box color palette to all objects, which are currently set to the background color, on all layers.

Additional Options

You can also select Update Visibility check boxes and Update Enabled Status along with any of the Color Preferences options. Update Visibility check boxes control the processing of the visibility check boxes around the Display Colors Setup dialog box color matrix. When you select this option, the visibility check boxes surrounding the color matrix are checked if data exists on a layer, and unchecked if no data exists on a layer.

Update Enabled Status specifies color assignment based on layer settings in the Enable/Disable Layers dialog box. This dialog box is accessible from the Layers Setup dialog box. To assign colors to enabled, nonelectrical layers containing data, select Update Enabled Status. Layers that do not contain data are disabled. For more information, see the Hiding or Displaying Non-electrical Layers topic.

Making All Objects Invisible

To make objects invisible, change the color of all objects, on all layers, to the background color. This option is helpful because it is a quick method for clearing the display.

Choose the Display Colors command on the Setup menu, then select the color you want for the background color. Choose Assign All in the Display Colors Setup dialog box, then choose Assign Background Color to All Objects. All objects become invisible. You can then use the Automatically Make Objects Visible options to make objects visible.

For more information, see the Making Objects Invisible in the Display topic.

Font Selections

Text strings and labels in your designs can use stroke font and/or system fonts that are installed on your system.

- You can set fonts for each text string and label you create in your design, choosing stroke font or system fonts for each selection. You can also have a combination of stroke font and system fonts within the same design.
- You can search for fonts used for text strings or labels, and can then open the item Properties to apply a different font name and style to all objects that you select for modification.
- Designs to be output to printers and plotters can also use both stroke fonts and system fonts. When making font selections, consider the following:
  - System font text is supported in RS-274X Gerber format when Fill mode is on and is output as a set of filled polygons unless.
  - System fonts are not supported in the RS-274D CAM output format. If you attempt to use this output format with system fonts, the program displays a warning message. If you proceed, system fonts will not be output. Instead, you should use the RS-274X format with system fonts.
- If the design uses fonts or character sets that are not installed on your system, a font substitution process begins automatically when the file is loaded. During this process, you are asked to choose fonts to substitute for those that are missing from your system.
- For systems using languages that do not include stroke font, English stroke font is used.
- Non-ASCII symbols, such as +/-, ohm, and degrees are available on your system for the installed fonts you select. If the character sets you choose are not available, a blank space or blank text box appears where the symbols should be. In this case, choose character sets that are available on your system to enable the symbols to display in your design.

When you open an existing design that was created on a system without system font supported, you must choose whether to use stroke or system fonts for every text string or label in the design. To convert the existing text or label fonts to another font, use the Properties dialog box. To find fonts easily, use the Find dialog box.

Managing Font Substitutions

If you open a design that uses fonts or character sets that are not installed on your system, empty boxes appear where you expect to find text or symbols, and the font replacement dialog box appears.

You can replace missing fonts automatically or manually, or you can skip the replacement process for fonts you identify. When you choose to replace fonts manually, you are asked to

confirm your font selections before the replacement process initiates. If you choose to skip font replacement, you do not confirm your selections to start the replacement process, or if you cancel the replacement, the loading of the design for display on your system is cancelled, and the original design file is preserved.

Layers

Layer Modes

PADS Layout supports two layer modes: default layer mode (30 layers) and increased layer mode (up to 250 layers). In default layer mode, the 30 layer maximum can consist of up to a maximum of 30 electrical layers or a combination of electrical and nonelectrical layers. In increased layer mode, the maximum number of electrical layers is 64 and the maximum number of nonelectrical layers is 186, for a total of 250 layers. The total number of layers includes associated layers such as mask, silkscreen, drill drawing, and assembly layers.

You change from default layer mode to increased layer mode by clicking the Max Layers button in the Layers Setup dialog box. Changing from default layer mode to increased layer mode increases all nonelectrical layer numbers by 100.

Tip: Once you change the design to increased layer mode, you cannot return to default layer mode.

In default layer mode, layer number 20 is used for placement outlines. In increased layer mode, layer 120 is used for placement outlines. Layer 25 in default layer mode, or 125 in increased layer mode, is used for oversizing thermals and antipads.

In default layer mode, you can only import, add, or load other default layer mode items, such as files or library items, to your design. You cannot load increased layer mode objects into the default layer mode design. In increased layer mode, you can load both default and increased layer mode objects into your design.

Tip: You cannot export a design with more than 30 electrical layers to a PADS Layout PADS-format ASCII file prior to version 4.0.

You don't have to convert existing default layer mode libraries, reuse files, or archived designs to increased layer mode. You can use existing library decals, drawings, and reuses that are saved in default layer mode for both default and increased layer mode designs. You can cut from a design in default layer mode and paste into a design in increased layer mode. If you make sure that you have consistent layer definition in libraries, reuses, and designs, no problems with layer matching will occur.

Either default or increased layer mode is specified in each design or design fragment such as .pcb, .asc, .stp, .dxf, .reu, and high speed .edp files, library decals and drawing items, copy/paste buffer, and external CAM documents.

Objects Associated with Layers

The following layer properties are changed during layer reassignment:

- Name
- Plane status
- Layer (routing) direction
- Dielectric constant and thickness (associated with upper copper layer)
- Associated nonelectrical layers (for top and bottom)
- List of assigned nets
- Colors, including outline colors

Tip: Because both layer names and colors are reassigned, an object remains in the same color after reassignment.

The following objects have an assigned layer and are reassigned during layer reassignment:

- Traces
- 2D lines, free and in decal (part outlines)
- Copper, open and closed, free and in decals, but not pin-associated copper
- Keepouts, free and in decals
- Copper cutouts, free, associated, and in decals
- Texts, free, combined with 2D lines, and in decals
- Attribute labels
- Pour outlines and plane areas
- Pour or plane area hatch outlines and hatch voids
- Conditional clearance rules
- Layer mask in routing rules

The following objects have an assigned layer and are *not* reassigned during layer reassignment:

Tip: Before deleting an electrical layer, make sure that you first delete all of the following objects from, and any references to, the layer.

- Pad definitions on absolute layers in pad stacks (for decals and vias)
- Pin-associated copper in decals
- Start and end layers for partial vias
- Drill pairs

Associating Component and Documentation Layers

Use Associations in the Layers Setup dialog box to associate layers. Layer association outputs text, line, or shape items on manufacturing plots for the component layers, top or bottom, without requiring you to enter the items directly on the layer. For information, see "TrueLayer and Layer Associations" on page 69.

When you associate a documentation layer dedicated to a manufacturing plot type with a component layer, the CAM output process automatically includes the associated documentation layer items with the plot type. You can associate the following layer types with a component layer: Paste Mask, Solder Mask, Silkscreen, and Assembly.

A list below each plot type lists each documentation layer you designated for the plot type. You may have more than one documentation layer, but you can only associate one layer at a time for automatic CAM output with the manufacturing plot type.

Layers 23 to 29 are default documentation layers for manufacturing plot types, including two for silkscreen: top and bottom.

When you create a CAM document for a silkscreen type, Layer 7 is automatically included in the document description. The same applies to the other manufacturing plot types.

TrueLayer and Layer Associations

When you select TrueLayer and flip a part to the opposite side, you do not have to modify the original part pad stacks. CAM information is flipped with the original part. You do not need to define a bottom paste mask or solder mask for SMDs since these parts exist on one layer at a time. For through-hole solder mask pad definitions, you can define unique pad shapes for each side.

If the solder mask on through-hole parts is identical on both the top and bottom side of the board, define the solder mask top pad definition, and you can associate the same solder mask layer to both the top and bottom.

If you want unique solder mask shapes for though-hole parts mounted on the bottom side of the board or for test point vias on the bottom side of the board, define the solder mask bottom layer in the part pad stacks and then use the associations for each layer. In all cases, the correct documentation layer is included automatically in your plot when you define the masking plot documents in CAM.

To clear the TrueLayer option, see the Starting PADS Layout topic for instructions on command line options.

Copper Pours and Plane Areas

Connecting Nets to a Copper Plane

If you are setting up a copper area as a power or ground plane, and you plan to dedicate the entire level to the plane, you don't need to draw the copper area. Instead, use the Layers Setup dialog box to define the level as a dedicated plane layer. Then assign the netnames that will connect to the plane layer. With this method you can include more than one netname to join to the plane.

When you use this technique, PADS Layout creates a plane plot film for the layer that defines the entire level as copper. The border is defined by the board outline, and the interior includes the oversized antipad insulators and the thermal connections where the copper joins pins associated with the plane by netname. The Plane Thermal option in Properties dialog boxes controls the thermal generation for individual pins of a plane net and signals CAM and plane checking to generate a thermal relief for the pin. You can also run plane checking from Verify Design.

You can determine whether you are successfully tied to the plane. If you are routing from an SMD to a via and the View Nets default is set to Partial, the ratsnest connections from the vias won't appear.

Drawing a Plane or Using a CAM Plane

You can also draw a poured copper area as a plane that connects power or ground nets. Using this method, you don't have to assign the layer as Plane in the Layers Setup dialog box, you can leave it as a routing layer. Rather than having CAM produce the plane plot, you are actually drawing the copper area yourself; the film for this layer is output as a routing plot.

This is a more true-to-plot display; the thermal connections are visible as part of the copper display on that layer. You can also run other traces on the level as long as they don't disrupt the connectivity of the plane. The copper pour routine insulates traces from the plane copper when it pours the copper fill.

You can only associate *one* netname per copper area to automatically join to the plane. You aren't limited to one copper area per layer; you can draw two on a layer side by side, one net

named for power and one for ground. You also have to rehatch copper pour areas every time you load the file or repour every time you move same-level routing because these edits increase file size.

Differences Between Copper, Copper Pour, and Plane Areas

There are many differences among copper, copper pour, and plane areas. The following table explains the differences and may help you understand when to use one instead of the other.

Table 4-2. Feature Comparison of Copper, Copper Pour, and Plane Areas

Feature	Copper	Copper Pour	Plane Area
Can be filled without assigning a net	X		
Is filled automatically	X		
Can be created in the PCB Decal Editor	X		
Can be created on non-electrical layers	X	X**	
Can have thermals on routed pads		X	
Can only be created on a No Plane layer		X	
Must encompass an electrical object of the net to which it's assigned when using the *Remove isolated copper* option		X	X
Avoids objects of any net to which is isn't assigned		X	X
Has thermal spokes based on the Thermal Options		X	X
Can generate thermals and antipads from the design rules*		X	X***
Is not filled automatically but must be poured		X	X
Can only be created on a Split/Mixed layer with pre-assigned net(s)			X
Can generate custom thermals and antipads from pad stack settings*			X
Can automatically create cutouts around embedded planes*			X
Can remove unused pads on inner layers*			X
Can be automatically generated based on the shape of the board outline			X
Can be split into two areas using the Auto Plane Separate tool			X
Can save hatch outlines with the design			X

*These features are controlled by settings.

**Copper pours placed on a non-electrical layer must be flooded using the right-click menu.

***If you use the *Use design rules for thermals and antipads* setting in the Split/Mixed Plane Options, antipads use the Copper-to-Drill clearance rule rather than the Copper-to-Pad clearance.

Generating Thermals

Component pins and vias automatically receive thermals if they are associated with a plane net and if they have pads on the associated CAM plane layer. For component pins and vias that receive thermals, the Plane Thermal option is automatically turned on in the Pin Properties dialog box.

When you redefine a plane net so that it is not associated with a CAM plane layer, the thermals are removed for all of the pins in the net. The Plane Thermal option is automatically cleared in the Pin Properties dialog box.

If during routing you add a via to a plane net, the via automatically receives a plane thermal. The Plane Thermal option is automatically selected in the Via Properties dialog box.

You can control the thermal view using the Routing tab.

For information on how thermals translate to and from SPECCTRA, see the *SPECCTRA Translator Help*.

Check the Flood Priorities by selecting the shape, opening the Properties and clicking on the Options button. The Flood Priority numbers should be lowest for inner nested planes and greatest for the outer planes.

Symptoms of Poor Thermal Results

If the following symptoms occur, see Troubleshooting Plane Area Fills:

- Split/mixed plane or copper pour will not flood
- Split Plane will not flood
- Problems with split/mixed planes
- Unable to flood split plane or copper pour

CAM Planes

When you generate thermals for CAM plane plots, PADS Layout looks for plane netnames associated with CAM plane layers. PADS Layout also checks that pins or vias with pads on the CAM plane layer have the Plane Thermal option selected in the Pin Properties and Via

Properties dialog boxes. Use the Show General Plane Indicators option in Tools > Options > Thermals tab to display CAM plane thermals.

When a pin exists in a net that is associated with a CAM plane layer and Plane Thermal is selected for the pin, a thermal appears on the pin. The ratsnest connections still appear. Use plane check in Verify Design to verify thermal generation for CAM plane plots.

Copper Pour

Control of thermal generation for copper flood is based on unrouted connections. Unlike CAM planes, copper flood does not use the Plane Thermal option. PADS Layout looks not only for same netname, but for unrouted connections leading to through-hole component pins as points to connect with a copper pour thermal. If the pin is routed, for example, on a different layer, and is no longer showing a ratsnest connection, you will not generate a copper pour thermal relief. You must have an unroute to generate copper pour thermals.

For copper pour thermal generation around component pins, the unrouted connection rule holds true. When there is no unrouted connection, a copper pour thermal is not produced around the pin. When a route and unroute exist on a pin, a copper pour thermal is produced.

The rule changes for vias. During copper pour, vias will always get copper pour thermals if the netname of the via matches the netname of the surrounding copper pour area, whether they show unroutes or not.

After the copper is poured and the copper pour thermals are installed, the ratsnest connections still appear. Use Nets from the View menu, and then the Traces option to hide unroutes. Use Verify Design from the Tools menu to check connectivity for copper pour areas.

Design Rule Versus Pad Stack - Thermals and Antipads

There are two ways to apply thermal and antipad settings. One is connection-based and uses the design rule hierarchy to apply the settings. The second is component-based and uses the pad stacks to customize the thermals and antipads of component pins. You can use a combination of design rule and pad stack thermal and antipad settings.

Restriction: Copper pours only use the design rule thermals and antipads. Pad stack thermals and antipads only apply to Plane areas and CAM planes.

Design Rule Thermals and Antipads

Design rule thermals are defined by the settings of the Thermals Options and the Copper-to-Pad or Copper-to-Via clearance rules. In the Thermals Options you define the spokes, and using the clearance rules, you define the gap between the pad and the copper.

Design rule antipads are defined by the Drill to Copper clearance rule. You can use the hierarchy of the design rules to create different gaps between the pad and copper of thermals

and the drill and copper of antipads. For example, using the Net level of the Rule hierarchy, you could apply a larger gap to any pad connected to a particular net.

You can only apply separate spoke settings to drilled and non-drilled thermals. The spoke settings globally apply to all pads of either type. If you need to customize the spoke settings of individual pads, you should use the pad stack thermal and antipad settings.

Pad Stack Thermals and Antipads

The pad stack thermals and antipads offer full control over all aspects of thermals and antipads. Unlike the design rule thermals and antipads, you can control the spokes of individual pads. However, pad stack thermal and antipad settings don't apply to copper pour areas, only to Plane areas and CAM planes. If you apply pad stack thermals and antipads to decals in the library, those settings may not apply to different PCB designs. You may want to only apply the pad stack settings to instances of the decal in the design rather than the decal in the library.

Priority

Although you can use a combination of design rule and pad stack thermal and antipad settings, the pad stack settings take priority over the more global design rule settings.

Pad Sizes and Pad Stacks

A *pad* is a small area of copper that acts as a conductor for component pins and vias. The pad ensures connectivity between the trace entering the drill hole and the copper plating that lines the inside of the hole.

Pads are classified as two types:

- Through hole pads—are used for components that mount with pins that go through the board. When through hole pads are drilled and plated, a small ring of copper remains and ensures connectivity between the trace entering the drill hole, the copper plating that lines the inside of the hole, and the pad on the opposite side of the board. Vias are considered through hole pads, but may be created to go through only certain layers.
- Surface mount pads—are used for components that have pins, which are glued to an outside layer of the board. Routing to vias provides connectivity to other layers.

Pad Stacks

On a two-layer board, PADS Layout sees a component pin or via drill hole as having a separate pad on each end, the top layer and the bottom layer. You can set different shape, size, and diameter values for each one. PADS Layout can assign another, separate pad with its own size and shape for each layer the hole passes through. If you add any inner routing layers to the design, you can define pads on those layers. The resulting tree of two or more pads is called the *pad stack*.

Pad stacks are classified into two categories:

- Component pad stacks

Component pad stacks are used for component pins and are either surface mount, with no drill diameter, or through hole. Pad stack information for a component is stored with its part decal information.

- Via pad stacks

Via pad stacks are used for feed-throughs and can be through hole or partial. *Partial vias* begin or end on an outer or inner layer. Partial vias are used on multilayer boards and are created by drilling laminate layers separately for layer-dedicated vias, then pressing them together and drilling the through holes. If a via connects an outer layer and an inner layer, it is called a *blind via*. If a via connects two inner layers, it is called a *buried via*. The via type determines whether a via is through hole or partial. The via description is the combination of type, plating, and pad stack information.

You can edit pad stacks by layer, so you can set component or via pads to zero, turning them off, on layers where they are not needed to create more routing real estate around a drill hole. You can assign different shapes to them for different routing or photoplot applications. The resulting configuration of size, shape, diameter, and layer description for a pad stack is called its *pad stack information*. For more information, see the Editing Pad Stacks topic.

For information on installing vias, see the Via Setup chapter topics.

Pad Stack Default Layers

The default layer list features a special generic layer list. It lists Mounted Side, Inner Layers, and Opposite Side. This setup allows quick customization of the electrical layers and will be flexible if you choose to mount the component on the bottom side, or add extra inner layers.

Mounted/Opposite Side

If you customize the Mounted Side or Opposite Side, your customized pad will flip with the component if you choose to mount the component on the bottom side of the board. However, if you choose to customize the specific layer of your top or bottom side of the board, that customized pad will remain on that layer even though you flip the component.

Inner Layers

If you customize the Inner Layers, you can add layers to your design and your pad customizations will propagate to any added. However, if you choose to customize each specific layer, and you increase the layers of your design, your customized pads will remain only on those specific internal layers.

Pad Stacks and Antipad Definitions

Setting an antipad definition for the inner layer modifies the photoplot output for CAM planes and split/mixed planes. If you want a unique antipad on split/mixed plane layers, add a new layer for the split/mixed plane before defining the antipad.

Pad Stacks and Associated Copper

When defining associated copper for a terminal, you should define a zero size, square shape pad in the terminal pad stack on the layer of the associated copper. The zero size, square shape pad is interpreted by routing commands to be the routing target for the associated pad copper on that layer.

Drill Size

During manufacturing, the interior surfaces of drill holes are coated with metal plating. For vias, the plating enables connectivity when the layers are pressed together. Plating reduces the diameter of drilled holes. The size difference does not affect vias as much as component holes, where a smaller diameter can hinder part insertion.

PADS Layout assumes that the drill size you define for a pad stack is the finished hole size, after plating. Manufacturing should use a larger drill bit than the specified drill size specified so that once the plating is added, the resulting inner diameter is at, or close to, the original finished specification. So that you can use the actual drill bit size in clearance checking, PADS Layout has a universal drill oversize setting on the Design tab of the Options dialog box, which adds a fixed amount of diameter to all drill size definitions. The combination of the pad stack drill size and the drill oversize setting is the diameter used by the batch DRC routines. This is also the drill hole size that displays on a pin or via.

In most cases, manufacturers use drill sizes equal to the pad stack drill size plus twice the thickness of the plating. To determine what value to enter, know how your board manufacturer chooses drill diameters.

Surface Mount Device Pads

Surface Mount Device (SMD) pads are usually rectangular, although you can assign any shape. When defining an SMD pad, set the pad stack for all other layers to a round, size 0 pad, drill size 0, and clear Plating. The drill oversize parameter does not apply. If associated copper is used to define the SMD pad, set the pad stack for that layer to a square shape, size 0.

When the drill size is 0 for a pad with a square shape, autorouters recognize it as an SMD pad. Use the Via under SMD routing command to place vias directly under SMD pads. For more information, see “Interactive Routing” on page 211. You can undersize the SMD pads for the paste mask photoplot during the CAM process.

Plane Connections

Pins that are supposed to connect to the plane are usually plotted for manufacturing using spoked thermal relief pads. Pins that are insulated from the plane are plotted using their pad diameters as clearance diameters, rather than copper areas, when they pass through the plane layer. In this case, they are called *antipads*.

For information on setting up CAM and copper pour planes see the Connecting a Net with a Plane topic.

Displaying Pad Stacks with Thermal Connections

For CAM plane layers, use the Show General Plane Indicators option in the Thermals tab of the Options dialog box to see which pins have thermals. Pins with thermals have a Plane Thermal setting, which determines if the thermal is generated for the pin. The Plane Thermal setting signals CAM output to assign a D-code for a thermal relief aperture around the pins.

Copper Pour Plane Pad Stacks and Remaining Ratsnest Display

For copper pour plane connections, unrouted connections are used to control thermal generation. Even after the connection is established, the net you are connecting with a plane still appears as an unrouted connection. These connections signal the copper flood operation and generate a copper pour thermal relief around the pins. The copper pour area draws a screen representation of the thermal.

Plating and Clearance Checking

You can turn off Plating for component pad stacks by clearing Plated in the Pad Stacks Properties dialog box. Nonplated holes are drilled to true drill diameter, without oversize. They are drilled after the plating process. Select Plating for nonelectrical drill holes, like mounting holes, which are basically parts with one pin. Otherwise, PADS Layout assumes that all via holes are plated and the Drill Size value is used for plated holes only.

The batch clearance checking functions consider the added drill oversize value when flagging errors. If Plated is cleared, the check applies to the true drill value.

Slotted Holes

Slotted holes are oval mounting holes in a printed circuit board. Slotted holes have orientation and offset properties, but have the same unit and range as the associated pad's orientation and offset. You can use slotted holes with only oval and rectangular pad shapes. Therefore, you can only define slotted holes for component pins. All pads in the pad stack should be oval or rectangular. You can create custom thermal or antipad definitions for slotted holes.

For custom antipads, the default antipad shape around the slotted hole is always oval. For custom thermals, the default pad shape around the slotted hole depends on the pad shape on the specific layer.

For custom thermals, settings on the Pad Stacks Properties dialog box control spoke angle and width. The Clearance Rules dialog box pad to copper clearance controls the calculation of the outer width.

Result: A custom thermal or antipad for a slotted hole has the same orientation and offset as the slot.

Other information on slotted holes includes:

- Slotted holes are displayed in the same color as drills.
- The Drill Oversize option in the Options dialog box > Design tab applies to plated slotted holes.
- Drill-to-drill clearance checking checks slotted holes.
- You can use slotted holes as test points.

Slotted Hole Geometry

A slotted hole length, orientation, and offset are the same as those for a pad:

0 £ length £ 1000 mils

-500 £ offset £ 500 mils

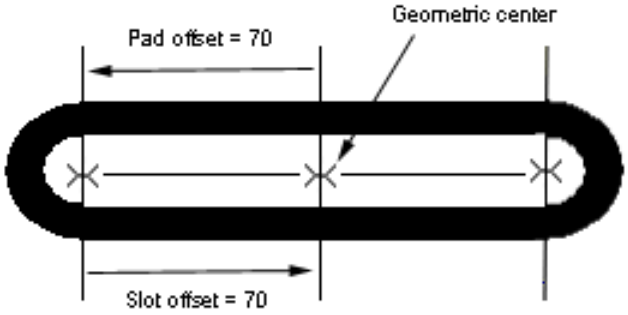
0 £ orientation £ 179.999 degrees

Slotted Hole Offset Versus Pad Offset

You can change the offset of oval pad shapes to move the electrical center of the pin (as well as the center of the drill). Since slotted holes are considered drills, the electrical center is also considered the center of the slotted hole. If you moved the pad offset to the far end of the pad, you would quickly move the slotted hole outside the pad boundary. Instead of moving a pad offset, you can use a slotted hole offset to move the slotted hole.

Slotted hole offset moves the center of the slotted hole relative to the electrical center of the pin–always in the opposite direction of the pad offset. For example, if you want to move the electrical center of a 200x60 mil pad 70 mils to the left, set a pad offset of 70. To center the slotted hole, set a slotted hole offset of 70. See the graphic below for more information. The maximum amount of offset you can set is one half the length.

Figure 4-1. Pad Offset Versus Slot Offset



CAM Output and Slotted Holes

Drill Drawings

Slotted holes are shown on drill drawing as 2 drill symbols, one at each end of the centerline of the slotted hole. The true outline (edge-to-edge representation) of the slotted hole is then draw around the two drill symbols.

Figure 4-2. Drill Drawing of a DIP14 with 30x120 Mil plated Slotted Holes at Each End

Slotted holes are shown in the drill chart as a value in the Size column. The size is the edge-to-edge size of the slotted hole, meaning that the width of the slotted hole is also the drill size. The Quantity column shows the number of slots.

Table 4-3 shows a sample Drill chart of a DIP14 with a 30x120 mil plated slotted holes at each end

Table 4-3. Sample Drill Chart

SIZE	QTY	SYM	PLTD
35	14	+	PLTD
30x132	2	X	PLTD

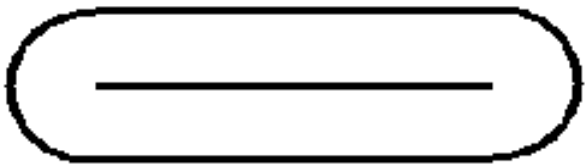
NC Drill

Two drill symbols for each end of a slotted hole are created in the NC drill data. Slotted holes are output according to pin type output (plated/nonplated). Slotted hole test points are output according to test point output. For example, if you output plated pins and test points, slotted holes that are plated and/or test points will also be output. Complete NC routing data for slotted holes is not output in the NC Drill data. You must use a CAM tool such as CAM350 with the slotted hole Gerber data to create the NC drill data for slotted holes.

Gerber Output

Slotted holes are represented two ways in Gerber data. The first is as a centerline with endpoints that are one half the drill size from the ends of the slotted hole. The second ways is as a closed, unfilled oval, showing the outer edge of the slotted hole. The centerline of this oval is the outer edge of the slotted hole. Both are drawn with the smallest round aperture available. You can use a CAM tool such as CAM350 with the slotted hole Gerber data to create the NC drill data for slotted holes.

Figure 4-3. Gerber Output of Slotted Holes



Using Slotted Holes in CAM350

The type of slotted hole representation you use in CAM350 depends on how you fabricate slotted holes.

- To create slotted holes with a series of drills, use the centerline format and the Gerber to Drill feature in CAM350.
- To mill slotted holes, use the outer edge format and Gerber to Mill in CAM350.

Drill drawings and NC Drill data of slotted holes are not supported when PADS Layout designs are imported into CAM350.

Pad Stack Report

You can create a text format, pad stack report file for a selected component or via or all components and vias.

A sample file output appears below.

```
PAD STACK LISTING
Finger Format — Size Shape Orientation Length Offset Drill
Pad Format —— Size Shape Drill
Shape =[OF]OvalFinger,[RF]RectFinger,[R]Round,[S]Square,[A]Annular,[O]Odd
Pad stack for Via: STANDARDVIA
START(1)55C 37
INNER 70 C  20 (inner layers not otherwise described)
END(2)   55 C
Pad stacks for Part Decal: DIP14
Used by Part Types: 7400
PIN 0 (All pins not otherwise described) (Plated)
TOP(1)   60 C  37
INNER 80 C  (inner layers not otherwise described)
BOTTOM(2)60C
PIN 1 (Plated)
TOP(1)   60 S  37
INNER 80 C  (inner layers not otherwise described)
BOTTOM(2)60C
Pad stacks for Part Decal: DIP14\SO
Used by Part Types: 7400
PIN 0 (All pins not otherwise described) (Plated)
TOP(1)   24 R  90 60 0 0
INNER 0  C  (inner layers not otherwise described)
BOTTOM(2)0C
```

Control of Solder Mask and Paste Mask

There are several methods to control solder or paste mask openings in the gerber outputs. You can add physical shapes as openings on the mask layers, and you can also generate the openings when creating the gerber outputs. And since multiple values can exist in different areas, there is a hierarchical rule about which values take precedence. You can use any combination of these methods.

Physical Shapes in the Work Area

When you create a physical shape (opening) on the mask layer using the following methods, you can view it in the Decal Editor and the Design Editor.

In the Pad Stacks

You can add the solder and/or paste mask layer to a pad stack and create the shape and size of the opening. And since you can create all of a component’s pins at the same time, you can

potentially add mask openings to all the pins of a component at once. This method limits you to the pad shapes that are available to pad stacks.

You can do this at the design level to make it design specific, or at the library level to make it available for each new design that uses the decal.

In the Decal Editor

When you edit the decal in the Decal Editor, you can add a copper shape to a solder mask or paste mask layer and that shape becomes the opening in the mask. There are two ways to create the shape:

- You can draw the shape freehand using copper. There is no limit to the shape like in the pad stacks.

  Tip: Use this method to create gang solder mask relief by drawing a copper rectangle over a whole row of pads.

- You can use the Generate Drafting Shape command to generate a copper shape based on the shape of a terminal that you can add to the solder or paste mask layer. This has the same effect as adding the shape to the pad stack.

Generated Shapes in the CAM Output

When you generate these shapes in the CAM Output you cannot view them in either the Decal Editor or the Design Editor. They are only visible in the CAM Preview.

In the CAM Output

When you create the solder or paste mask output, you can add the pads (from the component layer) and the pad shapes become the mask openings. You can globally oversize or undersize the pad shapes with a setting in the Plot Options.

In an Attribute

You can apply a unique over or under size setting to special components using an attribute. You can assign an oversize or undersize by adding the *CAM.Solder mask.Adjust* or *CAM.Paste mask.Adjust* attribute to a decal. The attribute value applies to all pads of the component.

Mask Hierarchy

There is a hierarchical order to the solder mask and paste mask values when there are conflicts. When multiple values exist at different hierarchy levels, the highest priority value is used; the others are ignored.

Table 4-4. Solder & Paste Mask Size Priorities

Priorities Highest to Lowest
1. Pad sizes on the top and bottom solder and paste mask layers in the pad stack. If a pad size is defined on both the component level and the decal level, the priority order is: • Component level value • Decal level value **Tip:** The CAM.Solder mask.adjust and CAM.Paste mask.adjust attributes apply not only to pad shapes on the Solder mask or Paste mask layers, but in the absence of those pad shapes they can also apply to top and bottom layer pads applied to the solder and paste mask layers during the creation of the CAM document. See also: Customizing Pad Stacks of Decal Pins 2. Component level over/undersize values of the CAM.Solder mask.adjust, and CAM.Paste mask.adjust attributes. This applies to Components and Vias. See also: Working with Object Attributes **Warning:** Because these two items are of equal priority, when they are both defined, they are both used to generate the mask shape.
1. Decal level over/undersize values of the CAM.Solder mask.adjust, and CAM.Paste mask.adjust attributes. See also: Creating Attributes in the PCB Decal Editor
2. Over/undersize value of the global CAM, Plot Options > Over(Under)size Pads By value in combination with the CAM.Apply Oversize To All Pads attribute. See also: Plot Options Dialog Box
3. PCB level over/undersize values of the CAM.Solder mask.adjust, and CAM.Paste mask.adjust attributes. See also: Creating Attributes in the PCB Decal Editor

Related Topics

Creating a Solder Mask Gerber-format File

Creating a Paste Mask Gerber-format File

Tenting Vias With Solder Mask

Chapter 5
Drafting

This chapter covers the drafting commands which are available on the Drafting Toolbar. Use drafting commands to create and edit drafting objects, including the board outline, copper areas, keepout areas, simple line shapes, text, and all other items not generally associated with part placement or routing. This chapter also discusses copper pour and split plane operations.

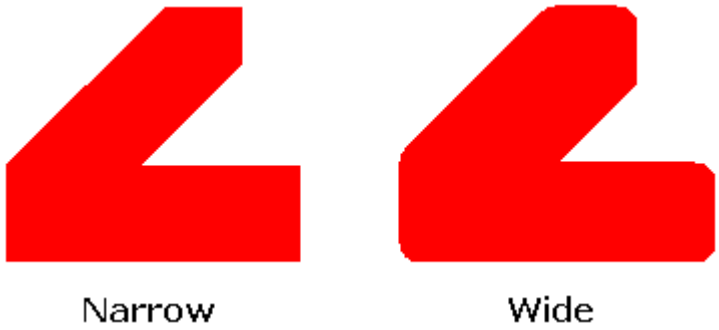
Edge Precision of Drafting Shapes

You often need to precisely control the edge of drafting shapes. There are many factors which contribute to that edge. Those factors are listed below.

Line Width

The precision of edges is controlled by the line width used to create the shape. The default line width is set on the Drafting/Text and Lines page of the Options dialog box. However, you can change the line width prior to, or after creating your shape. Use a very narrow outline width to achieve a sharp corner or increase the value for more blunt corners. All corners are rounded with a radius equal to one half of the outline width.

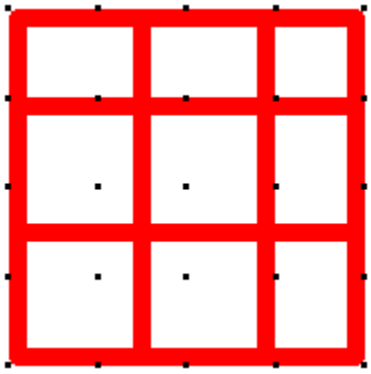
Figure 5-1. Narrow Versus Wide Outline Width



Design and Hatch Grids

Lines are used to create the outline of the shape and are also used to create the fill inside electrical drafting items. Lines or outlines are placed on the design grid. Then shapes are filled using horizontal lines placed on the Hatch Grid.

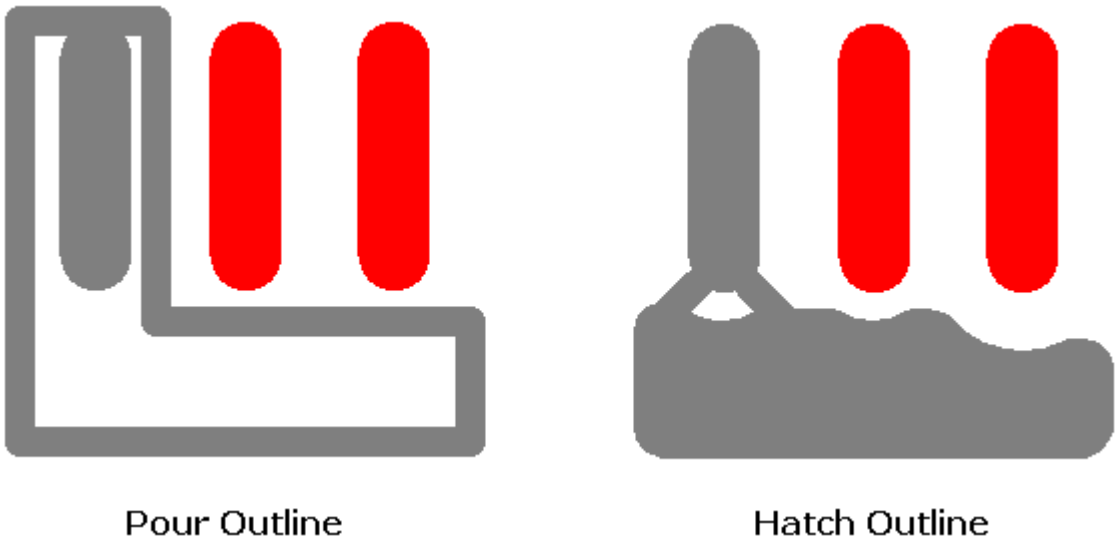
Figure 5-2. Hatch Grid



Clearance Rules

If you are creating a copper pour or a plane area and your drafting shape is near another object, the resulting poured outline could be displaced by the clearance rules of that object.

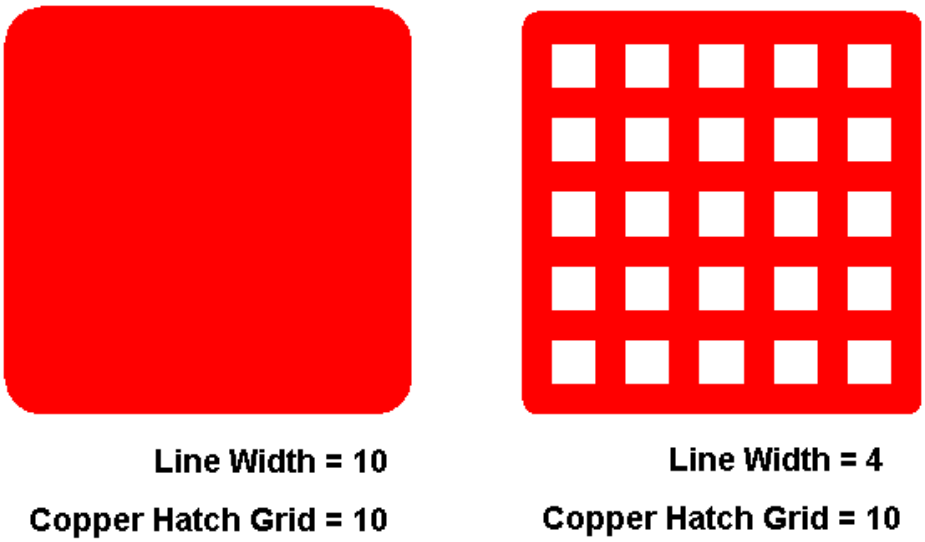
Figure 5-3. Clearance Rule Effect on the Hatch Outline



The Fill of Copper, Copper Pour Areas, and Plane Areas

The fill of copper, copper pour, and plane shapes is generated using drafting lines drawn on the copper hatch grid. If the shape is created using a line width value less than the copper hatch grid value, it results in a hatch pattern. If you use a line width value equal to, or greater than the copper grid value, it results in a solid copper shape.

Figure 5-4. Example of Solid and Hatch Pattern



Shapes can be flooded with copper in varying degrees of hatch through to a solid pattern. The pattern used by default when you flood a shape is determined by the relative values of two settings in the Options dialog box:

- The Default width value on the Drafting / Text and Lines options page
- The Copper hatch grid value on the Grids options page

You can set line width and grid settings before or after creating the object. After you've drawn the shape, you can change the settings in the Drafting Properties. If you need to customize the copper hatch grid for a copper pour shape, you can set the grid value in the Flood & Hatch Options dialog box.

For more information on creating shapes with the desired flood pattern, see Creating a Copper Shape, Creating a Copper Pour Area and Creating a Plane Area.

Split Planes

PADS Layout supports the creation of split planes and mixed planes. A split plane is a plane layer with one or more isolated areas of copper that have different net assignments. A mixed plane is a plane layer with one or more plane areas and any number of signal routes.

In many designs it is common to require large copper areas for voltage and ground nets. To do this, create an internal layer dedicated to a single net (typically power or ground nets). Although it is common to have one plane layer dedicated to a single net, designs with multiple voltage requirements require you to separate, or split, the plane layer into isolated areas. When designing very dense designs, use split plane layers for normal signal routing.

Layer Definition in Setup includes an option for identifying split/mixed plane layers and the 2-D drafting commands contain options for creating closed shapes to define plane areas and voids (areas with no copper). When you route on a layer identified as a split/mixed plane layer, clearances are automatically created around the trace and pin pair: the traces are actually plowed through the plane area.

Split plane operations also take full advantage of design rule-driven spacing and thermal relief generation. When you create plane areas with embedded traces, their spacing can be based on custom thermal clearances, design rule clearances or global clearances.

See also: If you use split/mixed planes and you also route using SPECCTRA, see "SPECCTRA and Split/Mixed Planes" on page 237.

Associating Nets to Copper Planes - Plane Layer vs. Poured Copper

If you are setting up a copper area as a power or ground plane, and you plan to dedicate the entire level to the plane, you don't need to draw the copper area. Instead, use the Layers Setup

dialog box to define the level as a dedicated plane layer. CAM processing treats plane layers as negative image layers for photoplot output. You can then use the same dialog box to assign the netnames that will connect to the plane layer. With this method you can include more than one netname to join to the plane.

When you use this technique, PADS Layout CAM processing creates a plane plot film for the layer that defines the entire level as copper. The border is defined by the board outline, and the interior includes the oversized antipad insulators and the thermal connections where the copper joins pins associated with the plane by netname. The Plane Thermal option in the Properties dialog box controls thermal generation for individual pins of a plane net, and signals CAM and plane checking to generate a thermal relief for the pin. In CAM options for plane layers, preferences are set for using custom thermal settings and via flood over.

You can determine whether you are successfully tied to the plane. If you are routing from an SMD to a via and View Nets default is set to Partial, the ratsnest connections from the vias won't appear. You can also run a plane check under Verify Design.

Drawing a Split Plane Instead of Using a CAM Plane

You can also physically draw a poured copper area as a plane that connects power or ground nets. Using this method, you don't have to assign the layer as Plane in the Layers Setup dialog box, you can leave it as a routing layer. Rather than having CAM produce the negative image plane plot, you are drawing the copper area yourself; the film for this layer is output as a routing plot. You do not assign a net to a routing layer as you would to a plane layer.

This is a more true-to-plot display; the thermal connections are visible as part of the copper display on that layer. You can also run other traces on the level as long as they don't disrupt the connectivity of the plane. The copper pour routine insulates traces from the plane copper when it pours the copper fill.

You can only associate one netname per copper area to automatically join to the plane. You aren't limited to one copper area per layer; you can draw two on a layer side by side, one netname for power and one for ground. You also have to rehatch copper pour areas every time you load the file or repour every time you move same-level routing because these edits increase file size.

Plane Thermal Indicators

Plane thermal indicators are graphic images that show the thermal attribute status of pads. There are two types of plane thermal indicators, general and specific, as well as a plane antipad

thermal indicator. Sometimes during editing, thermal markers are covered; redraw the screen to view all of the markers.

Table 5-1. Plane Thermal Indicators

Indicator	Description
General Plane Thermal	General plane thermal indicators show that a connection to a CAM or split/mixed plane exists somewhere within a pad stack. This indicator shows as a small "x" in the center of the pad: 
Specific Plane Thermal	Specific plane thermal indicators show that a specific pad on a specific layer is connected to a plane. These indicators appear in the color assigned to the board outline in the Display Colors Setup dialog box. There is a different specific plane indicator for CAM planes and split/mixed planes: CAM Plane Indicator:  Split/Mixed Plane Indicator: 
Plane Antipad	Pads that do not belong to a CAM or split/mixed plane net and are present on one of these layer types appear with a circle to represent the antipad. These pads use the color specified for that layer in the Display Colors Setup dialog box.

Understanding How to Manipulate Split Plane Thermals and Antipads

The following conceptual information applies to the creation of Plane areas and the use of Custom Thermals. For procedures to create plane areas, see Creating a Plane Area in the User Guide and for custom thermals, see Creating Thermals in the Pad Stacks in the User Guide.

In this section:

Common Split Plane Thermal and Antipad Clearance Scenarios

The following sections discuss the four most common setting-combinations that affect a design's thermal and antipad clearances. Understanding these situations allow you to predictably create the thermal and antipad clearances that you desire.

In this section:

Default Thermals and Antipads

Setup required:

1. *Remove Unused Pads* check box selected (enabled)
2. *Use Design Rules for Thermals and Antipads* check box cleared (disabled)
3. No custom thermals defined in Pad Stacks Properties
4. No custom antipads defined in Pad Stacks Properties

The Remove Unused Pads setting removes the pad from any component pin or via on a split plane layer if that pin or via is not thermally tied to the plane or there are no routes coming off that pin or via on the plane layer.

Default thermals and antipads use the following basic formulas for calculating the clearance to the drill for antipads and the clearance to the pad for thermals:

- Default component through-hole antipad diameter = (2 x Copper-to-Pad clearance value) + pad size
- Default via antipad diameter = (2 x Copper-to-Via clearance value) + via pad size

Example: +5V via in a +12V plane area with 35 mil pad, 20 mil drill, and 8 mil copper-to-via clearance value would result in a 51 mil diameter antipad, (2 x 8) + 35 = 51. Default copper to pad or copper to via clearance on thermals is simply controlled by the copper-to-pad or copper-to-via values.

The rules hierarchy influences the default antipad and thermal clearance sizes. A Net rule with a higher copper-to-pad value than the default rule will create larger antipad sizes and greater clearance on thermals for the pins of that net. A net rule created for the plane net will not affect the antipad clearance size, but will affect the copper clearance on thermals. Net rules for the opposing nets will affect antipad size.

Conditional rules have some special behavior. A Conditional rule in which a net is set against a layer will utilize the default antipad calculation. However, a Conditional rule in which one net is set against another net will not affect the antipad size unless the *Use Design Rules for Thermals and Antipads* setting is enabled - see the next section.

Design Rules for Thermals and Antipads

Setup required:

1. *Remove Unused Pads* check box selected (enabled)
2. *Use Design Rules for Thermals and Antipads* check box selected (enabled)
3. No custom thermals defined in Pad Stacks Properties
4. No custom antipads defined in Pad Stacks Properties

The thermals and antipads are now independently controlled by the Design Rules. This method allows you to create large antipad sizes without affecting the clearance on pads with thermals. The antipad size created affects both vias and component pads.

- Antipad Diameter for component pads and vias = (2 x Copper-to-drill) + Drill size

Thermal clearance is controlled by design rules copper-to-pad and copper-to-via settings. The rules hierarchy influences the default antipad and thermal clearance sizes. A net rule for the plane net affects the antipad size of any opposing net if you adjust the drill-to-copper value.

Example: A plane area is defined for +12v, a net rule for +12v with an increased drill-to-copper value will cause the antipad size to increase for all non-connected pins or vias within the +12v plane area.

Conditional rules have some special behavior. A Conditional rule in which a net is set against a layer will utilize the default antipad calculation - see the previous section. However, a Conditional rule in which one net is set against another net will not affect the antipad size unless the *Use Design Rules for Thermals and Antipads* setting is enabled.

Custom Thermals and Antipads

Setup required:

1. *Remove Unused Pads* check box selected (enabled)
2. *Use Design Rules for Thermals and Antipads* check box cleared (diabled)
3. You have custom thermals defined in Pad Stacks Properties
4. You have custom antipads defined in Pad Stacks Properties

No design rule settings will impact the custom antipad size you have defined for the decal(s). The thermal or antipad clearances will be based solely on your custom pad stack settings. Any clearance checks will also be based on these settings. Note, the *Remove Unused Pads* check box must be selected in order for the custom antipads to be created.

Hierarchy Between Custom Thermals and Antipads and the Use of Design Rules for Thermals and Antipads

Setup required:

1. *Remove Unused Pads* check box selected (enabled)
2. *Use Design Rules for Thermals and Antipads* check box selected (enabled)
3. You have custom thermals defined in Pad Stacks Properties
4. You have custom antipads defined in Pad Stacks Properties

All custom antipad settings are ignored on pins and vias. The antipad size are determined by the copper-to-drill value in design rules.

- Antipad Diameter for component pads and vias = (2 x Copper-to-drill clearance value) + Drill size

A custom thermal's inner diameter is still used (the actual pad size), however the copper-to-pad clearance is used for component pins and the copper-to-via clearance is used for vias. The outer diameter value from a custom thermal is ignored in favor of the design rules.

Common Thermal Style Scenarios

The following sections discuss the three most common situations that affect thermal styles such as orthogonal, diagonal, flood over and no connect. Understanding these situations will allow you to predictably create the thermal styles that you desire.

In this section:

- Thermal Style Changes in Tools > Options > Thermals Page
- Hierarchy between Custom Thermals and Thermal Style Changes in Tools > Options > Thermals Page
- Hierarchy Between Custom Thermals, Style Changes in Tools > Options > Thermals Page and Flood-over Via in Plane Area Polygons

Thermal Style Changes in Tools > Options > Thermals Page

Setup required:

1. You have made thermal style changes in Options > Thermals page
2. No custom thermals are defined in Pad Stacks Properties
3. The *Flood over vias* check box is cleared (disabled) in the Drafting Properties-Plane Area > Flood & Hatch Options dialog box

The thermal style you choose affects all component pins and all vias for all layers and for all copper pours and split planes. You can set different thermal styles for each of the four pad shapes: round, square, oval, and rectangle. You can also differentiate between through-hole and surface mount components.

For example: It is possible to have a through-hole round pad with orthogonal spokes and a through-hole oval pad with a flood-over. You could also setup a through-hole round pad to be diagonal while an SMD round pad is set to no-connect.

Hierarchy between Custom Thermals and Thermal Style Changes in Tools > Options > Thermals Page

Setup required:

1. You have made thermal style changes in Options > Thermals page
2. You have custom thermals defined in Pad Stacks Properties
3. The *Flood over vias* check box is cleared (disabled) in the Drafting Properties-Plane Area > Flood & Hatch Options dialog box

For those components with custom thermals defined, those settings will be adhered to. This only applies to Split Planes, copper pours do not adhere to custom thermals. All other components and/or vias will follow the thermal styles defined by the Thermals page in Tools > Options. You can create a flood-over of specific pins or vias by creating a custom thermal and reducing the outer diameter size to be equal to the inner diameter size. You may find this very helpful if, for instance, you only want the two mounting hole pins of your component to flood-over when connecting to GND.

Hierarchy Between Custom Thermals, Style Changes in Tools > Options > Thermals Page and Flood-over Via in Plane Area Polygons

Setup required:

1. You have made thermal style changes in Options > Thermals page
2. No custom thermals are defined in Pad Stacks Properties
3. The Flood over vias check box is selected (enabled) in the Drafting Properties-Plane Area > Flood & Hatch Options dialog box

For those components with custom thermals defined, those settings will be adhered to. This only applies to Split Planes, copper pours do not adhere to custom thermals. For those vias that are within the plane area polygon in which you specified for “Flood Over Vias” they will all be flooded over, no custom thermals will be adhered to within that plane area polygon. To set Flood Over Vias select the Plane area polygon in the plane area outline mode, right-click and click Properties, click on the Options button and in the Flood & Hatch options dialog box, select the Flood over vias check box. The thermal styles defined on the Thermals page are adhered to by all remaining components and vias.

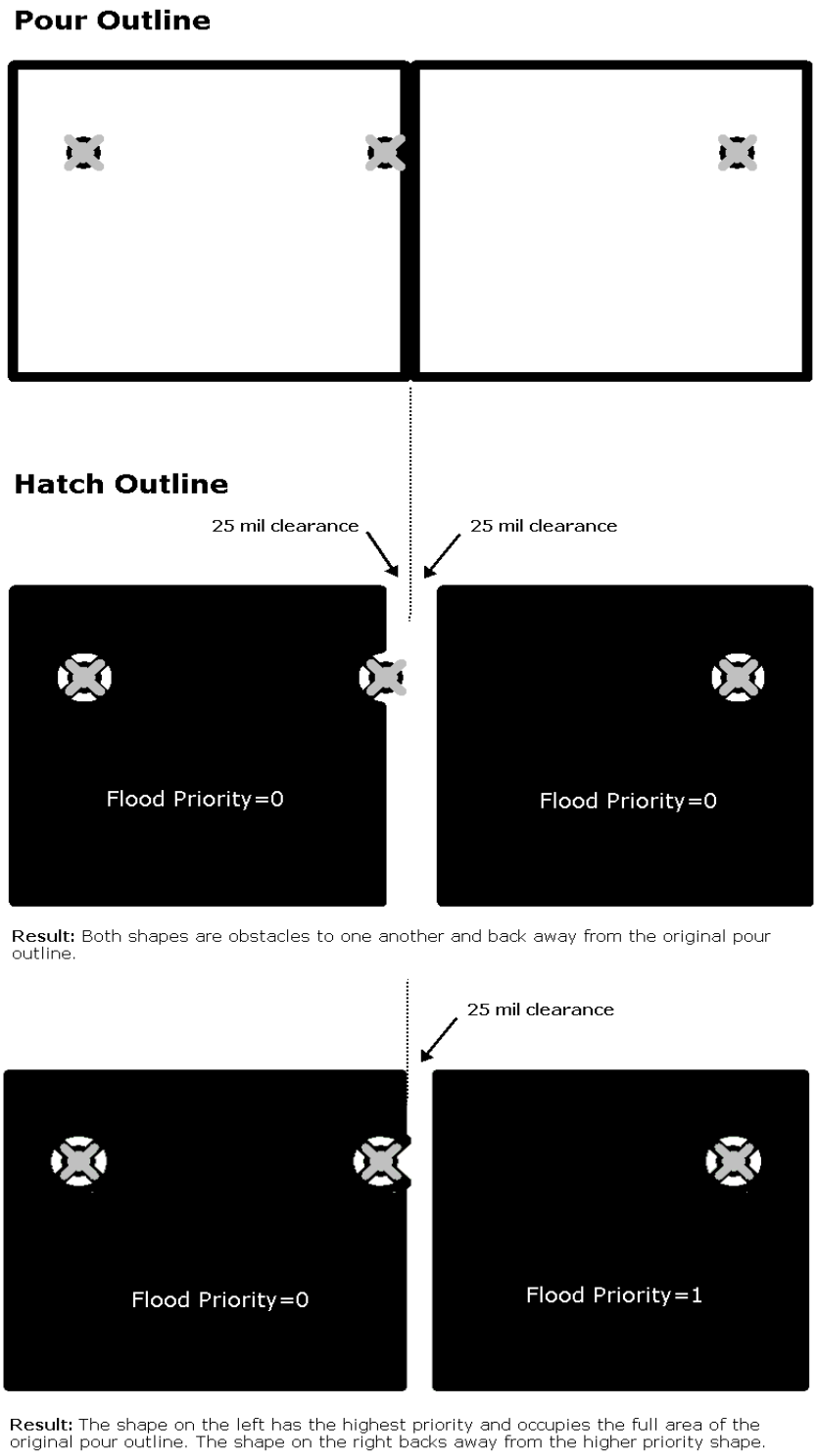
Copper Pour and Plane Area Flood Priorities

To determine which shape should be flooded first, you must set a flood priority to each shape. A shape with a lower priority number will be flooded before an object with a higher priority number. Copper pours or plane areas on different layers are processed independently.

Scenario 1

You have adjacent shapes with a 25mil copper-to-copper clearance rule value.

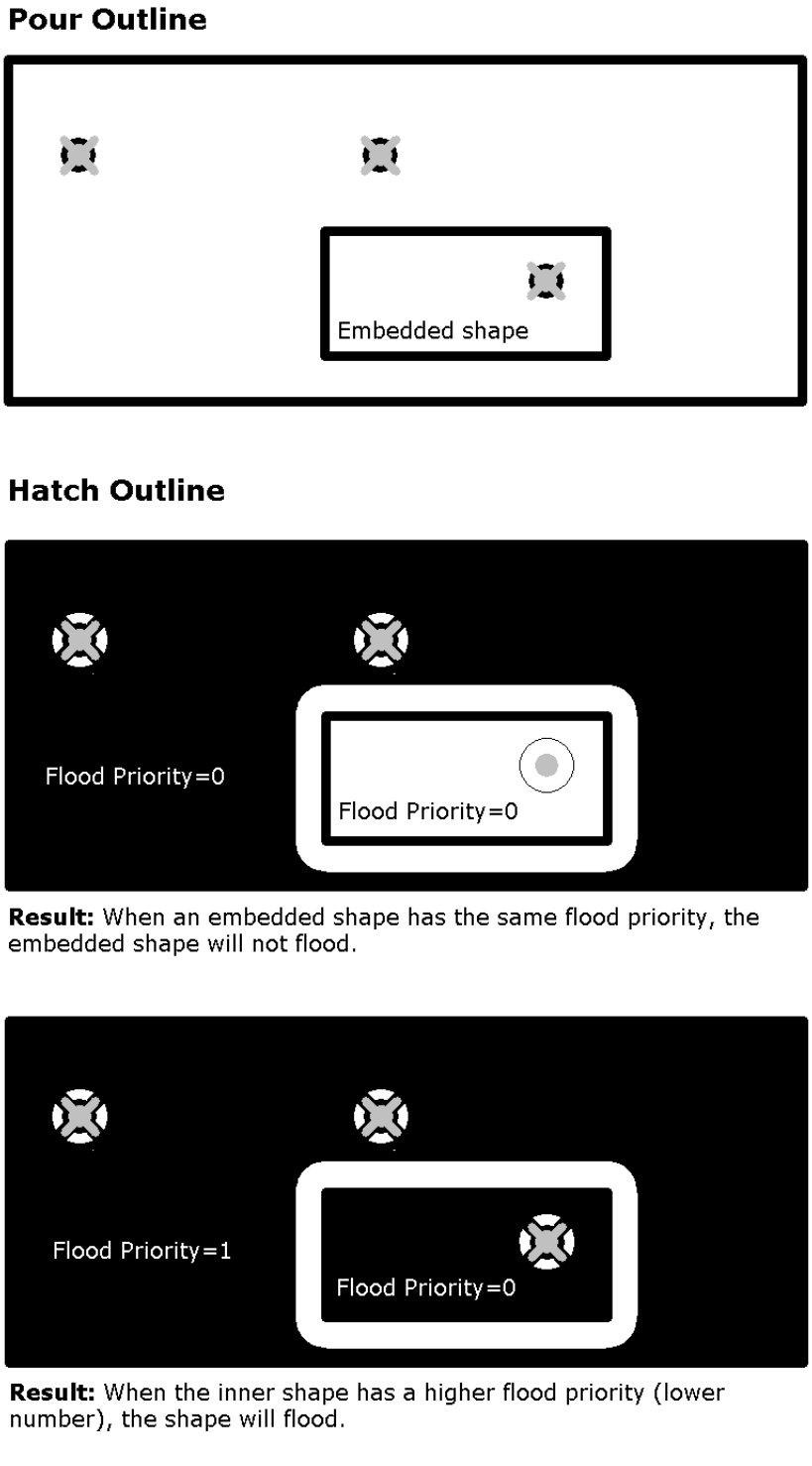
Figure 5-5. Scenario 1 with Same and Different Flood Priorities



Scenario 2

You have an embedded shape within a copper pour or plane area.

Figure 5-6. Scenario 2 with Same and Different Flood Priorities



Result: When an embedded shape has the same flood priority, the embedded shape will not flood.

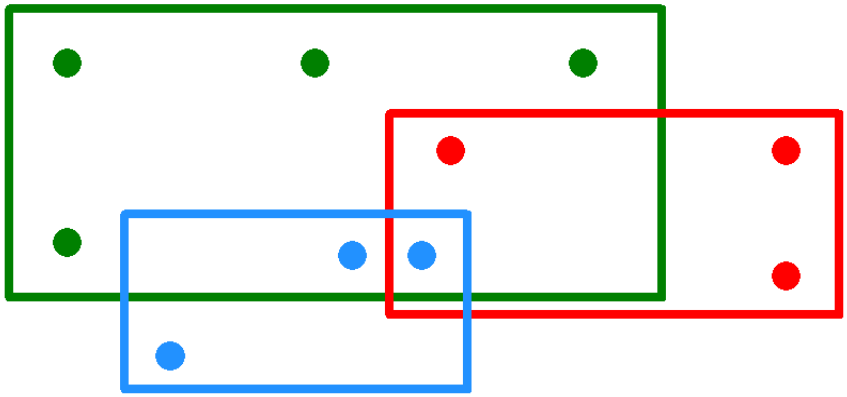
Result: When the inner shape has a higher flood priority (lower number), the shape will flood.

Scenario 3

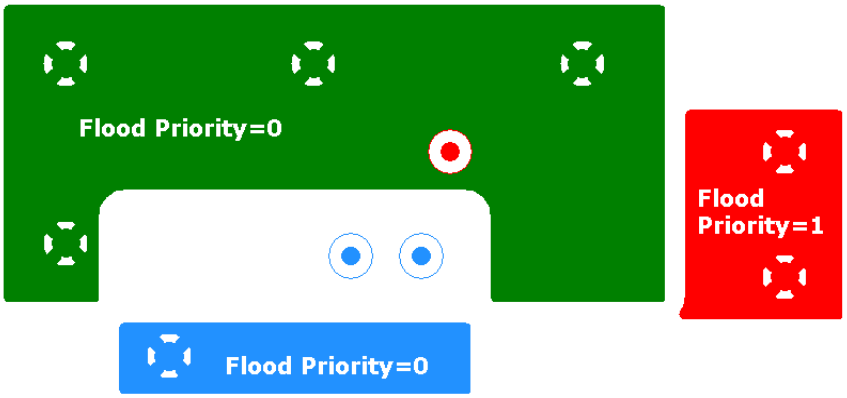
You have multiple overlapping shapes.

Figure 5-7. Scenario 3 with Wrong and Correct Flood Priorities

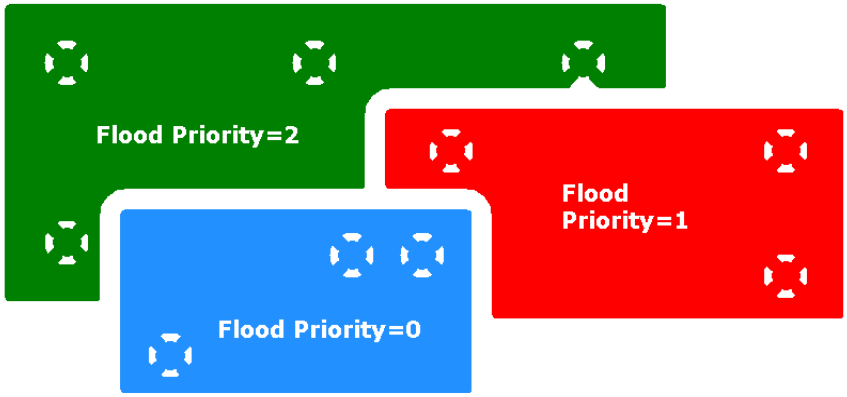
Pour Outlines



Hatch Outlines



Result: Shapes with the same flood priority are an obstacle to each other and if you give a shape the wrong flood priority some pads that are meant to connect are isolated.



Result: Using the right combination of flood priorities ensures the correct result.

Related Topics

Setting Flooding Order of Overlapping Copper Pour and Plane Areas

Flooding a Copper Pour Area

Flooding a Plane Area

Thermal Generation

Note the following information for thermals:

- Component pins and vias automatically receive thermals when they are associated with a plane net and have pads on the associated split/mixed or CAM plane layer. For component pins and vias that receive thermals, the Plane Thermal option is automatically selected in the Pin Properties dialog box. If you redefine a plane net so it is not associated with a split/mixed or CAM plane layer, the thermals are removed for all of the pins in the net, and the Plane Thermal option is automatically cleared in the Pin Properties dialog box.
- If during routing you add a via to a plane net, the via automatically receives a plane thermal. The Plane Thermal option is automatically selected in the Via Properties dialog box.

You can control thermal viewing using the Thermals page of the Options dialog box.

For information on how thermals translate to and from SPECCTRA, see “Links” on page 231.

And CAM Planes

When you generate thermals for CAM plane plots, PADS Layout looks for plane netnames associated with CAM plane layers and checks that pins or vias with pads on the CAM plane layer have the Plane Thermal option selected in the Pin Properties and Via Properties dialog boxes. Use the *Show general plane indicators* option in the Thermals Options to display which pins have plane thermals.

If a pin exists in a net that is associated with a CAM plane layer and Plane Thermal is on for the pin, a thermal appears on the pin. The connections still appear. Use the Plane Thermal option in Verify Design to verify thermal generation for CAM plane plots.

CAM Plane Thermals

CAM plane thermals are displayed in the work area and the CAM preview area as indicated below. These calculations are also used for output during printing, pen plotting, and photo plotting operations:

- The outer diameter of the thermal matches the width of the aperture set in the Photo plotter Setup dialog box.

- The inner diameter is 75% of the outer diameter.
- The number of spokes is always four, arranged diagonally.
- The spoke width is 1/6 of the outer diameter.

Tip: The inner width for custom CAM plane thermals is set as the pad size defined in the Pad Stacks Properties dialog box. The outer width is set as the default same-net pad to corner rule.

And Copper Pour

Thermal generation control for copper flood is based on unrouted connections. Unlike CAM planes, copper flood does not use the Plane Thermal option. PADS Layout look, not only for the same netname, but for unrouted connections leading to through-hole component pins as points to connect with a copper pour thermal. If the pin is routed, for example, on a different layer, and is no longer showing a ratsnest connection, you will not generate a copper pour thermal relief. You must have an unroute to generate copper pour thermals.

For copper pour thermal generation around component pins, the unrouted connection rule holds true for copper pour planes. When there is no unrouted connection, a copper pour thermal is not produced around the pin. When a route and unroute exist on a pin, a copper pour thermal is produced.

There is a slight divergence of the rule for vias. During copper pour, vias will always get copper pour thermals if the netname of the via matches the netname of the surrounding copper pour area, whether they show unroutes or not.

After the copper is poured and the copper pour thermals are installed, the ratsnest connections still appear. In View/Nets, select the Traces option to hide unroutes. Use Verify Design on the Tools menu to check connectivity for copper pour areas.

Connecting Planes and Nets

A plane is a large copper area that provides access to universally necessary nets, like power or ground. One design may use several plane areas. Planes are usually located on inner layers that are dedicated to the plane only, although you can place them on outer layers. The plane area can occupy all or only part of the layer it is on. A plane with two or more partial planes each servicing a different net is called a split plane.

Plane areas defined with Copper Pour may have insulated traces and vias passing across the plane area, as long as the traces do not divide the plane to break connectivity.

Pins that are supposed to connect to the plane are usually plotted for manufacturing using spoked thermal relief pads. Pins that are insulated from the plane are plotted using their pad diameters as clearance diameters, instead of copper areas, when they pass through the plane layer. These are called antipads.

Establishing a plane area and connecting the appropriate nets to it is usually one of the first routing tasks in the design process. The following two methods establish plane areas:

- Define a Layer as a "Plane" Type
- Draw a Copper Pour Area

For more information, see the Connecting a Net with a Plane topic.

Displaying Connectivity with General Plane Indicators

For CAM plane layers, use Show General Plane Indicators in the Thermal page of the Options dialog box to see which pins have thermals. The Plane Thermal option determines whether the thermal is generated for the pin. The Plane Thermal option signals CAM output to assign a D-code for a thermal relief aperture around the pins. Set the Plane Thermal option using the Properties dialog boxes for pins, vias, and jumper pins.

If you set up a plane connection successfully, a D-Code number is assigned for a thermal relief pad in your photoplotter aperture table, one that matches each pad size required.

Copper Pour Plane Connections and Ratsnest Display

For Copper Pour plane connections, the unroutes are unused to control thermal generation. Even after the connection is established, the net you are connecting with a plane still appears as an unrouted, ratsnest connection. These unroutes signal the copper flood operation and generate a copper pour thermal relief around the pins. The copper pour area draws a screen representation of the thermal.

Scaling 2D Line Objects and Dimensions

Use Scale in the object Properties to approximate arcs that are too large for the PADS Layout database or to define and scale fabrication documentation. An arc is too large if its radius is greater than 14 inches or its center is outside the database coordinate area. The database coordinate area is (-28, -28) to (28, 28) inches. Arcs are approximated with several straight segments. When you scale text or dimensions, combine them with other line objects. Line widths are not scaled.

Scaled objects have the same origin as the scale model. When combined text or dimension text is scaled, the maximum text height is 1000 mils and the maximum text width is 50 mils. The scale model must be either a 2D line or a dimension object. You cannot select board outline, copper, copper pour, or keepouts as the original model.

Location of Scaled Objects

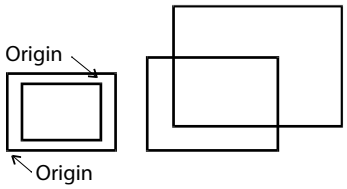
The scaled shape is placed on the same layer as the original shape model. If the new shape is a copper and the original layer is nonelectrical, the scaled shape is moved to the top layer. Board outline objects, however, are always placed on <All Layers>.

The resulting location of the scaled objects also depends on the following factors:

- If only one object is selected, the scaled object's origin is placed at the same location as the scale model's origin. If this places part of the scaled object outside of the database coordinate area, the object is centered within the database coordinate area.
- If multiple objects are selected, each scaled object is placed at the same origin as the scale model. If this places any part of the scaled objects outside the database coordinate area, Scale is canceled.
- If the scale model contains multiple 2D line objects and/or dimensions and you want to maintain their relative positions, combine the objects before using Scale. For examples, see the following figures.

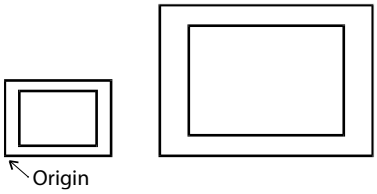
In the following figure, the two noncombined objects are scaled. The objects have individual origins, as shown on the left. These different origins are used when scaling, creating overlapping objects, as shown on the right.

Figure 5-8. Noncombined Objects



In the following figure, the two combined objects are scaled. The objects have a single origin, as shown on the left. This origin is used when scaling, creating a larger copy of the original objects, as shown on the right.

Figure 5-9. Combined Objects



Scaling and Copper Pour/Plane Areas

If the scaled shape is a copper pour shape and the original layer is a split/mixed plane, then the scaled shape is defined as a plane area on the split/mixed plane. If the scaled shape is a copper pour shape and the original layer is not a split/mixed plane, then the scaled shape is defined as a copper pour and you must assign a net to the copper pour.

Scaling and Keepouts

If the scaled shape is a keepout, you must define keepout restrictions for the scaled shape. If you do not define keepout restrictions, the default restrictions are used.

Dimensioning

Dimensioning measures distances or angles on points you select, creates a text string containing the measured value, and creates the associated extension lines and arrows. This information is considered mechanical documentation for PCB designs.

Table 5-2 shows the setup options that let you match existing drafting standards and control how to add new dimensions.

Table 5-2. Dimensioning Set Up Options

Option	Description
Options	Establishes the appearance of newly added dimensions. For more information, see the Options Dialog Box, Dimensioning / General Page.
Snap Mode	Controls how you select data items.
Edge Preference	Sets whether to measure lines from edge or centerline.

Dimensioning also lets you create Baseline dimensions for annotating multiple items from the same starting point. Use Continue to create daisy-chained dimensions.

Tip: You can use Dimensioning within the PCB Decal Editor. However, saving the decal converts dimensions to 2-D lines and text.

Dimensioning Modes

All of the various data items, routes, parts, and drafting items in PADS Layout are referred to as objects. For dimensions, the extension lines, dimension lines, arrows, and text strings that make up dimensions are called *dimension elements*. Collectively, they create a *dimension object*.

Tip: When selecting and modifying, the software considers dimension lines as part of the arrows.

The following lists examples of how you can combine these separate elements to indicate dimensions:

- The standard, or default, dimension is comprised of lines extending from each point of measurement, a dimension line with arrows at each end running between the extension lines, and a text string identifying the length.
- Datum line dimensions are comprised of one extension line and one text string.
- Leader line dimensions consist of a dimension line with an arrow at one end and a text string at the other end. This category includes radius type dimensions.

You can combine dimension elements in various ways to create a dimension object. Many of the modification commands use these combinations, for example:

- You can select the entire dimension object from one of its selected elements using Select Parent.
- If you open the Properties of a selected dimension element, a Parent button appears in the subsequent dialog box. Click the Parent button to consider the entire dimension object for modifications.

Chapter 6
Design Checking

This chapter discusses how to check the design and the netlist. This chapter also covers how to set up design rules for your design. You can also import rules from the schematic. This chapter also discusses the DFT, Design for Test, option of PADS Layout and automatic test point insertion.

Design Rules

This section covers how to set up design rules for your design. You can also import rules from the schematic.

Design rules represent constraints, such as clearances and available routing layers, that support manufacturing, signal integrity, and other design requirements. Design rule checking can be done as you interactively place and route the design (called on-line design rule checking or just DRC), or after you complete design operations.

Most rules that you set up are used by On-line Design Rule Checking and can actively prevent you from making design errors. Some rules are only checked as a post-process using the Verify Design utility. Lastly, some rules are only used by the advanced capabilities of PADS Router.

Setup Strategy

Your strategy in setting up the rules for your design, should be to first create a set of default rules. The default rules are a set of rules that apply to all the objects in your design that will not have any other rules assigned. Objects with hierarchical rules take precedence over the default rules. For example, a pin pair rule with a Trace Width of 10 will override the default rule with a Trace Width of 5.

> **Exception**: Larger clearance rules always have precedence despite the order of the hierarchy. For example, if you use a Default clearance rule of 50 mils, but a higher Net clearance rule of 10 mils, the more restrictive default clearance rule of 50 mils has precedence and is applied to the net even though a hierarchical Net clearance was created. Clearance rules should be less restrictive at lower levels of the hierarchy. They should be a minimum clearance value.

After the Default rule set has been created, create hierarchical rules based on the requirements of your design. Any hierarchical rules that are created are based on the Default set.

Design Rule Hierarchy

The design rule hierarchy enables you to specify a default set of rules for the entire design and additional rules for individual objects or collections of objects. For example, you might specify in the default design rules a small trace width that would be adequate for signal nets, and then specify a large trace width for a class composed of power nets.

Because objects can be subject to multiple design rules, the design rule hierarchy specifies the priority of the rule types, so you can predict which design rule is applied to an object.

Basic Object Hierarchy

The basic object or relational hierarchy can be easily visualized in the Rules dialog box. The lowest level of the hierarchy begins with the Default level. Once you set the Default rules, they apply to all design objects, unless you customize the rules at a higher level of the hierarchy. In the figure below, each object to the right of the Default rules takes precedence over the object before it. For example, the Net level has a higher priority than the Class level, which has only a higher priority than the Default level. The Component level of the hierarchy has the highest priority in the hierarchy.

Figure 6-1. Basic Object Hierarchy

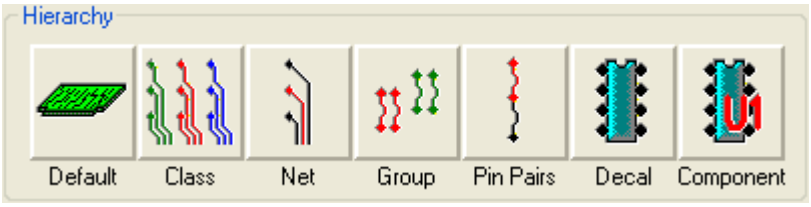


Table 6-1. Hierarchy Descriptions

Hierarchy level	Description
Default	Rules that apply to an object if there are no other individually defined rules.
Class	Rules for a collection of nets, called a *class*, that need identical rules.
Net	Rules for a specific net.
Group	Rules for a collection of pin pairs, called a *group*, that need identical rules.
Pin Pair	Rules for a specific pin pair. Usage example: A net needs different trace widths for certain pin-to-pin connections within it. You can apply rules to only those selected pin-to-pin connections.
Decal	Rules for all components using a specific decal.
Component	Rules for a specific component.

Complete Hierarchy

The complete hierarchy is not limited to the seven buttons in the Rules dialog box. Conditional Rules add many levels in between and are used extensively in the complete design rule hierarchy.

- **Conditional design rules**—Rules that come into effect when an object is adjacent to another object named in the conditional rule and/or on a layer named in the conditional rule.

  Usage Example: Change a trace at any point in its length depending on the layer on which it resides. Net A may be assigned a default clearance of 12, but may need to reduce to 8 when it enters layer 4. In this case, you can assign 12 as the default for net A and make a conditional rule which applies an 8 mil clearance to net A against layer 4.

Although the hierarchy is more extensive than the basic object hierarchy as seen in the Rules dialog box, the relationships that are found in the basic object hierarchy are still maintained by the conditional rules. For example, Net rules have a priority over Class rules in the basic hierarchy. So, a net to net conditional rule also has priority over a class to class rule.

The following list is the complete hierarchy when all possible design rules are involved. This lists the priority for all rules from lowest to highest. Default rules, at level 1, have the lowest priority and represent the lowest level of the rules hierarchy. Component rules, at level 32, have the highest priority and represent the highest level of the rules hierarchy.

Note

Associated nets rules are not part of the rules hierarchy.

1. **Default** rules
2. Default clearance-rules on a specific layer (**Conditional**)
3. **Class** rules
4. Class clearance-rules on a specific layer (**Conditional**)
5. **Net** rules
6. Net clearance-rules on a specific layer (**Conditional**)
7. **Group** rules
8. Group clearance-rules on a specific layer (**Conditional**)
9. **Pin Pair** rules
10. Pin Pair clearance-rules on a specific layer (**Conditional**)
11. Class against Class rules (**Conditional**)
12. Class against Class rules on a specific layer (**Conditional**)
13. Net against Class rules (**Conditional**)
14. Net against Class rules on a specific layer (**Conditional**)
15. Net against Net rules (**Conditional**)
16. Net against Net rules on a specific layer (**Conditional**)
17. Group against Class rules (**Conditional**)
18. Group against Class rules on a specific layer (**Conditional**)
19. Group against Net rules (**Conditional**)
20. Group against Net rules on a specific layer (**Conditional**)
21. Group against Group rules (**Conditional**)
22. Group against Group rules on a specific layer (**Conditional**)
23. Pin Pair against Class rules (**Conditional**)
24. Pin Pair against Class rules on a specific layer (**Conditional**)
25. Pin Pair against Net rules (**Conditional**)
26. Pin Pair against Net rules on a specific layer (**Conditional**)
27. Pin Pair against Group rules (**Conditional**)

28. Pin Pair against Group rules on a specific layer (**Conditional**)
29. Pin Pair against Pin Pair rules (**Conditional**)
30. Pin Pair against Pin Pair rules on a specific layer (**Conditional**)
31. **Decal** rules

    Restriction: Decal rules are not used in PADS Layout.
32. **Component** rules

    Restriction: Component rules are not used in PADS Layout.
33. **Differential pair rules**

    Restriction: Only the gap value applies to the hierarchy; it is considered as the equivalent of the trace to trace clearance.

Design Rule Categories

There are five possible categories of design rules. All categories are not available at all levels of the basic hierarchy. For example, the Fanout and Pad Entry categories are only available at the Default, Decal, and Component levels.

Figure 6-2. Rule Category Icons

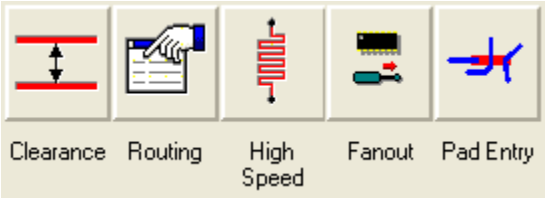


Table 6-2. Design Rule Category Descriptions

Rule Category	Description
Clearance rules	Set the minimum allowable air gap between various object types in the design, such as trace to trace and via to trace. **Tip:** You can also define conditional clearance rules that apply when an object is adjacent to a specified object, or when an object is on a specified layer. For example, you can use conditional rules to specify different clearance rules for a net when it is routed on different layers.
Routing rules	Topology type, assign and prohibit via types, layers available for routing, specify maximum number of vias per net, specify topology types, and allow or prohibit autorouting.

Table 6-2. Design Rule Category Descriptions (cont.)

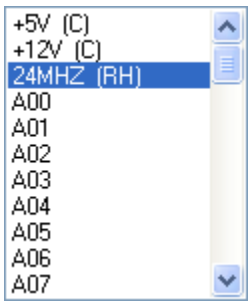
Rule Category	Description
High Speed rules	Parallelism, shielding, geometric and electrical constraints, and length matching. Set the minimum and maximum parameters for advanced design rules, such as parallelism, delay, and capacitance. You can pass these rules from the schematic or assign them in PADS Layout. You can also pass clearance and routing rules to autorouters. The router must be able to interpret the passed rules. **Tip:** You can also define conditional high-speed rules that apply when an object is adjacent to a specified object, or when an object is on a specified layer. For example, you can use conditional rules to specify different high-speed rules for a net when it is routed on different layers.
Fanout rules	Adding copper shapes automatically to SMD pads to make routing easier and to ensure that connections are made. **Restriction:** These rules are only used by PADS Router
Pad Entry	Location and angles for pad-to-trace intersections, and rules for placing vias on SMD pads. **Restriction:** These rules are only used by PADS Router

Non-Default Rules Indicators

When you change rules in the hierarchy from the default rules, two types of indicators are displayed to indicate that rules have been customized from the default rules.

- The categories of rules customized at the current rule level are displayed parenthetically beside the class, net, group, or pin pair name in the list.

Figure 6-3. Non-Default Parenthetical Indicators



C = Clearance rules

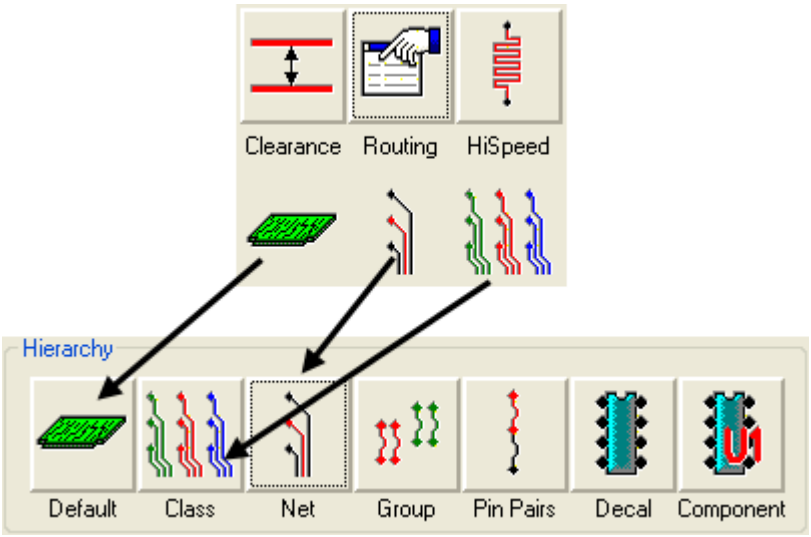
R = Routing rules

H = High speed rules

Example: A net listing followed by (RH) has non-default routing and high speed rules definitions.

- The icon that appears below each rule category button indicates whether the rules are customized at the current rule level or if it is only customized at another rule level.

Figure 6-4. Non-Default Icon Indicators



The meaning of the icon corresponds to the button in the Hierarchy area of the Rules dialog box.

Example: If a green polygon appears below the Clearance button, then the default values apply at the current level of the hierarchy for the object you've selected.

Advanced Rules Option

The Advanced Rules option is a license-enabled (check Installed Options on the Help menu to see if you have this option) modular add-on that extends the restricted rule set to allow use of the full rule set. The full set of constraints allow for control of more complex designs. Without the Advanced Rules option, you only have access to the Default, and Net design rules. With the Advanced Rules options, your license enables the Class, Group, Pin Pairs, Decal, Component, Conditional Rules, and Differential Pairs Rules.

When you use Advanced Rules, the Hierarchical dialog boxes on the Design Rules dialog box are available. You can read in hierarchically assigned rules from a netlist and edit and save rule changes using the hierarchy.

If you do not have the Advanced Rules option, the Hierarchical dialog boxes on the Design Rules dialog box are not available. However, schematic-applied hierarchical rules reside in the netlist, and they can be passed to any autorouter that can interpret them.

Design Rule Checking

On-line Design Rule Checking (DRC) provides continuous rule checking during routing and placement. On-line DRC does not check clearance for text, open associated copper, or free copper belonging to a decal. Table 6-3 shows the four DRC modes.

Table 6-3. Design Rule Checking Modes

Mode	Purpose
Prevent Errors	Prevents you from completing any operation that will create errors. You cannot paste items from the paste buffer. You cannot create or change (move, split, or miter, for example) the following items: board outlines, board cutouts, text, copper, and keepouts. To edit the Properties of these items, turn off On-line DRC using the modeless command, DRO. You can also right-click and click Ignore Clearance to temporarily override DRC and complete an operation. When you are done, On-line DRC automatically resets to Prevent mode.
Warn Errors	Warns that a design rule violation has occurred by highlighting the rule obstacles.
Ign Clrn	Ignores clearance checking.
Off	DRC is completely deactivated. When On-line DRC is off, program performance is enhanced. However, you are limited to the Route command for trace editing. When you turn On-line DRC on again, PADS Layout pauses to map the board. The undo buffer is cleared when you turn On-line DRC off, but is still available for further edits.

To turn On-line DRC on use the Design tab in the Options dialog box or the modeless command, DRP or DRW. After activating On-line DRC, you can change the setting using the dialog box, or the modeless commands.

When On-line DRC is activated, an octagon shaped *guard band* appears at the end of the route to indicate any clearance violation. You can hide the guard band using the Show Guard Band option in the Routing tab of the Options dialog box.

On-line Design Rule Checking is available on most PADS Layout configurations. For Advanced Design Rules package users, all hierarchical rules, except High Speed EDC settings, are monitored.

Clearance Checking Against Text

For clearance checking against text, a rectangle referred to as an *extent box* is drawn around the text string, calculated as a smallest rectangle that will contain the text string. If a text string is rotated, the extent box is also rotated.

Text strings included in clearance checking are those that exist on the same level as other checked objects, or created to appear and on all layers.

Tip: Text belonging to a decal is not checked for clearance by On-line DRC.

Clearance Checking Against Copper

Clearance checking against copper includes closed or open copper that is not part of the current net being routed and exists on the same layer as other checked objects. Closed copper in a decal that is associated with a terminal is checked for clearance by On-line DRC.

Tip: On-line DRC does not check the clearance of open associated copper and free copper belonging to a decal.

Design for Test

To support In Circuit Testing (ICT) procedures, PADS Layout's DFT Audit can help you manage in-circuit test points. Using parameters that you set, DFT Audit can analyze all nets in the design, automatically assign test point attributes to the appropriate vias and component pins on accessible (adaptable) nets, add test points to adaptable nets that are already routed, and report inaccessible (non-adaptable) nets. For non-adaptable nets, DFT Audit can add test point vias and place them outside the board outline. These capabilities help you consider ICT early in the design process, improving your productivity by reducing potential iterations of a manual DFT Audit.

To manually assign a component pin or via as a test point, you add a test point setting to the object. A test point can be one pin of a multiple pin component, the only pin of a single test point component, or a via.

DFT Audit assigns vias and component pins as test points rather than adding several single pin components. Therefore, you avoid backward annotation of test point information to the schematic. For more information, see "Test Point Definition" on page 115.

When you run DFT Audit, PADS Layout automatically transfers the design to PADS Router. Using parameters that you set in PADS Layout, PADS Router analyzes all nets for adaptability and adds test points to routed adaptable nets. Note that PADS Router may reroute nets during DFT Audit. When PADS Router is done, it transfers the design back to PADS Layout. For nets that PADS Router determines to be non-adaptable, PADS Layout can optionally add test points, which are placed outside the board edge. When DFT Audit finishes, the DFT Audit Board Report appears.

To access the DFT Audit, click DFT Audit on the Tools menu. For information about running DFT Audit, see the Performing a Test Point Audit topic.

Tip: DFT Audit tolerates slotted holes, but doesn't test them for adaptability.

While you can run DFT Audit in either PADS Layout or PADS Router, the dialog box used to set an DFT Audit option depends on the program you are running. For information, see the "Mapping PADS Layout DFT Audit settings to PADS Router" section in the "Routing Setup" chapter in the *Routing Concepts Guide*.

Test Point Definition

Figure 6-5 show the different parts of a test point, while Table 6-4 describes the parts.

Figure 6-5. Different Parts of a Test Point

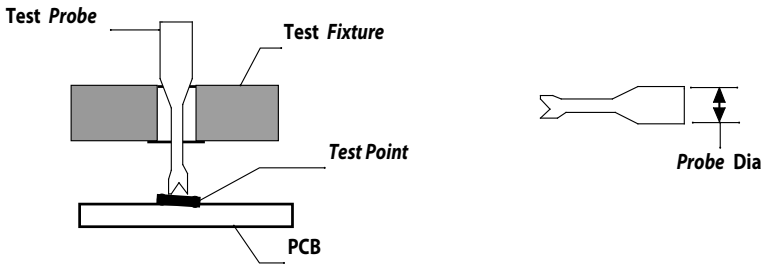


Table 6-4. Parts of a Test Point

Part of Test Point	Description
Test Probe	Also known as the probe, nail, nail pin, or tester pin. This object accesses the test point through the test fixture and makes contact between the test point on the PCB and the test equipment.
Test Fixture	A thick metallic plate attached to the ICT equipment that is customized for each PCB. The test fixture accurately positions test probes to their respective test points on the PCB. Test fixtures can be designed for a single side of the PCB, typically the bottom side, or for both sides of the board, which is called a clam fixture.
Test Point	The point on the net you are accessing, typically a via or component pin.
Head Type	The type of head style or contact point on the test probe. The head is the part of the probe that makes contact with the test point. You cannot set the Head Type in PADS Layout.
Nail Diameter	An ASCII string assigned to a test point via or pin that equals the probe diameter.
Nail Number	A unique label assigned to a probe.

DFT-Related Options

In addition to adding more test point capabilities through DFT Audit, PADS Layout contains features in other functionality to support test points. This test point information is discussed with the specific PADS Layout topic and context-sensitive help is available from any dialog boxes you may encounter when working with test points. Table 6-5 shows impacted functionality.

Table 6-5. DFT-Related Options

Option	Impact
Add a test point	Manually adds a test point attribute to an existing via, jumper pin, or component pin.
ASCII I/O	Exports and imports test points.
Cluster Placement and Cluster options	Prompt you when you move, disperse, or collapse clusters with a locked test point.
CAM	Select Items dialog box displays test points. NC Drill Options dialog box plots test point locations.
Compare Test Points	Compares test point locations in two files.
ECO	Delete Connection, Delete Net, Delete Part, and Change Part handle test points differently.
End Test Point	Manually ends a route with a test point via on a dangling route.
Find	Finds by test points.
Modifying	Prompts you if you modify an object with a test point attribute; for example, changing the pad stack of a component pin that is a test point or changing the via type. See the Modifying Via Properties, Customizing Pad Stacks of Decal Pins, and Modifying Pin Properties topics.
Moving	Prompts you if you move a locked test point. This includes all moving commands for via, pin, cluster, union, or reroute, including spinning, rotating, and flipping objects. See the To Move Components, Moving, Dispersing, or Aligning a Component, Cluster, or Union with a Locked Test Point, Cluster Placement, and To Move a Trace Segment topics.
Reports	Extended reports for test points, including more keywords.
Routing tab	Displaying test points and locking test point locations.
SPECCTRA Translator	The SPECCTRA Translator supports via keepouts and a net of unused pins.
Verify Design	Checks for test point probe violations on the entire design.

Design for Fabrication

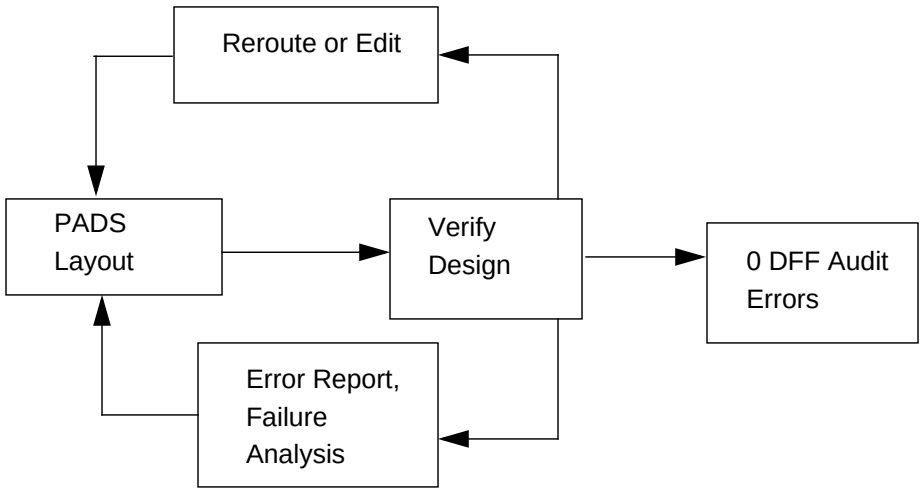
To support fabrication design rules, PADS Layout provides fabrication checks with Verify Design. This functionality, called DFF Audit, lets you either check for fabrication errors within PADS Layout or backward annotate errors from the CAM product, CAM350. DFF Audit detects potential errors in a design, so that you can identify these problems prior to board fabrication.

The checking within PADS Layout uses the CAM document definitions to determine if fabrication errors exist. The CAM documents determine the photoplot output and include layer composites, oversize, suppression, and other masking preferences. All electrical layers are analyzed to check acid traps and copper sliver fabrication. To check mask sliver and solder mask bridge fabrication, the solder mask layers are analyzed. To check silkscreen over pads, silkscreen layers are compared to solder mask layers.

Design for Fabrication Workflow

Figure 6-6 shows the basic workflow for auditing your design for fabrication.

Figure 6-6. Design for Fabrication Workflow



Fabrication Checks Definition

The following sections describe the DFF Audit options in the Fabrication Checking Setup dialog box. The DFF Audit options are:

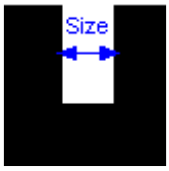
- Acid Traps
- Slivers
- Solder Bridges
- Starved Thermals
- Annular Ring
- Silkscreen Over Pads
- Trace Width/Pad Size

Acid Traps

An acid trap is a location where, due to the surface tension of the etching, acid gets trapped in an area. This acid causes over-etching, which hurts yield. The acid trap runs on all visible electrical layers as defined by CAM documents.

Acid Trap Maximum Size indicates the maximum size of the acid traps to flag. The area of pools that are flagged will be less than this value (Figure 6-7).

Figure 6-7. Acid Trap Maximum Size



Acid Trap Maximum Angle is an angle from 1 to 89 degrees. Any copper items (traces, pads, or any other objects that exist on the layer) that form an angle smaller than this are flagged as an acid trap (Figure 6-8).

Figure 6-8. Acid Trap Maximum Angle

Slivers

Copper slivers (Figure 6-9) are areas in the copper that are so narrow they may flake off. This check detects potential slivers on the electrical and composite layers in the design.

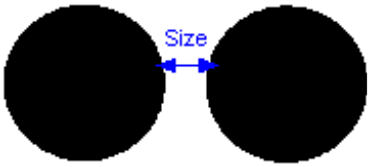
Minimum Copper indicates the maximum size of the copper slivers to flag. This flags slivers of a width less than this value. This check runs on all visible electrical layers as defined by CAM documents.

Figure 6-9. Sliver

Mask slivers in the solder mask layer are areas where the solder mask is so narrow they may flake off. These flakes float around and may drop into an area that needs to be soldered later, resulting in a bad board.

Minimum Mask (Figure 6-10) indicates the maximum size of the slivers to flag. This flags slivers of a width less than this value. This check runs top and bottom solder mask layers, if visible as defined by CAM documents.

Figure 6-10. Minimum Mask Size



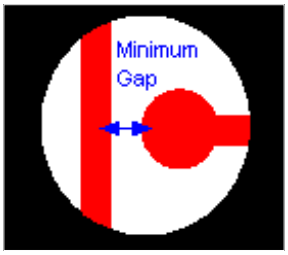
Solder Bridges

When a mask layer is created, openings for pads may be oversized too much and expose an adjacent trace or other conductive object. Therefore, during fabrication, the copper for that pad may become too close and create a bridge to the adjacent object. Solder bridges are usually caused by problems during mask data creation.

The CAD system used may be unable to validate that what was created is going to work.

The Minimum Gap (Figure 6-11) is the maximum distance the solder can bridge and cause a connection to an adjacent object within the same mask opening. If the adjacent object is farther from the pad than this distance, even if the mask layer exposes it, it will not be identified as a bridge.

Figure 6-11. Minimum Gap

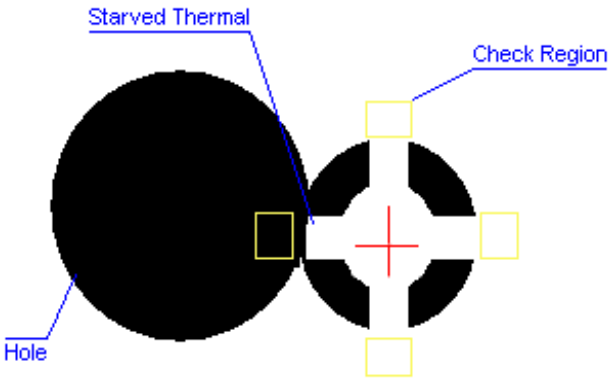


Starved Thermals

Many designs are plagued by thermal pad problems for negative CAM planes because the CAD system did not verify whether the thermals were going to make good connections to the copper plane. The Starved Thermals fabrication check verifies whether each thermal connection to the negative CAM plane is valid, or if it has been constricted by adjacent data that is too close or overlapping – effectively starving out the ties. This check runs on all visible CAM negative plane layers as defined by CAM documents.

Starved Thermal Minimum Clearance is the percentage of the area next to the spoke of the thermal that must not be blocked by another object. Any smaller opening is considered starved.

Figure 6-12. Starved Thermals



Starved Thermal Minimum Spokes is the number of thermal spokes that cannot be blocked by another object. Any less will be considered starved. The number of spokes is specified as EVERY, meaning all spokes must not be blocked, or as an integer from 1 to 4.

Annular Ring

The Annular Ring area lets you set up annular ring checks (described in Table 6-6) by comparing data on different layers. This test checks both the size and the offset between the two layers. Layers to be tested are derived from CAM documents and pad stack data. This area

provides selections for pad, mask, and drill checks. When drill sizes are analyzed for annular rings, the drill oversize setting on the Tools > Options > Design tab is not considered.

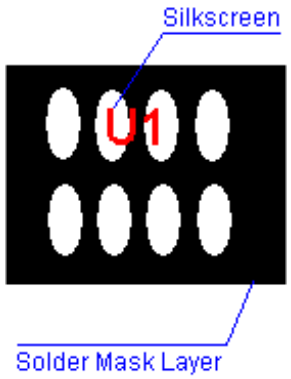
Table 6-6. Annular Ring Checks

Annular Ring Check Type	Checks
Annular Ring–Pad to Mask	The clearance between a pad and its solder mask opening. The offset and the annular ring are checked against the specified clearance value. This check is run on top and bottom electrical layers against their associated solder mask layers.
Annular Ring–Drill to Mask	The clearance between a drill and its solder mask opening. The offset and the annular ring are checked against the specified clearance value. This check is run on the top and bottom drill layers against their corresponding solder mask layers.
Annular Ring–Drill to Pad	The clearance between a drill and its associated pad. The offset and the annular ring are checked against the specified clearance value. This check is run on each specified layer.

Silkscreen Over Pads

The Silkscreen Over Pads check lets you set up the clearance for comparing data on silkscreen layers against top and bottom electrical layers. This check analyzes both the size and the offset between the two layers. Layers to be tested are derived from CAM documents and pad stack data. This check is run for top and bottom electrical layers against their associated silkscreen layers.

Figure 6-13. Comparing Data on Silkscreen Layers



Trace Width/Pad Size

The Trace Width/Pad Size check runs minimum trace width and minimum pad size checks for electrical layers.

The Trace Width check detects small electrical traces on the electrical layers in the design. Minimum Trace indicates the maximum size of the traces to flag. Traces with a width less than this value will be flagged. This check runs on all visible electrical layers as defined by CAM documents.

The Pad Size check detects small pads on the electrical layers in the design before the board is manufactured. Minimum Pad indicates the maximum size of the pads to flag. Pads with a diameter less than this value will be flagged. This check runs on all visible electrical layers as defined by CAM documents.

Chapter 7
ECO Process

This chapter discusses Engineering Change Order (ECO) operations, the types of data that are included when comparing a design and a schematic, and the file formats used when updating a design with changes from a schematic or updating a schematic with changes from the design.

ECO operations include any processes that modify the connection list or parts list. These operations include deleting, adding, and changing various aspects of decals, parts, nets, pin pairs, pad stacks, attributes, or design rules. You must be in ECO mode before you can make such edits. In ECO mode, PADS Layout records your changes in a text file called the ECO file. You can use the ECO file as a reference to update, or backward annotate, the schematic.

Tip: You can change alternate decals outside of ECO mode allowing attribute changes even if the attribute is ECO-registered. For example, you can change the decal for U1 from a DIP 14 with a Geometry.Height attribute set at 200, to a SOIC 14 with a Geometry.Height attribute set at 100. Because you are not in ECO mode, the change is not recorded in the ECO file. After changing decals, locate possible errors of this type by comparing the designs using the Compare/ECO Tools dialog box in PADS Layout or the ECOGEN executable file in DOS.

ECO Registration

ECO-Registered Parts

A part is ECO-registered when you edit the part with the Library Manager, select ECO Registered Part on the General tab of the Part Information dialog box, and then save the part to the library.

When updating a design with changes from a schematic or updating a schematic with changes from a design, you can exclude or include non-ECO-registered parts from ECO processing. In the ECO Options dialog box, do one of the following:

- If you want ECO processing to exclude non-ECO-registered parts, select Output Only ECO Registered Parts.
- If you want ECO processing to include non-ECO-registered parts, which includes non-electrical parts, clear Output Only ECO Registered Parts.

You must be in ECO mode to add, delete, rename, or alter a part regardless of whether it is registered or not.

Avoid connecting non-ECO-registered parts, such as mechanical hardware, to ECO-registered netlist items, such as GND. A conflict may occur when an ECO-registered item is connected to a non-ECO-registered item.

ECO-Registered Attributes

Both ECO-registered and non-ECO-registered attributes can be added, deleted, or changed in ECO mode. To turn on ECO Registration for attributes, use the Objects tab of the Attribute Properties dialog box. Also turn on ECO Registration for any attributes you want to backward annotate to the schematic. Via attributes are not registered attributes, therefore you can add, delete, or change them in ECO mode or non-ECO mode.

You can exclude non-ECO-registered attributes from ECO processing by clearing the Compare Only ECO Registered Attributes option in the Compare/ECO Tools dialog box.

When updating a design from a schematic, a report automatically appears indicating the ECO registration of imported attributes. When an attribute does not exist in the Attribute Dictionary, it is added with ECO registration turned off. If the attribute already exists in the dictionary, the existing attribute and ECO registration in the dictionary are used.

For more information, see the Setting Attribute Properties topic.

Predefined Netnames

When you add a net to the design using the Add Route, Add Connection, or Copy Route command, the Derive Net Name from Pin Function option controls how the added net is named. When this option is enabled, the pin's pin name determines the net name. If no pin name exists, or this option is disabled, PADS Layout automatically generates the net name.

Use the Derive Net Name from Pin Function option when creating the design on the fly instead of creating the design from the netlist generated by the schematic tool. For example, in a typical BGA design you manually connect each die part's substrate bond pad to a BGA component pad using the Add Route or Add Connection command. To give the added net a meaningful name, enable Derive Net Name from Pin Function to derive the net name from the die part pin name, for example, GND.

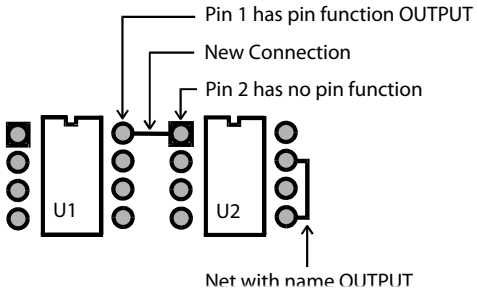
Adding a Connection

This section describes the command behavior for other possible cases of adding a connection between two pins.

In example 1 shown in Figure 7-1, Pin 1 and Pin 2 are not in a net. A pin name (function) exists for one of the pins.

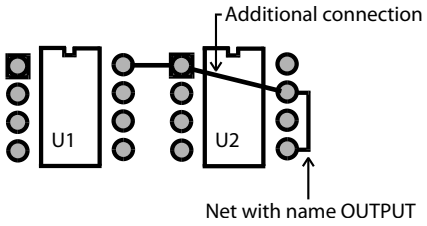
Connecting Pin 1 and Pin 2 opens the Define Name of Net dialog box.

Figure 7-1. Adding a Connection When Pins are Not in a Net



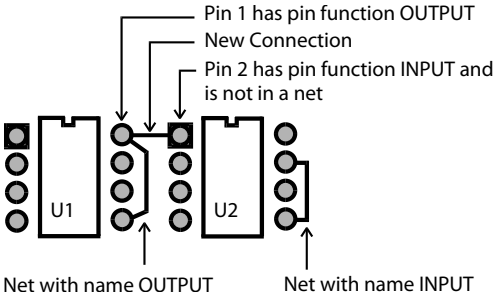
Clicking Add Pins to OUTPUT in the Define Name of Net dialog box adds Pin 1 and Pin 2 to netname OUTPUT (Figure 7-2).

Figure 7-2. Adding Pin 1 and Pin 2 to Netname OUTPUT



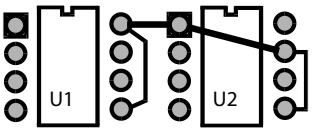
In example 2 shown in Figure 7-3, one of the pins is in a net. One of the pins has a pin function defined.

Figure 7-3. Adding a Connection When One of the Pins is in a Net



When you connect Pin 1 and Pin 2, the Define Name of Net dialog box appears. Choose the name you want to assign to the merged nets. The nets are then merged as shown in Figure 7-4.

Figure 7-4. Merging Nets

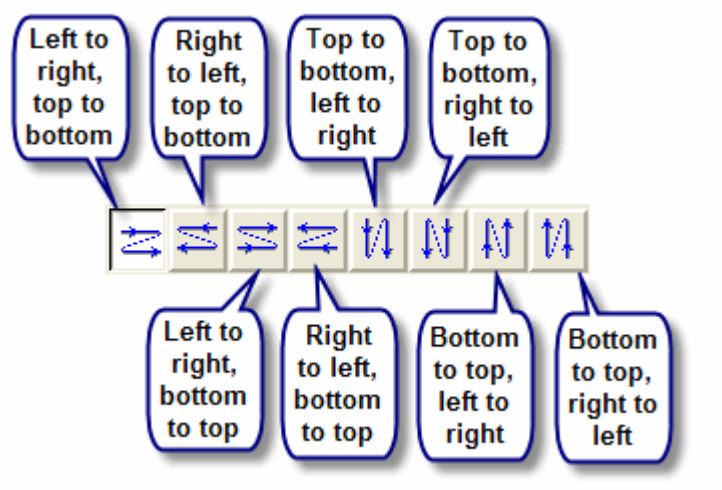


AutoRenumber Sweeps

The result you get when you autorenumber depend on the matrix created by the cell size and the directional pattern of the renumbering sweep. The values used in the Cell Size area of the AutoRenumber dialog box allows you to create custom renumbering sweeps of the board. Here we provide two usage examples and then recommend a best practice.

You can select one of eight different directional numbering patterns. Choose the pattern that best suits the orientation and shape of your board.

Figure 7-5. AutoRenumber Patterns

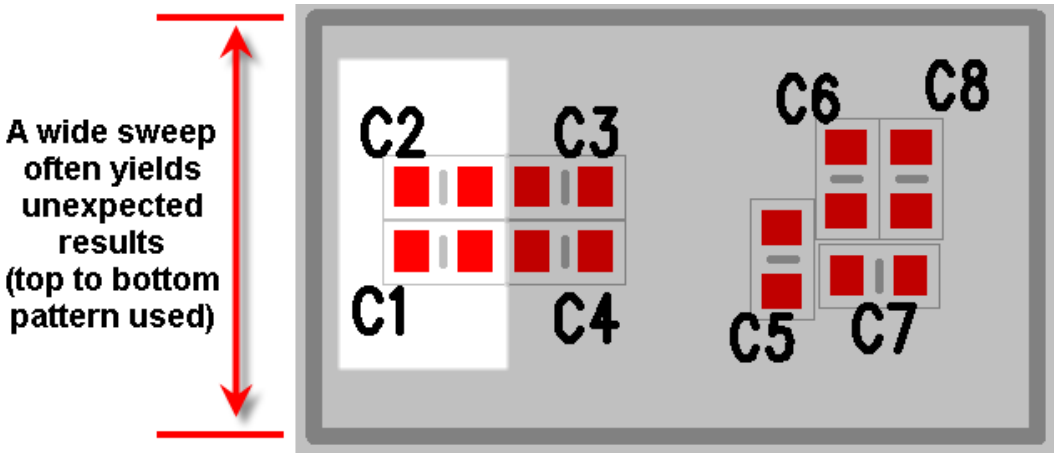


Example 1

In this example, you accept the default *Cell Size* settings of the AutoRenumber dialog box.

The *On x* and *On y* values are set to encompass your entire board. This produces one long, wide sweep. The result of using a wide sweep frequently results in unexpected numbering of the reference designators.

Figure 7-6. Wide Sweep Results

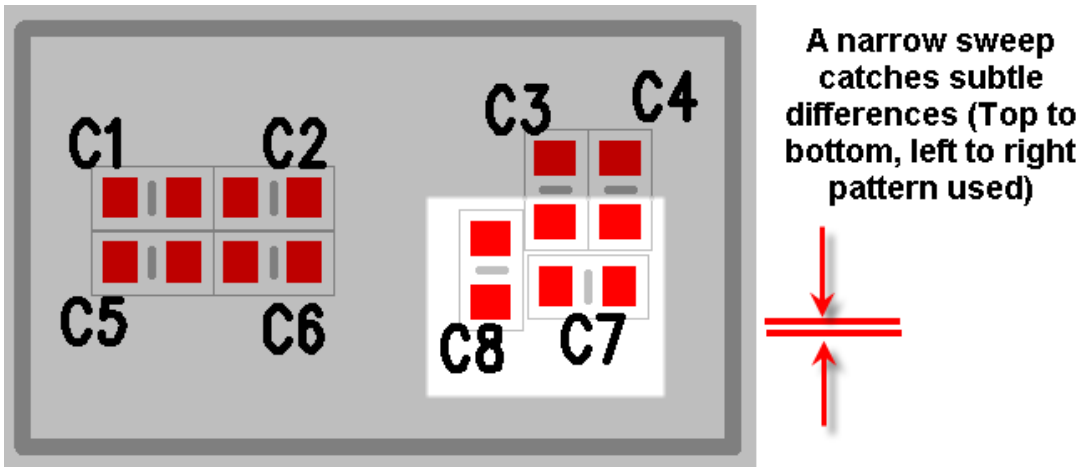


Example 2

In this example, you type a small value for the *On y* value of the *Cell Size*.

The On y value is something small like 10 mils and it catches even the smallest variations in the alignment of components. The result of using a sweep that is too narrow results in numbering of components that don’t appear to be offset.

Figure 7-7. Narrow Sweep Results



Best Practice

Based on the examples shown, the best practice to follow with renumbering is to create cells that do span your entire board, but in narrow sweeps. You may need to adjust the width of your sweep to achieve the desired result. When renumbering, the sweep uses the bottom pin of components as the origin.

For example, your board measures 4000 mils wide and 2000 mils high. You are renumbering left to right, top to bottom. You want to set your cell size to *On x*=4000 and *On y*=10.

Tip: If the reference designator renumbering isn't exactly perfect, you can always manually renumber a few components that weren't renumbered according to what you expected.

Related Topics

Changing the Reference Designators of Multiple Components in ECO Mode (Autorenumbering)

Recording Versus Generating an ECO File

It is always best to record the engineering changes you make to your design. When you record the changes, an exact list is kept of the before and after change to objects.

For example, the .eco file will list that capacitor C42 was renamed to C71. If instead, you generate a list of engineering changes by simply comparing two designs, there is no way for the software to determine the changes that were made to parts that are electrically identical. If there are four identical capacitors in the design with identical connections, it might report that C28 changed to C71 when that isn't exactly what happened. Electrically, there's nothing wrong with these arbitrary decisions since the capacitors have the same value and are attached to the same

nets. But if you compared how those capacitors are placed in the layout relative to how they’re placed in the schematic, it might not make sense and it could lead you to be concerned that something terrible has occurred with the synchronization of the design.

So, depending on how you create the .eco file, a nenumbered back annotation can have different effects, one of which can be inconvenient to cross probing or double-checking the results of the renumbering - comparing the schematics to the layout. Using a recorded .eco file is the best option and can be used in all three methods of back annotation including manual and automated methods. If you record the renumbering of the reference designators in the .eco file while you use the autorenumber tool, you will have a perfect (was...is) before and after listing of the changed reference designators.

But if you generate the .eco by comparing netlists, you will not have an exact (was...is) before and after list of the discretes in the schematic. Discretes that are identical components and have identical net connections are indistinguishable in the netlist. Electrically the changes will be correct. But since there is no placement information in netlists, an .eco file generated by comparison won’t be able to determine the exact refdes exchange of these discretes.

Comparing and Updating Designs

You can compare two versions of a design and create the files needed to update the original design to match the new design. Before comparing a schematic to a PCB layout, create a PADS-format ASCII netlist file (.asc) by generating a netlist in PADS Logic, DxDesigner, or other schematic tool.

When you compare the updated schematic to the original PCB layout and then update the PCB layout to match the schematic, the process is called *forward to layout* or *forward annotation*. Similarly, when you compare the updated PCB layout to the original schematic and then update the schematic to match the PCB layout, the process is called *backward from layout* or *backward annotation*.

If PADS Layout and the schematic tool are on the same computer, you can use more convenient automated tools to compare and update design versions. If PADS Layout and the schematic tool are not on the same computer, you can use PADS Layout to compare two versions of the design. See the following for more information:

- For DxDesigner, see Working with DxDesigner.
- For PADS Logic, see Working with PADS Logic.

Design comparison can handle unused pins nets. An unused pins net contains all the component pins with no assigned net and groups them into one large net.

Design comparison does not do any of the following:

- Add pins removed from logic nets to the unused pins net.
- Use the reuse definition; the actual elements in the physical design reuse are used during comparison.

During comparison, it is assumed the new design contains the most current Attribute Dictionary. If an attribute is not ECO registered in the new design, the attribute is backward annotated only if you clear Compare only ECO Registered Parts check box on the Comparison tab of the Compare/ECO Tools dialog box. If an attribute is ECO registered in the new design, the attribute is backward annotated and the value in the old design is updated, but the ECO registration for the attribute in the old design is not updated.

Attribute values for the Number, Decimal Number, or Measure type properties are automatically converted during the ECO process. For example, when a frequency value is entered as 100 at the schematic or library, it is converted to .1 kHz by default. Also, leading and trailing zeroes are truncated. For example, the decimal number 123.400 becomes 123.4. Although these conversions are correct, design comparison and the ECO process detect and report these conversions as differences. Therefore, a design populated with attributes could have thousands of warnings.

To avoid these warnings, you can do one of the following:

- Define your attributes as Free Text type in the Attribute Dictionary. To take advantage of the math functions in the Attribute Manager dialog box, go to the Attribute Dictionary and change the type to Number, Decimal Number, or Measure. Then, before comparing or beginning an ECO, set the type back to Free Text.
- To use the Number, Decimal Number, or Measure types, make sure the attributes are ECO registered and then update the schematic with changes from PADS Layout. The values are converted in PADS Layout and can be transferred back to the schematic. This synchronizes the schematic and PCB layout.

Differences Report

The Differences Report (Layout.rep) reports the results of a comparison of two design files. (You perform the comparison using **Tools > Compare/ECO** or the ecogen command in DOS.) For example, you can compare a new design (one with changes) to the original design (before changes). The Differences Report lists the changes found in the new design as compared in the original design.

The Differences Report has the following sections:

Part Differences

Lists part type information and part placement information in separate sub-sections.

The part type information sub-section lists the reference designator and the part type for both the old and new designs. Parts that exist only in the old design are listed under the New Design column as <none>. Parts that exist only in the new design are listed under the Old Design column as <none>. Parts that are renamed are listed on the same line. Parts that have new part types are listed on the same line. Parts that have new assigned decals are listed on the same line. Parts that are identical by reference designator and part type in both designs are not listed.

The part placement information sub-section lists the differences for each part's x/y coordinates, glue status, and mirror (flip) status. Part placement information is reported only for parts that exist in both the old design and the new design.

Net Differences

Lists names of the nets that do not exist. Lists the nets that match, but have different names, including nets in the old design that have been combined in the new design. A net split operation appears as pin differences. Nets are listed alphabetically under the Old Design column, except where multiple nets are combined, when they are listed in succession. Nets that do not exist in the old design are listed at the end of this section.

Swapped-Gate Differences

Lists any gates from the old design that are swapped with gates in the new design. The report lists reference designators for the parent components in the design followed by the pins in the gate.

Swapped-Pin Differences

Lists any swapped pins in the old design that are swapped with pins in the new design. This list provides reference designators for the components in the design followed by the swapped pins.

Unmatched Net Pins in Old Design

Lists any connected pins in the old design that are missing or connected to other nets in the new design. These are the pins that are deleted from nets during the ECO process. This list provides net names in the old design followed by unmatched pins in the net. If the net does not exist in the new design, all pins in the net are listed.

Unmatched Net Pins in New Design

Lists any connected pins in the new design that are missing or connected to other nets in the old design. These are the pins that are added to nets during the ECO process. This list provides net names in the new design followed by unmatched pins in the net. If the net does not exist in the old design, all pins in the net are listed.

Attribute Differences

Lists each object under the following headings: Attribute Name, Old Value, and New Value. Attribute differences are included only for objects that exist in both the old design and the new design. If an attribute is missing in either design, the value is listed as <no attr>. If the attribute exists, but has no value, it is listed as <no value>.

Tip: To generate a report containing the mechanical (nonelectrical) parts in the design, clear Compare Only ECO Registered Parts on the Comparison tab of the Compare/ECO Tools dialog box.

Unmatched Net Pin Pairs in Old Design

Lists any pin pairs in the old design that are missing, connected to other nets, or connected to the same scheduled net in a different place in the new design. These would be the pin pairs that would be deleted from nets during the ECO process.

The report lists net names in the old design followed by the unmatched pin pairs in the net. If the net is missing in the new design, then all the pin pairs in the net are listed.

Unmatched Net Pin Pairs in New Design

Lists any connected pin pairs in the new design that are missing, connected to other nets, or connected to the same scheduled net in a different place in the old design. These would be the pin pairs that would be added to nets during the ECO process.

The report lists net names in the new design followed by the unmatched pin pairs in the net. If the net is missing in the old design, then all the pin pairs in the net are listed.

Rules Differences

This section reports differences in design rules between two designs. This section lists each object that has rules differences as a subheading. The subheading has three columns: Rule Name, Old Value, and new Value. (The Rules Differences section heading shows the units in which the values for rules are displayed.)

The subheading for each object type includes:

- The object type
- The object name in the original design
- The object name in the new design (if the name is different)
- The rule type

For example:

```
RULES DIFFERENCES (Values in mils)
          Old Object Name -> New Object Name -> Rule Type
          Rule Name           Old Value         New Value
NETCLS    CLASS1 -> MYCLS    ROUTING
          PIN_SHARE           ON                OFF
          VALID_LAYERS        Top               Bottom
```

This example shows a routing rule set change for the CLASS1 class. The change renamed CLASS1 to MYCLS and included changed in the pin share and valid layers rules. No other routing rules were changed.

Differences in Net Classes

This section lists names of net classes that:

- Do not exist in one design or another. (Classes that do not exist in the original design appear at the end of the section.)
- Match but have different names

For example:

```
NET CLASS DIFFERENCES
Old Design      New Design
CLASS1          MYCLASS
CLASS2          <none>
<none>          NEWCLS
```

This example shows that in the new design:

- The CLASS1 net class was renamed to MYCLASS.
- The CLASS2 net class was removed.
- The NEWCLS net class was added.

Class Nets That Were Removed

This section reports nets that are in the original design but were removed from the new design. (The net class in the original design has nets either not included in the new design or included in different net classes in the new design.)

This section lists:

- Each net class in the original design from which nets were removed and the names of those removed nets
- All of the nets in the net class if the net class does not exist in the new design

For example:

```
REMOVED CLASS NETS

CLASS1          $$$1879     $$$1906     $$$1928
                $$$1920
CLSMESH         GNC         VCC
```

This example shows:

- Nets $$$1879, $$$1906, $$$1928, and $$$1920 in the CLASS1 net class of the original design are missing from that net class in the new design.

- Nets GND and VCC in the CLSMESH class of the original design are missing from that net class in the new design.

If you use the ECO operation to update the original design, the operation deletes the nets from net classes in the original design.

Class Nets That Were Added

This section reports nets that are not in the original design and are added in the new design. (The net class in the new design has nets either not included in the original design or included in different net classes in the original design.)

This section lists:

- Each net class that has added nets in the new design and the names of added nets
- All of the nets in the net class if the net class is new (does not exist in the original design)

For example:

```
ADDED CLASS NETS

CLASS1          ANDROID     BAJOR     SPOT
                DATA00      DATA01
CLSMESH         ZORG        GND2
```

This example shows that in the new design:

- ANDROID, BAJOR, SPOT, DATA00, and DATA01 nets were added to CLASS1.
- ZORG and GND2 nets were added to the CLSMESH class.

Differences in Pin-Pair Groups

This section lists pin-pair groups that:

- Do not exist in one design or the other. Pin-pair groups that do not exist in the original design appear at the end of the section.
- Match but have different names.

For example:

```
PIN-PAIR GROUP DIFFERENCES
Old Design        New Design
GROUP1            <none>
GROUP2            GROUPB
<none>            NEWGRP
```

This example shows that in the new (changed) design:

- The GROUP1 pin-pair group was removed.
- The GROUP2 pin-pair group was renamed to GROUPB.
- The NEWGRP pin-pair group was added.

Pin Pairs That Were Removed From Groups

This section reports pin-pairs that were in the original design but removed from the new design. (The pin-pair group in the original design has pin-pairs either not included in the new design or included in different groups in the new design.)

This section lists:

- Each group in the original design from which pin-pairs were removed and the names of removed pin-pairs
- All of the pin-pairs in the group if the group does not exist in the new design

For example:

```
REMOVED GROUP PIN-PAIRS

GROUP1         U2.2-U1.2 U3.3-U2.2 U3.2-U4.2
               U1.5-U2.5 U4.5-U1.5
GLMESH         R102.1-C2.2
```

This example shows that in the new (changed) design:

- Five pin-pairs were removed from GROUP1.
- The pin-pair R102.1-C2.2 was removed from the GLMESH group.

Pin-Pairs That Were Added to Groups

This section reports pin-pairs in the new design that are not in the original design. (The pin-pair group in the new design has pin-pairs either not included in the original design or included in different groups in the original design.) An ECO operation adds these pin-pairs.

This section lists:

- Each group with added pin-pairs in the new design and the names of added pin-pairs.
- All of the pin-pairs in the group if the group is new (does not exist in the original design).

For example:

```
UNMATCHED GROUP PIN-PAIRS

G         U2.2-U1.2 U3.2-U2.2
DASL      R29.2-U23.17
```

This example shows that in the new (changed) design:

- Two pin-pairs were added to the G group.
- The pin-pair R29.2-U23.17 was added to the DASL group.

File Formats for Passing Data Between PADS Layout and the Schematic Tool

You can use PADS-format ASCII files to pass data between PADS Layout and the schematic tool. This section describes only the ECO format. Some design modification data, such as design rules, are written to the PADS-format ASCII netlist file. For information about the PADS-format ASCII file format, see the PADS-ASCII Format Specification.

When updating a design with changes from a schematic, you can pass the following data: connectivity changes, part type changes, placement changes, and design rule changes.

When updating a schematic with changes from PADS Layout, you can pass the following data: reference designator renames, gate swaps, pin swaps, net splits/joins/renames, decal assignments, and design rule changes.

Tip: Some schematic capture tools may not support back annotation of all design changes.

Modification Data Types

Several types of modification information can be passed between a design and a schematic. Table 7-1 summarizes the availability of each modification data type when transferring data between PADS Layout and the schematic tool.

Table 7-1. Modification Data Types

Category	Data Type	Data Flow Direction Forward (to layout) or Backward (from layout)
Net Modification Commands	Add pin to net	Forward for all/Backward for PADS Logic only
	Join two nets together	Forward/Backward
	Delete Pin from Net	Forward for all/Backward for PADS Logic only
	Split net into two nets	Forward/Backward
	Rename net	Forward for all/Backward for PADS Logic only
	Net scheduling (ePD only)	Forward
Part Operations	Add part	Forward for all/Backward for PADS Logic only
	Delete part	Forward for all/Backward for PADS Logic only
	Change part type	Forward for all/Backward for PADS Logic only
	Change decal	Forward for all/Backward for PADS Logic only
	Rename part	Forward/Backward
	Move part	Forward for all/Backward for DxDesigner only

Table 7-1. Modification Data Types (cont.)

Category	Data Type	Data Flow Direction Forward (to layout) or Backward (from layout)
Gate and Pin Swapping	Swap two gates Swap two pins	Forward/Backward Forward/Backward
Attributes	Add/Modify/Delete attributes	Forward/Backward
Design Rules	Add/Modify/Delete design rules (contained in PADS-format ASCII netlist file)	Forward/Backward

PADS Layout can process all items and apply corresponding changes to the design when importing the ECO file (forward annotation).

ECO File Format

The ECO file format is similar to the PADS-format ASCII database format. Each type of modified data begins with a line of header information. The key information is located in a header line between asterisks (*).

The starting header, which appears at the beginning of the file, identifies the file as Forward/Backward Annotation file:

```
*PADS-ECO-V3.0[-<units>]*
```

where <units> may be MILS, INCHES, or METRIC.

Once the header is found in an annotation file, the modification information following the header is processed.

The end of the file (EOF) information marker is:

```
*END*
```

An annotation file may have one or more entries from the ECO process, or it may contain more than one section of modification information. The file is read until the EOF marker is reached.

Add Pin to Net

The header line is:

```
*NET*
```

The header line for the net to which you are adding the pin always follows this.

```
*SIGNAL* <netname> [<width>]
```

where:

<netname>	Net to which to add pins
<width>	Trace width with which to associate the connections. <width> is optional and maintained only for compatibility with older software versions. Note: In the case of adding a new net for forward annotation, if the trace width described in the *SIGNAL* line does not match the default value for the design, a special Clearance Rule containing the width value is created and attached to the net. Other rule settings match rule settings from the default PADS Layout design.

Example signal name header:

```
*SIGNAL* VCC
```

The format for the pin information is just a list of pins separated by spaces, on one or more lines; for example:

```
Y4.3  U7.3  U6.1  U6.8  U9.1
```

To add pins to a new net, list the pins under a new *SIGNAL* header.

```
*SIGNAL* GND
U4.7  U6.8  U8.7  U7.8  U9.7
```

If the netname does not currently exist in the design, it is added.

Delete Pin from Net

The header line is:

```
*DELPIN* DELETE CONNECTIONS TO PIN
U17.1 CLK
```

where:

- “U17” is the part reference name.
- “1” is the pin number to disconnect.
- “CLK” is the net to which pin is connected (optional)

To delete multiple pins from the net, list the pins on separate lines under the same *DELPIN* header.

Exclude Nets

The header line is:

```
*EXCNET* MyClass
```

The format for the exclude information is:

```
A02
A03
A04
```

where A02, A03, and A04 are the names of the nets to exclude from the class named MyClass.

List the nets to remove from the class named in the header; only those nets listed are removed, all other nets remain with the class.

This applies to Net Class rules only.

Join Two Nets Together

The header line is:

```
*JOINNET*
```

The format for the join information is:

```
OLDNET0 OLDNET1
```

where OLDNET0 and OLDNET1 are the names of the nets to combine. The new combined net uses the OLDNET1 netname.

A connection is added between the nets using two random pins in the selected nets. The trace width of the added connection is the same as that of a connection in the first net (OLDNET0).

Exclude Connections

The header line is:

```
*EXCCON* GROUP_0
```

The format for the rename net information is:

```
J1.4 U1.9
U6.2 U7.2
```

where J1.4 and U1.9, and U6.2 and U7.2 are the names of the pin pairs to exclude from the pin pair group named GROUP_0.

List the pin pairs to remove from the group named in the header; only those pin pairs listed are removed, all other pin pairs remain with the group.

Split Net into Two New Nets

The header line is:

```
*SPLITNET*
```

The format for the split net information is:

```
*SIGNAL* CLK
```

where CLK is the old netname of the net that was split into two sections.

This is followed by the list of pins remaining in the old net:

```
U15.3  U14.3  U16.3  U5.3  U11.2
*SIGNAL* CLK-A
```

where CLK-A is the name of the new net followed by a list of pins that were separated from the old net.

```
U13.3  U12.3  U18.3  U19.3  U20.3  U21.3
```

To split multiple nets, list the old and new net pin lists under a new *SPLITNET* header.

Rename Net

The header line is:

```
*RENNET*
```

The format for the rename net information is:

```
VCC +3.3V
```

where VCC is the old net name and +3.3V is the new net name.

To rename multiple nets, list the old and new net name pairs on separate lines under the same *RENNET* header.

Add Part

The header line is:

```
*PART*
```

The format for the part information is:

```
U1 74LS00
```

where U1 is the part reference name and 74LS00 is the part type name.

Added parts are placed at the lower left corner of the board outline. If the board outline does not exist, parts are placed at the user-defined origin.

To add multiple parts, list them on separate lines under the same *PART* header.

Delete Part

The header line is:

```
*DELPART*
```

The format for the part information is:

```
U4 74LS02
```

where U4 is the part reference designator to delete and 74LS02 is the part type name (not required).

To delete multiple parts, list them on separate lines under the same *DELPART* header.

Tip: Before deleting pins, disconnect them from nets in the design.

Rename Part

The header line is:

```
*RENPART*
```

The format for the rename part information is:

```
U7 U1
```

where U7 is the old name and U1 is the new name.

To rename multiple parts, list them on separate lines under the same *RENPART* header.

All parts under the same *RENPART* header are renamed simultaneously; therefore, to swap the reference designators between two parts, the following input is accepted:

```
U1 U2
U2 U1
```

To facilitate renaming parts, duplicate names are not checked until all renaming is complete. This lets the example above run without a conflict over U2. If any error is encountered, parts in the list are not renamed.

Change Part Type

The header line is:

```
*CHGPART*
```

The format for the change part information is:

```
U2  7400  7404
```

where:

- U2 is the reference name of the part.
- 7400 is the old part type.
- 7404 is the new part type.

To change multiple parts, list them on separate lines under the same *CHGPART* header.

Change Decal

The header line is:

```
*CHGPART*
```

The format for the change part information is:

```
U2  7400@DIP14  7400@SOIC14
```

where:

- U2 is the reference name of the part.
- 7400 is the old part type.
- 7400 is the new part type.
- DIP14 is the old decal.
- SOIC14 is the new decal.

To change multiple parts, list them on separate lines under the same *CHGPART* header.

Move Part

The header line is:

```
*MOVEPART*
```

The format for the move part information is:

```
U2 [LOC:-2300,-16700,90] [FLP:ON] [GLU:OFF]
```

where:

- U2 is the reference name of the part.
- “LOC:-2300,-16700,90” is the new part location information: x coordinate, y coordinate, and orientation
- “FLP:ON” is the Flip side flag (ON to flip, OFF to not flip)
- “GLU:OFF” is the Glued flag (ON to glue, OfF to not glue)

Tip: While each move part information parameter is optional, at least one parameter must be specified.

The part location coordinates are relative to the design origin.

The part orientation is specified in degrees. The orientation range is 0 degrees to 359.999 degrees, with a precision of 0.001 degree (values are rounded to the closest 0.001 degree).

To move multiple parts, list them on separate lines under the same *MOVEPART* header.

Change Design Origin

The header line is:

```
*MOVEPART*
```

The format for the change design origin information is:

```
ORIGIN:-18000,-16250[,PERM]
```

where:

- “ORIGIN:-18000,-16250” is New design origin information: x coordinate and y coordinate.
- “,PERM” denotes whether to permanently change the design origin (PERM to change permanently, omit PERM to change only until the header of the next ECO command).

The ECO file units apply to the coordinates.

Swap Two Gates

The header line is:

```
*SWPGATES*
```

The format for gate swapping information is:

```
U1.A  U4.B
```

where

- U1 and U4 are part reference designators.
- A and B are gates.

To swap multiple gates, list them on separate lines under the same *SWPGATES* header.

Gates are swapped sequentially, in the order listed.

Make sure that the part types of the parts are the same and that the gates are swappable. If they are not, they will still be swapped, but a warning message appears.

Swap Two Pins

The header line is:

```
*SWPPINS*
```

The format for pin swapping is:

```
U5 1.2
U5 4.5
```

where:

- U5 is the reference designator for the part.
- 1.2 represents the pin pair to swap.
- 4.5 represents the other pin pair to swap.

You can swap pins only within a gate of the same part.

Make sure that the pins are swappable in the part type description. If they are not, they will still be swapped, but a warning message appears.

To swap pins on other parts, list the pin pairs under a new *SWPPINS* header.

General Attributes

The following command adds or modifies a list of attributes for a single object:

```
*SET_ATTRIBUTE* <object type> <object name>
"<attr name>" <value>
```

where:

<object type>	Can be PIN#, PART, PARTTYPE, DECAL, NET, NETCLASS, or PCB
<object name>	Name for the corresponding object, DEFAULT for PCB type
<attr name>	Full, structured name of the attribute, enclosed by quotation marks
<value>	Attribute value written as a text string

For example:

```
*SET_ATTRIBUTE* DECAL DIP14
"Geometry.Height" 300
"CAM.AutoInsertable" NO
*SET_ATTRIBUTE* NETCLASS DATABUS
"CAE.Frequency" 66 MHz
```

You can list multiple attributes on separate lines under the same *SET_ATTRIBUTE* header. A space or spaces separate the attribute name and value strings. Because attribute names can contain spaces, quotation marks are required around the attribute name. The end of the line terminates the attribute value string, so quotation marks are not required around the attribute value string.

Because attribute names themselves can contain quotation marks, repeated quotation marks are used to represent a quotation mark within the attribute name. For example, a yes/no type decal attribute named Double”Quotes can be set in the ECO file with the following statements.

```
*SET_ATTRIBUTE* DECAL DIP14
“Double”“Quotes” NO
```

Attribute types are not tracked in the ECO file. If, during the ECO process, a new attribute name is encountered, Free Text type is assigned and the default attribute hierarchy is applied depending on the object type.

Table 7-2 represents default rules for adding new attributes to the Attribute Dictionary.

Table 7-2. Default Rules for Adding New Attributes to the Attribute Dictionary

Object Type	Hierarchy Level
Pin	None
Net	Net Class
Net Class	None
PCB	None
Component	Part type Decal
Decal	None
Part Type	None

To delete a list of attributes for a single object:

```
*DEL_ATTRIBUTE* <object type> <object name>
<attr name>
```

Example:

```
*DEL_ATTRIBUTE* NET POWER
CAE.Voltage
```

Only the attribute name is required, therefore, the attribute name does not need quotation marks around it.

Rules Changes

The ECO file shows changes when you change design rules in the following ways:

- Add an object to or delete it from a rule set (General Rules only)
- Change values for rules

- Add a rule to or delete it from a design
- Split a rule (General rules only). You might change a single rule with multiple object assignments in the original design to multiple rules with assignment subsets in the new design.

Add Object Assignments to a Rule Set (General Rules Only)

When you add object assignments to a general rule, the ECO file includes the *CREATE_GENERAL_RULES* statement and adds the name of the object to the list of all objects assigned to the rule. For example:

```
*ADD_GENERAL_RULES* CLEARANCE
HIERARCHY_OBJECT NET:OLDNET1
HIERARCHY_OBJECT NET:OLDNET2
HIERARCHY_OBJECT NET:NEWNET
```

This example shows the output in the ECO file for a change in the new design that added a net (NEWNET) to the CLEARANCE rule set. In the original design, this rule set included only two members (OLDNET1 and OLDNET2).

Delete Object Assignments from a Rule Set (General Rules Only)

When you delete object assignments from a general rule, the ECO file shows the *DELETE_GENERAL_RULES* statement and lists a HIERARCHY_OBJECT statement for each object that was deleted. For example:

```
*DELETE_GENERAL_RULES* CLEARANCE
HIERARCHY_OBJECT NET:OLDNET2
```

This example shows the output in the ECO file for a change in the new design that deleted OLDNET2 from the CLEARANCE rule set.

Changes in Values for a Rule

When you modify values for a general or conditional rule, the comparison operation detects those changes. Values changes can occur when you assign objects to or delete object assignments from general rules. When the comparison operation detects a values change between two matched rules, the ECO file includes:

- A *MODIFY_GENERAL_RULES* statement or a *MODIFY_CONDITIONAL_RULES* statement
- A HIERARCHY_OBJECT statement for each object assigned to the rule
- The entire set of values for the new rule

For example:

```
*MODIFY_GENERAL_RULES* CLEARANCE
HIERARCHY_OBJECT NET:NET1

MIN_TRACK_WIDTH 6
REC_TRACK_WIDTH 8
MAX_TRACK_WIDTH 12
TRACK_TO_TRACK 12
VIA_TO_TRACK 12
VIA_TO_VIA 12
PAD_TO_TRACK 12
PAD_TO_VIA 12
PAD_TO_PAD 12
SMD_TO_TRACK 12
SMD_TO_VIA 12
SMD_TO_PAD 10
SMD_TO_SMD 12
...
```

The example shows a change in values for NET1. The output lists all of the values for NET1, including a changed value for SMD_TO_PAD in the new design (10 instead of 12).

Add a Rule

When you add a rule (where no rule of the same type and object assignments already exists), the ECO file includes:

- A *CREATE_GENERAL_RULES* statement or a *CREATE_CONDITIONAL_RULES* statement
- A HIERARCHY_OBJECT statement for each object assigned to the rule
- The entire set of values for the new rule

For example:

```
*CREATE_GENERAL_RULES*  CLEARANCE
HIERARCHY_OBJECT NET:NET3
MIN_TRACK_WIDTH 6
REC_TRACK_WIDTH 8
MAX_TRACK_WIDTH 12
TRACK_TO_TRACK 12
VIA_TO_TRACK 12
VIA_TO_VIA 12
PAD_TO_TRACK 12
PAD_TO_VIA 12
PAD_TO_PAD 12
SMD_TO_TRACK 12
SMD_TO_VIA 12
SMD_TO_PAD 12
SMD_TO_SMD 12
...
```

This example shows a change that added a new clearance rule and assigned NET3 to it.

Split a Rule

During comparison and update, a change can “split” a single general rule with multiple object assignments into multiple rules with assignment subsets. The comparison operation matches the “old” single rule with only one of the corresponding “new” rules. In addition, the operation modifies the matched rule and adds the unmatched new rules.

Sample Backward Annotation File

```
*PADS-ECO-V3.0-mils*
*REMARK*
*SIGNAL* I 12
U17.16 U12.16
*JOINNET*
DATA0DATA5
*RENNET*
SIGHSIGHB
*RENNET*
STROBEL CLK0
DATAADATA0
*DELPIN*
P1.6 ENABLOW
U1.3 ENABLOW
P1.19 CLK
U11.A1 CLK
*PART*
C30 DCAP1
C31 DCAP1
*DELPIN*
U2.1 ADD0
U2.2 ADD1
U4.2 ADD1
U2.3 DATA0
U2.4 STROBB
U4.4 STROBB
U2.5 GND
U2.6 SIGH
P1.4 SIGH
U2.7 GND
U2.8 CLONE
U4.12 CLONE
U2.9 ZED011
P1.5 ZED011
U2.10 INT3
U2.13 ENABL0
U2.11 VCO
U2.12 GND
U2.14 VCC
*DELPART*
U5 7404
*RENPART*
U7 U4
U4 U7
*CHGPART*
U11 74007410
```

```
*END*
```

Chapter 8
Physical Design Reuse

This chapter discusses creating, modifying, and using physical design reuses. A physical design reuse contains a collection of items, called elements. You can manipulate this collection as one item. Physical design reuse elements include: components, routes, vias, and text items. Properties and attributes of these objects, such as test point status, are included in the physical design reuse.

You can also save a physical design reuse to a file, then add it to other designs, effectively reusing proven and tested elements and shortening the design cycle of new designs. A design may have a common circuit that you want to repeat a number of times in the same design. To do this, you can create a physical design reuse. Place the components and interconnect traces for the common circuit, save that as a physical design reuse, and then do one of the following:

- Add the physical design reuse to your design for the repeated circuits.
- Create copies of the reuse using objects that already exist in the design.

Each time you create a physical design reuse, it is assigned a *reuse type*, a unique name that describes it. The reuse type is similar to the library part type. Each physical design reuse also has an origin. The origin is visible only when you reset the origin of the reuse.

Tip: You cannot display the reuse name and reuse type in the design.

Adding a Physical Design Reuse

When you add a physical design reuse to a design, the design and the physical design reuse are compared and results are recorded in the report file. Information is processed in the following order:

- Compare layer definition
- Compare part types
- Compare PCB decals
- Add components
- Add pin pairs
- Add routes and design rules
- Add polygon and text items

When you add a second instance of a physical design reuse to the same design, the checking that occurs (described below) is skipped. However, if you copy a physical design reuse and paste it into a different design, the checking is performed.

A report file is created logging any errors or warnings. The report file is named Layout.err and located in \PADS Projects. If an error is encountered, the add reuse process cancels. If you receive warnings, you can choose whether to cancel the process or add the reuse.

Compare Layer Definition

The layer arrangements in both the physical design reuse and the design are compared. If both are identical, the physical design reuse is added.

Errors include:

- Layer counts not matching.
- Layer types not matching. Documentation layer type differences are not treated as errors. The layer type in the reuse is ignored in this case.

Warnings include:

- Layer names not matching. The layer name in the reuse is ignored. The report file details the layer name change.
- Other Layer Definition Data. Layer thickness, layer associations, and so on, are stored in the reuse, but the design settings are always used. This includes netnames associated with plane layers.

Compare Part Types

The part type names in the physical design reuse and the design are compared. If both are the same, the two part type definitions are compared. The definitions must match exactly, or the process cancels. Any differences found are recorded as errors in the report file for each part type.

Compare PCB Decals

If the PCB decals share the same name in the physical design reuse and in the design, the two decal definitions are compared. The definitions must match, or the add reuse process cancels. Slight differences in the decal definitions are tolerated. The location and orientation of the reference designator, part type, and attribute labels do not have to be identical. The attribute name and value, however, must match. Also, if the decal definitions differ, but the pin counts match, the decal definition in the physical design reuse is added to the design with a new name (its original name with an "A" suffix).

When the reuse is added, the PCB decal definition in the design is used.

When the decal pin counts do not match errors are recorded in the report file. When PCB decals are renamed warnings are recorded in the report file.

Add Components

After the Reuse Properties dialog box settings are accounted for, component elements are added. If a part cannot be added, the add reuse process cancels.

Warnings include:

- Start at, Increment by, or Suffix/Prefix assigns a used reference designator. If a current setting in the Reuse Properties dialog box creates a reference designator conflict, a message immediately appears and you can return to the Reuse Properties dialog box to choose different settings.
- Start at or Increment by assigns a new reference designator. If a reference designator is renamed (for example with a prefix), a message is recorded in the report file.

- Same or Next Highest assigns a new reference designator. If a reference designator is renamed, a message is recorded in the report file.
- Assign Prefix assigns a new reference designator. If a reference designator is renamed, a message is recorded in the report file.
- Assign Suffix assigns a new reference designator. If a reference designator is renamed, a message is recorded in the report file.

Add Pin Pairs

Pin pairs are added to the design. Pin pairs are also added to the design to merge reuse nets with existing nets in the design. If all pin pairs can be added to the design without conflict, the Add Reuse process continues. If a pin pair cannot be added, the Add Reuse process continues but the net is not added to the design.

When a net is renamed, a message is recorded in the report file.

Add Routes and Design Rules

Route elements are added based on the same rules and checking procedures for the route copy command. If any portion of the route patterns in the physical design reuse cannot be added to the design, the Add Reuse process continues and a message is recorded in the report file.

Nets added to the design assume the design rules for the net with which they merge. Otherwise, nets assume the default net rules. Rules saved with the physical design reuse are not used.

Jumpers are added to the design similarly to components and vias. The pad stacks and components are added as defined in the reuse definition, including any settings for thermal status, test point status, and so on.

Warnings include:

- Single pin nets not added without an existing net. Single pin nets are only added to the design when a net of the same name already exists in the design. Otherwise, single pin nets are not added. If single pin nets are removed, a warning is recorded in the report file.
- Via definitions not matching. If via definitions are the same in the physical design reuse and the design, the two via definitions are compared. The definitions must match or the via is added to the design with a new name (its original name with an "A" suffix). When the reuse is added, the via definition in the design is used. If a new via definition is created, a message is recorded in the report file.
- Add route fails. If any portion of the route cannot be added, a message is recorded in the report file describing the failure.

- Add jumper fails. If a jumper cannot be added, a message is recorded in the report file describing the failure.

Add Polygon and Text Items

Polygon and text items are added to the design. There are no restrictions on adding polygons to a design, since adding a reuse must be performed with DRC off.

Tip: Pour cutouts are saved in the physical design reuse; however, board cutouts are not.

Elements in a Physical Design Reuse

The elements below can be included in a physical design reuse. Properties and attributes of these elements, such as test point status, are included in the physical design reuse. Each element is assigned a unique ID in the reuse. You must completely select an item to include it in a reuse. Partially selected items are not included. Physical design reuse elements include:

- Components
- Routing Objects
- Drafting Objects
- Unions and Arrays
- Net-based Design Rules

The physical design reuse opens as a new design. The start-up file information and design rules saved with the physical design reuse load as well. When you load a physical design reuse, single pin components are added to the ends of the single pin nets to preserve the net objects. These components have no drill size, pad size, outline, and so on. They are glued by default, non-ECO registered, and located at the 0,0 origin. The component name includes the entire netname, or reasonable portions of it, to easily identify the component to net associations.

Component Elements

Table 8-1 shows component elements eligible for inclusion in a physical design reuse.

Table 8-1. Component Elements Used in Physical Design Reuse

Eligible Element	Description
Reference Designator	Includes the reference designator of a component.
Part Type Definition	Includes all details of the part type definitions assigned to components.
Location	Includes the location of component elements relative to the origin of the physical design reuse; including X,Y coordinates, rotation, and the side of the design on which the component is mounted.
Reference Designator and Type Label Properties	Includes each component element's reference designator and part type labels.
PCB Decal and Alternates	Includes the PCB decal and alternates assigned to each component element.
Component Attributes	Includes all attributes associated with each component element as well as their labels.
Other Properties	Includes other details of the component and its pins, including the test point status, whether the test point is Top Access, thermal eligibility, and thermal and antipad pad stack for each pin.

Routing Objects

The path of route segments, or the route pattern, is stored with width, layer, and netname information. For a physical design reuse, it is assumed that a net consists of one or more pin pairs. Nets are not included directly in a reuse. Instead, the physical properties of the net such as traces, vias, jumpers, coppers, and signal names are included, so the net is preserved.

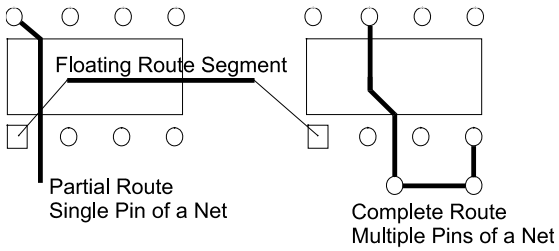
Tip: Because only net properties are included in a physical design reuse, unrouted pin pairs are not included. The signal name of a pin pair, however, is included and is assigned to pins of the pin pair.

Table 8-2 shows route elements you can include in a physical design reuse.

Table 8-2. Route Elements Used in Physical Design Reuse

Route Element	Description
Complete Routes and Route Paths	Contiguous routed or unrouted segments of a pin pair between two component pins. This includes all trace segments, vias, and jumpers.
Partial Routes	Incomplete, but contiguous, route segments of a pin pair that start or terminate on one component pin. The unrouted segments are also saved in the physical design reuse.
Floating Segments	Segments of a pin pair that neither start nor end on a component pin. The path of the segments is stored in the physical design reuse with width, layer, and netname attributes. This may require generating new pin pairs when the physical design reuse is added to the design.
Multiple Pins of a Net	Complete or partial route of the same net, with more than one component pin of the net included in the selection. When included in the physical design reuse, pin pairs may be created between the pins included in the physical design reuse in order to associate and preserve the route paths.
Single Pin of a Net	Partial route of the same net with only one component pin of the net included in the selection. When included in the physical design reuse, a pin pair is created. If you cannot create the pin pair, the route is not included in the physical design reuse.
Vias	Including any via in the route with the via name as an attribute.
Via Definitions	Including the via definition (pad stack) in the physical design reuse for each element added to the physical design reuse.
Via Attributes	Including test point status, Top Access status for test point, thermal status, and user-defined attributes.
Teardrop Settings	Including teardrop status of route end points at vias and pads. Teardrop display is maintained according to the Tools > Options > Routing > Generate Teardrops setting.
Jumper Component and Pins	Including a jumper component with its reference designator, reference designator labels, pad stack definition of its vias (pins), and attributes.

Figure 8-1. Route Elements



Drafting Objects

The definition of the path of the line item, or shape, with width, layer, and netname stored in the physical design reuse. This includes both open and closed shapes.

Table 8-3 shows line elements eligible for inclusion in a physical design reuse.

Table 8-3. Line Elements Used in Physical Design Reuse

Line Element	Description
2D Lines	Including the shapes of all drawing pieces, layer for all drawing pieces, and a list of all text items associated with the 2D line.
Dimensions	Including the X,Y location, layer, and individual dimension element settings, like text and arrow extensions.
Text Items	Including the text string, X,Y location, orientation, layer, height and width, mirror flag, and justification settings.
Complete Copper Polygons	Size and shape of polygons, stored with line width and netname.
Complete Copper Pour Polygons	Shape of pour polygons, stored as corner coordinates with polyline width and netname. Hatch outlines are not included in a physical design reuse. Editing to hatch outline is lost when you add a physical design reuse to a design.
Complete Split Plane Polygons	Shape of a split plane polygon, stored as corner coordinates with attributes for polyline width and netname. Hatch outlines are not stored in a physical design reuse, but pour outlines are.
Cutouts and Keepouts	Any completely selected keepout, copper pour cutout, and copper cutout. Board cutouts are not stored in a physical design reuse.

Board outlines, board cutouts, hatch outlines, hatch voids, via thermals, and pad thermals are not included in the physical design reuse.

Unions and Arrays

Any completely selected union or component array is stored in a physical design reuse. Clusters are not included in a physical design reuse.

Net-Based Design Rules

Design rules for nets and net classes, default rules, and net-based pin pair and group rules are included in the physical design reuse. No other design rules, including pin pair-based rules, are included. Design rules are not added to a design, but are used for comparison purposes.

Make Like Reuse

Make Like Reuse clones a selected physical design reuse using existing components and their logical interconnects as the elements for the physical design reuse. All other physical design reuse elements (such as traces, vias, coppers, 2D lines, and text) are created.

The Make Like Reuse process maps components and interconnects of the physical design reuse circuit to a subcircuit within the design, searching for a match. Components are filtered based on the part type, the number of connections, the decal type, and the value/tolerance attributes. Nets are filtered based on the number of connections. Successive passes filter components and nets based on neighboring net and neighboring component characteristics, including terminal types and connection count. Parallel circuits are handled, and the filtering is independent of reference designators and netnames.

Default reuse properties are assigned to the new physical design reuse, including the reuse name. The Designator Enumeration setting and Net Preferences are ignored since the elements already exist in the design.

Make Like Reuse skips glued components, components that are union or cluster members, component elements in another physical design reuse, and components with attached traces.

You can create a like physical design reuse using any of the following:

- Shortcut menu command
- Make Like Reuse in Verb Mode
- Make Like Reuse in Object Mode

For more information, see their relevant topics in *PADS Layout Help*.

Deselection Report

Creates the report file report.rep in \PADS Projects and opens the file in the default text editor. The file contains a list of items removed from the selection because they were not valid for inclusion in the physical design reuse.

Tip: The deselection report and the selection report are created using the same filename. If you want to save this file, do so in the default text editor using a different filename.

Selection Report

Creates the report file report.rep in \PADS Projects and opens the file in the default text editor. The file contains a list of items included in the physical design reuse.

Tip: The deselection report and the selection report are created using the same filename. If you want to save this file, do so in the default text editor using a different filename.

Make Like Reuse Report

Creates the report file Layout.err in \PADS Projects and opens the file in the default text editor. The file contains operations and matching results from the Make Like Reuse command.

Chapter 9
Working with Attributes and Labels

This chapter discusses creating, modifying, and using attributes and labels.

You can use attributes to associate information with an object in the design. Attributes are made of two parts, an attribute name and its corresponding value. For example, you can create an IsSMD attribute to keep track of which parts are SMD and which are not.

You can assign attributes to the following objects:

- PCB (the board)
- Part Type
- Decal
- Part
- Net Class
- Net
- Pin, including jumper pin
- Via

Every attribute you add to a design is added to the Attribute Dictionary. Attributes are assigned for the entire design. Once you name an attribute and set its properties, that name and those properties apply throughout the design.

Attributes

Attributes Workflow

This is the general process for adding attributes to a design:

1. Create attributes. You can create attributes using the Attribute Dictionary. You can also create attributes and assign them to objects using the Object Attributes dialog box; however, you cannot modify the properties of the attribute with this dialog box. Therefore, it is easier to create all your attributes in the Attribute Dictionary. For more information, see the Adding a New Attribute topic.
2. Define the attribute properties. You must set the kind of value the attribute should have, the design objects to which you want to assign the attribute, and the hierarchy for the attribute. For more information, see the Setting Attribute Properties topic.
3. Assign attributes to objects in the design. For more information, see the Working with Object Attributes topic.
4. You can assign attributes to multiple objects of multiple types. For more information, see the Using the Attribute Manager topic.
5. When it is necessary to change the attributes assigned to objects use the Properties dialog boxes.

Attribute Hierarchy

The attribute hierarchy is the search order in which PADS Layout searches the database to find an attribute value. You can assign attributes using the default hierarchy or you can change the hierarchy, creating your own search order. The lowest level to which you can assign an attribute is the PCB. An attribute applied to the PCB applies to every object on the board, unless you set an attribute at a higher hierarchy level. When you set an attribute at a higher level in the hierarchy, it overrides the PCB level.

The levels in the attribute hierarchy are object dependent, that is, each object has a different hierarchy. You can modify the hierarchy for every attribute using the Objects tab on the Attribute Properties dialog box. Table 9-1 shows the attribute hierarchy for each object type.

Table 9-1. Attribute Hierarchy

For Object	Hierarchy is
PCB	None. This is the lowest level. Attributes assigned at other levels in this hierarchy override attributes assigned at this level.
Part Type	PCB
Decal	PCB
Part	Decal Part Type PCB
Net Class	PCB
Net	Net Class PCB
Via	Net Net Class PCB
Pin	Net Net Class Part Decal Part Type PCB

If you assign attributes to multiple levels and then delete an attribute, the attribute from the next level in the hierarchy is assumed. For example, if you assign a component attribute at the Part Type level and at the PCB level, and you delete the attribute at the Part Type level, the attribute at the PCB level is then applied to the component.

Passing Attributes

As shown in Table 9-2, you can pass attributes between PADS Layout and many other programs. PADS Layout provides a default set of units (and unit prefixes) that are accepted as input and used for output. For more information, see "Default Units" on page 186.

Table 9-2. Passing Attributes to Other Programs

Program	Attribute Passing
DxDesigner	You can pass certain attributes between PADS Layout and DxDesigner. This means you can send attributes to and accept attributes from DxDesigner. For more information, see "Passing Attributes Between DxDesigner and PADS Layout" on page 243.
PADS Logic	You can pass the Value and Tolerance part type attributes in PADS Logic forward to PADS Layout as part of a netlist. PADS Layout can accept the attributes from the netlist.
IDF	The default attribute Geometry.Height is automatically exported to IDF. This attribute replaces the ZHEIGHT functionality used in previous versions of PADS Layout. For more information, see "Exporting IDF Files" on page 319.
BoardSim	You can pass the default attributes Value, Tolerance, Voltage, and PowerGround to BoardSim. For more information, see the Creating HyperLynx BoardSim - HYP Files topic.

Attribute Dictionary

Use the Attribute Dictionary to create attributes in a design. PADS Layout provides default attributes that are applied to every new design. Although the attributes are provided, they are not assigned to any objects.

You can automatically load attributes for part types and decals from the current libraries into PCB designs using the Attribute Dictionary dialog box. You can load attributes when you open files or you can load the attributes after you open a file. When you load attributes, the following actions occur:

- The Attribute Dictionary is updated with new attributes.
- Each new attribute uses the default hierarchy.
- Attributes are added to the current part types and decals, as appropriate, in the open design.
- ECO registration is turned on for all new attributes.
- ECO commands are not stored in the ECO journal file for these updates.

For more information, see the Using the Attribute Dictionary topic.

Default Attributes

PADS Layout provides default attributes that apply to every design. You can change the default attribute dictionary to match library attributes or to suit other design needs. For more information, see the Setting Attribute Properties topic.

PADS Layout also provides a default set of units (and unit prefixes) that are accepted as input and used as output. For more information, see "Default Units" on page 186.

Do not edit the ASSEMBLY_OPTIONS attribute. PADS Layout automatically maintains this attribute.

Default Attribute Properties

Table 9-3 lists default attribute properties. In the column titles, S = System, H = Hidden, RO = Read-only.

Table 9-3. Default Attribute Properties

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
Value	Free Text***	Part, Part Type	Yes	Part, Part Type	Yes	No	No
Tolerance	Free Text***	Part, Part Type	Yes	Part, Part Type	Yes	No	No
HyperLynx. Model	Free Text***	Part, Part Type	Yes	Part, Part Type	Yes	No	No
HyperLynx. Model File	Free Text***	Part, Part Type	Yes	Part, Part Type	Yes	No	No
HyperLynx. Function	List*	Pin	Yes	None	Yes	No	No
HyperLynx. Frequency	Frequency (Measure)	Net, Net Class	Yes	Net, Net Class	Yes	No	No
HyperLynx.Duty Cycle	Percentage (Measure)	Net, Net Class	Yes	Net, Net Class	Yes	No	No
HyperLynx.Type	List**	Net, Net Class	Yes	Net, Net Class	Yes	No	No
HyperLynx. Default IC. Model	Free Text***	Net, Net Class	Yes	Net, Net Class	Yes	No	No

Table 9-3. Default Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
HyperLynx. Default IC. Model File	Free Text***	Net, Net Class	Yes	Net, Net Class	Yes	No	No
HyperLynx. Default IC. Model Pin	Free Text***	Net, Net Class	Yes	Net, Net Class	Yes	No	No
Part Number	Free Text***	Part, Part Type	Yes	Part, Part Type	No	No	No
Description	Free Text***	Part, Part Type	Yes	Part, Part Type	No	No	No
Cost	Free Text***	Part, Part Type	Yes	Part, Part Type	No	No	No
Manufacturer #1	Free Text***	Part, Part Type	Yes	Part, Part Type	No	No	No
Manufacturer #2	Free Text***	Part, Part Type	Yes	Part, Part Type	No	No	No
DIE.xxx	Free Text***	Decal	No	Decal	Yes	Yes	Yes
ASSEMBLY_ OPTIONS	Free Text***	PCB, Part	Yes	PCB	Yes	Yes	Yes
PowerGround	Yes/No	Net, Net Class, PCB	Yes	Net, Net Class	Yes	No	No
Voltage	Measure	Net, Net Class	Yes	Net, Net Class	Yes	No	No
Geometry. Height	Size/Dim. (Measure)	PCB, Part, Decal, Part Type	No	Part, Decal	Yes	No	No

*HyperLynx.Function Values:

SIM_OUT
SIM_BOTH
SIM_IN

**HyperLynx.Type Values:

Clock
Strobe
Data

Address
Power Supply
Analog High Speed
Analog Low Speed
Do Not Analyze

*** Free Text attributes are not case-sensitive.

Other Attribute Properties

Table 9-4 list other attribute properties. In the column titles, S = System, H = Hidden, RO = Read-only.

Table 9-4. Other Attribute Properties

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
AutoDimensioning. Line_Layer	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_Layer	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning.Arc_RadiusMode	Yes/No	PCB	No	PCB	Yes	No	No
AutoDimensioning. Marker_Shape	Yes/No (each)	PCB	No	PCB	Yes	No	No
AutoDimensioning. Marker_Size	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Marker_Width	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_Shape	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_Length	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_Size	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_LineWidth	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_TailLength	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Arrow_TextGap	Measure	PCB	No	PCB	Yes	No	No

Table 9-4. Other Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
AutoDimensioning. Text_Height	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_Width	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_Suffix	Free Text†	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_NumberPrecision	Measure*	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_AngularPrecision	Measure*	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_ DisplacementCase	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_ DisplacementValue	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_ DefaultOrientation	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_DefaultPosition	Number	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_ManualMove	Yes/No	PCB	No	PCB	Yes	No	No
AutoDimensioning. Text_NoGenerate	Yes/No	PCB	No	PCB	Yes	No	No
AutoDimensioning. Extension_Draw1	Yes/No	PCB	No	PCB	Yes	No	No
AutoDimensioning. Extension_Draw2	Yes/No	PCB	No	PCB	Yes	No	No
AutoDimensioning. Extension_Width	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Extension_ PickPointGap	Measure	PCB	No	PCB	Yes	No	No

Table 9-4. Other Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
AutoDimensioning. Extension_LineGap	Measure	PCB	No	PCB	Yes	No	No
AutoDimensioning. Preview_Type	Number	PCB	No	PCB	Yes	No	No
DFT.Nail Count Per Net	Number	Net, Net Class, PCB	Yes	Net, Net Class	Yes	No	No
DFT.Nail Diameter	Free Text†	Pin, Via	No	Pin, Via	Yes	No	No
DFT.Nail Number	Free Text†	Pin, Via	No	Pin, Via	Yes	No	No
DFT.Generate Test Points	Yes/No	PCB	No	PCB	Yes	Yes	Yes
DFT.Probe to Trace Clearance	Size/Dim. (Measure)**	PCB	No	PCB	Yes	Yes	Yes
DFT.Probe to Pad Clearance	Size/Dim. (Measure)**	PCB	No	PCB	Yes	Yes	Yes
DFT.Allow Stubs	Yes/No	PCB	No	PCB	Yes	Yes	Yes
DFT.Stub Length	Size/Dim. (Measure)***	PCB	No	PCB	Yes	Yes	Yes
DFT.Use Via Grid	Yes/No	PCB	No	PCB	Yes	Yes	Yes
DFT.Grid X-Coordinate	Size/Dim. (Measure)***	PCB	No	PCB	Yes	Yes	Yes
DFT.Grid Y-Coordinate	Size/Dim. (Measure)***	PCB	No	PCB	Yes	Yes	Yes
Strategy.SplitPairs.Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.SplitPairs. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.SplitPairs. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.SplitPairs. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes

Table 9-4. Other Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
Strategy.SplitPairs. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.Fanout. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Fanout. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.Patterns. Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Patterns. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Patterns. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Patterns. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.Patterns. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Patterns. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.Route.Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Route. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Route. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes

Table 9-4. Other Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
Strategy.Route. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.Route. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Route. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.Optimize. Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Optimize. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Optimize. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Optimize. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.Optimize. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Optimize. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.Miters. Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Miters. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Miters. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.Miters. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.Miters. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.Miters. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes
Strategy.TestPoint.Pass	List****	PCB	No	PCB	Yes	Yes	Yes
Strategy.TestPoint. Protect	Yes/No	PCB	No	PCB	Yes	Yes	Yes

Table 9-4. Other Attribute Properties (cont.)

Attribute	Type	Objects	ECO	Hierarchy	S	H	RO
Strategy.TestPoint. Pause	Yes/No	PCB	No	PCB	Yes	Yes	Yes
Strategy.TestPoint. Priority	Number	PCB	No	Net, Net Class, Part, PCB	Yes	Yes	Yes
Strategy.TestPoint. Intensity	List*****	PCB	No	PCB	Yes	Yes	Yes
Strategy.TestPoint. PlanePriority	Number	PCB	No	PCB	Yes	Yes	Yes

† Free Text attributes are not case-sensitive.

* Possible Dimensioning text number and text angular precision values range from 0 to 8.

** Possible DFT.Probe to trace and probe to pad clearance values range from 1 to 1000.

***Possible DFT.Stub length, grid x-coordinate, and grid y-coordinate values.

****Possible Strategy.XXX.Pass values:

Values shown in Table 9-5 are available for all Strategy.*XXX*.Pass attributes, where XXX is the name of the pass type for PADS Router to perform: Split Pairs, Fanout, Patterns, Route, Optimize, Miters, and Test Point.

Table 9-5. Values for Strategy XXX Pas Attributes

Status	Represents
Done	Indicates PADS Router completed this pass.
Yes	Indicates PADS Router should perform this pass.
No	Indicates PADS Router should not perform this pass.

*****Possible Strategy.XXX.Intensity values:

These values are available for all Strategy.*XXX*.Intensity attributes, where XXX is the name of the pass type for PADS Router to perform: Split Pairs, Fanout, Patterns, Route, Optimize, Miters, and Test Point.

Default Attribute Usage

Table 9-6 shows how default attributes are used.

Table 9-6. Default Attribute Usage

Attribute	Used For
Value	Replaces the value attribute that is usually assigned to the part name on the schematic level. When you bring parts into a design, value is converted to the new attribute format.
Tolerance	Replaces the tolerance attribute that is usually assigned to the part name on the schematic level. When you bring parts into a design, tolerance is converted to the new attribute format.
HyperLynx.Model	Lists the part model in the .ref file used for BoardSim simulations.
HyperLynx.Model File	Lists the part model file in the .ref file used for BoardSim simulations.
HyperLynx.Function	Lists the BoardSim simulation mode for pins, in, out, or bidirectional.
HyperLynx.Frequency	Lists the working frequency for nets in BoardSim simulations.
HyperLynx.Duty Cycle	Lists the percentage of time the signal is in high state. Used for BoardSim simulations.
HyperLynx.Type	Lists the signal type used for BoardSim simulations.
HyperLynx.Default IC.Model	Lists the default IC model for the net. Used in BoardSim simulations.
HyperLynx.Default IC.Model File	Lists the file name for the model in the above attribute. Used in BoardSim simulations.
HyperLynx.Default IC.Model Pin	Lists the specific pin for the model in the above attribute, if applicable. Used in BoardSim simulations.
Part Number	Used for part ordering, accounting, and so on.
Description	Describes the purpose of the part.
Cost	Specifies the cost of the part.
Manufacturer #1	Specifies the primary manufacturer of the part.
Manufacturer #2	Specifies a secondary manufacturer of the part.
ASSEMBLY_OPTIONS	Indicates whether the part is part of an assembly variant. Existing assembly variants are converted.

Table 9-6. Default Attribute Usage (cont.)

Attribute	Used For
PowerGround	Identifies nets as ground and power nets.
Voltage	Describes the voltage of the net.
Geometry.Height	Describes the part height (height above the PCB).

Other Attribute Usage

Table 9-7 shows how other attributes are used.

Table 9-7. Other Attribute Usage

Attribute	Used For
AutoDimensioning.Line_Layer	Indicates the layer on which the dimensioning lines appear.
AutoDimensioning.Text_Layer	Indicates the layer on which the dimensioning text appears.
AutoDimensioning.Arc_RadiusMode	Indicates whether to measure a radius or a diameter when you dimension a circle.
AutoDimensioning.Marker_Shape	Indicates the shapes of the alignment tool, horizontal, vertical, or diagonal crosshairs, and a square or circular bull's-eye.
AutoDimensioning.Marker_Size	Indicates the size of the alignment tool.
AutoDimensioning.Marker_Width	Indicates the line width of the alignment tool.
AutoDimensioning.Arrow_Shape	Indicates whether to draw open arrows, closed arrows, or datum lines.
AutoDimensioning.Arrow_Length	Indicates the length of the arrow.
AutoDimensioning.Arrow_Size	Indicates the width (height) of the arrow.
AutoDimensioning.Arrow_LineWidth	Indicates the line width of the tail and arrow lines.
AutoDimensioning.Arrow_TailLength	Indicates the minimum length of the arrow tail.
AutoDimensioning.Arrow_TextGap	Indicates the spacing between the tail and the measurement text.

Table 9-7. Other Attribute Usage (cont.)

Attribute	Used For
AutoDimensioning.Text_Height	Indicates the height of dimensioning text.
AutoDimensioning.Text_Width	Indicates the width of one character in dimensioning text.
AutoDimensioning.Text_Suffix	Indicates the suffix to appear after the dimensioning measurement.
AutoDimensioning.Text_NumberPrecision	Indicates the number of decimal places, in mils, for linear measurements.
AutoDimensioning.Text_AngularPrecision	Indicates the number of decimal places, in degrees, for angular measurements.
AutoDimensioning.Text_DisplacementCase	Indicates the position of the dimensioning text.
AutoDimensioning.Text_DisplacementValue	Indicates the custom position of the dimensioning text.
AutoDimensioning.Text_DefaultOrientation	Indicates the orientation of the dimensioning text.
AutoDimensioning.Text_DefaultPosition	Indicates the position of the dimensioning text.
AutoDimensioning.Text_ManualMove	Attaches the dimensioning text to the pointer when you add dimensions.
AutoDimensioning.Text_NoGenerate	Creates only extension lines and arrows when you add dimensions.
AutoDimensioning.Extension_Draw1	Draws an extension line for the first point you select.
AutoDimensioning.Extension_Draw2	Draws an extension line for the second point you select.
AutoDimensioning.Extension_Width	Indicates the width of the extension line.
AutoDimensioning.Extension_PickPointGap	Indicates the gap between the selection point and the end of the extension line.
AutoDimensioning.Extension_LineGap	Indicates the overhang of the line beyond the arrow.

Table 9-7. Other Attribute Usage (cont.)

Attribute	Used For
AutoDimensioning.Preview_Type	Indicates the contents of the preview window at various orientations depending on the current preference settings.
DFT.Nail Count Per Net	Indicates the maximum number of test points on a net.
DFT.Nail Diameter	Indicates the probe, or nail diameter, size for a test point.
DFT.Nail Number	Indicates the ID of the probe in the test fixture.
DFT.Generate Test Points	Indicates whether test points should be created.
DFT.Probe to Trace Clearance	Indicates the minimum probe-to-trace clearance.
DFT.Probe to Pad Clearance	Indicates the minimum probe-to-pad clearance.
DFT.Allow Stubs	Indicates whether *stubs* should be created.
DFT.Stub Length	Indicates the maximum stub length.
DFT.Use Via Grid	Indicates whether to use the via grid when adding and placing test points.
DFT.Grid X-Coordinate	Indicates the via grid size along the X axis.
DFT.Grid Y-Coordinate	Indicates the via grid size along the Y axis.
Strategy.SplitPairs.Pass	Indicates whether PADS Router should perform the split pairs pass and whether it is complete.
Strategy.SplitPairs.Protect	Indicates whether PADS Router should protect traces routed during the split pairs pass.
Strategy.SplitPairs.Pause	Indicates whether PADS Router should pause routing after completing the split pairs pass.
Strategy.SplitPairs.Priority	Indicates the routing order of nets for the split pairs pass.

Table 9-7. Other Attribute Usage (cont.)

Attribute	Used For
Strategy.SplitPairs.Intensity	Indicates the *intensity*, or level of effort, for PADS Router to use when performing the split pairs pass.
Strategy.Fanout.Pass	Indicates whether PADS Router should perform the fanout pass and whether it is complete.
Strategy.Fanout.Protect	Indicates whether PADS Router should protect traces routed during the fanout pass.
Strategy.Fanout.Pause	Indicates whether PADS Router should pause routing after completing the fanout pass.
Strategy.Fanout.Priority	Indicates the routing order of nets for the fanout pass.
Strategy.Fanout.Intensity	Indicates the *intensity* for PADS Router to use when performing the fanout pass.
Strategy.Fanout.PlanePriority	Indicates the routing order for plane nets for the fanout pass.
Strategy.Patterns.Pass	Indicates whether PADS Router should perform the patterns pass and whether it is complete.
Strategy.Patterns.Protect	Indicates whether PADS Router should protect traces routed during the patterns pass.
Strategy.Patterns.Pause	Indicates whether PADS Router should pause routing after completing the patterns pass.
Strategy.Patterns.Priority	Indicates the routing order of nets for the patterns pass.
Strategy.Patterns.Intensity	Indicates the *intensity* for PADS Router to use when performing the patterns pass.
Strategy.Patterns.PlanePriority	Indicates the routing order for plane nets for the patterns pass.
Strategy.Route.Pass	Indicates whether PADS Router should perform the route pass and whether it is complete.

Table 9-7. Other Attribute Usage (cont.)

Attribute	Used For
Strategy.Route.Protect	Indicates whether PADS Router should protect traces routed during the route pass.
Strategy.Route.Pause	Indicates whether PADS Router should pause routing after completing the route pass.
Strategy.Route.Priority	Indicates the routing order of nets for the route pass.
Strategy.Route.Intensity	Indicates the *intensity* for PADS Router to use when performing the route pass.
Strategy.Route.PlanePriority	Indicates the routing order for plane nets for the route pass.
Strategy.Optimize.Pass	Indicates whether PADS Router should perform the optimize pass and whether it is complete.
Strategy.Optimize.Protect	Indicates whether PADS Router should protect traces routed during the optimize pass.
Strategy.Optimize.Pause	Indicates whether PADS Router should pause routing after completing the optimize pass.
Strategy.Optimize.Priority	Indicates the routing order of nets for the optimize pass.
Strategy.Optimize.Intensity	Indicates the *intensity* for PADS Router to use when performing the optimize pass.
Strategy.Optimize.PlanePriority	Indicates the routing order for plane nets for the optimize pass.
Strategy.Miters.Pass	Indicates whether PADS Router should perform the miters pass and whether it is complete.
Strategy.Miters.Protect	Indicates whether PADS Router should protect traces routed during the miters pass.
Strategy.Miters.Pause	Indicates whether PADS Router should pause routing after completing the miters pass.

Table 9-7. Other Attribute Usage (cont.)

Attribute	Used For
Strategy.Miters.Priority	Indicates the routing order of nets for the miters pass.
Strategy.Miters.Intensity	Indicates the *intensity* for PADS Router to use when performing the miters pass.
Strategy.Miters.PlanePriority	Indicates the routing order for plane nets for the miters pass.
Strategy.TestPoint.Pass	Indicates whether PADS Router should perform the test point pass and whether it is complete.
Strategy.TestPoint.Protect	Indicates whether PADS Router should protect traces routed during the test point pass.
Strategy.TestPoint.Pause	Indicates whether PADS Router should pause routing after completing the test point pass.
Strategy.TestPoint.Priority	Indicates the routing order of nets for the test point pass.
Strategy.TestPoint.Intensity	Indicates the *intensity* for PADS Router to use when performing the Test Point pass.
Strategy.TestPoint.PlanePriority	Indicates the routing order for plane nets for the test point pass.

Assigning Attributes

If the object to which you want to assign an attribute isn't selectable, select the related object according to Table 9-8. For more information, see the Setting Attribute Properties for assigning to the same object types and Using the Attribute Manager topics to assign to multiple object types.

Table 9-8. Assigning Attributes

Object	What to Select
PCB	Any object
Part Type	A design component
Decal	A design component
Part	A design component

Table 9-8. Assigning Attributes (cont.)

Object	What to Select
Jumper	A Jumper
Pin	A Pin
Net Class	A net that is a member of the class
Net	A Net
Via	A Via

Using Attribute Values

Attribute values can be 2047 characters long. You can use any printable character, including spaces, in an attribute value. A space, however, can't be the first or last character in the value. When you enter an attribute value, the exact value you type (in dialog boxes, ASCII files, or the library) is stored. This means that capitalization, leading and trailing zeros, embedded spaces, specific unit prefixes, and the presentation of the Yes/No value are all stored exactly as you type them. Leading and trailing spaces are not saved. Invalid values are not saved.

Exact values are saved for the following attribute types:

- Number
- Decimal umber
- Yes/No
- Measure

Exact values are not saved in the Attribute Dictionary entry for the List attribute type or when you set Limits for Number, Decimal Number, and Measure attribute types. For the Number attribute type, leading zeros are removed. For Decimal Number and Measure attribute types, leading zeros are removed, trailing zeros after the decimal point are removed, and numbers greater than 14 characters are rounded. Numbers with more than 14 zeros may be converted to scientific notation.

When using attribute values:

- Automation does not pass the exact attribute value for Yes/No, Number, or Decimal Number attribute types. Automation does, however, pass the exact attribute value for the Measure attribute type.
- Automation ignores whether an attribute is read only, system, or hidden. Therefore, Automation can change all attributes (properties, Attribute Dictionary entry, and value), regardless of their state.

Special Attribute Measurements

You can enter complex units like ounces/sq. foot for copper thickness; however, PADS Layout does not input, process, or output prefixes for complex units.

- Size/Dimension. PADS Layout accepts as input, processes, and outputs Size/Dimension units using the values shown in Table 9-9.

Table 9-9. Size and Dimension Measurements

Unit set on Global Tab (Options)	Output Example	Comment
Mils	12 mil	1 mil = $25.4*10^{-6}$ m
Metric	3 mm	1 mm = $1*10^{-3}$ m
Inches	2"	1" = $25.4*10^{-3}$ m

- Percentages. PADS Layout accepts as input, processes, and outputs a percentage, like 10%, if you use the percent symbol (%).

Number/Decimal Number Attribute Values and ECO

Attribute values for the Number, Decimal Number, or Measure type properties are automatically converted during the ECO process. For example, a frequency value, if entered as 100 at the schematic or library, is converted to .1 kHz by default. Also, leading and trailing zeroes are truncated. For example, the decimal number 123.400 becomes 123.4.

Although these conversions are correct, Compare Design (in DxDesigner Link), Compare Netlist, and the ECO process, detect and report these conversions as differences. Therefore, a design populated with attributes could have thousands of warnings. To avoid this, you can either:

- Define attributes as Free Text type in the Attribute Dictionary. When you want to take advantage of the math functions in the Attribute Manager dialog box, go to the Attribute Dictionary and change the type to Number, Decimal Number, or Measure. Then, before performing a comparison or beginning an ECO, set the type back to Free Text.

or

- Use the Number, Decimal Number, or Measure types. Make sure the attributes are ECO-registered and then perform a backward annotation. The values are converted in the design and backward annotated to the schematic. The schematic and PCB layout will now be synchronized.

Exact Attribute Value Examples

Tables 9-10 through 9-13 show examples of exact attribute values.

Table 9-10. Yes/No Examples

You Type	V 3.5 and Higher Import and Export	V 3.0 Imports and Exports
y	Y	Yes
NO	NO	No
true	True	Yes
1	1	Yes

Table 9-11. Number Examples

You Type	V 3.5 and Higher Import and Export	V 3.0 Imports and Exports
0001	0001	1

Table 9-12. Decimal Number Examples

You Type	V 3.5 and Higher Import and Export	V 3.0 Imports and Exports
0001.5	0001.5	1.5
0.123456789	0.123456789	0.123457
0.000001	0.000001	1E-006
1d3	1d3	1000
12.3e7	12.3e7	1.23E+008
121.	121.	121
1.230000	1.230000	1.23

Table 9-13. Measure Examples

You Type	V 3.5 and Higher Import and Export	V 3.0 Imports and Exports
10	10	10V
1000V	1000V	1kV
1e-5V	1e-5V	10uV
12 volt	12 volt	12V
7 MILLIVOLT	7 MILLIVOLT	7mV

List Exception

Although exact List type values are not saved, list entries are changed to match the Attribute Dictionary entry. For example, if the Attribute Dictionary entry for a list type attribute has Intel, IBM, and AMD as list choices, and you enter intel as a value, PADS Layout changes the entry to Intel. The lowercase *i* is changed to uppercase. Table 9-14 lists exceptions for voltage measures.

Table 9-14. Measure, Voltage Attribute Exceptions

You Type	V 3.5 and higher Import and Export	V 3.0 Imports and Exports	Comment
" 10V"	"10V"	"10V"	Spaces before 10V are removed in both 3.0 and 3.5 **Note:** Quotation marks used only to show spaces
"10V "	"10V"	"10V"	Spaces after 10V are removed in both version 3.0 and 3.5 **Note:** Quotation marks used only to show spaces
ten volt			Invalid string. Input is ignored and no value is attached to the attribute.

Measure, Geometry.Height (Size/Dimension) Attribute Exceptions

If a unit of measure is not specified for a Size/Dimension Measure type attribute, PADS Layout does not save the exact value. The value is not saved because a number in the attribute value without a unit creates confusion if you change the current units.

Table 9-15. Measure, Geometry Height Attribute Exceptions

V 4.0 and higher Input		V 4.0 and higher Output for Current Units		
String	**Current Units**	**Mils**	**Inches**	**Metric (mm)**
10	mils	10mil	0.01"	0.254mm
0.1	inches	100mil	0.1"	2.54mm
10	metric	39.37mil	0.3937"	10mm

Tip: Using a non-standard format for your value might cause PADS Layout to change the value even if you do not change the current units. For example entering a value of 0001.2000mil, causes PADS Layout to change the value to 1.2mil.

Default Units

You can include a unit with an attribute value. PADS Layout provides a default set of units (and unit prefixes) that are accepted as input and used as output. PADS Layout uses *Systeme Internationale* units, or SI units. Units are exported with attributes and are converted appropriately. User-defined units (dollar, yen, feet, pound, ounces/sq. foot, and so on) are not converted. Also, you cannot use prefixes with user-defined units.

The following units are supported, but are either enabled or disabled for actual use within PADS Layout. For more information, see "Unit Prefixes" on page 187. To change the units that appear in this list, see "Customizing Units for Attributes" on page 188.

Tip: An extra comma (,) means that you can enter the abbreviation for the unit without a prefix. For example, you can enter O for Ohm in an attribute value; it is a valid value. You cannot, however, add F for Farad in an attribute value; it is not a valid value. You must use Farad with a prefix.

Supported Units

Table 9-16 lists units that are supported in PADS Layout.

Table 9-16. Supported Units

Abbreviation	Unit	Enabled	Allowed Prefixes	Quantity
O	Ohm	Yes	u,m,,k,M,G	Resistance
F	Farad	Yes	p,n,u,m	Capacitance
H	Henry	Yes	n,u,m	Inductance
Hz	Hertz	Yes	,k,M,G	Frequency
A	Ampere	Yes	u,m,,k	Electric current
V	Volt	Yes	n,u,m,,k	Voltage
W	Watt	Yes	p,u,m,,k,M,G	Power
s	Second	Yes	p,n,u,m	Time
g	Gram	No	u,m,,k	Mass
Wb	Weber	No	p,n,u,m	Magnetic flux
T	Tesla	No	p,n,u,m,,k	Magnetic flux density
C	Coulomb	No	p,n,u,m	Charge
S	Siemens	No	n,u,m	Electric conductance

Table 9-16. Supported Units (cont.)

Abbreviation	Unit	Enabled	Allowed Prefixes	Quantity
J	Joule	No	p,n,u,m,,k,M,G	Energy
N	Newton	No	u,m,,k	Force
Pa	Pascal	No	,k,M	Pressure
K	Kelvin	No	u,m,	Temperature
rad	Radian	No	u,m,	Plane angle
sr	Steradian	No	u,m,	Solid angle
cd	Candela	No	u,m,,k	Luminous intensity
lx	Lux	No	u,m,,k	Illumination
lm	Lumen	No	u,m,,k	Luminous flux
mol	Mole	No	u,m,,k	Amount of substance
Gy	Gray	No	U,m,,k	Absorbed dose
Bq	Becquerel	No	U,m,,k	Activity
Sv	Sievert	No	U,m,,k	Dose equivalent
m	Meter	No	P,n,u,m,,k	Distance
l	Liter	No	U,m,	Liquid

Unit Prefixes

Every unit has a set of prefixes that are valid, as shown in Table 9-17. Each prefix has a symbol and is a power of ten. You can't use prefixes with user-defined units, such as dollar, yen, feet, pound, and so on.

Tip: PADS Layout never exports the prefixes h, da, d, and c.

Table 9-17. Unit Prefixes

Symbol	Prefix	Power of Ten
Y	Yotta	+24
Z	Zetta	+21
E	Exa	+18
P	Peta	+15

Table 9-17. Unit Prefixes (cont.)

Symbol	Prefix	Power of Ten
T	Tera	+12
G	Giga	+9
M	Mega	+6
k	kilo	+3
h	hecto	+2
da	deca	+1
d	deci	-1
c	centi	-2
m	milli	-3
u	micro	-6
n	nano	-9
p	pico	-12
f	fempto	-15
a	atto	-18
z	zepto	-21
y	yocto	-24

Customizing Units for Attributes

You can customize (enable or disable) the supported units by modifying the powerpcb.ini file in C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Programs. To customize the .ini file:

1. Open the .ini file in a text editor, such as Notepad.
2. Add a new attribute unit section by typing the header **[SI Units]**.
3. Make modifications as described in the "Enabling Units" and "Disabling Units" topics.
4. Save the .ini file.

Enabling Units

To enable units, delete the *ignore;* variable from the line. For example, the line for the Gram unit reads:

Gram=ignore;u,m,,k

Modify the line so it reads:

Gram=u,m,,k

Disabling Units

To disable units, add the *ignore;* variable to the line. For example, the line for the Farad unit reads:

Farad=p,n,u,m

Modify the line so it reads:

Farad=ignore;p,n,u,m

It is recommended that you leave the unit prefixes intact even when disabling the unit. This makes it easier to enable the unit later because you won't have to specify prefixes again.

.ini File Format for Units

Units are specified in the [SI Units] section of the .ini file in the following format:

```
<full unit name>=[ignore;][input:<prefix list>;][output:
<prefix list>]
```

where:

<full unit name>	Specifies the name of the unit.
[ignore;]	Specifies whether to ignore the unit. If this variable is included, the unit is ignored. To enable the unit remove this variable.
[input:<prefix list>;]	Specifies that you are creating a list of prefixes that are valid for input into PADS Layout.
[output:<prefix list>]	Specifies that you are creating a list of prefixes that are valid for output from PADS Layout.

Tip: You can list the valid prefixes after the equal sign (=) and those prefixes are used for both input and output.

Creating Attributes in the PCB Decal Editor

You can use attributes in the PCB Decal Editor, but they differ in concept from attributes in the Layout Editor:

- The only attributes you can create in the PCB Decal Editor are decal attributes, which are associated with the physical decal.
- The Attribute Dictionary doesn't exist in the PCB Decal Editor; therefore, attributes in the PCB Decal Editor are text strings only. They don't have properties, types, hierarchies, and other settings that attributes in the Attribute Dictionary have.

When you use Edit Decal to enter the PCB Decal Editor, upon exit, you are asked whether to apply the changes you made. If you apply the changes when you return to the Layout Editor, any attributes you added in the PCB Decal Editor are added to the Attribute Dictionary. The attributes are added at the Decal level of the attribute hierarchy and assigned to the appropriate objects.

Tip: When you enter PCB Decal Editor using the Tools menu command, any attributes you create in the PCB Decal Editor are *not* added to the Attribute Dictionary.

If you created an attribute in the PCB Decal Editor that exists in the Attribute Dictionary, the existing attribute in the dictionary is maintained. The attribute is assigned to the part that uses the decal. If a label for the attribute exists, it is associated with the attribute in the Attribute Dictionary.

If you created an attribute in the PCB Decal Editor that does not exist in the Attribute Dictionary, a non ECO-Registered attribute is created with the Free Text type. The attribute is then assigned to the part that uses the decal. If a label for the attribute exists, it is associated with the attribute created in the Attribute Dictionary.

Non-Decal Attributes

A decal label is always associated with an attribute name. At any time, that attribute may be a decal attribute, or it may not. For example, you may create a decal attribute called My Attribute and then create a label for it. If you delete the decal attribute, the label remains, but it is now associated with a non-decal attribute with the name My Attribute.

The reverse is also true. You can create a label and associate it with an attribute that isn't a decal attribute (by choosing an attribute from the Library Attribute Manager or by typing an attribute name in the Add New Decal Label dialog box). The label is now associated with a non-decal attribute.

Labels

Label Defaults

If you create a label, but don't provide display information in the decal or in the component, PADS Layout uses default visibility properties. The default visibility properties match part type display in PowerPCB 2.1. The same position, text height, and width are used.

The default position for the first label is at the decal origin with no orientation, and it uses the default height and width. If a label already exists in the first position, the second position is used. The default position for the second label is under the first label position.

Figure 9-1 shows how a new height attribute label is placed in the second default position. Note that the second position is based on the first position, not where other labels are placed.

Figure 9-1. New Height Attribute Label (Second Default Position)

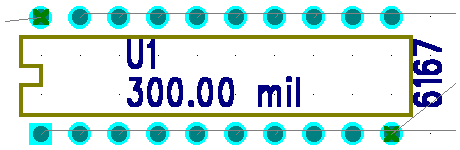
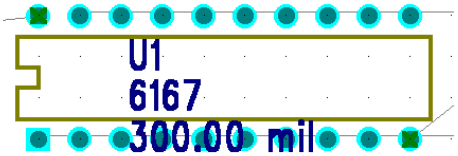


Figure 9-2 shows how a new height attribute label is placed in the third default position because the first and second positions are already filled.

Figure 9-2. New Height Attribute Label (Third Default Position)



Justification Examples

You can justify free text and attribute, reference designator, and part type labels. You can set justification options when you create the label or text. For more information, see Justifying a Label.

In Figure 9-3, a label appears within a part outline. The label uses the default justification, which is Left, Down; meaning that the label was placed by its lower left corner.

Figure 9-3. Label Within a Part Outline

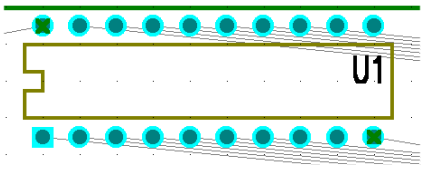


Figure 9-4 shows what happens when you change the Height and Width of the label. The label now overlaps the part outline.

Figure 9-4. Label After Changing the Height and Width (with Overlap)

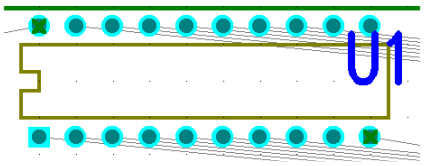
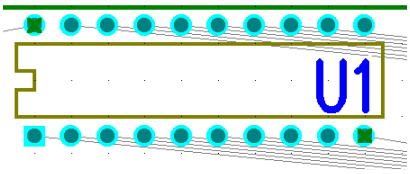


Figure 9-5 shows how you can change the Height and Width of the label and prevent the label from overlapping outlines. By changing the justification to Top, Right you change the location by which the label is placed, preventing overlap.

Figure 9-5. Label After Changing the Height and Width (no Overlap)



Right Reading Examples

You can set the left or right reading status for free text and attribute, reference designator, and part type labels using the Label Properties dialog box. You can also use the Properties dialog box for any of the objects. For more information, see Adding a New Part Label or the appropriate Properties object topic in *PADS Layout Help*.

Figures 9-6 through 9-11 show different reading orientations.

Figure 9-6. Normal Rotation

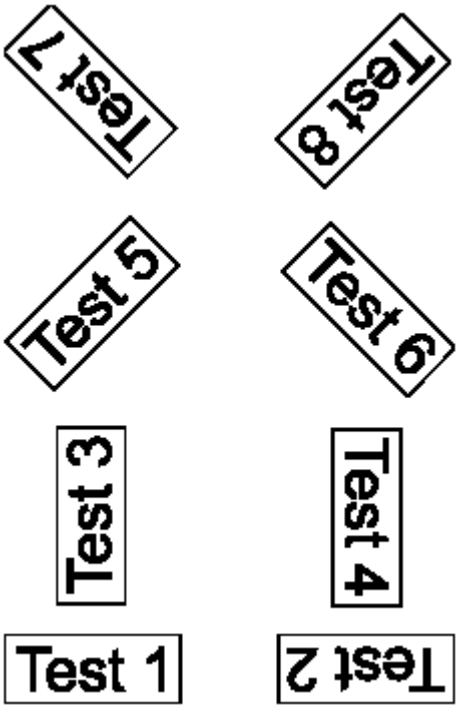


Figure 9-7. Mirrored

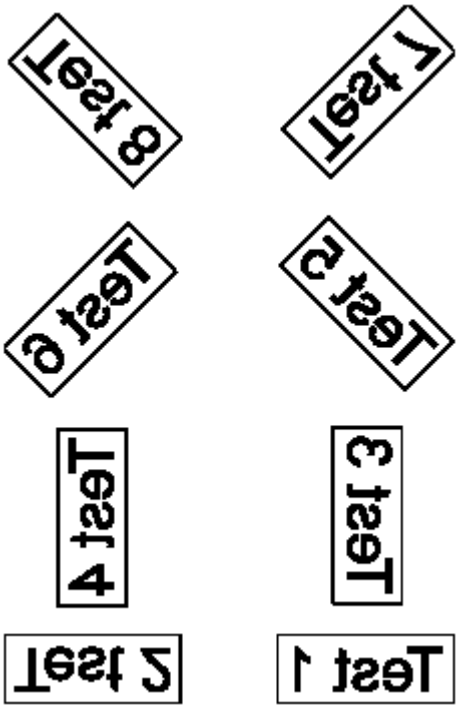


Figure 9-8. Right Reading - Orthogonal

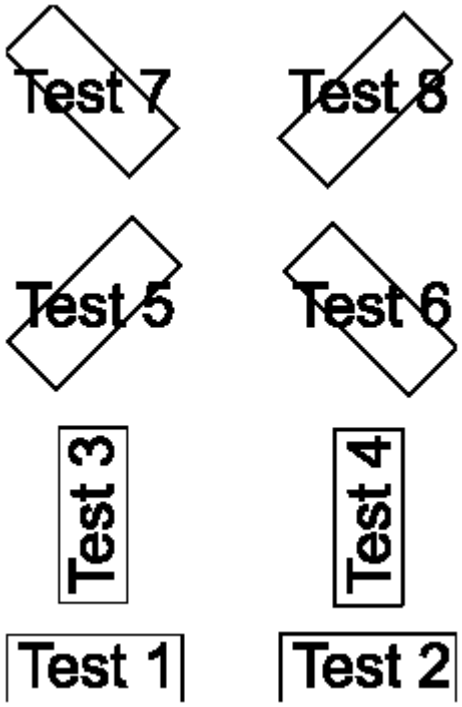


Figure 9-9. Right Reading - Orthogonal, Mirrored

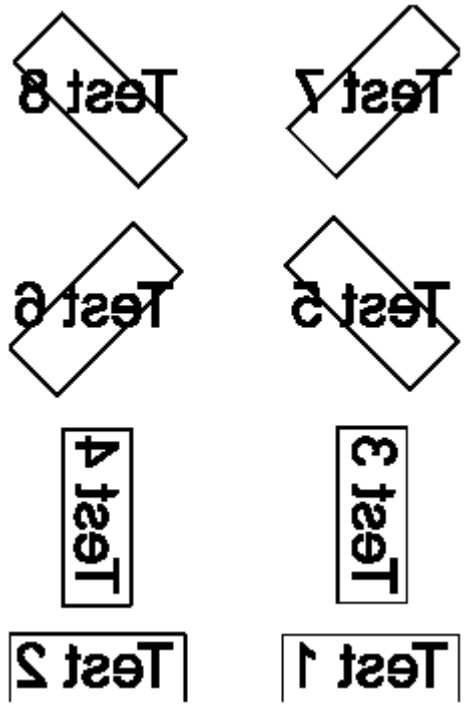


Figure 9-10. Right Reading - Angled

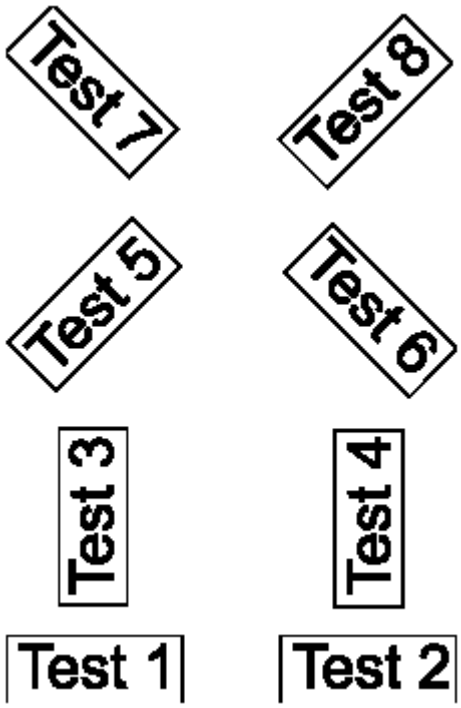
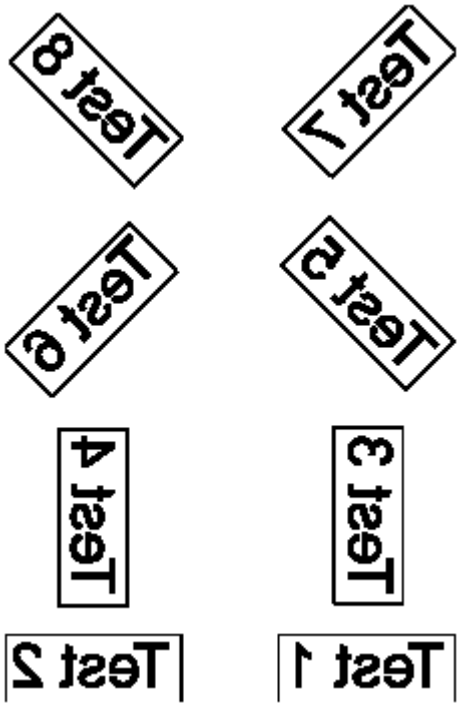


Figure 9-11. Right Reading - Angled, Mirrored



Managing Reference Designators

Since reference designators are now handled the same as attribute labels, you have more flexibility in visibility and placement. You can now:

- Create a reference designator label and set it up for silkscreen layers. For more information, see the To Move Reference Designators for Silkscreens topic.
- Create a second reference designator label and set it up for assembly drawing layers. For more information, see the Generating a Second Set of Reference Designators for Assembly Drawings topic.
- Hide reference designators. For more information, see the To Hide Reference Designators topic.

Labels in the PCB Decal Editor

Work with labels in the PCB Decal Editor exactly as you do in the Layout Editor. Labels in the PCB Decal Editor offer you greater flexibility; you can display either decal attributes or an attribute from an object that uses the decal. You can, for example, create a label for the part attribute Cost. Since Cost is not a decal attribute, you create the attribute in the Attribute Dictionary in the Layout Editor, and assign a placeholder label in the PCB Decal Editor. For more information, see the Creating Placeholder Attribute Labels topic.

When you create labels, they may not be visible. Turn on label visibility in the Display Colors Setup dialog box. On the Display Colors Setup dialog box, you can set the color for reference designators, part type, and attribute labels.

Tip: Labels created prior to version 3.0 for part names, reference designators, and terminal numbers are converted to current version labels when you open the part in the PCB Decal Editor.

If you created a placeholder label for an attribute that exists in the Attribute Dictionary, the label is associated with the existing attribute, but it is not visible. The attribute is not, however, assigned to the part using the decal. You must manually assign it.

Chapter 10
Placing Parts

This chapter explains part placement guidelines, aids to part placement, placement commands, automatic placement, and dispersion.

Parts placement depends on many factors. Besides connection length and spacing tolerances that won't create interference, manufacturing considerations such as accessibility by pick-and-place machines and solder treatment also affect placement. Once you determine your manufacturing strategy and import your netlist, use the interactive placement tools to optimize your placement scheme.

Tools to aid in parts placement include:

- Find function that locates and selects parts by reference designator or location
- Separate grid settings for x and y coordinates, set by a modeless command
- Interactive part alignment function
- Nudge interactive parts shoving, with DRC enabled
- Net length minimization
- Connection length minimization
- Automatic and interactive part, pin, and gate swapping
- Automatic part renumbering

Remember to backward annotate engineering change order (ECO) changes to the schematic.

Placement Guidelines

A good placement scheme:

- Observes all the restrictions placed on component location and trace routing.
- Provides the easiest routing solutions.

Follow this general order for placement:

1. Place all components that have a fixed location, such as connectors, ICs that require a specific location, and mounting holes.
2. Because of their typically heavy connectivity pattern, connectors need sufficient room between other connectors and the first row of ICs to exit the connector. Allow at least .5 between the connector and the nearest row of ICs.
3. Place components that have engineering restrictions, such as parts that must be a certain distance from each other, parts that can't be located next to each other, and parts that must be near the connector.
4. Place components with no overriding placement requirements. Follow these general guidelines:
    - A part's pads should fall on the grid used during autorouting. Some standard increments are 100, 50, 25, 20, 16.67, 12.5, or 10 mils. Set up the design grid before placing components.
    - The flow of connections should be vertical and horizontal. Minimize diagonal connections.

- Component pads should line up vertically and horizontally so they do not block routing channels. If two ICs are side by side with their pads staggered 50 mils, the pads of one IC block the routing channels of the other.
- For manufacturing reasons, it is better to have all of the ICs on the board oriented in the same direction, rather than some vertical and some horizontal.

Placement and Length Minimization

Before you begin placement, set the topology types you want to use for nets, net classes, or the Default level of the rules hierarchy. Topology is the pattern of the trace and the order in which to connect pins in a net. Length minimization reorders pin connections in a net to support the topology you set. Length minimization does not change the netlist, it just finds better places to make connections required by the netlist. For example, if you specify the "minimized" topology for a net, length minimization finds the pin order that produces the shortest pin-to-pin connections.

When you move a part around the layout, length minimization happens on the fly. If the topology type is set to something other than "protected," you can see ratsnest connections linking and unlinking pins on the moved part to the nearest viable terminals on other parts. Also, as you move the part, a running measurement called New Length/Old Length appears on the message line. 100 equals 100 percent of all nets connected to the part when you picked it up. If the percentage becomes less than 100, the length is getting better. If the percentage becomes more than 100, the length is getting worse. When you have a placement you think will work, you can also run length minimization for the entire board by using the Length Minimization command.

Generally, placement that minimizes connection length should be your primary consideration. However, using minimum connection length to determine part location can result in dense areas of connections, usually in the center of the board. These connections can outnumber the routing channels available in the area. Consequently, after you minimize the connection length, study the board to see whether there are critical dense areas. Don't hesitate to make local adjustments to the component locations to spread the connections away from the dense areas.

Part swapping also depends on topology type. No ECO file is required or generated for length minimization or part swapping. On the other hand, gate swapping, pin swapping, and ECL terminator swapping, are all recorded in an ECO file for backward annotation. There may be, however, some high-speed or critical nets in your design for which you want to set to the "protected" topology type before you begin placement.

Controlling Length Minimization

There are two levels of control for length minimization.

- Global options on the Design tab of the Options dialog box that determine if and when length minimization occurs during a move.
- Length minimization, on the fly or during a move, which is more memory intensive.

The more detailed controls are configured on the default or hierarchical rules level, using the Routing Rules dialog box. Once you select a hierarchical level to work on, a single net, for example, you can use Routing Rules to set the topology type for the net. With local controls you can set topology selectively for critical nets. You can set some nets to one topology type and other nets to another type. For more information, see the Routing Rules Dialog Box topic.

If you use the high-speed rules option, you can set topology rules at the hierarchical level.

Placement Related ECOs

An engineering change order (ECO) is a change in design that either comes to the board from the schematic or occurs in the layout and effects the schematic. Because ordinary length minimization and part swapping do not change the netlist, you don't need to backward annotate them to the schematic. You must, however, record in an .eco file length minimization and placement optimization tools like gate and pin swapping and reference designator renumbering. For this reason, these commands are located on the ECO Toolbar. When you access the ECO Toolbar you are automatically prompted to make this file.

Moving Items

With Move by Origin

When you select several objects at once, you can use different origins for the move. If you select Move by Origin in the Design tab of the Options dialog box, the pointer snaps to the center of the group, calculated from the extents of the group, and uses it as the origin for the move. If you clear Move by Origin, the origin is the selection point for a drag move, wherever the pointer is when you click Move.

With Stretch Traces During Component Move

When you select a component with partial traces connected to the pins of the component, the traces stretch to remain connected to the pins as long as you have Stretch Traces During Component Move selected in the Design tab. Each trace connected to a pin of the moving component connects to the new pin position. If DRC is off, straight-line segments are added. If DRC is on, traces are rerouted with smoothed patterns using AutoRoute. When Stretch Traces

During Component Move is not selected, unroutes are created between the tacks of the trace connections.

Interactive Placement Tools

You can place parts to exact tolerances using the alignment and nudge tools.

- Alignment snaps selected parts along an imaginary line coming from the side or center of a guide component.
- Nudge eliminates overlaps and clearance violations by pushing parts aside to make room for a crowded component.

Tip: For nudging, all component elements are considered glued. Align and Nudge are unavailable when you select physical design reuse elements.

Nudging Parts

When parts are close enough to violate either pad-to-pad or body-to-body clearance rules, they are considered overlapping. For more information, see To Nudge Overlapping Parts topic.

Nudge and Design Rule Checking (DRC)

When DRC is set to Prevent, PADS Layout actively prevents overlapping by canceling the completion of an illegal move and returning the selected component to Move mode.

This automatic checking can be further refined using Nudge. Nudge is a shove function for parts; it resolves overlapping by moving surrounding parts away from the part you are trying to place. Nudge has three operational modes that you set on the Design tab of the Options dialog box. As shown in Table 10-1, Nudge operation is dependent on the current DRC setting:

Table 10-1. DRC Setting and Nudge Modes

DRC Setting	Prevent	Warn	Off or Ign Clrn
Auto	Overlapping parts are automatically moved and unhighlighted. If relocation is not possible due to available space, a message appears.		
Warn	The Nudge dialog box appears for user interaction.		

Table 10-1. DRC Setting and Nudge Modes (cont.)

DRC Setting	Prevent	Warn	Off or Ign Clrn
Off	No Nudge operations occur. The selected part remains attached to the cursor.	The Clearance Violation dialog box appears. The choices are: Ignore–Places part without adjusting overlapping parts. Explain–Lists the intersecting parts. Cancel–Cancels the command; the part remains attached to the cursor.	The part is placed without moving overlapping parts.

When you use Nudge for placement, PADS Layout approximates the component size. For example, if you placed an L-shaped component, PADS Layout approximates that part area as a square, not as an L. Nudge then uses the greater of the pad-to-pad or body-to-body clearances defined in Design Rules setup to determine the distance that must exist between two components.

In many cases, several parts are adjusted to accommodate the moved part. Nudge does not move glued parts or parts outside the board outline. Nudge treats test points as glued objects. Nudge does not move parts inside the board outline outside the outline. When an overlap exists, the Nudge Parts and Unions dialog box appears.

Nudge does adhere to the current grid setting, so be sure you set the grid fine enough for the command place within the restrictions of your routing strategy.

Component Arrays

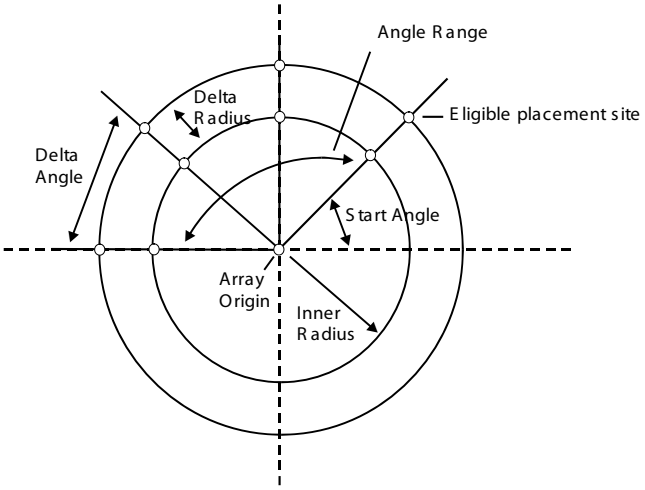
You can arrange parts by creating arrays. A component array is a union with members placed on sites of a user-defined matrix. You can create planar arrays or circular arrays. For more information, see the To Create a Component Array topic.

Defining Arrays

Defining a Polar Grid or Circular Array

Figure 10-1 shows the different values that you set to create a polar grid or a circular array. In a circular array, components are placed equal distances from each other on one or several rings. By turning on the polar grid and Snap to Grid, you can create drafting items with corners located at the nodes of the polar grid.

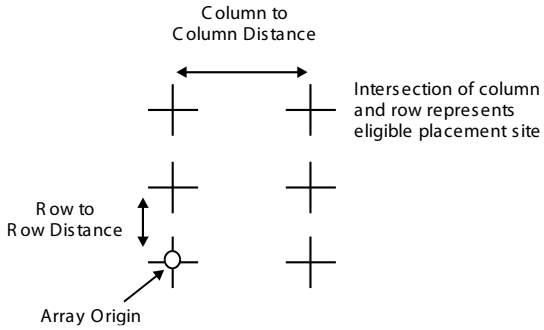
Figure 10-1. Polar Grid Array



Defining a Planar Array

Figure 10-2 shows the different values that you set to create a planar array. A planar array is a union of components that are placed on the intersections of equidistant parallel lines. The parallel lines can exist in both the X and Y directions.

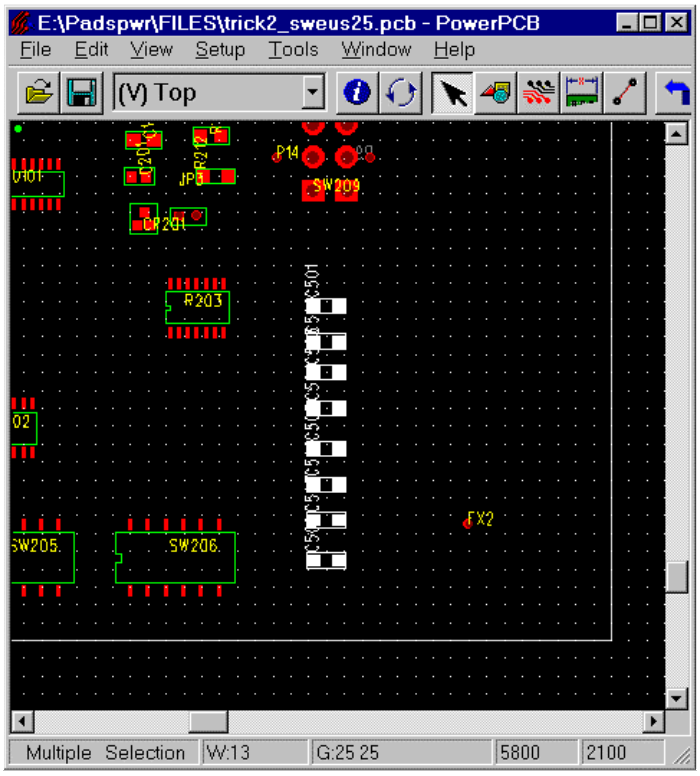
Figure 10-2. Planar Array



Component Array Examples

Figure 10-3 represents an original PCB design with eight components selected to create an array.

Figure 10-3. PCB Design with an Array



The following examples represent the same eight components using different grid and array settings.

Example 1

Figure 10-4 shows a planar array of the original eight components with a minimum body-to-body clearance of 6 mils and 45-degree rotated components:

Figure 10-4. An Example Planar Array (Minimum Body-to-Body Clearance of 6 mils)

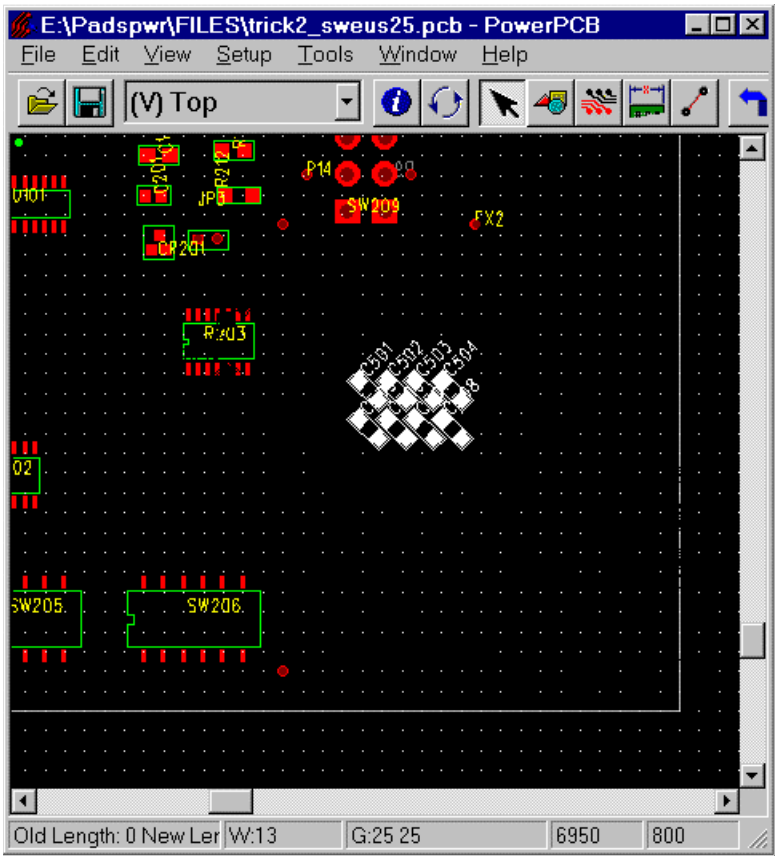


Table 10-2 show the settings used on the Planar Array tab to create the planar grid in Figure 10-4.

Table 10-2. Planar Array Settings (6 mils)

Option	Setting
Row to Row	193.7 (mils)
Column to Column	140.24 (mils)
Number of Columns	4
Place by Columns	On
Rotate	On
Orientation	45 (degrees)
Align by	Origin

Example 2

Figure 10-5 shows the example planar array of the original eight components with a minimum body-to-body clearance of 200 mils.

Figure 10-5. An Example Planar Array (Minimum Body-to-Body Clearance of 200 mils):

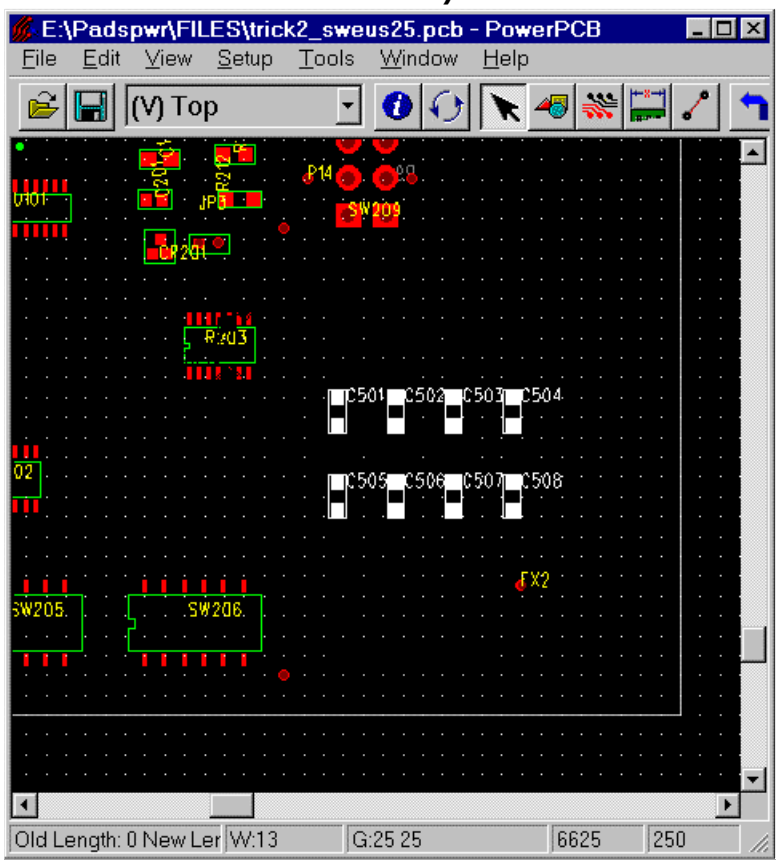


Table 10-3 show the settings on the Planar Array tab used to create the planar grid in Figure 10-5.

Table 10-3. Planar Array Settings (200 mils)

Option	Setting
Row to Row	430.15 (mils)
Column to Column	293.4 (mils)
Number of Columns	4
Place by Columns	On
Rotate	On
Orientation	0.000 (degrees)

Table 10-3. Planar Array Settings (200 mils) (cont.)

Option	Setting
Align by	Origin

Example 3

Figure 10-6 shows an example circular array of the original eight components with a minimum body-to-body clearance of 200 mils:

Figure 10-6. An Example Circular Array

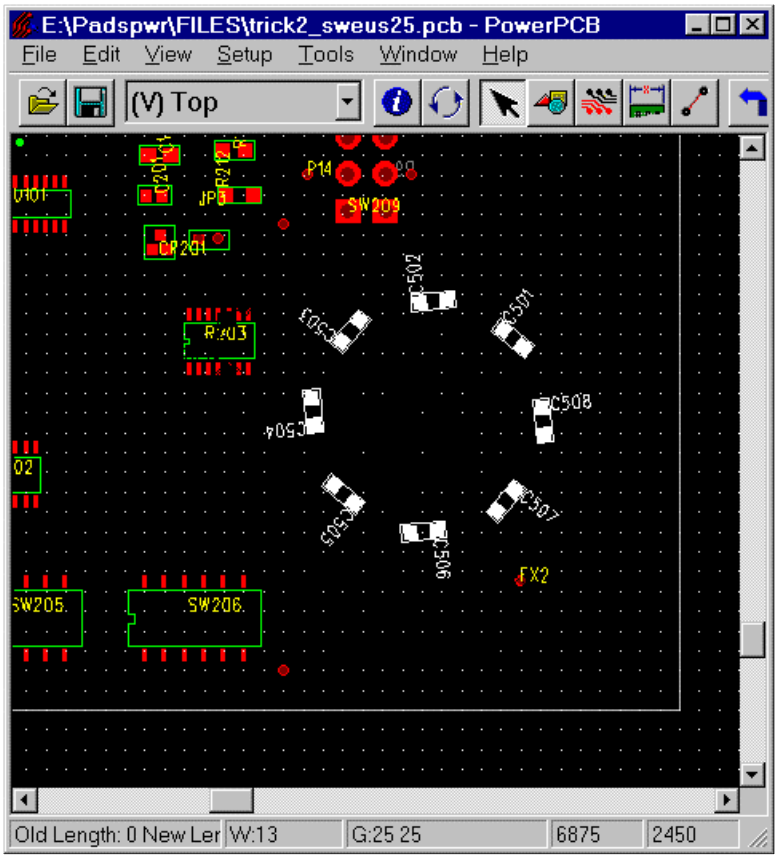


Table 10-4 shows the settings on the Planar Array tab used to create the circular array in Figure 10-6.

Table 10-4. Circular Array Settings

Option	Setting
Inner Radius	576.68 (mils)
Delta Radius	0 (mils)
Start Angle	5 (degrees)
Angle Range	360 (degrees), Locked

Table 10-4. Circular Array Settings (cont.)

Option	Setting
Delta Range	45.000
Sites per Ring	8
Direction	Counterclockwise
Rotate	On
Orientation	0.000
Align by	Origin

Polar Grid and Radial Move Example

Figure 10-7 shows an example polar grid definition.

Figure 10-7. An Example Polar Grid Definition

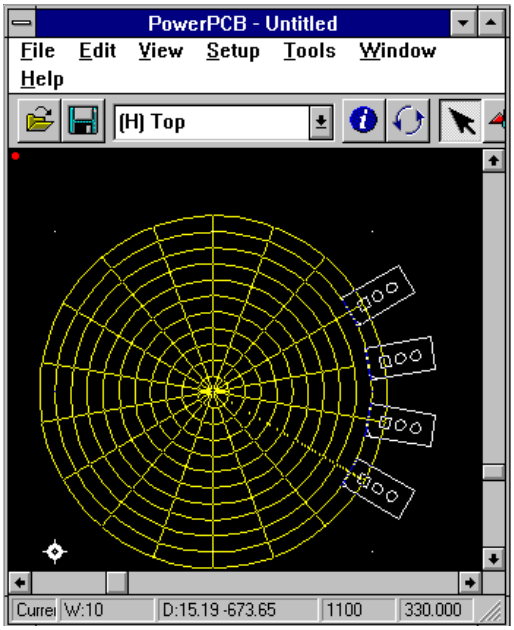


Table 10-5 shows the Radial Move settings used to create the polar grid in Figure 10-7.

Table 10-5. Radial Move Settings

Option	Setting
Polar Grid Origin	X=1000, Y=1000
Inner Radius	100 (mils)
Delta Radius	100 (mils)
Start Angle	30.000 (degrees)

Table 10-5. Radial Move Settings (cont.)

Option	Setting
Angle Range	360.000 (degrees)
Delta Angle	20.000 (degrees)
Sites Per Ring	18
Direction	Clockwise
Auto Rotate	On
Disperse	On
Use Discrete Radius	On
Use Discrete Angle	On
Polar Orientation	Let Me Specify, 0.000 (degrees)

Using the Radial Move Shortcut Menu

Right-click to open the Radial Move shortcut menu (Table 10-6).

Table 10-6. Radial Move Shortcut Menu Items

Command	Description
Properties	Queries the component or union. For more information, see the Modifying Component Properties topic.
Rotate 90	Rotates the part while calculating the orientation of the part relative to the radial direction, from the grid origin to the current position of the part's move origin. Each part or union rotates using its individual origin point. For more information, see the To Rotate an Object topic.
Flip Side	Flips each object individually around its radial direction. For more information, see the To Flip a Component topic.
Move by Cursor Location	Moves objects according to the relationship between the cursor location and the origin of the moved object. The default move preference is found in the Options Dialog Box, Design Page.
Move by Origin	Moves objects by the origin of the object. The default move preference is found in the Options Dialog Box, Design Page.
Move by Midpoint	Moves objects by the midpoint of an object or a group of objects. Snaps to the current pointer position. The default move preference is found in the Options Dialog Box, Design Page.

Table 10-6. Radial Move Shortcut Menu Items

Command	Description
Ignore Clearance	Temporarily ignores clearances for design rules. The default On-line DRC setting is found in the Options Dialog Box, Design Page.

Cluster and Union Placement

Use cluster placement features to create associations or groupings of connected parts. Two object types are used:

- Unions

  Unions are user-created part associations that have a strict relationship with each other. An example of this is associating a filter capacitor to reside on top of an IC. When you move or place a selected union, the physical relationship between the parts, or *union members*, remains unchanged. Unions are not created using the Cluster Placement routines, but they are considered during cluster creation and automatic placement operations. For information, see the Managing Unions topic.

- Clusters

  Clusters are collections of individual parts, unions, and other clusters, based on connectivity. A series of ICs and associated discrete components that make up a memory array could make up a cluster. Clusters differ from unions because the parts that belong to the cluster, or *cluster members*, are rearranged within the cluster to improve placement.

  Clusters are useful in very large designs or in designs where areas of the board are separated into different functions. You can create or modify clusters manually or automatically. You can also automatically place individual parts using this feature. For more information, see the Placing Parts Automatically and Moving Clusters Interactively topics.

Any operation that creates, displays, or modifies clusters (except the Cluster Manager) automatically places you in Cluster View mode. For more information, see the Cluster View Mode topic.

Cluster Display Settings

The circle that represents a cluster is assigned in the Display Colors Setup dialog box as an Outline property on the Top layer. The text representing the cluster name is assigned as a Text property on layer 20.

The connection lines originating from the center of the cluster (the same as in the normal view) are assigned with the Connection setting in the Display Colors Setup dialog box.

Chapter 11
Interactive Routing

This chapter covers the interactive routers, routing setup procedures, making changes during routing, and making changes after routing.

Interactive Routing Modes

Depending on the options you enable, you may have one, two, or three modes available for interactive routing. Table 11-1 lists the interactive routing modes.

Table 11-1. Interactive Routing Modes

Routing Mode	Description
Basic Route Editor	Use the pointer to indicate each corner in the trace. For more information, see "Route Command" on page 219.
Dynamic Route Editor and Bus Router	Dynamic Route Editor (DRE) is an interactive autorouter that follows the direction of the pointer as you move it, seeking optimal paths and installing corners as the route progresses. For more information, see "Dynamic Route Editor (DRE)" on page 219. The bus router is a dynamic route editing tool that creates data lines, memory arrays, or similar connections where several routes need to flow together from one set of parts to another. For more information, see "Bus Router" on page 219.
Dynamic Autorouter	This is a pin-to-pin autorouter that automatically adds traces between pin pairs. Other routes are moved as required. For more information, see "Dynamic Autorouter" on page 219.
BGA Route Wizard	The Advanced Packaging Toolkit's BGA Route Wizard provides automated features including creating connections between BGAs and SBPs, creating BGA grid array fanouts, routing die-to-BGA connecting traces, and generating plating tails. For more information, see the Using the BGA Route Wizard topic.

Before creating new routes, consider the initial setup procedures that are common to all modes.

Routing Setup Considerations

Before using any of the interactive routers, consider the setup options below:

- Angle mode
- Starting layer selection
- Via selection
- Trace width
- Length minimization
- Display control
- Protecting routes and unroutes

Angle Modes

All routing modes are subject to the angle constraints for pad entry that you set in the Line/Trace area in the Design tab of the Options dialog box. As shown in Table 11-2, the routing modes are Orthogonal, Diagonal, and Any Angle. The modes determine how a trace follows the cursor from corner to corner and enters a pad on completion. To set these modes quickly type the AO, AD, or AA modeless commands.

Table 11-2. Angle Modes

Mode	Description
Orthogonal (AO)	Prevents you from entering diagonal lines and 45-degree corners.
Diagonal (AD)	Limits you to 45- and 90-degree turns.
Any Angle (AA)	Allows placing routes at any angle.

Tip: The dynamic autorouter and Dynamic Route Editor do not use Any Angle mode. Diagonal and Any Angle both create 45-degree angle traces.

Starting Layer for Routing

To set the current routing layer before you start routing, select the layer from the Layer list on the toolbar. A layer's current direction setting appears next to its entry in the Layer list. When you select a connection and select a routing mode, the route begins on that layer.

When you end a route on a layer different from where you began it, the layer you ended on becomes the active layer and appears in the Layer list. The next connection you select starts on

this active layer. You can set a different active layer by using the list or by using the modeless command: type L; type the Layer number, and press Enter.

Vias

A via is a drilled and plated hole used to connect copper objects on different layers of a printed circuit board.

In general, there are two types of vias:

- Through-hole vias —Vias that are drilled through every layer of the board.
- Partial vias —Vias drilled through a select number of layers. Partial vias include:
  - Blind vias— Vias that pass through an outer layer and connect to an inner layer.
  - Buried vias— A partial via that passes through only the inner layers.

Vias have a variety of uses in a layout design:

- In Routing. The most common use for vias is in routing traces, to continue the connectivity of traces to other layers. When you add a via during routing, PADS Layout connects it to the trace and considers it a routing via.
- In stitching and shielding. Vias used aside from routing are sometimes called free vias because they can be added anywhere in the design.

  Free vias are more often called stitching vias because when added to a design, they may resemble a row or pattern of stitches. In addition, they are often used to “stitch together” a copper to a plane. For example, you might add vias to a small copper area to give it more numerous connections to a ground plane, insuring a solid connection.

  When you add these vias, PADS Layout assigns them the Stitching property, which distinguishes them from routing vias.

There are several methods you can use to place stitching vias in a layout design:

- Add single vias repetitively. Vias appear on your cursor, ready to be placed. For information, see Add Stitching Vias.
- Surround a net, pin-pair, or copper area with vias to act as a via shield (**Add Via Shield** command). For example, you might place a via shield around a pin-pair to create an RF trough.

  For information, see Adding a Via Shield.
- Stitch a copper or copper hatch shape with a pattern of vias (**Via Stitch** command). There are two modes for stitching the patterns:
  - You can fill the shape with vias.

For information, see Filling a Shape with a Pattern of Vias.

- You can also add vias inside the perimeter of the shape.

  For more information, see Placing Vias Inside the Perimeter of a Shape.

Via Types for Routing

Use the Vias dialog box to determine what kind of via to install when you change levels while routing. Use the V modeless command to open the dialog box.

Table 11-3 lists the via routing modes. To set the via type quickly, use the VA, VP, or VT modeless commands.

Table 11-3. Via Routing Modes

Mode	Description
Automatic (VA)	PADS Layout chooses from all vias, through or partial, that can handle the particular layer change. If PADS Layout finds partial vias dedicated to the layer change, it chooses from them even if the via is not one that is specifically selected in the Vias dialog box. If PADS Layout can't find a dedicated partial via, it selects any through vias for a through or partial layer change. It then checks the Routing Rules dialog box for vias that are allowed for the net you are routing. If more than one via still passes, PADS Layout installs the one with the smallest drill diameter or smallest pad size. Automatic allows only vias that begin and end on the layer pair shown on the Routing tab of the Options dialog box. To use automatic via mode, the layer pair for routing and the layer pair for the partial via definition must match. For example, if you have a partial via set up for layers 1 through 4, and the layer pair for routing is set for layers 1 through 8, automatic mode will not insert a via.
Partial (VP)	The automatic via selection still occurs, but it is limited to the partial vias only.
Through (VT)	The list of through vias becomes active. Highlight the via you want to use as the default and click Apply. This is the via which is installed every time you change levels between layer pairs.

In any case, the via must not create a clearance violation according to the default clearance rules or the clearance rules attached to the net you are routing, whichever takes precedence. If DRC is set to Prevent Errors, the layer won't change if the trace or via creates a clearance problem.

You can change an installed via to another type using the Via Properties dialog box. For more information, see the Modifying Via Properties topic. You can also change the via type during routing. For more information, see the To Change the Via Type While Routing topic.

Allow Vias by Net

Use the Routing Rules dialog box to control which vias are eligible for which nets. This is one of the main criteria for the automatic via selection.

Trace Width

You can pass a value for trace width with the netname from the schematic. You can also enter or edit the width as a line in the schematic ASCII file for nets. If widths are assigned to nets when the netlist is read in, the connections automatically assume those widths when routing. If no widths are associated, PADS Layout provides a default width of 12 mils for all nets. You can change the default trace width at any time by changing the Recommended Trace Width. For more information, see the Clearance Rules dialog box topic.

The recommended width is in effect unless you override it locally. For example, a net may have a width of 12 assigned to it, but you need a certain pin pair within the net to have a width of 10. Use the Pin Pairs Properties dialog box to assign a local trace width to the pin pair in the Trace Width box.

You can see a connection's assigned width when you select it. The width appears on the status bar under the pin pair.

If you use the High-Speed Rules option (Advanced Rules), you can assign different widths within a net by making the width assignments on the pin pair or group of pin pairs level. You can also join netnames in net classes with trace width rules applied to them. Changes made to the class name change all the nets in the class.

Length Minimization

When you start interactive routing, the flightline to the terminating pin may jump from the pin to which it is connected to another pin. This on-the-fly length minimization shows that the pin indicated by the flightline is part of the same net and is closer to the cursor than the original terminating pin.

You can disable length minimization by setting the topology type to "protected." In this case, the flightline appears but maintains its original connection. You can set this property for net, class, or the Default level of the rules hierarchy. For more information, see the Routing Rules dialog box topic.

Display Control

Colors

Use Display Colors on the Setup menu to set different colors for routed traces on different levels. You can also set different colors for copper and for pads on pins or vias per level. If

Active Layer Comes to Front is selected on the Global tab of the Options dialog box, when you start a routing command on that level all the traces drawn in the color for the current layer come forward and overlap traces on other levels.

To selectively hide or display traces or connections or to display nets with specific colors, use the View Nets dialog box. You can change the appearance of nets, net classes, and nets with rules. For more information, see the To Assign Colors to Nets topic.

Display of Trace Width

To speed scrolling, panning, and screen regeneration, you can set a minimum displayed width for all lines. Any line widths under this width value appear as 1-pixel centerlines.

You can set the Default Width in the Drafting tab of the Options dialog box or use the R modeless command. Set the value to your narrowest line width or less to see the design's true widths.

Trace Length Monitor

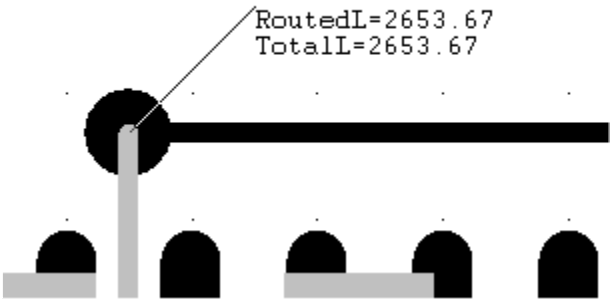
The trace length monitor calculates and shows the trace length (on the status bar and near the pointer) as you route. When net length rules are specified, then red or green text in the monitor indicates whether the length is within the specified rules:

- Red text indicates violations of the rule.
- Green text indicates trace lengths within the rule.

The status bar shows detailed length information.

The trace length monitor shows the routed length and the estimated *total length* of the net (Figure 11-1). For more information, see the Using the Trace Length Monitor topic.

Figure 11-1. The Trace Length Monitor



Setting the Colors for the Trace Length Monitor

You can set the colors of the text in the monitor that indicate whether the length is within the specified rules. Set the colors in the powerpcb.ini file. In the .ini file, look for the [monitor] section that resembles the following:

```
[monitor]
normal = 10
norules = 15
error = 12
```

The numbers represent colors from the [Colormap] section of the .ini file. To change the colors used in the monitor, change the numbers in the [monitor] section to a color from the [Colormap] section. Values may be 0-31.

Effects of Reroute and Smooth on Trace Length

During reroute operations, the trace length monitor may show a length greater that the actual final length. In addition, for dynamic routing, smoothing operations and pad entry corrections may be performed on traces after you finish routing (depending on your Routing / General options). These operations may change trace length slightly.

Trace Length Monitor Reports

As mentioned, the trace length monitor shows the net length rules, the current routed length, and the estimated *total length*. The rules that are reported are *net length rules*. As shown in Table 11-4, the trace length monitor reports different information depending on the length rule.

Table 11-4. Trace Length Monitor Reports

Length Rule	Information in Report
Minimum/Maximum Length Rules	Shows the current routed length, the total length, the minimum net length rule, and whether it's applied at the net or pin pair level of the rules hierarchy.
Matched Length Rules	Shows the current routed length, the total length, and the matching net length rule.
Differential Pair Rules	Shows the current routed length, the total estimated length, the minimum length rule, the maximum length rule, the gap rule, and whether it's a net or pin pair rule.
No Length Rules	Shows the current routed length and the total estimated length.

Interactive Routers

You can start the interactive routers using verb mode selection or object mode selection. For more information, see the Routing the Design chapter.

Route Command

The command for manually entering traces is Route. The basic route editor associates a netname, layer color, width, and edge-to-edge clearance to a drawn trace as basic properties. You can add arcs, miters, and vias while a trace is in progress or after it is completed.

If you enabled On-line Design Rules Checking, the basic route editor responds to clearance rules as they apply to the Dynamic Route Editor. Click Verify Design from the Tools menu at any time to check the design for clearance violations.

Dynamic Autorouter

The dynamic autorouter activates a single-layer, pin to pin autorouter for a selected connection or pin pair.

You must have On-line DRC active to use the autorouter. It operates on one layer only, and observes the current design grid. Once started, it makes several attempts to find a path. If it cannot complete the trace, it leaves the unrouted connection.

Dynamic Route Editor (DRE)

The Dynamic Route Editor (DRE) is an interactive autorouter that follows the direction of the cursor as you move it, seeking optimal paths and installing corners as the route progresses. DRE is a gridless, shape-based router. If the routing grid is set at less than clearance values, DRE uses the clearance settings to decide the path. You must set On-line Design Rule Checking to Prevent to enable DRE.

Because DRE installs its own corners, it lets you quickly route a selected connection by pulling the head of the trace through obstacles with the cursor, guiding the path you want to use. Although you can, you do not have to manually install corners. DRE also moves established traces out of the way and makes room for a path you are routing, providing there is room to move without creating clearance violations.

You can also reroute segments with DRE. For more information, see the To Reroute with Sketch Route topic.

Bus Router

The bus router is a dynamic route editing tool that creates busses. Use the bus router to quickly route data lines, memory arrays, or similar connections where several routes need to flow together from one set of parts to another.

Tip: You need Dynamic Route Editing (DRE) to use the bus router.

To use the bus router, identify multiple pin pairs to route and place one route. This route acts as a *guide route*. The other routes, called *follow routes*, are based on the established path of the guide route. The bus router creates the guide route using the same automatic corner creation and trace shoving as Dynamic Route Editor.

To take full advantage of the bus router's automatic routing features, make sure selected pin pairs are adjacent to each other at both ends of the connection and that connections do not cross. Selecting pins with connections that cross each other invokes a manual bus route mode where you dynamically route each connection in sequence. For more information, see the Bus Routing topic.

Selection Rules for Bus Routing

The bus router creates routes from selected pins, vias, and tacks. You can set the Selection Filter to select only these objects. You can also right-click and click Select Pins/Vias/Tacks when nothing in the work area is selected. The bus router does not work with selected connections or traces.

While selected pins are the most commonly used objects for bus routing, adding selected vias and tacks lets you route a portion of a bus, end it, and then begin routing again.

Selection Rules

The following rules govern selection for bus routing:

- Select at least two objects. This can be a combination of pins, vias, and tacks.
- Selected pins can have through-hole or surface mount pads. Selected surface mount pads must exist on the same layer.
- Selected pins must all belong to the same part.
- Selected objects must be assigned to a net.
- Nets to which the selected objects are assigned cannot be assigned to a plane layer.
- Selected objects cannot have a differential pair attribute.
- Selected vias or tacks must be attached to a dangling route.

Object Filtering

When you start the bus router, PADS Layout automatically clears the following:

- Pin selections on multiple parts. PADS Layout determines which part has the most selected pins and deselects pins belonging to other parts.
- If two or more parts have the same number of selected pins, the part added to the design is chosen.
- Pins assigned to a plane net.
- One of two objects if they belong to the same net.

Active Layer Setting

When you select surface mount pads or tacks at the ends of a trace, PADS Layout detects the layer on which these objects exist. PADS Layout automatically sets this as the active routing layer.

Tip: If a selected pin has a layer restriction on the active layer, it is deselected.

Using the Bus Router

You can start the bus router from the shortcut menu for Object mode selection or use the Bus Route button from the Design Toolbar for Verb mode selection.

Adding Corners

- Keep guide routes as short as possible.

Long guide routes can be difficult for the bus router to resolve, so indicate corners for bus turns as often as possible. If a Bus Route failure message appears on the status bar, press Backspace to undo the bus to the last indicated position. Then insert the corner at a different location.

- Switch the guide properly.

Proper control of the guide route is necessary for successful bus routing. The guide route should be internal when you indicate a corner. If you use an external guide route to create a corner, the internal follow routes shrink to make a turn, as long as there is sufficient space.

Controlling the Guide Route

While bus routing, you can cycle through all of the connections in the bus to make a different connection the guide route. To set the next object in the bus as the guide route, right-click and click Next Guide. To set the previous object in the bus as the guide route, right-click and click Previous Guide.

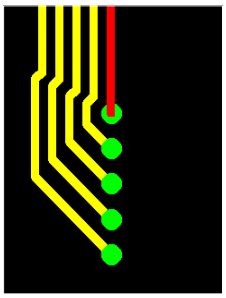
Via Pattern Mode

You can choose from five via patterns when you add vias to a bus. PADS Layout uses the via pattern you choose until you choose a different pattern or exit the program. To set a via pattern while bus routing, right-click and click a pattern from the Via Pattern Mode menu. The available via patterns are:

Parallel

Vias are added in a column, parallel to the direction in which the bus is traveling. Parallel is the default via pattern. If the guide route is one of the middle routes, the remaining routes adjust on either side of the pattern (Figure 11-2).

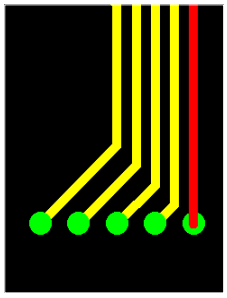
Figure 11-2. Routing when Via Pattern is Parallel to Bus Direction



Perpendicular

Vias are added in a row, perpendicular to the direction in which the bus is traveling. When the guide route is one of the middle routes, the remaining routes adjust on either side of the pattern (Figure 11-3).

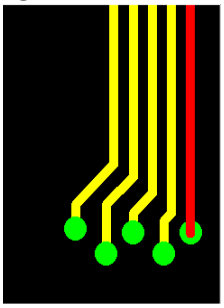
Figure 11-3. Routing when Via Pattern is Perpendicular to Bus Direction



Staggered

Vias are added in staggered rows to minimize space requirements. If the guide route is one of the middle routes, the remaining routes adjust on either side of the pattern (Figure 11-4).

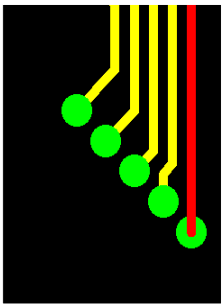
Figure 11-4. Routing when Via Pattern is Staggered



Minus 45 Degrees

Vias are added in a diagonal, at a negative 45-degree angle to the direction in which the bus is traveling. If the guide route is one of the middle routes, the remaining routes adjust on either side of the pattern (Figure 11-5).

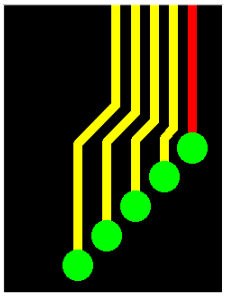
Figure 11-5. Routing when Via Pattern is Minus 45 Degrees



Plus 45 Degrees

Vias are added in a diagonal, at a positive 45-degree angle to the direction in which the bus is traveling (Figure 11-6).

Figure 11-6. Routing when Via Pattern is Plus 45 Degrees



Making Changes During Routing

Changing the Layer While Routing

To change layers while routing, use the L modeless command while the trace is active. PADS Layout installs a via at the last corner and move your trace to the selected layer. The Via Type dialog box settings control the type of via you place. The trace color changes accordingly and the new layer appears in the Layer list.

You can also use the Layer command in the shortcut menu. If you change repeatedly between two layers, you can set a layer pair and use the layer toggle shortcut key, Ctrl+T.

If the layer change is allowed, PADS Layout installs a via at the last corner when you change layers either while routing or after entering the trace. You can set PADS Layout to install a selected via by default. Since partial vias undergo a checking process, you cannot set them this way by default; they are chosen automatically depending on the circumstances of layer change.

Conditions for Changing Layers

Operations for changing layers in interactive routing are the same for the Route command and Dynamic Route Editor.

When you change layers in PADS Layout, you must meet several requirements before you can place a via. First, PADS Layout checks the layer change to see whether it is a legal *drill pair*, the start and finish layer numbers. This check affects multiple layer boards. Establish pairs by clicking Drill Pairs on the Setup menu according to your projected manufacturing plan. Once you enter drill pairs, PADS Layout warns you if you're joining two layers that don't match your manufacturing scheme.

For more information, on setting up through hole vs. partial vias, see the To Change the Layer While Routing topic.

Using the Layer Pair

If you're working extensively between two layers, set them as the default layer pair and use Toggle from the shortcut menu or press Ctrl+T to install a via and change to the other level.

Set the layer pair on the Routing tab of the Options dialog box. You can also use the PL modeless command.

To change levels outside the layer pair, use the L modeless command or the Level command on the shortcut menu.

Changing the Via Type While Routing

The new via type is used until you change it. You can leave the Vias dialog box open to quickly change from one via type to another. Minimize the dialog box or move it around the work area as needed.

Changing the Trace Width While Routing

To change the trace width of a route in progress, use the modeless command or the Grid/Width dialog box. The current routing width changes from the last corner, leaving segments before the last corner at their original width. All subsequent segments draw at the new width, until you reach the next pin in the net. The width stays associated with the connection until you change it. The new width becomes the effective width for the rest of the connection, but it does not change the Recommended Trace Width setting for the net. If you end with a partial route and use it later, the width you set separately is still in effect.

To change the width of a trace once it is routed, select the item and edit the Properties. You can also click Find from the Edit menu to find all traces of a similar width, select them, and change them to a new width.

Ending a Trace on a Different Net

You can end a trace on a segment with a different netname, but you must use the Add Route button on the ECO Toolbar. For more information, see the Adding a Route in ECO Mode topic.

Making Changes After Routing

There are several editing tools that clean up or modify routed trace patterns or completely reroute a trace.

Rerouting with Route or Dynamic Route

You can reroute traces by starting and ending a new replacement trace anywhere along the existing trace. You must meet two conditions:

- Existing and new traces have the same netname.
- Copper sharing is enabled for the net.

To enable copper sharing for a net, select Copper Sharing for Routing in the Net Rules dialog box. To enable copper sharing for all nets, click Copper Sharing for Routing in the Default Rules dialog box. For more information, see “Design Rules” on page 105.

Use Transparent Mode (T modeless command) to view obstacles that may lie under traces on the current active layer.

Adding Stitching Vias

You can use Add Via to add a stitching via. Stitching vias interact with many commands in PADS Layout:

- The All Except Connected Plane Nets option in the View Nets dialog box disables unroute display to stitching vias that are embedded in copper pour, split mixed planes, or CAM planes. Also, Length Minimization calculates both the pads and drills of the stitching via to determine the unroute visibility to a stitching via.
- When you copy trace patterns, stitching vias are also copied. Copied stitching vias snap to the via grid when pasted.

Adding Tacks

When you are in Dynamic Route Editor, the Add Tack command adds a tack to a selected trace at the selection point. You can delete a manually added tack by selecting it and pressing Delete. If the tack is not required for a route direction change, then pressing the Delete key deletes the corner beneath the tack and moves the tack to the next "U" point in the route.

The selection filter contains a check box for tacks so you can set the filter to select only tacks.

Adding Test Points

You can manually add a test point along an existing segment or trace of a route.

Add Test Point in Verb Mode

Test points, by default, are not glued and are bottom accessible. The inserted via is the type that you choose in the Audit Rules tab of the DFT Audit dialog box. For more information, see the Performing a Test Point Audit topic.

You can also make an existing via or pin, making it a test point. For more information, see the Modifying Via Properties or Modifying Pin Properties topics.

Vias Under SMD Pads

You can right-click and click Add Via Under SMD when you select an SMD pad or multiple pads. First, however, you must edit the clearance rules for the nets that get the via. Every net can have:

- Clearances between objects in the net and anything else
- Clearances between objects in the same net

One of the clearance rules for same nets applies to the minimum SMD to Via distance. You must set this value to zero to allow the via under the SMD pad; so that Add Via Under SMD can

function. Also consider other via rules, such as the via can't create a clearance violation. It must be allowed for the net and in drill pairs.

Connecting SMD Pads to Planes

Routing an SMD pad to a plane involves placing a via to the plane under or somewhere near the pad. When vias are in plane nets, they receive a thermal relief pad for the plane layer based on the pad outer diameter for the layer, defined in the via's pad stack.

When you route SMD pads to a plane layer, use the View Nets dialog box to check whether you routed all the required SMD pads. This check lets you turn off all unrouted connections in the display except the SMD-to-plane connections. These turn off as you link them to the plane with vias. The connection remains to define the thermal relief locations for copper flooding. CAM plane and split/mixed plane layer thermals are created based on the Plane Thermal options in the Via Properties dialog box. Use the Show Plane Thermals option in Thermals tab of the Options dialog box to display CAM plane and split/mixed plane layer thermals.

When you install the via, either under the pad or a short distance away, the connection for that pin-to-pin connection disappears when you route the connection to the plane layer.

Route Protection

You can set options that prevent the modification of routed and partially routed connections. This feature protects critically placed routes during interactive routing and in batch routing in automatic routers. Protection passes bidirectionally to and from the SPECCTRA Autorouter. You can also protect unrouted connections and the unrouted portions of partial routes.

While in a protected state, you cannot make changes to a trace or unroute. For more information, see the To Break a Physical Design Reuse topic. Components, vias, test points, and other design items associated with protected routes or unroutes are also protected.

You can set the display to differentiate between protected routes and unprotected routes. For more information, see “Display Control” on page 216.

Protecting Routes

Protecting routes involves selecting the object and using the Properties to enable protection. You can set protection for fully routed connections and the routed portion of partial routes at a pin pair- or net-level.

Protecting Unroutes

Protecting unroutes involves selecting the object and using the Properties to enable protection. Before you protect the unrouted portion of a partial route you must protect the routed portion.

You can then set protection for fully unrouted connections and any unrouted portion of partial routes at a pin pair- or net-level.

Connecting a Net with a Plane

Establishing a plane area and connecting the appropriate nets to it are usually the first routing tasks in the design process. The following two methods establish plane areas:

- Define a Layer as a Plane Type

Use Layer Definition from the Setup menu to set an entire layer to Plane type. Use the Net Associations command to define one or more netnames to connect to this layer with thermal reliefs.

With this method, you do not assign a netname to the plane area. The connection occurs when you define the layer and associate the netname. The copper and other plane details do not appear in the .pcb file. The copper area, antipads, and thermal relief connections appear only on the final printout or photoplot for the manufacturing output.

- Draw a Copper Pour Area

Use this method when you want insulated traces passing through the plane area. The copper pour area does not require you to define its layer as a plane layer. Instead, assign the netname of the net to which you want to connect to the copper pour outline. You are limited to one net per pour area. When you flood the area, you can see the copper hatch and thermal reliefs; they appear on the printout or plot.

Viewing Plane Layer Connectivity

For CAM plane layers, use the Show Plane Thermals option in the Thermal tab of the Options dialog box to see which pins have thermals. The Plane Thermal option determines whether the thermal is generated for the pin. The Plane Thermal option signals CAM output to assign a D-code for a thermal relief aperture around the pins. Set the Plane Thermal option using the Properties dialog boxes for pins, vias, and jumper pins.

If you set up a plane connection successfully, a D-Code number is assigned for a thermal relief pad in your photoplotter aperture table, one that matches each pad size required. For more information, see Interpreting CAM Plane Thermal Graphics.

Ratsnest Display Remains on Connected Copper Pour Plane Pad Stacks

For Copper Pour plane connections, the unroutes are unused to control thermal generation. Even after the connection is established, the net you are connecting with a plane still appears as an unrouted, ratsnest connection. These unroutes signal the copper flood operation and generate a copper pour thermal relief around the pins. The copper pour area draws a screen representation of the thermal.

Setting Pins and Vias as Thermals

Jumper pins, pins, and vias can all be thermals. For these objects to be eligible for a thermal, you must click to select the Plane Thermal check box in the Jumper Pin Properties, Pin Properties, and Via Properties dialog boxes.

If you select more than one pin or via where both plane nets and nonplane nets are included, and not all pins and vias are marked as eligible for indicators, the Plane Thermal check box is unavailable. It is neither on nor off. If you click to select Plane Thermal, the selected plane net pins or vias are updated.

Chapter 12
Links

This chapter describes how design and routing information is passed between PADS Layout and linked products.

SPECCTRA Link

The SPECCTRA Link sends PCB design file data to a SPECCTRA design file, and then imports the SPECCTRA results back to a PCB design file. SPECCTRA translates route protection status, physical design reuses, test point settings on both vias and component pins, decal keepouts, board keepouts, and board cutouts. Route protection status is passed to and from SPECCTRA for routed traces and vias using SPECCTRA's *fix* attribute for wires and *route* attribute for vias. Protection for unrouted traces is passed to SPECCTRA only for whole connections, such as routed links that start and end on component pins. For more information, see “Passing Route Protection Status to SPECCTRA” on page 233.

When you use the SPECCTRA command from the Tools menu, you can run the entire process of converting, routing, and reloading into PADS Layout automatically.

To control the SPECCTRA interface you can also start the Link and SPECCTRA executables manually. When you run SPECCTRA manually, you can choose the conversion direction: TO SPECCTRA or FROM SPECCTRA.

The Link features an automatic DO File editor to edit existing or new DO files. Parameters are automatically saved between sessions.

Unused Pins Net

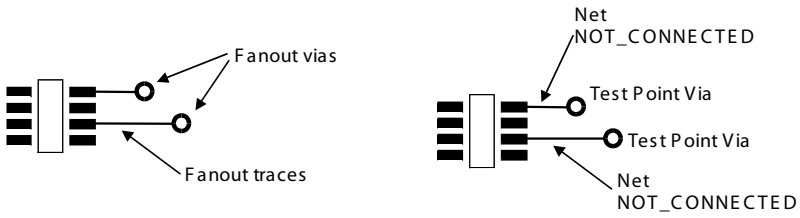
PADS Layout passes unused pins, or pins that are not connected to a net, to SPECCTRA. These pins and their fanouts are added to a net of unused pins called +UNUSED_PINS+ in SPECCTRA (this was formerly called *UNUSED_PINS*). When you return the routed design to PADS Layout, you can also pass the unused pins and fanout information. The +UNUSED_PINS+ net information is translated into the NOT_CONNECTED net in PADS Layout. You can change the name of the NOT_CONNECTED net in the SPECCTRA Options dialog box, but the net should have a unique name.

The NOT_CONNECTED net is a normal net inside PADS Layout that contains the fanout via, the trace, and the unused pin. Because this net is a normal net in PADS Layout, pins in it are no longer treated as unused pins in subsequent SPECCTRA sessions. Use the existing DO file to protect this net from being routed as a normal net in subsequent sessions. To recreate this net in SPECCTRA, delete it in PADS Layout before routing the design.

Tip: PADS Layout cannot define a connection for a single pin net; therefore, if the +UNUSED_PINS+ net contains only a single component pin and fanout, PADS Layout cannot interpret it.

Figure 12-1 demonstrates fanouts on unused pins in SPECCTRA. It also demonstrates how PADS Layout interprets those fanouts and unused pins. Because SPECCTRA now translates test point attributes on component pins as well as on vias, the following figure demonstrates how SPECCTRA passes test points assigned to the fanouts of the unused pins.

Figure 12-1. Fanouts on Unused Pins



Passing Data to SPECCTRA

Passing DFT Audit Settings to SPECCTRA

You can pass DFT Audit test point placement options to SPECCTRA for its test point placement routine. SPECCTRA generally uses these options when applicable. Options such as test point clearances are passed to SPECCTRA. But options such as nail diameters and fixture drill sizes are used only by DFT Audit. Because they do not affect SPECCTRA's test point insertion, this type of information is not passed.

SPECCTRA bases its clearance calculations on DFT Audit options. SPECCTRA's clearance options are set in the Setup Test Point Rule area of the SPECCTRA DO File dialog box. For more information about DFT Audit, see "Design for Test" on page 114.

Tip: SPECCTRA does not allow assigning multiple test points to net. This setting is not passed to SPECCTRA.

Passing Keepouts to SPECCTRA

The PADS Layout to SPECCTRA Link supports via keepouts and passing other types of keepouts (for wires, bends, components, and pins) to SPECCTRA. Copper shapes on any level are passed in this way. By manipulating this existing keepout functionality, you can pass other specific keepouts to SPECCTRA.

PADS Layout supports keepouts in the Layout Editor.

Passing Slotted Holes to SPECCTRA

Nonplated slotted holes are converted to keepouts on all layers. Plated slotted holes use a circular drill at the electrical center.

Passing Route Protection Status to SPECCTRA

You can set options in PADS Layout that prevent modifying routed traces and vias in SPECCTRA. This feature protects critically placed routes during interactive routing and in batch routing in automatic routers. Protection is passed to and from SPECCTRA. Components, vias, test points, and other design items associated with protected routes or unroutes are also protected.

Passing Protected Traces and Vias

Routed traces with protected status are passed to SPECCTRA with a *fix* attribute. You cannot modify fixed traces, and the router cannot route to this trace.

Vias attached to at least one protected trace are passed to SPECCTRA with a *route* attribute. Although you cannot modify these vias, you can route to them to complete a connection.

Table 12-1 lists the rules used by the SPECCTRA Link to determine via attributes.

Table 12-1. SPECCTRA Link Rules

Via has protected traces	Locked test point status	Glue status	SPECCTRA via status
Yes	Any	Any	Route
No	Yes	Any	Route
No	No	Yes	Protect
No	No	No	Normal (by default)

Passing Protected, Unrouted Traces

Route protection status is passed to SPECCTRA only for those PADS Layout unrouted traces that are whole connections, which start and end on component pins. Unrouted connections are passed to SPECCTRA as a *fromto* with a *fix* attribute.

Passing Protected Components with Routed Traces

A component is passed to SPECCTRA with *lock* status as long as the component has at least one pin attached to a trace with route protection.

Receiving Protection Status for Routed Traces from SPECCTRA

- SPECCTRA wires that have a *fix* attribute are returned to PADS Layout as traces with route protection.
- Vias with a protect attribute in SPECCTRA are transformed to glued status in PADS Layout.
- Vias that have a *route* attribute in SPECCTRA are not given special protection in PADS Layout.

Table 12-3 through Table 12-4 provide lists of the passing and returning protection status for PADS Layout traces, vias, and unroutes.

Traces

Table 12-2. Protection Status for Traces

Status in PADS Layout	Status passed to SPECCTRA	Protection in SPECCTRA	Status passed from SPECCTRA	Status returned to PADS Layout
Regular trace	Normal type wire	---	Normal type	Regular trace
Regular trace	Normal type	Protect	Protect type	Regular trace
Regular trace	Normal type	Fix	Protect type	Regular trace
Protected trace	Fix type	---	Fix type	Protected trace
Protected trace	Fix type	Protect/Unprotect	Fix type	Protected trace
Protected trace	Fix type	Fix/unfix	Fix type	Protected trace

Vias

Table 12-3. Protection Status for Vias

Status of trace in PADS Layout						
Has Protected Traces	Locked TP status	Glued Status	**Status passed to SPECCTRA**	**Protection in SPECCTRA**	**Status passed from SPECCTRA**	**Status returned to PADS Layout**
Yes	Any	Any	Route	Any	Route	Normal*
No	Yes	Any	Route	Any	Route	Normal*
No	No	Yes	Protect	---	Protect	Glued
No	No	Yes	Protect	Unprotect	Normal	Normal**
No	No	Yes	Protect	Unfix	Protect	Glued
No	No	Yes	Protect	Fix, protect	Protect	Glued
No	No	No	Normal	---	Normal	Normal
No	No	No	Normal	Fix, protect	Protect	Glued

* If defined in the original PADS Layout design, the glue status of the via is lost. The test point is preserved by SPECCTRA and is returned to the PADS Layout design.

** You can delete the via in SPECCTRA if the route, edit, or clean commands are performed in SPECCTRA after the Unprotect command.

Unroutes

Table 12-4. Protection Status for Unroutes

Type of Unroute in PADS Layout	Passed to SPECCTRA as	Returned from SPECCTRA as
Not protected unroute	No special handling	No special handling
Protected totally unrouted connection	Fromto type, Fix	No special handling. Not protected in PADS Layout
Protected unroutes of partially routed connections	No special handling	No special handling

Passing Physical Design Reuses to SPECCTRA

Routes that are elements of a physical design reuse are passed to SPECCTRA with a *fix* attribute. Therefore, you cannot modify or route them. SPECCTRA only connects to vias and coppers in a physical design reuse if they are passed with a *route* attribute. Table 12-5 provides a summary of Physical Design Reuse Processing.

Table 12-5. Physical Reuse Processing

Reuse Element	Passed to SPECCTRA as	Returned from SPECCTRA
Component	lock_type fix	No (Warning issued; whether it has changed, or not, the new placement is not returned to PADS Layout.)
Via	type route	No
Trace	type fix	No
Jumper	lock_type fix	No
Copper with netname	type route	No

Passing Jumpers to SPECCTRA

You can pass jumpers to SPECCTRA. SPECCTRA will not attempt to route PADS Layout jumpers.

Tip: Do not use SPECCTRA jumpers; they are not backward compatible.

SPECCTRA and Split/Mixed Planes

To attach power nets to plane polygons in SPECCTRA, use one of the following split plane definition methods:

- If you typically define split planes after you route your designs in SPECCTRA, see the Defining Split Planes After Routing in SPECCTRA topic.
- If you typically define split planes before you route your designs in SPECCTRA, see the Defining Split Planes Before Routing in SPECCTRA topic.

SPECCTRA translates split/mixed plane layers without routing, named copper, or plane polygons as power layers. SPECCTRA does not consider power layers as routing layers and, therefore, cannot route on these layers. This minimizes the layer count passed to the router. For example, you can route a four-layer design with two power layers in a SPECCTRA configuration licensed for two routing layers.

SPECCTRA regards the entire plane as the area in which to connect component pins to all plane nets. The SPECCTRA fanout and route commands connect SMD component pins by routing short traces from the pins to vias to satisfy a connection to the plane.

For best results:

- Perform a multipass fanout of power pins before you execute multiple route passes by inserting the number of fanout passes in the command. For example, change "fanout (pin_type power)" to "fanout 5 (pin_type power)".

Select the proper fanout options as defined in the "Do File Editor" topic in the *SPECCTRA Link Help*.

- Avoid assigning design rules to nets that are associated with split/mixed plane layers. If design rules are present the design will not open in SPECCTRA. The SPECCTRA Link automatically removes or ignores design rules associated with split/mixed plane layers.

Routed Traces on PADS Layout Split/Mixed Plane Layers

With PADS Layout you can route traces on split/mixed plane layers. This is also possible in SPECCTRA, but the file is automatically adjusted to achieve the proper results in SPECCTRA. These changes result in behavior modifications of the routing commands in SPECCTRA, possibly causing unexpected routing patterns.

Adjustments to the Design

SPECCTRA translates PADS Layout split/mixed plane layers with routing or named copper as mixed, rather than power layers. Therefore, SPECCTRA routes on these layers, if necessary, to complete the design.

Behavior Changes in SPECCTRA

Mixed layers are translated as routing layers, increasing the layer count passed to SPECCTRA. For example, a four-layer design with two routing layers, one power layer, and one mixed layer is considered a three routing layer design and cannot be opened in a SPECCTRA configuration that is licensed for only two routing layers.

SPECCTRA considers plane polygons on mixed layers as areas available for connecting component pins to the plane net, but does not consider them obstacles to routing. Therefore, the fanout and route commands can add routes that pass through the plane polygons on SPECCTRA mixed layers.

Isolated instances of routing failure may occur. Failures may include the failure of the fanout command to connect pins to the plane polygons, or when the route command moves a connected plane net pin outside the area defined by the plane polygon, thus isolating the pin from the plane net.

To avoid the behavior changes, use one of the following procedures in your design process:

- Split the planes after routing. For more information, see "SPECCTRA and Split/Mixed Planes" on page 237.
- Unroute the offending traces before proceeding. Prior to passing the design to SPECCTRA, unroute any traces in the split/mixed layers. Remove any named copper. After routing in SPECCTRA completes and the design returns to PADS Layout, reroute the traces you previously unrouted and restore the named copper shapes to the proper split/mixed layer.

PADS Layout to SPECCTRA Rules Conversion

Design Rules are converted into one of two types of SPECCTRA rules using the keywords RULE and CIRCUIT. The type used in the conversion is indicated for each SPECCTRA rule.

Routing, High-Speed, and Clearance Rules

Routing, high-speed, and clearance rules are supported for classes, nets, groups, and pin pairs. Additionally, as Table 12-6 shows, PADS Layout default rules are passed when possible.

Table 12-6. Routing Rules

PADS Layout Routing Rules	SPECCTRA Routing Rules
Copper Sharing	(rule (tjunction on/off))
Priority	(circuit (priority #))
Selected Layers*	(circuit (use_layer # # #))
Selected Vias*	(circuit (use_via a b c))

Table 12-6. Routing Rules (cont.)

PADS Layout Routing Rules	SPECCTRA Routing Rules
*Selected Layer and Selected Via rules require the Advanced Rules option in SPECCTRA.	

The Link passes default routing rules to SPECCTRA. Copper Sharing is passed at the PCB rule level.

The Link also passes Priority, Selected Layers, and Selected Vias by creating an artificial class containing all nets you did not assign to a class. This artificial class is named CLASS_*nnnnnn*, where *nnnnnn* is a number from 0 to 9999999. The default routing rules are then passed to SPECCTRA in this artificial class, as shown in Table 12-7.

Table 12-7. High-Speed Rules

PADS Layout High-Speed Rules	SPECCTRA High-Speed Rules
Min/Max Length	(circuit (length *max min*) (type actual))
Stub Length	(rule (max_stub #))
Match Lengths	(circuit (match_fromto_length *on/off*) (tolerance *dist*))
Shielding w/Gap	(circuit (shield on (use_net *net*))) (rule (shield_gap *dist*)) Net must be a power net or SPECCTRA will fail on input.
Parallelism Length and Gap	(rule (parallel_segment (gap *dist*) (limit *dist*)

The Link passes the default high-speed rules to SPECCTRA. It passes Parallel Length, Tandem Length and Gap, Stub Length, and Min/Max Length by creating an artificial class containing all nets you did not assign to a class. This artificial class is named CLASS_*nnnnnn*, where *nnnnnn* is a number from 0 to 9999999. The high-speed rules are then passed to SPECCTRA in this artificial class.

Clearance Rules

As shown in Table 12-8, SPECCTRA accepts only one trace width per rule so the recommended width is passed.

Table 12-8. Clearance Rules

PADS Layout	SPECCTRA
	(rule (width *dist*)
Trace to Trace spacing	(clearance *dist* (type wire_wire))
Via to Trace spacing	(clearance *dist* (type via_wire))
Via to Via spacing	(clearance *dist* (type via_via))
Pad to Trace spacing	(clearance *dist* (type pin_wire))
Pad to Via spacing	(clearance *dist* (type pin_via))
Pad to Pad spacing	(clearance *dist* (type pin_pin))
Smd to Trace spacing	(clearance *dist* (type smd_wire))
Smd to Via spacing	(clearance *dist* (type smd_via))
Smd to Pad spacing	(clearance *dist* (type smd_pin))
Smd to SMD spacing	(clearance *dist* (type smd_smd))
Board to Trace spacing	(clearance *dist* (type area_wire))
Board to Via spacing	(clearance *dist* (type area_via))
Board to Pad spacing	(clearance *dist* (type area_pin))
Board to SMD spacing	(clearance *dist* (type area_smd))
Smd to Via same net	(clearance *dist* (type smd_via_same_net))
Smd to Crn same net	(clearance *dist* (type smd_to_turn_gap))
Pad to Crn same net	(clearance *dist* (type pad_to_turn_gap))
Via to Via same net	(clearance *dist* (type via_to_via_same_net))
Drill to Drill spacing	(clearance *dist* (type drill_gap))
Body to Body spacing	Unsupported

The Link passes default clearance rules to SPECCTRA at the PCB rule level.

Tip: Copper is translated to a SPECCTRA *keepout*. In addition, a board outline is translated as a *boundary*. Both keepouts and boundaries use the area clearance mentioned above.

Text

Text translates as a keepout. During generation of the SPECCTRA keepout, the maximum text clearance defined in the following clearance rules is used to expand the SPECCTRA keepout area to allow for the required clearance:

- Text-to-Trace spacing
- Text-to-Via spacing
- Text-to-Drill spacing
- Text-to-SMD spacing

Differential Pairs

SPECCTRA does not support minimum/maximum length directly for differential pairs. However, you can add these restrictions using routing rules applied to the pair of nets.

The following SPECCTRA syntax supports net pairs:

```
(pair (nets nm1 nm2 (gap dist)))
Pin pairs use the syntax:
(pair (wires nm1 nm2 (gap dist)))
```

Conditional Rules

Conditional rules are separated into two categories: high-speed and clearance. SPECCTRA can accept conditional rules for layers. However, any other rules can only be conditional if they use net classes. A class can contain only nets, not pin pairs or groups. Therefore, any conditional rule that includes a group or pin pair, and is a high-speed rule, cannot be passed to SPECCTRA.

Nets in conditional rules are automatically placed in dummy SPECCTRA classes using the naming convention NEW_CLASS_# as follows:

```
(class NEW_CLASS1 $$$2016)
```

High-Speed Conditional Rules

A high-speed conditional rule is converted like a nonconditional high-speed rule, except that it must be of the form *class* vs. *class*, as follows:

```
### PADS Layout Conditional Rule ###
(class_class
```

```
(classes CLASS_0 CLASS_1)
(rule (parallel_segment (gap dist) (limit dist)))
(rule (tandem_segment (gap dist) (limit dist)))
)
##############################
```

Clearance Conditional Rules

Conditional clearance rules must be of the form *object vs. layer*, as follows:

```
##########################
### PADS Layout Group Rule ###
(group GROUP_0
(fromto U2-2 U1-2
)
(fromto U4-9 U1-9
)
(fromto U3-10 U4-10
)
(layer_rule 1
(rule
(width dist)
(clearance dist (type wire_wire))
(clearance dist (type via_wire))
etc.
```

Tip: Conditional rules vs. text are ignored. SPECCTRA does not recognize text.

DxDesigner Link

DxDesigner Link is a stand-alone application linking PADS Layout or PowerPCB to DxDesigner for easy passing of data and cross-probing. For more information, see the *DxDesigner Link Help*.

When using DxDesigner Link, set up a project in DxDesigner before using the link.

Attribute Conversion

Attribute values for the Number, Decimal Number, or Measure type properties are automatically converted during the ECO process. For example, a frequency value entered as 100 at the schematic or library, is converted to .1 kHz by default. Also, leading and trailing zeroes are truncated. For example, the decimal number 123.400 becomes 123.4.

Although these conversions are correct, Compare Design, Compare Netlist (in PADS Layout), and the ECO process, detect and report these conversions as differences. Therefore, a design populated with attributes could have thousands of warnings.

To avoid this, you can either:

- Define attributes as Free Text type in the Attribute Dictionary. When you want to use the math functions in the Attribute Manager dialog box, go to the Attribute Dictionary and change the type to Number, Decimal Number, or Measure. Then, before performing a comparison or beginning an ECO, set the type back to Free Text.

or

- If you use the Number, Decimal Number, or Measure types, make sure the attributes are ECO registered and then perform a backward annotation. The values are converted in PADS Layout and backward annotated to the schematic. The schematic and PCB layout will now be in sync.

Passing Attributes Between DxDesigner and PADS Layout

You must specify all attributes to pass to PADS Layout in the DxDesigner configuration file. The pads.cfg file is the standard prefabricated configuration file. This allows passing general attributes to the default PADS Layout Attribute Dictionary. DxDesigner supports the following object types for general attributes: PCB, Part, Net, and Pin.

The pads.cfg file is used for PCBFWD and PCBBCK. An optional control exists in the pads.cfg file to control whether error output for PCBFWD contains a list of attributes that are not passed. Uncomment the *ExcludeNums 7129* line at the beginning of the pads.cfg file to exclude a list of attributes that are not passed from the error output for PCBFWD.

When you execute the Forward or Backward Annotation functions, process indicator dialog boxes appear. These dialog boxes allow you to cancel the action, view reports and monitor progress of the operation.

For more information, see the "Configuration File Structure" in the *Schematic Editor Help* in DxDesigner.

Managing the Selection List

Cross-probing by selecting objects is possible from within either DxDesigner or PADS Layout. Set the Selection Filter for the type of object to select within the applications from which you make selections. The matching object is selected in the other application, regardless of its Selection Filter setting.

Both applications support multiple selections. Clicking an unused area in either application, clears the selection list in both applications.

Selecting from PADS Layout

Selecting Components

Selecting components in PADS Layout results in selecting matching components in DxDesigner. For connectors and unpacked components, the selected part may correspond to more than one DxDesigner item. Corresponding DxDesigner items are listed in the DxDesigner Selections list in the Selections tab.

Selecting Nets

Selecting nets in PADS Layout results in selecting matching nets in DxDesigner. A netname in PADS Layout may correspond to more than one instance of the netname in the DxDesigner schematic. Corresponding DxDesigner items are listed in the DxDesigner Selections list, which is in the Selections tab.

Selecting Pins

A single PADS Layout pin corresponds to a single DxDesigner pin. Selecting a pin in PADS Layout results in selecting the pin in DxDesigner. Selecting multiple pins in PADS Layout results in selecting multiple pins in DxDesigner. Corresponding DxDesigner items are listed in the DxDesigner Selections list, which is on the Selections tab.

Selecting from DxDesigner

Components

Selecting components in DxDesigner results in selecting matching components in PADS Layout.

Nets

Selecting nets in DxDesigner results in selecting the same nets in PADS Layout. If you area select nets in DxDesigner, only one of the nets is selected in PADS Layout.

Pins

When you select a pin in DxDesigner, it selects both the component and the pin in PADS Layout.

Troubleshooting DxDesigner Link

The following errors appear when the Automation server, which enables PADS Layout and DxDesigner to communicate, encounters a problem:

Unable to Connect to VNSD

When you start DxDesigner by clicking DxDesigner on the Tools menu, this error may appear. It does not indicate a PADS Layout or DxDesigner session problem.

Server Busy

This message, or the message below, appears when one application tries to call another application before the Automation server finishes with the previous action.

This action cannot be completed because the "DxDesigner - sch\Sample" application is not responding. Select Switch To and correct the problem.

DxDesigner Closed

If, from DxDesigner, you manually close a DxDesigner session and that session was started in PADS Layout, PADS Layout is not immediately notified that the connection is closed. The following messages appear when you try to send a message to a closed DxDesigner session:

The OLE Automation Server DxDesigner has been terminated, either by the user closing DxDesigner, or by an unexpected event.

Please refer to the On-line Help topic Troubleshooting OLE for additional information.

Debugging information: DxDesigner cause exception code 130:
Unable To Allocate Object Collection.

To reconnect to DxDesigner, click Connect in the DxDesigner Link dialog box.

No Cross-Reference File

If no cross-reference file exists to map the design to the schematic, the following error message appears.

The DxDesigner/PADS Layout Cross Reference File C:\<path>\DRPSCH.PXR could not be properly loaded.

See the Report File - \PADS Projects\CROSSTBLERR

Aborting…

Chapter 13
CAM and CAM Plus

This chapter covers the different output methods you have available. You can create output from a printer, plotter, photoplotter, or NC drill device. This chapter includes information on setting up your printing or plotting device, previewing the output, and creating the output. In addition, you can create computer-aided manufacturing (CAM) output files that are compatible with a variety of automatic assembly and pick-and-place machines.

You use the CAM command on the File menu to produce laser copy, plots, Gerber files, drill drawings, and other manufacturing outputs.

A plot type, drill, silkscreen, or routing, together with an associated output device setup is called a *CAM document*. Using CAM documents, you can keep and quickly run any number of plots you've configured. For example, your document list might include Routing Top, Laser and Routing Top, and Photoplot. The plot configurations are part of the CAM document. You can define a document and print documents singly or in batch mode directly from the CAM dialog box. CAM documents support 30 layers.

Your configurations are saved within the .pcb file, so each file has its own CAM documents list. You can use Import and Export to move a CAM documents list, including the aperture list and drill feed and speed table, between .pcb files.

Tip: You cannot import a Perform or Work aperture list.

For steps detailing how to define CAM documents, see the Defining CAM Documents topic.

Associated Copper and CAM

CAM interprets pads and other objects differently when they are associated with copper in the PCB Decal Editor. For more information, see Associating Copper with Terminals topic. Associating copper shapes and open copper is one way to create hard breakouts in decals.

- Terminals are interpreted as vias.
- Closed copper shapes are interpreted as pads.
- Open copper (a path drawn with copper) is interpreted as a trace.

Interpreting CAM Plane Thermal Graphics

CAM plane thermals are displayed in the CAM Preview dialog box (and in the Layout editor) as indicated below. These calculations are also used for output during printing, pen plotting, and RS-274-X photoplotting operations:

- The outer diameter of the thermal matches the width of the aperture set in the Photo plotter Setup dialog box.
- The inner diameter is 75% of the outer diameter.
- The number of spokes is always four, arranged diagonally.
- The spoke width is 1/6 of the outer diameter.

Tip: The inner width for custom CAM plane thermals is set as the pad size defined in the Pad Stacks Properties dialog box. The outer width is set as the default same-net pad to corner rule.

CAM Document Creation Workflow

1. Create a CAM document. You can create CAM documents using the Define CAM Documents dialog box. For more information, see the Defining CAM Documents topic.
2. Define properties of the CAM document. You must set the CAM document name, document type, and fabrication layers.
3. Assign layers and items to the CAM document. You can assign layers and attributes using the Select Items dialog box. For more information, see the Making Design Objects Visible in CAM Documents topic.

4. Select plot options. You can set the positioning, suppression, and CAM plane layer options using the Plot Options dialog box. For more information, see the Plot Options dialog box topic.
5. Assign output device for the CAM document. You can easily do this using the Define CAM Documents dialog box.
6. Set up output device. Once you have selected the output device, you can set up the device by using the Define CAM Documents dialog box.

Interpreting the CAM Document Types

When you add a CAM document, you must select the type of CAM document that you want to create. Each CAM document type has a specific set of options.

Tip: The default settings of the CAM document types may not suit your decal layer usage. Always verify the document in the CAM Preview window to ensure that all necessary design elements are displayed for the type of document you are generating.

Custom—Create your own definition for a document. Use the Select Items dialog box to specify which objects appear in the document.

CAM Plane—Document CAM Plane layers. CAM planes are always solid copper planes in PADS Layout. To improve visibility in the work area, CAM planes are negative images and take on the background color of PADS Layout. In the CAM Preview window options, you can invert the negative image of the CAM document. By default, conductive elements appear white in the CAM Preview window. Layer 25 (125 in extended layer mode) is frequently used for CAM plane objects where added copper is the opposite - not copper (negative layer). Layer 25 can also be used for defining the oversize of thermals and antipads (an old method), however, the CAM Plane custom thermal setting in the Plot Options dialog box is a better method. Use the Select Items dialog box to add layer 25 if needed.

Routing/Split Plane—Document layers that contain plane areas and/or routing. Use this type for Split/Mixed and No Plane layers. Conductive elements appear black in the CAM Preview window.

Silkscreen—Document top and bottom silkscreen layers. Layer 20 (120 in extended layer mode) is frequently used for placement/nudge outlines as an alternative to the silkscreen layer outline. Use the Select Items dialog box to add layer 20 if needed. Silkscreen items appear black in the CAM Preview window.

Paste Mask—Document top and bottom paste mask layers. Black areas are paste locations. White is the paste mask. Paste is only used for surface mount components. Pads of through-hole components are not shown.

Solder Mask—Document top and bottom solder mask layers. Black areas are solder locations. White is the solder mask.

Assembly—Document assembly drawings. Component outlines may have been created on the top layer, silkscreen layer, and/or layer 20. Use the Select Items dialog box to add the layers you need. Assembly items appear black in the CAM Preview window.

Drill Drawing—Document drill locations. A drill table is added automatically and appears at a location specified in the Drill Drawing Options dialog box.

NC Drill—Generate an NC Drill file. This document is not intended for viewing. It contains the x and y location and drill size of each hole required in the design.

Verify Photo—View and verify any existing CAM document. The document to be viewed can be from a different design. Verify Photo supports the macros, aperture selection, and fill area commands of PADS Layout Gerber outputs. Verify Photo plots can only process RS-274-X Gerber files created by PADS Layout.

RS-274-X Format

The RS-274-X format is based on the *Gerber Format Guide* (Document Number: 0000-00-RM-000, Part Number: 414-100-002) by Gerber Systems Corporation. RS-274-X in PADS Layout is a data format created in CAM for photoplotters. Clicking the RS-274-X output format from the Photo plotter Advanced Setup dialog box creates a Gerber file with the information shown in Table 13-1.

Table 13-1. RS-274-X Fields

Field	Description
AM	Aperture macro
AD	Aperture description
FS	Format statement
MO	Units mode
IN	Image name
LN	Layer name
LP	Layer polarity
G36,G37	Fill area controls

These features are allowed for the 9500, 9800, 9900, GPC, and Insight/2020 photoplotters that also support the G74,G75 multiquadrant circular interpolation function codes. Some of these

parameters appear in the Gerber file depending on photo plotter settings; however, the parameters in Table 13-2 are always present in the Gerber file.

Table 13-2. Gerber File Fields

Field	Description
The name of the design	%IN *job name* *% where *job name* is the name of the design.
The mode parameter	Indicates the units, for example, %MOIN*% or %MOMM*%.
A format statement parameter	Describes the format options selected in the Photo plotter Advanced Setup dialog box, for example %FSLAX45Y45*%.

RS-274-X File Details

Aperture Table

All simple apertures; such as round, oval, or rectangular; are described in the output Gerber file as a %ADDxx*% parameter. The Photoplotter Aperture Report file is also created.

Aperture Macro

Copper areas associated with pins are output as unique aperture flashes:

- Line-shaped polygonal copper with 50 vertexes or less and circular copper are supported.
- Line-shaped polygonal solid copper with 50 vertexes or less and circular solid copper are supported.
- Arc-shaped copper outlines are aperture macros in Gerber using polygonal approximation with up to 50 vertices. The %AM*% parameter limits the number of vertices to 50.
- Thermal reliefs for CAM planes are output as macro flash.

Tips:

- PADS Layout does not support copper cutouts for coppers associated with component pins.
- PADS Layout does not support hatched coppers associated with component pins. All associated coppers appear and output in CAM as solid, regardless of the grid spacing and line width.

Fill Area

Solid, not hatched, copper and copper pour areas are output in fill area mode (G36,G37 brackets). Hatched areas are output in vector format. Circular copper areas, arc-shaped polygons, and circular and polygonal copper cutouts are all supported.

Verify Photoplots

Verify Photoplots Document Type supports the macros, aperture selection, and fill area commands of PADS Layout Gerber outputs. Verify Photoplots can only process RS-274-X Gerber files created by PADS Layout.

CAM Plus Assembly Machine Interface

The CAM Plus command generates computer-aided manufacturing (CAM) output files that are compatible with a variety of automatic assembly and pick-and-place machines. Supported formats include Dynapert, Siemens, Universal, and Quad.

> **Tip**: You must prepare an information file called part.def before using CAM Plus. This may require assistance from manufacturing engineering to determine which parts to insert by which machines, and to set standard bin assignments. Also, place all design components at their final locations before you start CAM Plus.

Batch Mode and Mask Mode

CAM Plus uses the part.def file differently depending on whether you click Masked in the Parts list in conjunction with Batch Part Definition File.

- With Masked selected and this option on, CAM Plus sorts parts to all machines as defined in the part.def file, ignoring non-defined parts, Part D.
- With Mask selected, this option off, and a specific machine selected, CAM Plus sorts only parts defined in part.def as assigned to that machine.
- With something other than Masked selected and this option on, CAM Plus sorts parts assigned in part.def to their respective machines, but includes undefined parts in files for all machines.
- With something other than Masked selected, this option off, and a specific machine selected, CAM Plus sends all parts to the selected machine.

Board Offset Values

The Board Offset command defines the offset from the machine’s location dowel to the 0,0 system origin board. This offset value converts the design coordinates to the machine origin.

- The default is 0 offset, equivalent to treating the CAD system origin as the origin of the machine. Another way to think of this is that the X and Y offset are the distance from the location dowel of the machine to the PADS Layout system origin. If the location dowel is in the lower left corner, this will be a positive value.
- The maximum offset is 10 inches.
- All distance values are given in inches; for example, 1250 is 1.25 inches.
- You may need to define a new board offset for each machine.

Step and Repeat Values

The Step and Repeat command defines whether to treat the board as a single design when creating the output program file, or to insert a number of boards simultaneously. If the latter, you also define the number of steps in the X and Y directions and the step and repeat interval to use.

CAM Plus uses current design units for the offset and step and repeat values. Each machine type has its own unit type and the data is always converted to the unit type appropriate for the machine regardless of the current design units.

The program generates assembly program files for inserting parts on all boards.

Table 13-3. Step and Repeat Fields

Field	Description
X Count	Number of copies in X direction
Y Count	Number of copies in Y direction
X-Step	The step distance in the X direction; the distance between the origins of each board
Y-Step	The step distance in the Y direction; this is the distance from the origins of each board

The default is no step and repeat, equivalent to a step of 1 in X and Y.

The maximum number of steps in X or Y is 20. The maximum step and repeat offset is 10 inches.

Output Format

Files are produced for all parts of a selected class: masked, through hole, SMT, top, bottom, and so on. All parts in the class are included in this output file, whether or not their insert class is defined as belonging to the specific machine.

When an output is performed, the resulting file or files are stored in a program-created folder under \PADS Projects\cam. The subfolder matches the design's file name. For example, if the design loaded when CAM Plus is started is test.pcb, the results are stored in \PADS Projects\cam\test.

CAM Plus Report File Names

This section explains the file names created by CAM Plus for various insertion machines. The prefix name is that of the machine. The prefix for each machine type is shown under the heading for each machine in the "Supported Machine Formats" on page 256. The last two characters in the output file name (shown as xx) are determined by the components selected.

Table 13-4 shows the first letter in the pair of ending characters.

Table 13-4. Output File Name Ending Characters (First Letter)

Letter	Description
M	If the output selected is mask
S	If surface mount is selected
T	If through hole is selected
A	If all parts are selected

Table 13-5 shows the second letter is the pair of ending characters.

Table 13-5. Output File Name Ending Characters (Second Letter)

Letter	Description
T	If components on the top layer are selected
B	If components on the bottom layer are selected

Universal Tooling and Universal Axial Output

Universal-specific instructions are active if you click Universal machine formats, or if you check batch part.def file.

Status Messages

Displays the current state of output.

Supported Machine Formats

The following are supported machine formats:

- Siemens HS-180
- Siemens MS-72k
- Quad 100
- Daum SMF
- Zevatech PPM-9
- Universal Machine Output Files
- Universal Sequencer File
- Philips CSM Format
- Parts List

Siemens HS-180

Output File Name = SIHS18xx.BP

As shown in Table 13-6, the header information of the Siemens HS-180 output contains six lines of data:

Table 13-6. Siemens HS-180 Output File Header Information

File Name	Input ASCII File Name
U CAD-ASCII	The type of data
1.0	Version number
comment	General comments for example, PADSCIM OUTPUT
date	File creation date
num_parts	Number of parts inserted by the HS-180 machines, followed by the number of parts inserted by each machine in the line (not supported)

A single line entry, in the following format, exists for each part:

```
Line_number, Part_Type, X_loc, Y_loc, Rotation, Code, Ref_Name
```

Table 13-7 describes the fields in each line entry.

Table 13-7. Siemens HS-180 Output File Line Entry Fields

Field	Description
Line_number	The order of the parts, starting with 1
Part_Type	The part type information; discrete parts include part values
X_lOC	The X insertion coordinate of the part (in mils)
Y_lOC	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees
Code	(blank)
Ref_Name	The reference designator

Siemens MS-72k

Output File Name = SIMS72xx.BP

As shown in Table 13-8, the header information of the Siemens MS-72 output file contains ten lines of data.

Table 13-8. Siemens MS-72 Output File Header Information

Data	Description
BP/MS72	The type of data
V1.0	The version number
file name	Input ASCII file name, that is test.ast
comment	General comment, for example, PADSCIM OUTPUT
date	File creation date
component file	(left blank)
cluster file	(left blank)
ink spot	(always set to "0")

A single line entry, in the following format, exists for each part:

```
Line_number, Part_Type, X_loc, Y_loc, Rotation, Ref_Name, M_Code
```

Table 13-9 describes the fields in each line entry.

Table 13-9. Siemens MS-72 Output FIle Line Entry Fields

Field	Description
Line_number	The order of the parts, starting with 1
Part_Type	The part type information. Discrete parts include part values
X_lOC	The X insertions coordinate of the part (in mils)
Y_lOC	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees
Ref_Name	The reference designator
M_Code	The part is to be inserted, always set to asterisk (*)

Quad 100

Output File Name = QUADxx.100

There is no header information in the file. A single line entry, in the following format, exists for each part:

```
Ref_Name, X_loc, Y_loc, Rotation, Part_Num
```

Table 13-10 describes the fields in each line entry.

Table 13-10. Quad 100 Output File Line Entry Fields

Field	Description
Ref_Name	The reference designator
X_lOC	The X insertion coordinate of the part (in mils)
Y_lOC	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees
Part_Num	Part number (from the Part Definition File)

Daum SMF

Output File Name = DAUMxx.SMF

Table 13-11 shows the header information of the DAUM.SMF output file, which contains 24 lines of data.

Table 13-11. Daum SMF Output File Header Information

Data	Description
Machine directory	Directory where part information is held in the Daum machine
;	
;	
PLACER;	Standard entry
Innoveda;	Standard entry
;	
;	
N;	Optimize output (Y or N)
N;	Optimize feeder info (Y or N)
2.54;	Scale factor of output
rotation;	Degrees of rotation
axis;	Axis of rotation (N = none)
height;	Height of boards
Board_Length;	Length of board (not used)
Board_Width;	Width of board (not used)
X_Offset;	X offset from board origin
Y_Offset;	Y offset from board origin
0;	Step & repeat (not used)
0;	Step & repeat (not used)
0;	Step & repeat (not used)
0;	Step & repeat (not used)
Number_Parts;	Number of parts to insert
Number_Feeders;	Number of feeders to use
Version 1.0;	Version number

A single line entry, in the following format, exists for each part:

```
Ref_Name, part_type, value, case_type, tolerance, trash_box, part_number,
X_loc, Y_loc, rotation, feeder_num, feeder_type, current_component,
feeder_num, stop-value;
```

Table 13-12 describes the fields in each line entry.

Table 13-12. Daum SMF Output File Line Entry Fields

Field	Description
Ref_Name	The reference designator
part_type	The part type information
value	Discrete parts include part values
case_type	The type of case
trash_box	Waste bin for bad parts
part_number	Part number (from the Part Definition File)
X_lOC	The X insertion coordinate of the part (in mils)
Y_lOC	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees
Feeder_num	Feeder number as defined in PART.DEF.
feeder_type	Feeder type as defined in PART.DEF.
current_component	
Sequence number	
stop-value	Set to 0

The next section is the bin or feeder assignments. It contains the following entry for each unique part type:

```
feeder_number, feeder_type, component_type, value, case, tolerance,
trash_box, tool, parts, 0, feeder_number;
```

Each entry is the same as described above.

Zevatech PPM-9

Output File Name = ZEVAxx.PPM

The header information of the Zevatech output is one line of data, containing the version number and the date the report was generated. A single line entry, in the following format, exists for each part:

```
PLACE Ref_Name X_loc Y_loc Rotation 0 0 00 Part_Type
```

Table 13-13 describes the fields in each line entry.

Table 13-13. Zevatech PPM Output File Line Entry Fields

Field	Description
PLACE	The standard starting entry
Ref_Name	The part reference designator
X_loc	The X insertion coordinate of the part (in mils)
Y_loc	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees.
0 0 00	Fixed entries
Part_Type	Part type from .ast file. Discrete parts include part values

Universal Machine Output Files

Table 13-14. Universal Machine Output Files

Machine	Output file
Universal 6241B Axial Inserter/Sequencer	UN6241xx.PUT
Universal 6348A Radial Inserter/Sequencer	UN6348xx.PUT
Universal 6772A Multi-Module Dip Inserter	UN6772xx.PUT
Universal 6287A Axial Inserter	UN6287xx.PUT

There is no header information in the machine program format. A single line entry, in the following format, exists for each part:

```
line_number> X_loc Y_loc z_span depth_stop anvil good bad MC CC
```

Table 13-15 describes the fields in each line entry.

Table 13-15. Universal Machine Output File Line Entry Fields

Field	Description
line_number	The line number in the file
X_loc	The X insertion coordinate of the part (in mils)
Y_loc	The Y insertion coordinate of the part (in mils)
z_span	The programmed z span entry
depth_stop	The insertion depth for the part
anvil	The anvil number and dispensing head number
good	The good count, initially set to 0
bad	The bad count, set to 0
MC	The alphabetic modifier code for the Z-span
CC	The insertion command code

The bin use report is a text file, with the name UN6241xx.BIN, also stored in \PADS Projects\cam. This file is designed for use by the machine operator.
A single line entry, in the following format, exists for each part:

```
Part_name Bin_number Pre_allocation Check
```

Table 13-16 describes the fields in each line entry.

Table 13-16. Bin Use Report File Line Entry Fields

Field	Description
Part_name	Part to load in the bin, taken from the PADS Layout part type
Bin_number	The bin assignment
Pre_allocation	Indicates if the bin was pre-assigned in the part.def file or was assigned by the output program
Check	Enables the machine operator to mark if the part is loaded. There is no field to determine whether the machine operator is loaded

Universal Sequencer File

The sequencer file is a Universal program with the name UN6287.SEQ, used to program a sequencer machine that is Sequencer 1 compatible. There is no header information in the file. A single line entry, in the following format, exists for each part:

```
line> +00000 +00000 +00000 +00000 Bin_number +00000 +00000 B C
```

The following table describes the fields in each line entry.

Field	Description
line	The line number in the file
Bin_number	The bin number to load from

The sequence list is used to program other sequencing machines. It is an ASCII file containing the following information for each part:

```
Sequence: number Part Name: part_type
```

The following table describes the fields in each line entry

Field	Description
number	The insertion sequence number
part_type	The type of part to insert

Philips CSM Format

Output File Name = PHILIPxx.CSM

Table 13-17 shows the header information.

Table 13-17. Phillips CSM Output File Header Information

Data	Description
*DESIGN	Standard entry for start of header information
Formatvs = 1	
Board = Name	
Twelvenc = 1234 567 89012	
Maskcntr =	
Issuedat =	
Revisdat =	
Userid =	
Unit = M (M for metric, or 1 for inches)	
*COMPONENT	Standard entry for end of header information

A single line entry, in the following format, exists for each part:

```
Ref_Name= (Part_Type{value},Part_Number,,,X,Y,Rotation,Layer
```

Table 13-18 describes the fields in each line entry.

Table 13-18. Phillips CSM Output File Line Entry Fields

Field	Description
Ref_Name	The reference designator
Part_Type	The part type from the .ast; discrete parts include part values
Part_Number	Part number (from the Part Definition File)
X	The X insertion coordinate of the part (in mils)
Y	The Y insertion coordinate of the part (in mils)
Rotation	The rotation of the part in degrees
Layer	Top or Bottom layer

Parts List

This command outputs a general purpose parts list, in the following format, containing all parts on the board.

```
*** CIM Status Report ***
Ref Part Name Level Inserted by:
```

Table 13-19 describes the fields in each line entry.

Table 13-19. General Purpose Parts List Line Entry Fields

Field	Description
Ref	The component reference name
Part Name	The PADS Layout library part type
Level	The layer, either Top or Bottom
Inserted by	The name of the machine for inserting the part

Part Definition File

The Part Definition File, located by default in C:\MentorGraphics\<*latest_version*>PADS\SDD_HOME\Libraries, points particular part shapes to the specific machines that specialize in inserting them. For each part type that you insert automatically, an entry should exist in the part.def file. The entry is a single line, and the order in the file is not important. If an entry is not found, a warning is sent to the error file cimrep.err.

Usually the Read Part Definitions option is selected. The only time part.def would not be necessary is if you output to a selected machine.

Format of the part.def File

The entries in the part definition file are case sensitive. Also, no specific header data is required. Additional information depends on the machine inserting the part. A single line entry, in the following format, exists for each part:

```
part_type, ins=machine_name,
```

Table 13-20 describes the fields in each line entry.

Table 13-20. Part Definition File Line Entry Fields

Field	Description
part_type	The part type from PADS Layout.
machine_name	The name of the insertion machine that will insert the part.

Machine Codes

Every part requires a specific machine code entry. Table 13-21 shows the six-character codes for the supported machines.

Table 13-21. Machine Codes

Machine	Machine Code
Dynapert 318	dynpro
Philips CMS	philip
Daum	daum
Siemens_HS_180	sims18
Siemens MS 72	sims72
Zevatech PPM-9	zeva
Quad 00	quad
Universal 6241	un6241
Universal 6348	un6348
Universal 6772	un6772
Universal 6287	un6287
uParts	(no insert)

Some insertion machines require additional data for each part. The order in which each data item is added is not important. Each entry must be separated by a comma (,).

Dynapert

```
number= part_number
```

In the above line, part_number represents the company part or stock number.

Siemens

Parts inserted by Siemens MS-72 and HS-180 require no additional data.

Daum

```
type=type_name, number=part_number, val=value, tol=tolerance, feedtype=
feeder, pip= nozzle_type.
```

Table 13-22 shows the fields for each line entry.

Table 13-22. Part Definition File Daum Line Entry Fields

Field	Description
type_name	Brief description of the part: resistor, capacitor, and so on.
part_number	The company part or stock number.
value	The field used to describe discrete part values
tolerance	The field used to describe discrete part tolerances
feeder	The type of feeder to use to insert the part; options are 1, 2, or 3
nozzle_type	The nozzle number used to insert the part; options are 1 through 6

Quad

```
number= part_number
```

In the above line, part_number represents the company part or stock number.

Philips

```
number=part_number,val=value,tol=tolerance,
```

Table 13-23 shows the fields for each line entry.

Table 13-23. Part Definition File Phillips Line Entry Fields

Field	Description
part_number	The company part or stock number
value	The field used to describe discrete part values
tolerance	The field used to describe discrete part tolerances

Universal

```
bodydiam=body, leaddiam=lead, anvil=anvil, feednum= bin_number
```

Table 13-24 shows the fields for each line entry.

Table 13-24. Part Definition File Universal Line Entry Fields

Field	Description
body	The diameter size in mils
lead	The lead diameter in mils

Table 13-24. Part Definition File Universal Line Entry Fields

Field	Description
anvil	The bend style
bin_number	The number of any pre-assigned bins

Zevatech

```
part_type, ins=zeva
```

Bin Assignments

Assigning parts to bins is an important function in CAM Plus. Parts must be loaded into the bins before the insertion process starts. This can take several hours for complex machines. Commonly used parts are often kept permanently in some bins. You can define this in CAM Plus for Universal and Daum insertion machines when preassigned bins are used. Other parts are assigned to arbitrary bins.

CAM350

CAM350 is a pre-production CAM system that combines DFM analysis, DRC checking, test fixturing, planning, and tooling. The CAM350 products range from Gerber viewers to full-featured CAM editors that process PCB databases into usable fabrication and panel data. CAM350 lets you analyze your design for manufacturing issues prior to fabrication, drastically reducing cycle time and cost. The tools are based on a fully-intelligent CAD database and can input and output virtually any Gerber format, IPC-D-350/IPC-D-356 data, CAD database, or netlist.

CAM350 Link

CAM350 Link is a PADS Layout option that automatically translates a design database into a CAM350 (version 6.0 and greater) database. You no longer need to generate an ASCII file for translation; CAM350 Link uses the native PADS Layout file format and adds a .cam extension. In addition to converting designs for translation, you can launch CAM350 and load the current database directly from the CAM350 dialog box.

CAM350 Link also supports backward annotation of Design For Manufacture (DFM) errors to a PADS Layout database so that you can identify and correct DFM errors in PADS Layout instead of making the corrections in CAM350.

See the Defining CAM Documents topic for information on outputting CAM documents from PADS Layout. See the Setting Up the CAM350 Link topic for information on CAM350 Link procedures.

CAM350 Link Non-Supported Objects

The following objects are not passed from PADS Layout to CAM350 during CAM350 Link translation when translating a CAM document with nets and when translating CAD layers to CAM layers:

- Associated pin copper is merged into a single layer custom aperture. Associated pin copper is always merged to the pad layer during translation to CAM350. The result is that associated copper is not handled on layers other than the pad layer. In addition, associated copper is not distinguished from pad flashes.
- Combined text is not output with owning lines in CAM documents. When combined text and lines are translated to CAM350, combined text visibility is controlled by text visibility in the CAM document itself, not by the line visibility in the CAM document settings.
- The pad oversize option in the CAM documents is not supported when translating to CAM350.
- Hatched copper is not supported for PADS-format ASCII import. Hatched copper is converted to solid fill copper rather than hatched copper when translating to CAM350.
- Decal lines and text on the Top or Bottom electrical layers are not distinguished from lines and text for Top or Bottom silkscreen layers. Decal lines and text from the Top and Bottom electrical layers are moved to silkscreen layers during CAM350 translation. Visibility control for Top and Bottom lines and text can not be distinguished from visibility control for silkscreen lines and text.
- Jumper reference designators and outlines are moved to silkscreen layer. Jumper reference designators are converted to free text and jumper outlines are converted to

lines in CAM350. Jumper reference designators and outlines are also moved to the silkscreen layer during translation.

The CAM350 Analysis feature reports jumpers as net connectivity errors. CAM350 does not support CAM negative planes on outer mounting layers. CAM350 does not support a layer type of negative plane for outer mounting layers.

Test Points

Test points in CAM350 are used for Bed of Nails testing or Flying Probe in-circuit testing. Test points in PADS Layout are used for in-circuit testing using fixtures. Although the test configurations and rule data are similar, there is no direct mapping between PADS Layout test point configuration data and CAM350 configuration data. However, test point status is passed to CAM350 for information purposes. The test point status and the access side are translated from the ASCII file to the CAM350 database.

Test point configuration data and clearance rules are not passed to CAM350 from PADS Layout.

Probe sizes are not passed to CAM350. Only the test point location and probe side are passed.

CAM350 Link Document Conversion

PADS Layout CAM documents combine layer and data type specifications for output in a CAM photoplot file. A new layer is created in the CAM350 file for each PADS Layout CAM document. For more information on defining, adding, and editing CAM documents see the Defining CAM Documents topic.

CAM documents can be arbitrarily named. These arbitrary names are used for layer names in the CAM350 database to hold the CAM document content. The CAM350 Link extracts the PADS Layout data types from the PADS-format ASCII file for the layers specified in the MISC CAM section of the ASCII file. The resulting layers in the CAM350 database match the content of the Gerber photoplot file generated by PADS Layout CAM operations.

Tip: CAM document names are truncated to 16 characters due to the maximum file name length in CAM350.

Supported CAM documents for CAM350 Link include the following five types:

- Plane: ground plane (pads, vias, copper, lines, text)
- Routing: top (pads, vias, tracks, copper, lines, text)
- Silkscreen: (outline top)

top: (ref. des., part type)

silkscreen top: (lines, text, outlines)

- Pastemask:

top: (pads)

paste mask top: (copper, lines, text)

- Solder mask:

top: (pads, test points)

solder mask top: (copper, lines, text, test points)

Chapter 14
Programming

This chapter discusses macros and Basic scripting.

Macros

You can edit, run, and debug macros in the Macro tab. A macro is any combination of commands, keystrokes, and mouse clicks that you record to later replay. You can record virtually any set of procedural steps for replay, thus simplifying redundant activities such as setting preferences and settings.

The Macro editor supports standard editing features. It includes unlimited Undo/Redo operations, syntax highlighting, and provides context-sensitive help on the Macro language. You can also open multiple macros in the macro editor. You can move between open macros and edit them. You can nest macros within other macros.

The embedded Macro debugger lets you:

- Start, pause, and stop macro execution.
- Insert breakpoints into the macro for step-by-step execution.
- Report run-time errors.

Macro Text Colors

Text colors in the macro editor represent the following:

Table 14-1. Text Color Representations

Color	Meaning
Blue	Basic Keywords
Black	User Variables
Cyan	Basic Functions
Purple	Automation Objects or Members
Red	Errors
Green	Comments

Macro Editing Shortcuts

You use standard Windows editing functions when editing macros. Use the mouse to select areas of text.

Table 14-2 shows the default shortcut keys for macro editing functions.

Table 14-2. Default Macro Editing Shortcuts

Command	Description
Up, Down arrows	Move the insertion point one line up or down.
Left, Right arrows	Move the insertion point one character left or right.
Home, End	Move the insertion point to the beginning or the end of the line.
Page Up, Page Down	Moves the insertion point one page up or one page down.
Ctrl+Home	Moves the insertion point to the top of a file.
Ctrl+End	Moves the insertion point to the end of the file.
Ctrl+Left, Ctrl+Right	Moves the insertion point one word left or right.
Shift+pointer move command	Moves the text between the start and end insertion points.
Ctrl+C, Ctrl+Ins	Copies text selected in the Edit area to the Clipboard.
Ctrl+V, Shift+Ins	Pastes the contents of the Clipboard to the Edit area.
Ctrl+X, Delete	Cuts selected text from the Edit area to the Clipboard.
Alt+Backspace, Ctrl+Z	Undoes the last edit. This is a multilevel undo. **See also:** "Undo and redo" in *PADS Router Help*

Table 14-2. Default Macro Editing Shortcuts (cont.)

Command	Description
Ctrl+Shift+Z	Restores the edit rescinded by the Undo command. **See also:** "Undo and redo" in *PADS Router Help*
Ctrl+A	Selects all.
Ctrl+O	Opens the macro.
Ctrl+S	Saves the current macro.
Ctrl+P	Prints the current macro.

Reporting Run-Time Errors

If a run-time error occurs, the debugger suspends running the macro, and displays a message on the status bar. The Instruction pointer is set on the line that produced the error. After fixing the error, you can resume running the macro.

Recording Mouse Movements

When recording a macro, you can record mouse movements in these ways:

- Record movement from coordinate A to coordinate B without recording the intermediate coordinates (compressed).
- Record movement from coordinate A to coordinate B including all of the intermediate coordinates (uncompressed).
- Record movement from coordinate A to coordinate B using incremental coordinates (relative).
- Record movement from coordinate A to coordinate B using absolute coordinates.

Compressed Mouse Mode

Compressed mouse mode records only the start point and end point of a mouse movement. Compression is recommended under most circumstances because it significantly reduces the size of your macro file. Recording intermediate mouse movements increases the file size, but documents coordinate information if required for a special application.

Relative Mouse Mode

Relative mouse mode records the start point and end point of a movement in incremental coordinates instead of absolute coordinates.

Recording a session

You can record a session upon start-up using command line switches. You can record macros in the background or load and run a macro when you start the program.

A log file records all of your actions during a session. This file is useful when trying to reproduce problems during debugging. Events are logged to the log file and to the macro editor. Each time you create a log, the previous .log file is saved to a .bak file, so you can retrieve a record of your actions up to the last recorded session. The macro .log file is written to the folder specified in the General tab of the Options dialog box. The default folder is \PADS Projects.

To enable this feature, modify the shortcut and add a "–log" command line parameter.

Tip: If you use the PADS Router Link from PADS Layout, the .log file is not written.

For more information, refer to Recording a Session in *the User's Guide*.

Basic Scripting

PADS Layout includes an internal scripting capability, the Sax Basic Engine‰, by Sax Software Corporation. The Sax Basic Engine makes the Automation features within PADS Layout more accessible to everyone—developer and nondeveloper. This engine includes the entire development environment required to develop Basic scripts, including editor, debugger, interpreter, tracer, variable watch, dialog editor, Automation object browser, and new/open/save/print capabilities. These capabilities ensure that every user can work with Basic scripting without previous programming knowledge.

Basic is a scripting language developed to provide users with a unified language in Windows 98, Windows 2000, and Windows NT. More and more Windows applications like PADS Layout include Basic capabilities, such as Microsoft Word and Microsoft Excel', to let users customize their application with a standard scripting language.

Scripts written with the editor comply with all Microsoft requirements in terms of Basic syntax and, therefore, you can play these scripts in any other Basic interpreter, such as Word or Excel. However, you cannot run Basic scripts created outside of the Sax Basic Engine within the Sax Basic Engine because the Sax Basic Engine is a subset of Basic. You cannot, for example, run the Automation samples within the Sax Basic Engine.

You can create a script that calls another script. For example, ScriptA can call ScriptB. A script can also call a PADS Layout macro using the RunMacro() Automation call. A PADS Layout macro, however, cannot call a script.

You can also create a script that runs a series of scripts, or a "master" script. For example:

```
'$Include: "scriptA.bas"
'$Include: "scriptB.bas"
'$Include: "scriptC.bas"
Sub Main
                                                    Call scriptA
   Call script B
   Call script C
End Sub
```

As shown in Table 14-3, the editor displays source code using different colors.

Table 14-3. Text Colors Representation

Color	Represents
Blue	Basic Keywords
Black	User Variables
Cyan	Basic Functions
Purple	PADS Layout Automation Objects or Members
Red	Errors
Green	Comments

The color is context-sensitive. That is, when you place the cursor on the text and press F1, the correct help file opens to the correct help topic. For example, if the cursor is on a PADS Layout Automation Object when you press F1, the *PADS Layout Automation Server Help* appears.

For more information on the Basic editor and language, see the Sax Basic Editor Help. In the editor, right-click, point to Help, and then click Editor Help or Language Help. (C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Programs\sbe5_000.hlp).

In addition to the editor, PADS Layout has a script manager that lets you manage the existing Basic scripts listed in Table 14-4. Using this script manager, you can load your most frequently used scripts and run them independently of the editor itself.

Basic Sample Scripts

Table 14-4. Basic Sample Scripts

	Script	Description
00	What is a VB Script.BAS	Empty script demonstrating what a Basic script is and how to define it.
01	Using a Message Box.BAS	Demonstrates how to display an OK dialog box.
02	Using a Variable.BAS	Demonstrates a how to assign a value to a variable and how to get its value.
03	Using a VB Function.BAS	Demonstrates how to use a standard Basic function and display its result in a message box.
04	Using a PADS Layout Function.BAS	Demonstrates Basic interaction with a PADS Layout Automation function.
05	Using If and Then Statements.BAS	Demonstrates If, Then statements.
06	Using a Custom Dialog1.BAS	Demonstrates a simple dialog box using the dialog box editor.
07	Using a Custom Dialog2.BAS	Demonstrates a standard dialog box using the dialog box editor.
08	Using a Custom Dialog3.BAS	Demonstrates a complex dialog box using the dialog box editor.
09	Using It All Together.BAS	Provides a "real life" example. Lists all .pcb files in \PADS Projects. Selecting a file from the list opens that file in PADS Layout.
10	List of Comps and Nets.BAS	Lists all components and nets.
11	Select by Pin Count.BAS	Selects all parts that have the number of pins you enter.
12	Move by Pin Count.BAS	Moves all parts that have the number of pins you enter.
13	Width Table.BAS	Lists all possible width values. Selecting a width changes the current width.
14	Part List Report.BAS	Creates a Part List report from the open .pcb or .bga file in Microsoft Excel.
15	Pin List Report.BAS	Creates a Pin List report from the open .pcb or .bga file in Excel.

Table 14-4. Basic Sample Scripts (cont.)

	Script	Description
16	Via List Report.BAS	Creates a Via List report from the open .pcb or .bga file in Excel.
17	Select All Test Points.BAS	Selects all test points.
18	Part Web Search.BAS	Searches major semiconductor manufacturers' Web sites for the selected part's part type.
19	Off-Grid Pins.BAS	Lists all pins and vias that are off the current grid.
20	PADS Layout Script Wizard.BAS	Generates a Wizard dialog box, which you use to create a Basic report.

Basic Sample Scripts/RGL Reports

Table 14-5 describes scripts that are equivalent to existing Report Generation Language (RGL) reports. These files are located in C:\PADS Projects\Samples\Scripts\Layout\RGL_Samples.

Table 14-5. Sample Scripts Equivalent to RGL Reports

Script	Purpose
RGL.BAS	Contains a library of functions, which is used by the other scripts in this group; the scripts in this group must contain RGL.BAS to function.
Netlist Without Pin Info.BAS	Reports signals by netname without pin information.
Netlist With Pin Info.BAS	Reports signals by netname with pin information.
Part List 1.BAS	Reports parts by reference designator.
Part List 2.BAS	Reports reference designator by part type.
Test Points.BAS	Reports test point locations and netnames.
Jumper List.BAS	Reports jumper locations and netnames.
PowerPCB V2.0 Format netlist.BAS	Reports a PowerPCB V2.0 format netlist.
PowerPCB V3.0 Format netlist.BAS	Reports a PowerPCB V3.0 format netlist.
DFT Extended Test Point.BAS	Reports test points by nets, nets without test points, and the number of test points per net.

Chapter 15
OLE in PADS Layout

This chapter discusses object embedding and OLE automation capabilities.

With object embedding capabilities, you can embed an object into a design. You can also link objects to their source so linked objects automatically update each time you open the PADS Layout database. Finally, using Automation you can develop custom, "plug-in" applications using object-oriented programming (OOP) techniques with applications such as Basic, Microsoft® Word, and Visual C++®.

Table 15-1 shows PADS Layout OLE functions.

Table 15-1. OLE Functions

Function	Description
OLE Automation Server	Third party companies can integrate their products with PADS Layout, in the same way that PADS Layout integrates with DxDesigner. Using the OLE Automation server you can customize existing features and automate tasks using standard scripting languages, such as Basic. Descriptions of the PADS Layout OLE Automation server are provided in a help file, ppcbole.hlp, devoted to Automation samples and Automation server API commands.
Object Linking and Embedding	You can link or embed objects from other applications within a design.

Object Linking and Embedding

You can insert other files or applications as linked or embedded objects within a PCB design. For example you can insert a Microsoft Word document containing manufacturing information, a Microsoft Excel spreadsheet containing a Bill of Materials, or a DxDesigner schematic. PADS Layout does not need to understand the format of an inserted object because it communicates with the application that created the file. The source application tells PADS Layout what information to display and how to display it.

Tip: You cannot insert OLE linked or embedded objects in the PCB Decal Editor.

The following are restrictions:

- The insertion of PADS Logic, Layout or Router files as OLE objects in other files (including other PADS files) is not supported. Any PADS Logic, Layout or Router file inserted in another file will not behave properly and cannot be edited within the “container” application (Visual Editing).
- Nested embedding (inserting an object into an already linked or embedded object) is not supported.

Linking and Embedding Objects

If you insert an object whose source application is an OLE linking and embedding server, that application opens inside PADS Layout, but runs in background. The source application's toolbar takes over the PADS Layout toolbar. You can then work with the source application in

the same way as you would if you started it outside of PADS Layout. This is called visual editing.

When you click outside of the object, the PADS Layout toolbar takes over again. You can continue to design in PADS Layout. Because the source application continues to run in background you can click on the object and work in the source application at any time.

> **Tip**: Once you insert an object into a PADS Layout design, you can export it to a .ole file and import it into other PADS Layout designs.

Displaying OLE Objects

PADS Layout does not understand the format of an inserted object. It communicates with the application that created the OLE object to display the information. If the application that created the OLE object is installed and registered on your system, PADS Layout calls upon that application to display the OLE object in PADS Layout as it would appear in the source application. For example, a Microsoft Word document can appear within PADS Layout and the Word toolbars appear within PADS Layout.

If the source application is not installed and registered on your system, PADS Layout can display the inserted OLE object only as an icon, not as it would appear in the source application. PADS Layout also displays the OLE object as an icon if the object is an application.

OLE and View Menu Commands

You can use all of the View menu commands with OLE objects. You can zoom into and zoom out of the objects.

Turning Display Off

You can cancel OLE object display using the Display OLE Objects option on the Global tab of the Options dialog box. You may want to cancel OLE object display when PADS Layout contains many linked or embedded objects to increase redraw speed. Redraw speed decreases noticeably if the OLE Linking and Embedding servers, or source applications, which actually display the OLE items, are not optimized for remote display.

Changing Background Color

OLE objects appear with a solid white background. You may want to display the OLE object with a transparent background so you can see the bitmap background. If your object is a Word document, you may want a white background because black text on a transparent background results in black on black, or an invisible object. For more information on how to change the background color, see the Change the Background Color of an OLE Object topic.

Redrawing a Screen

When PADS Layout redraws, it redraws components first and then OLE objects. OLE objects always redraw in the same order and always redraw after PADS Layout objects; therefore, OLE objects always appear on top of components.

Other OLE Editing Commands

Once you insert an OLE object, the name of the object appears at the bottom of the Edit menu. For example, if you insert a video clip, Video Clip Object appears at the bottom of the Edit menu. A cascading menu appears to the right of the object name listing all the commands that you can perform on the OLE object. With a video clip object, you can Play, Edit, or Open the object.

When using OLE editing commands, you must remember:

- The commands that appear for each object depend on the object type; therefore, a Word object will not have the same options as a video clip.
- You cannot Undo or Redo actions performed on OLE objects; therefore, use care when editing OLE objects.

OLE objects have some common commands, which include:

- Edit
- Open
- Convert

Edit

The Edit command edits the OLE object in PADS Layout. You can edit the object using all of the source application's commands and tools. When you finish editing, click Close on the File menu. You return to PADS Layout and the OLE object is updated.

Open

The Open command opens the OLE object in the source application. You can then edit the object within the source application. When you finish editing, click Exit on the File menu. You return to PADS Layout and the OLE object is updated.

Convert

The Convert command converts an OLE object into another object. You can also convert the OLE object from appearing as an icon to appearing as the actual object; for example, a Word document instead of the Word icon.

> **Tip**: The object's source application determines the form to which you can convert the object.

Saving OLE Objects

Linked and embedded objects are automatically saved when you save a PADS Layout design. OLE objects are saved in the .pcb file. When you open the .pcb file, the OLE objects also automatically open.

You can, however, save the OLE objects in a separate file, with a .ole extension, by clicking Export on the File menu. For more information, see “Importing and Exporting Files” on page 39. You can open these .ole files in other applications that understand the .ole file format. For example, if you insert a Word document into PADS Layout and then save the Word object, you can later open Word and open the Word documents stored in the PADS Layout .ole file. You can also do this with a DxDesigner object.

OLE and CAM

You can print OLE linked or embedded objects with any Windows-supported printer. You cannot photoplot or plot OLE objects. Also, OLE objects do not appear when you preview prints. For more information, see the Plot Options dialog box topic.

Chapter 16
BGA Operations

This chapter discusses the ball grid array (BGA) advanced packaging toolkit.

If you have the Advanced Packaging Toolkit option, PADS Layout supports advanced packaging technology. In advanced packaging, a bare chip die is mounted directly onto a substrate and wire bonds are used to make the connections between the chip and the substrate paths. Routed traces connect the wire bonds to the package pins on the substrate. The BGA component is then mounted on the printed circuit board.

Table 16-1 shows the BGA wizards you can use to create advanced packaging:

Table 16-1. BGA Wizards

Wizard	Description
Die Wizard	You can create die part definitions parametrically or import the die description using GDSII or formatted ASCII files. The Die Wizard replaces Component IQ by providing die capture directly in the Layout Editor. This eliminates the need to transfer .ciq files.
Wire Bond Wizard	You can create and edit wire bonds using the Wire Bond Wizard. Additional enhancements in the Layout Editor give users more control over substrate bond pad placement and defining an unlimited number of wire bond rings. The Wire Bond Wizard replaces Library IQ, eliminating the need to transfer .liq files.
Die Flag Wizard	You can parametrically construct complex die flags and power rings. Predefined ring and spoke configurations, along with dynamic sizing controls, simplify creation. This satisfies the flag and ring requirements for most designs. To further streamline the design process, soldermask relief is automatically generated for the corresponding flag and rings.

For sequential steps for creating a BGA design, see To Create a BGA Design.

Dynamic Route Editor

The Dynamic Route Editor (DRE) is an autorouter that follows the direction of your pointer as you move it, seeking optimal paths and installing corners as the route progresses. You can use the DRE for routing BGAs.

Because the DRE installs its own corners, you can quickly route a selected connection by pulling the head of the trace through obstacles, with the pointer showing the path you want to use. You can also install corners manually. If sufficient space exists and clearance violations are not created, the DRE also shoves established traces and vias aside to make room for the path you are routing.

You can also reroute segments in DRE. For more information on the DRE see “Dynamic Route Editor” on page 288. For sequential steps for using DRE see Routing Dynamically.

BGA Route Wizard

The BGA Route Wizard provides automated features that reduce repetitive design tasks including creating connections between BGAs, pads, and SBPs; creating BGA fanouts; routing die-to-BGA connecting traces; and generating plating tails.

When you click the BGA Route Wizard button on the BGA Toolbar, the BGA Route Wizard dialog box appears. Use the BGA Route Wizard dialog box to generate connections only, or generate connections and routes.

This topic provides definitions of and visual references for the route segments and fanout patterns created during route processing.

BGA Route Patterns

The BGA Route Wizard generates route patterns on die and BGA layers only. BGA fanouts are generated on the BGA layer. This is performed on double-sided designs only. All other parts of route patterns (serpentine routes, plating tails, SBP fanouts, and any-angle coupling traces) are created on the die layer.

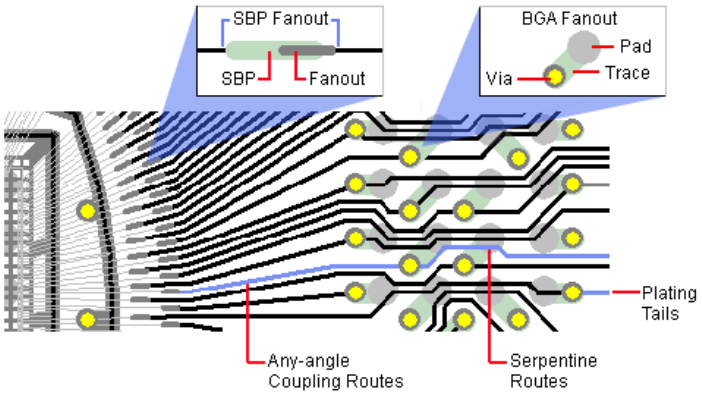
BGA Route Segments

BGA routes are made up of the segments listed in Table 16-2 and shown in Figure 16-1

Table 16-2. BGA Route Segments

Segment	Description
Serpentine Routes	Portions of the BGA route that connect any-angle coupling traces to the BGA array.
Plating Tails	Portions of the BGA route that connect BGA vias to a plating bar (bus bar).
SBP Fanouts	Single-segment fanouts on the SBP which attach to the any-angle coupling traces.
Any-Angle Coupling Traces	Portions of the BGA route that connect SBP fanouts to serpentine routes.
BGA Fanouts	Single-segment fanouts from BGA array pads to BGA vias. BGA array fanout patterns are created for double-sided packages only.

Figure 16-1. BGA Route Segments



BGA Fanout Patterns

Figure 16-2 and Figure 16-3 show the available BGA fanout patterns.

Figure 16-2. Herringbone and Diagonal

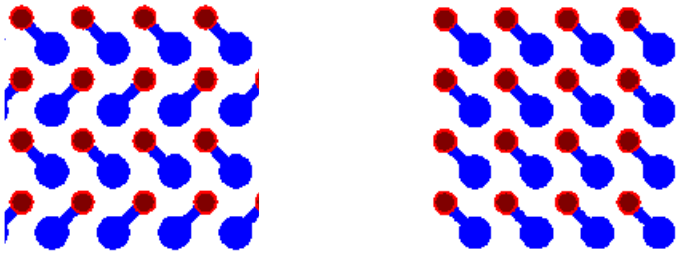
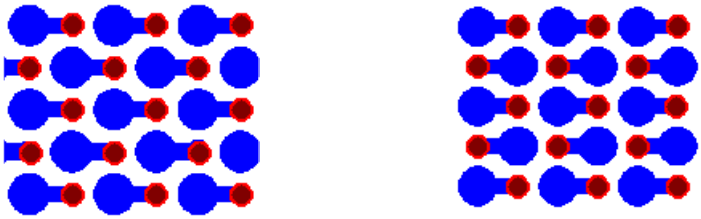


Figure 16-3. Vortex and Double Vortex



Herringbone and diagonal patterns are available for regular (non-staggered) arrays. Vortex and double vortex patterns are available for staggered arrays.

BGA Fanout Direction

The directional controls on the BGA Fanouts tab rotate the BGA fanouts in the directions listed in Table 16-3.

Table 16-3. BGA Fanout Rotation Directions

Direction	Description
Out	Rotates BGA fanouts to the outside of the design.
In	Rotates BGA fanouts toward the inside of the design.
Clockwise	Rotates BGA fanouts in a clockwise direction.
Counterclockwise	Rotates BGA fanouts in a counterclockwise direction.

BGA fanout rotation is based on the position of the BGA with respect to the die (Figure 16-4). For example, if the BGA is above the die, clockwise rotation turns the fanout to the left. If the BGA is below the die, clockwise rotation turns the fanout to the right.

Figure 16-4. BGA Fanout Rotation Directions

If there are more than two rows of BGA pads and you select either the Orthogonal for Outside Row or Orthogonal for Inside Row option, the direction options affect only BGA fanouts within the central rows.

Partitioning a Die

When using Generate Connections, use the Partitioning option on the Connections tab of the BGA Route Wizard dialog box to determine whether Select Sides or Select Quadrants appears on the Select Pads tab.

Side Partitioning

You can use Select Sides (side partitioning) with either Generate Connections or Generate Connections and Route. Side partitioning diagonally divides pads into four sets of pins, based on the die or BGA geometry (Figure 16-5 and Figure 16-6).

Figure 16-5. A Side-Partitioned Die

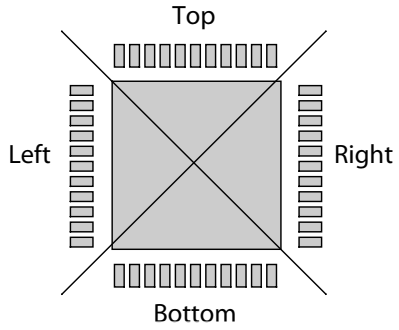
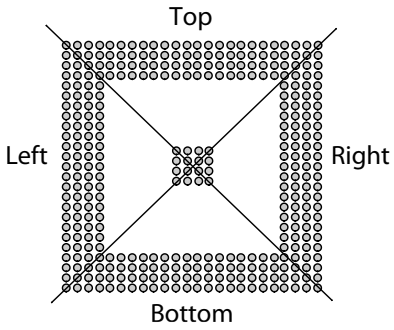


Figure 16-6. A Side-Partitioned BGA Array



Quadrant Partitioning

You can use Select Quadrants (quadrant partitioning) only with Generate Connections. Quadrant partitioning divides pads into four sets of pins along X and Y axes based on the die or BGA geometry (Figure 16-7 and Figure 16-8). This form of partitioning is especially useful for flip chip dies that have quadrant symmetry of functional BGA balls.

Figure 16-7. A Quadrant-Partitioned Die

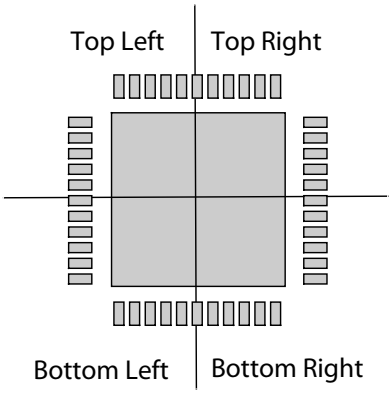
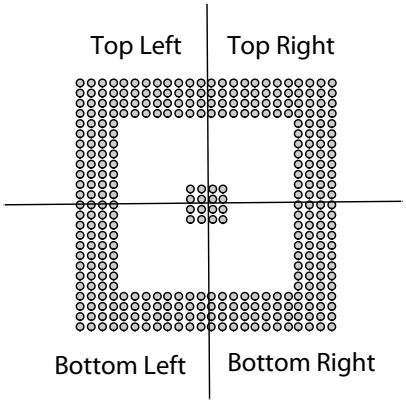


Figure 16-8. A Quadrant-Partitioned BGA Array



Die Wizard

You can create Die Part definitions parametrically or import the die description using GDSII or formatted ASCII files. The Die Wizard provides die capture directly in the Layout Editor.

Creating Die Information

You can design a die to use either wire bonding or flip chip attachment.

Use the Die Wizard to create the die part definitions. You can create the die part definitions parametrically or import the die description using GDSII or formatted ASCII files. You can modify imported information. While you are working with Die Wizard, you are provided with a real-time preview of the die part definitions you are creating in the die display area of the Die Wizard dialog box.

Caution

Beginning with PADS 9.0, die parts and flip chips are no longer identified by their family designations (DIE or FLP), but instead by the Special Purpose settings in the General tab of the Part Information dialog box. When you import an ASCII file created by a previous PADS version, these Special Purpose settings are automatically set for parts having the logic family DIE or FLP. The part's family designation remains the same.

Parametric construction is ideal for scenarios requiring package feasibility studies before the die is complete.

ASCII file data may come from the IC place and route design system, a spreadsheet program such as Microsoft Excel, or a text editor. For more information, see "Die Data ASCII File Format" on page 294.

GDSII files are binary files that instruct where to add items at x-y locations. GDSII Stream format is the standard file format for transferring or archiving 2D graphical design data.

For information on how to create a die, see the following topics:

- To Create a Die from a Text File
- To Create a Die Parametrically
- To Create a Die from a Text File

Die Data ASCII File Format

If you import an ASCII file to create the die part definition, you must implement the following rules:

- The first line of the file should specify the units (Mil, MM, Micron, or Inch).
- The file must be comma-delimited.
- The minimum information required for pad placement is for x and y. Other data can be missing but commas must be present to mark each field. Missing data can be supplied using the Die Wizard.
- The pad data must be in the following format to import correctly into the Die Wizard:

Table 16-4. Example Pad Information

Pad Number	Pad Function	x-Coordinate	y-Coordinate	Pad Length	Pad Width
1	GND	-3.66	3.865	0.07	0.07
2	PWR	-3.54	3.865	0.07	0.07
3	SIG003	-3.42	3.865	0.07	0.07

Tip: If the pad width is not present but the pad length is, the pad is circular.

The following is a sample file based off the data in Table 16-4.

```
MM,,,,,
1,GND,-3.66,3.865,0.07,0.07
2,PWR,-3.54,3.865,0.07,0.07
3,SIG003,-3.42,3.865,0.07,0.07
```

Wire Bond Wizard

Wire Bond Wizard provides automatic wire bond fanout capabilities. The Wire Bond Wizard, along with the Wire Bond Editor, provides all the wire bonding functionality needed to autogenerate wire bond fanouts and edit the wire bond die.

Using the Wire Bond Wizard, you can:

- Define SBP rings and their properties. For each ring, you set the parameters controlling the geometry of the guide, fanout properties, and wire bond placement strategy. For more information, see the Wire Bond Wizard Dialog Box topic.
- Set SBP-to-SBP clearance and Wire Bond rules using the Rules and Wire Bond Rules dialog boxes. For more information, see the Wire Bond Wizard Dialog Box, Rules Dialog Box, and To Define Wire Bond Rules topics.
- Assign CBPs to rings. Once you assign a CBP to a ring, the virtual SBP and wire bond are created for the CBP, and the SBP is automatically placed on the guide of the specified ring. You can also reassign CBPs using the CBPs tab on the Wire Bond Wizard dialog box.
- Generate a wire bond fanout between the component bond pads and the substrate bond pads. Placement of SBPs is part of the fanout generation. For more information, see "Applying the Wire Bond Fanout to the Design" on page 296.

For the sequential steps to create a wire bond fanout, see Wire Bond Fanout Workflow.

Wire Bond Wizard Preview Options

To preview the wire bond fanout, the SBP guides, and the component bond pad assignments (view the preplaced SBPs) use the preview capabilities of the Wire Bond Wizard.

When you use the Guide tab to change the number of and dimensions of the SBP guides, a preview of these changes appears in the work area.

Previewing component bond pad assignments displays the new substrate bond pads and wire bonds created as you use the Assign CBPs to Rings dialog box. The CBP assignment preview adjusts automatically when you make changes to the SBP guides or the shape or size of the substrate bond pads.

To preview the wire bond fanout that corresponds to currently set rules and options, use the Preview Fanout button. If you change any of the settings that affect the fanout pattern, the fanout changes to preview the component bond pad assignments. Use the Preview Fanout button to regenerate the preview of the wire bond fanout after changing the parameters.

You can also view a report about the previewed wire bond fanout.

Setting Rules for Wire Bond Fanout Creation

The rules in Table 16-5 are common for all rings and are essential for wire bond fanout creation:

Table 16-5. Ring Rules

Rule	How to Set
SBP–to–SBP clearance	Set with SMD-to-SMD clearance rules set on the Rules dialog box
WB–to–WB clearance	Set using the Wire Bond Rules dialog box
WB–to–SBP clearance	Set using the Wire Bond Rules dialog box
WB minimal and maximal lengths	Set using the Wire Bond Rules dialog box
WB maximum angle	Set using the Wire Bond Rules dialog box

Although the rules above apply to the entire wire bond fanout for all rings of the whole die component, you can define the following individually:

- The wire bond width for a ring
- Preferred SBP–to–SBP spacing for a ring
- Preferred WB–to–SBP spacing for a ring

Also, if you create the fanout on multiple layers, you can define the SBP–to–SBP clearance individually for each design layer.

Applying the Wire Bond Fanout to the Design

To create the wire bond fanout pattern, click Create Fanout on the Wire Bond Wizard dialog box. The substrate bond pads and wire bonds are created in the design, and the wire bond preference parameters are stored with the die component. You can generate a report as well.

When the generated wire bond fanout has been successfully applied to the design, all selections for the processed CBPs are cleared in the Generate Fanout For area.

DRC Considerations

The pattern generated by the wire bond fanout engine may not conform to all design and wire bond rules.

If Design Rule Checking (DRC) is off, the fanout pattern is saved in the design as it was generated, even if there are rule violations.

If DRC is on, any parts of the design that violate the rules are not created. If violations exist, the fanout pattern is not applied to the design, but can clear DRC and save the fanout in the design.

SBP Guides

An SBP guide defines a virtual snap line along which substrate bond pads are aligned during wire bond fanout generation and wire bond fanout editing.

Each die component can have its own set of SBP guides, unlimited in number. They display as lines 1 mil wide.

SBP Guide Shapes

Figures 16-9 through 16-13 show available SBP guide shapes.

Figure 16-9. Rectangle

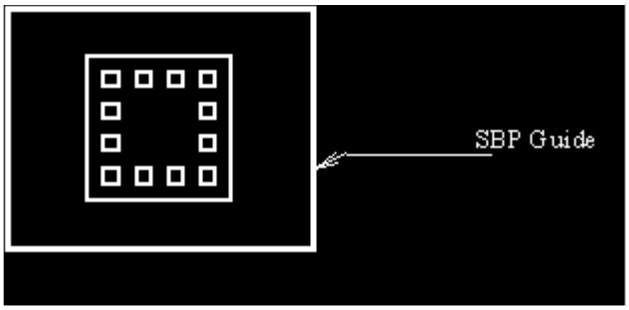


Figure 16-10. Rounded Rectangle

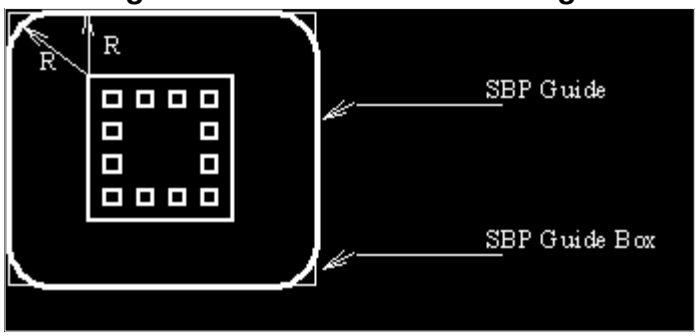


Figure 16-11. Tent

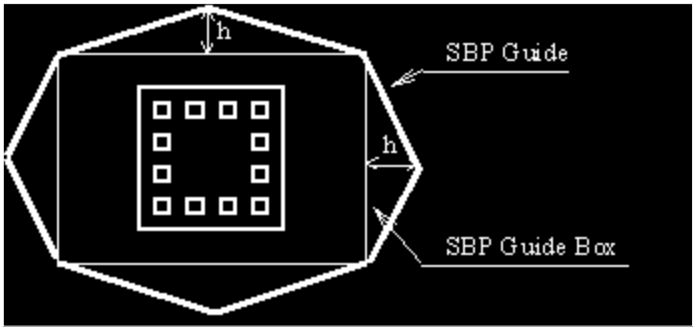


Figure 16-12. Arced

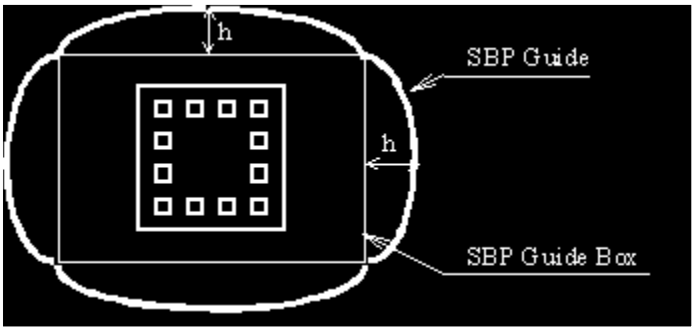
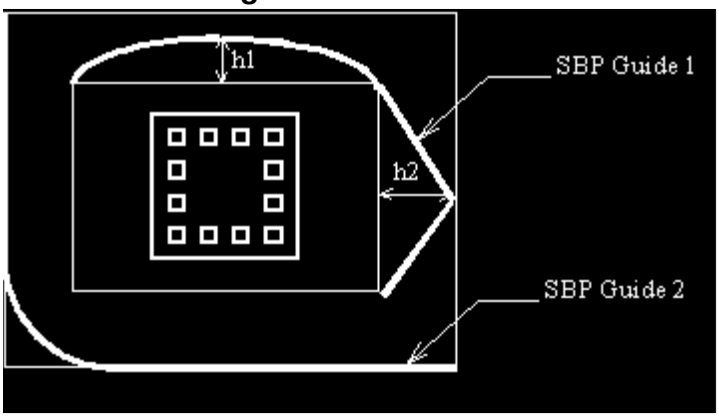


Figure 16-13. Mixed



SBP Guide Shape Definition

Shape definition is based on the SBP guide box and is set on the Wire Bond Wizard Guide tab.

- Define Rectangle using the parameters in Size and Shape.
- Define Rounded Rectangle using the parameters in Size and Shape.
- Define Tent using the parameters in Size, Shape, and Height.
- Define Arced using the parameters in Size, Shape, and Height.
- Define a Mixed shape using the parameters in Size, Shape, Height, By Side, and Shape By Side.

In Mixed shape geometry, the predefined patterns (rectangle, rounded rectangle, and so on) are still the base shape, but the patterns are defined per die side. The only restriction is that all patterns for Mixed shape must be based on the same guide box. You can create the SBP guide with one or more sides missing.

Height for the Arced and Tent shapes can be set independently for each die side.

The corner for Rounded Rectangle becomes rounded when both adjacent sides are defined with the Rounded Rectangle shape.

Placing SBPs

You can set general parameters for SBP placement that are common to all rings in the wire bond fanout.

If you select the Fit to Guide option, SBPs are fitted into SBP guide limits using preferred spacing values as they are defined for rings. If the SBPs do not fit, smaller spacing is attempted. The smallest value possible that is attempted for SBPs on a given side is the minimal spacing value set in the rules.

Fit to Guide mode does not take effect until you do both of the following on the Wire Bond Wizard Strategy tab:

- Enable and define the preferred spacings for SBP to SBP and WB to SBP.
- Clear the Force Preferred Spacing option.

If you select the Allow Gaps option, one of the following applies:

- Gaps are allowed when placing wire bonds in the wire bond fanout;
- Wire Bond Wizard evenly distributes all wire bond fanouts into a compact group.

Wire Bond Rules

Each die part in a design has its own set of wire bond rules. If a rule is not set for the current die, the rule is not checked. Use the Wire Bond Rules dialog box to define rules that apply to wire bonds in the currently open die part.

Checking Wire Bond Rules

You can define wire bond rules for individual die parts, but when you check wire bond rules, all die parts in the design are checked. For more information, see the Verifying the Design or Setting Up Wire Bond Checking topics.

You can also run wire bond rule batch checking at any time. For more information, see the Verifying the Design topic.

When DRC is on, the wire bond rules are checked along with other design rules (pad to pad, pad to trace, and so on). If DRC errors occur, a message appears on the status line.

Tip: You can modify wire bonds whether DRC is on or off. Pin names are checked for uniqueness whether DRC is on or off.

All rules are checked upon completion of the current operation, as they are in the Layout Editor. Additionally, wire bond rules are checked on the fly. During an operation, such as moving a substrate bond pad, any unsatisfied rule results in a violation and an error marker appears.

When checking is complete, a wire bond report appears.

Wire Bond Report

Wire Bond reports provide information on wire bond rules violations and compliance. When you check wire bond rules, the Wire Bond report is saved to a file named wbr_report.lst. The default location for the file is \PADS Projects.

The following is an example of a wire bond report:

```
Wire Bond Rule Checking Report - previewbgadieflag.pcb - Wed Feb 09
14:01:45 2001
Checking Die Part U1 <MWG122160ECG> <MWG122160ECG> ...
  Wire bond Rules
    Min Length: 30
    Max Length: 175
    Max Angle: 45.00
    WB to WB Clearance: 1
    WB to SBP Clearance: 1

  Die Part U1: NO errors found
```

Tip: The value Not Set appears for rules that are not set in the Wire Bond Rules dialog box.

Wire Bond Editor

In the Layout Editor, only substrate bond pads are available for selection and modification. The Wire Bond Editor is available only for die parts designed with the Advanced Packaging Toolkit. You can edit only one die part at a time.

The Wire Bond Editor opens (*explodes*) a selected die part, so you can move, add, delete, and edit individual component bond pads, wire bonds, and substrate bond pads. You can edit the die size by selecting Edit Die Size from the Wire Bond Editor shortcut menu. For more information, see the To Edit the Die Size topic.

Tip: You cannot edit the die outline by selecting it using the Wire Bond Wizard.

Editing Wire Bonds

You can edit wire bonds only in the Wire Bond Editor. For more information, see the Using the Wire Bond Editor topic.

You cannot edit wire bonds in the same way that you edit bond pads. You can alter the distance by which the wire bond is offset from the center of the bond pads. For more information, see the Wire Bond Properties Dialog Box topic. You can also alter the design rules for wire bonds. For more information, see the To Define Wire Bond Rules topic.

Use any of the‘ following methods to select a wire bond to edit:

- With nothing selected, click Select Anything or Select WBs on the shortcut menu and select the wire bond.
- Click Select WBs on the shortcut menu and area select to edit multiple wire bonds.
- With a component bond pad or substrate bond pad selected, click Select WB on the shortcut menu.
- Select the component bond pad or substrate bond pad to which the wire bond is connected. Right-click and click Properties. Click WB.

Die Flag Wizard

The Die Flag Wizard, a feature available from the BGA Toolbar, lets you create a die flag. A die flag is a conductive shape that lies on the surface of the substrate, under the die. The die flag is cutout with patterns that create a nonconductive area.

The die flag serves the following conductive and bonding functions for the die:

- Connecting to the back-based die, typically the ground
- Providing a heat sink and a pathway for heat dissipation
- Mounting the die to the substrate

While you are working with Die Flag Wizard, you are provided with a real-time preview of the shapes you are creating.

For information on how to create a die flag and rings, see the To Create a Die Flag and Rings topic.

Die Flag Parts

A die flag is composed of the following parts (Figure 16-14):

- Die flag ring

  The die flag ring surrounds the center paddle and attaches to it by spokes. This ring typically provides a ground connection. The outer boundary of the die flag ring defines the outer boundary of the entire die flag.

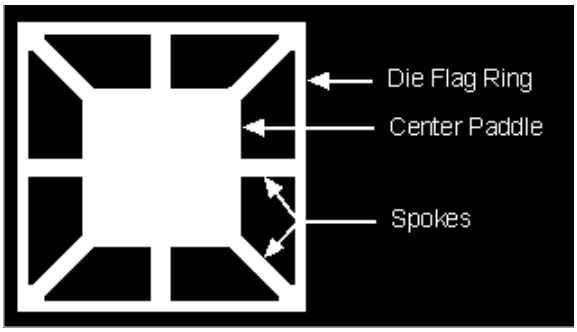
- Center paddle

The center paddle is a square or rectangular filled shape that lies at the center of the die flag. The term coverage describes the ratio of the center paddle area to the area of the entire die flag.

- Spokes

  The spokes connect the center paddle to the die flag ring.

Figure 16-14. Parts of a Die Flag



The two possible spoke configurations you can create are either:

- A corner configuration (Figure 16-15).
- A corner offset configuration (Figure 16-16)

Figure 16-15. Project from Corners

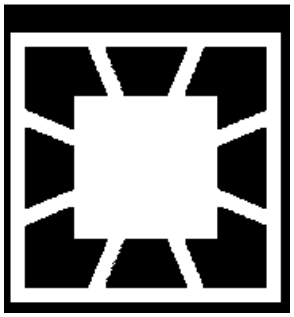
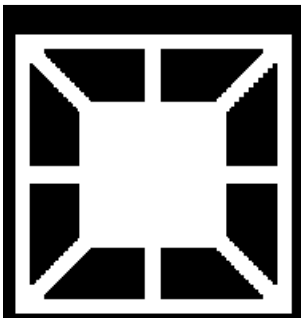


Figure 16-16. Offset from Corners



Rings

In addition to the die flag ring, other rings might surround the die flag. As shown in Figure 16-17 through Figure 16-19, these rings might have different shapes.

The first surrounding ring typically provides a power connection. Other surrounding rings may serve other purposes, such as ground connections and signal rings.

Figure 16-17. Rectangle Ring Shape

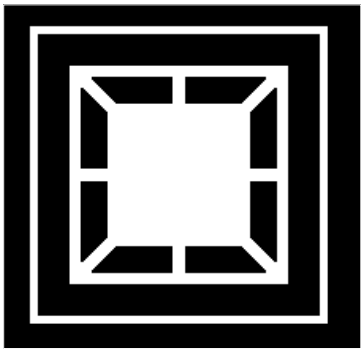


Figure 16-18. Rounded Rectangle Ring Shape

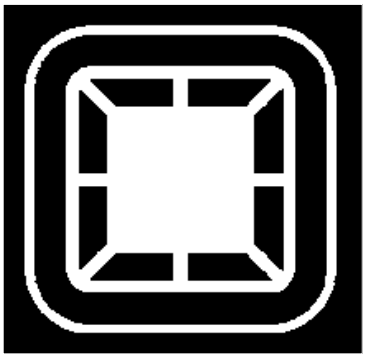
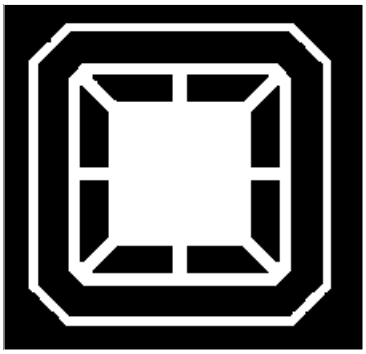


Figure 16-19. Chamfered Rectangle



Chapter 17
File Formats

This chapter discusses the common file formats for importing and exporting data in PADS Layout.

ASCII Format

You cannot always export all data to earlier ASCII formats. Earlier versions do not support several database structures. If your design exceeds any earlier database limits, either export is canceled or the unsupported data is not exported.

Layer-specific reference designator and part type references are converted to generic reference designator and part type settings when converting version 4.0 and higher CAM documents to earlier versions using ASCII Export. In addition, attributes and keepout settings are lost when exporting.

All information about part type and attribute labels is ignored. If a component has more than one reference designator label, one is selected and converted. Visibility and size status for the label is ignored; but the position is maintained.

Because jumpers, although not part of the part list, are considered vias, you can export them to ASCII.

PADS-format ASCII files created with previous versions are compatible with this version. However, importing an ASCII file that was not created with PADS Layout resets any rules or layer assignments that are not part of the ASCII file to the PADS Layout default. For example,

if you set the directions for the layer of an empty database to vertical on top and horizontal on the bottom, the layers revert to horizontal on top and vertical on bottom when you read a netlist and save the file in .pcb format.

To avoid overwriting PADS Layout rules when you read an ASCII file, use the Export command to output the rules as a rules.asc file. Read your ASCII file and then import the rules.asc file.

ASCII Messages

The following are possible ASCII messages:

- Part <Name> has more than 100 gates. The design cannot be exported to the selected format.

  The design has a decal, part type, or a component exceeding the limit for the specified format.

- The design has more than 30 electrical layers and cannot be exported to the selected format.

  You have tried to export an increased layer mode design to a previous version of PADS Layout.

- The selected format does not support more than 30 layers. Objects on layers <layer range> will be deleted.

  You have tried to export an increased layer mode design to a previous version of PADS Layout in the ASCII Output dialog box. PADS Layout versions prior to 4.0 support default layer mode, but not increased layer mode.

  If you press OK, the file is exported to PADS-format ASCII. Layers 1 - *n* are unchanged in the ASCII file; layers in the indicated layer range are deleted in the ASCII file and all objects on these layers are deleted as well. Press Cancel to return to the ASCII Export dialog box.

ASCII Parts and Connections Lists

The Export command generates PADS-format ASCII files that carry all design-related information associated with a .pcb file: positions for routing and vias, parts, copper, rules, GUI setups, and router strategy files.

Each information category appears after a unique control statement and with its own data format. This section covers formats necessary for creating the most commonly used PADS-format ASCII files: parts and connections lists. You can use any text editor to prepare these files.

Units Used in ASCII Parts and Connections Lists

As shown in Table 17-1, you can replace mils with metric or inches.

Table 17-1. Units in ASCII Parts and Connections Lists

Unit	Description
Mils	Data is expressed to the nearest whole milli-inch (THOU). Degrees are expressed to the nearest tenth of a degree. Allowed ranges are from 56000 to +56000, not to exceed a total of 56000. Allowed ranges in degrees are from 0.0 to 359.9 degrees.
Inches	Data is expressed to the nearest thousandth of an inch. Degrees are expressed to the nearest tenth of a degree. Allowed ranges are 56.000 to +56.000, not to exceed a total of 56.000. Allowed ranges in degrees are from 0.0 to 359.9 degrees.
Metric	Data is expressed to the nearest tenth of a millimeter in millimeters. Degrees are expressed to the nearest tenth. Allowed ranges are from 1422.4 to +1422.4, not to exceed a total of 1422.4
Basic	One database unit = 2/3 x 10^9 meters. Allowed ranges are from 213,360,000,000 to +213,360,000,000.

Parts List Section

The following shows the mandatory minimum content of a V4.0 parts list. The parenthetical statements are for information only and not included.

```
!POWERPCB-V4.0-MILS! (Initialization Header)
*Part*          (Part Section Header)
U1 7401         (Ref Des and Part Type)
U2 7401         (Ref Des and Part Type)
etc.
etc.
*End*       (Terminator of the parts list)
```

If you output an ASCII file parts list from a placed board, you can see the reference line summarizing the optional information you can include:

```
*REMARK* REFNM PTYPENM X Y ORI GLUE MIRROR ALT CLSTID CLSTATTR BROTHERID
LABELS
*REMARK* .REUSE. INSTANCE RPART
*REMARK* VISIBLE LX LY LORI LHEIGHT LWIDTH LLEVEL MIRRORED HJUST VJUST
RTREAD
*REMARK*
```

Table 17-2 shows the fields in the reference line.

Table 17-2. Reference Line Fields

Field	Description
REFNM	Unique reference designator name, up to 15 alphanumeric characters.
PTYPENM	Part type name and optional PCB decal information. The part type can consist of 40 characters. The PCB decal will be appended at the end of the part type separated by an at (@) sign. The PCB decal will be appended to the part type in the ASCII file only if a part decal has been modified using the Pad Stacks Properties dialog box. Value and tolerance are not specified in this field.
X and Y	Coordinates of the part placement origin in the design.
ORI	Orientation in degrees of the part in the design. Allowed values range from 0.0 to 359.999, in increments of 0.001 of a degree.
GLUE	Flag indicating whether the symbol is glued or not. Allowed values are G (glue) or U (unglued).
MIRROR	Flag indicating whether the symbol has been mirrored to the opposite side of the board. Allowed values are M (mirrored on bottom layer) or N (not mirrored on top layer).
ALT	Alternate decal number. This is a sequential number and follows the sequence as defined in the part type file; for example, DIP14:DIP14\SO:DIP14\SOL are 0, 1, 2, respectively. Allowed values are from 0 to 15.
LABELS	Number of part labels. Each label is a two-line entry

When the part is a member of a physical design reuse, the following line is added after the part header line:

```
*REMARK* .REUSE. INSTANCE RPART
```

Table 17-3 shows the fields in the part header line.

Table 17-3. Fields in the Physical Design Reuse Part Header Line

Field	Description
.REUSE.	Keyword preceded and followed by periods (.).
INSTANCE	Physical design reuse name.
RPART	Part reference designator within a physical design reuse.

Labels define the locations for reference designator, part type, and attributes.

```
* REMARK * VISIBLE LX LY LORI LHEIGHT LWIDTH LLEVEL MIRRORED HJUST VJUST
RTREAD
```

Table 17-4 shows the fields in the labels

Table 17-4. Fields in the Labels Line

Field	Description
VISIBLE	Label visibility type. Possible values are: VALUE, FULL_NAME, NAME, FULL_BOTH, BOTH, and NONE.
LX and LY	Coordinates of the label origin.
LORI	Relative label orientation. Precision is three digits after the decimal point.
LHEIGHT	Height of the label text.
LWIDTH	Line width of the label text.
LLEVEL	Layer on which the label is located. Possible values are from 0 to 30. 0 means all layers.
MIRRORED	Flag. Possible values are 0 and 1.
HJUST	Horizontal justification. Possible values are LEFT, CENTER, and RIGHT.
VJUST	Vertical justification. Possible values are UP, CENTER, and DOWN.
RTREAD	Right reading status. Possible values are ORTHO (orthogonal), ANGLED, or NONE.

The second line is the attribute name, for example, reference designator or part type.

PADS Layout uses this format to record part position and placement status. You can use it before you load a part list to prelocate and glue connectors or mounting holes or to orient or rotate a part away from the orientation as stored in the parts library.

Tip: The following characters are illegal for part names: carets (^), brackets ([ ]), ampersands (&), asterisks (*), braces ({}), parentheses (), at signs (@), question marks (?), equal signs (=), backward slashes (\), periods, commas, colons, and spaces.

Standard Header

Adding the *Standard* header after the *Part* header automatically creates a power (VCC) and ground (GND) connection netlist, generated from the Signal Pin (SIGPIN) data contained in the part type information for each part in the part list. This eliminates your inputting the power and ground nets. However, if the part type data does not contain the power and ground data, these nodes are not generated automatically.

Notes on Parts List Format

There is a maximum of six columns of mandatory and optional data.

To accept the data in a column, all previous columns must have entries. If no data exists for the previous column, enter zero (0).

For example, if all ICs are glued in the above parts list, G must exist in the GLU column for each IC. In addition, the X, Y, and Orientation columns require zero (0) entry for each IC in the parts list.

Sample Parts List

Here is an example of a short V4.0 parts list where some parts use both the mandatory and optional features and other parts use only the mandatory features:

```
!POWERPCB-V4.0-MILS! DESIGN DATABASE ASCII FILE 1.0
*Part*
P1        CON\26P\ED  15200    8500  0.0   G
U1        74LS00
U2        74LS04
U3        74LS00
U4        74LS00
U{5-20}   74LS04
C1        CL25
C2        CL25
C3        CL25
C{4-20}   CK05
R1        R1/4W
ATTRIBUTE VALUES
{
PART R1
{
Value 1K
Tolerance 5%
}
}
*End*
```

From the above parts list connector P1 uses library part CON\26P\ED, is located at X=15.2" Y=8.5" from the origin, is oriented the same as originally created, and is glued at the indicated position. U1-U20 are the ICs. They are not prelocated, are at their original orientation, and are free to move.

The entries for U{5-20} and C{4-20} demonstrate a shorthand way to enter a long series of part reference designators using the identical part, without using an individual line for each part. The format is as follows:

- The first letter is the reference designator stem or family, for example U.
- The range of part reference designations, for example 5-20, enclosed with brackets.
- The part type preceded by a space.

In the following example, U5 through U20 use a 74LS04 part type.

```
U{5-20} 74LS04
```

The following example uses the family and tag approach for identifying the resistor R1. R1/4W is the family. R1/4W is the part type for the entire family of quarter watt resistors. This part type name calls the part type from the library. Attributes value 1K and Tolerance 5% are associated with the part after it is imported, and they uniquely identify the resistors.

```
R1 R1/4W
```

Notes on Parts List Files

The X and Y distance between the system board origin and the part origin is in 0.001" units (mils). Because it is assumed that the system origin and the board origin are the same, the parts are positioned in the proper location on the board relative to the board origin. If the system origin is not the same as the board origin, the X and Y placement of the components is not on the board.

Always read the parts list before the connection list. When the connection list is read via Import ASCII, it is checked, by reference designator, for the availability of each part in the connection list. If the part reference designator is in the connection list but not in the parts list, a *part not found* message appears. Correct the error to accept that line of the connection list.

Use Compare Netlists after the parts and connections are imported to ensure that all items are correctly transferred to the database.

Parts List Error Messages

As the parts list is read, it is checked for data inconsistencies. An error file is created of any errors. In addition, the line of data is not accepted in the database. The following are possible error messages:

- No *PowerPCB* found at beginning of file.

  Add the correct header.

- Expecting a valid *XXX* command statement.

  Examine the list. Command statements starting with * are either in error or missing.

- No *End* command found at end of file.

  Add the *End* statement.

- Bad *Parts* ASCII Data Line Format U11.1

  Examine the line containing U11.1 for errors.

Connection List

The following format is mandatory for a connection list:

```
!POWERPCB-V4.0-MILS! DESIGN DATABASE ASCII FILE 1.0
*Net*
*Signal*    Name 1
Node 1      Node 2      Node 3
*Signal*    Name 2
Node 1      Node 2      Node 3
 "
 "
 "
*Signal*    Name 3
Node 1      Node 2      Node 3
*End*
```

Table 17-5 lists the fields in the connection list format.

Table 17-5. Connection List Format Fields

Field	Description
Net	Section command for the connection list.
Signal	Heading for each net. Starts the line containing the name of each individual net.
Name	Name of the net. This is required for each net. The maximum netname length is 47 characters. You can use any alphanumeric characters except brackets ({ }), asterisks (*), and spaces.
Node	Part reference designator and pad/pin number.

Notes on Connection List

A To-From connection represents the electrical connection between two or more physical nodes (part pin/pads).

A node consists of two items of data.

Use a space to separate two nodes. Within a node, use a period to separate the reference designator and the pad number.

Reference Designator Format

The reference designator entry names a part. You can use up to fifteen alphanumeric characters; however, the first character must be alphabetical.

Signal Statements and To-From Signal Names

A *Signal* statement and name are used to assign a signal name for a To-From tree and to identify electrically different To-From trees. A *Signal* statement must exist between each successive To-From tree/net. If a *Signal* statement is not present, the two trees are tied.

The signal name is from the logic diagram, such as SIGA, CLK, +5V, and GND. If the signal is unnamed, use the *Signal* statement to identify the connection and supply the name. The maximum signal name length is 47 characters. You can use any alphanumeric characters except space, tab, comma (,), period (.), braces ({}), asterisks (*), and ampersands (&).

For example:

```
*Signal* D0
U1.3  R1.1  U2.1  A 3-node To-From connection tree named D0.
```

Discrete Component Pad Numbering

Discrete components do not normally have pin numbers on the logic diagram. They must, however, have a pin number in the database.

Tying Nets Together

In logic diagram documentation, parts of a multiple-node connection tree can be depicted anywhere, on all sheets, and at various locations on a sheet. In preparing your To-From list, you do not need to enter the entire tree as one continuous entry. If you use a common signal name for each partial connection, you can enter the different parts of a common connection at different locations of the To-From list. PADS Layout will automatically tie the partial nets together.

To tie all entries in the To-From list give them the same signal name. For example:

```
*Signal*    D0
U1.3        R2.1  U1.6
*Signal*    D0
U5.2        R6.1
*Signal*    D0
U5.8
```

results in:

```
*Signal* D0
U1.3 R2.1 U1.6 U5.2 R6.1 U5.8
```

Use the common signal name technique (above) to enter a long multiple node connection tree that requires several lines of data in the To-From list. Use a signal name that is common on all lines of data.

Abbreviated Notations

Certain type connection trees that have a common pin, such as power trees, ground trees, and memory trees, can use an abbreviated or shorthand format for entering the tree. This abbreviated format uses the format X{F-L}.P and is interpreted as shown in Table 17-6.

Table 17-6. Abbreviated Notations

Shorthand	Description
X	Reference designator letter
F	First reference designator number of a continuous series of reference designations
L	Last reference designator number in the series
{ }	Abbreviated notation symbol
P	Common pin/pad number

A dash (-) separates F and L, a period (.) precedes P. For example, the following results in a common tree for GND, which is read U1 through U50, pin 7. PADS Layout expands the abbreviated format into the full GND connection tree:

```
*Signal* GND50
U{1-50}.7
```

Width

The Width entry is in thousandths of an inch. The first entry of a net establishes the width of all succeeding nets until the width changes. Therefore, arrange the Connection lists by width.

Sample Connection List

```
!PADS-POWERPCB-V4.0-MILS! DESIGN DATABASE ASCII FILE 1.0
*Net*
*Signal*    D0
P1.1 U1.1 U2.3 U4.5 U6.1 U7.2
*Signal     D1
U1.2 U2.4 U3.3
*Signal*    D2
U1.6 U2.5 U3.4
*Signal*    GND
U1.7 U2.7 U3.7 U4.7 P1.7
U8.7 U9.7 U10.7
*Signal*    PWR
U1.14 U2.14 U3.14 U4.14 P1.14
U8.14 U9.14 U10.14
*End*
```

The automatically generated power and ground connection list, generated from the SIGPIN data contained in the part type information will not contain power and ground nodes not identified in the part type data. Examples of this are power and ground pins of an I/O connector, the grounding of spare gates of an IC, and so on. The designer must enter this data in the power and ground net.

Connection List Error Messages

As the connection list ASCII file is read and merged with the parts list, it is checked for inconsistencies. If errors are found, the line of data is not accepted. The following are possible error messages:

- Part name not found P1

  The following line of connection data contains a node using P1; however, the parts list did not include P1.

- Bad *Signal data line

  The line does not contain a signal name.

- Mixing Nets: Node 1 Node 2 Node 3 Node 3 Node 4 Node 5

  Two different nets have a common node (node 3).

- Bad *Connections ASCII Data Line Format U20.A U20.5

The line of data is not in the required format. Alphanumeric pin/pad identifications are not allowed unless they are predefined in the part type file.

DXF Format

The DXF format handles database objects differently. DXF does not support physical design reuses. DXF also does not support route protection information; therefore, when you import or export a DXF file, all route protection information is removed. DXF does, however, store slotted hole information for both import and export. Because jumpers, although not part of the part list, are considered vias, you can export them to DXF.

PADS Layout does not import DXF files from PowerPCB 1.3 and earlier. Versions before PowerPCB 3.0 do not read the DXF files created by PowerPCB 4.0.

If you import a DXF file that contains an ellipse, or a scaled arc with a radius that is too big for the database, a dialog box appears asking you to enter an approximation error. The approximation error determines how the ellipse/arc will be broken into line segments.

For information on creating objects that properly translate to copper polygons and to avoid creating self-intersecting polygons, see "Defining Copper Objects in AutoCAD" on page 317.

DXF Messages

The following are possible DXF messages:

- Warning: Found a self-intersecting copper piece while importing hatched solid. File line: XXXX. Continue process?

  The DXF file contains a self-intersecting polygon. File line: XXXX refers to the line where the error occurs. Correct the polygons in the AutoCAD application. For information on creating objects that properly translate to copper polygons and avoiding creating self-intersecting polygons, see "Defining Copper Objects in AutoCAD" on page 317.

- Warning: Board cutout containing point <x,y> intersects existing board cutout containing <x,y>. New cutout not imported from DXF file.

  The Overlapping board cutouts are reported as warnings when importing a DXF file. Click OK to continue importing or click Cancel to cancel.

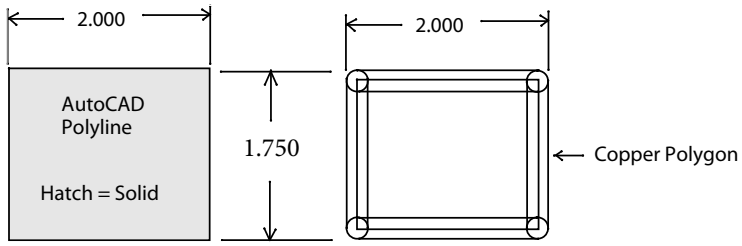
- Warning: Board cutout containing point <x,y> intersects the board outline. Cutout not imported from DXF file.

  The cutouts that overlap the board outline are reported as warnings. Click OK to continue importing or click Cancel to cancel.

Defining Copper Objects in AutoCAD

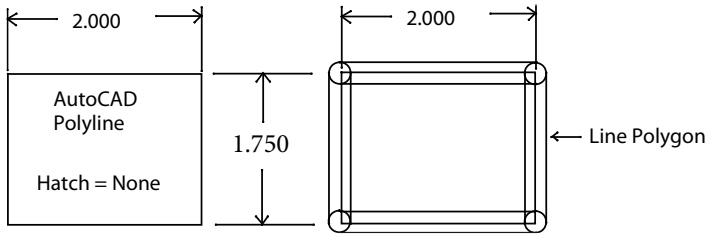
To define an object in AutoCAD that properly translates to a copper or copper pour polygon, create a closed polyline that does not self-intersect and assign a solid hatch type. If you create a polygon in this way, it converts to a copper polygon in PADS Layout (Figure 17-1). In addition, the copper polygon is created in PADS Layout so that the outside edges of the polygon are aligned with the edges of the shape in AutoCAD, ensuring the entire shape is dimensioned and checked correctly. The following graphic demonstrates this conversion.

Figure 17-1. Copper Object with Closed Polygon (Solid Hatch Type)



If you create a polygon without hatching in AutoCAD, the polygon converts to a 2D line in PADS Layout (Figure 17-2). In this case the centerlines of the 2D line are aligned with the edges of the shape in AutoCAD. The following graphic demonstrates this conversion.

Figure 17-2. Copper Object with Closed Polygon (No Hatching)



DXF Export of Filled Polygons

Filled copper polygons export as polylines with a solid hatch.

Legacy Layer Mapping

The following is a cross-reference of the DXF layer names used in the various releases of PADS products. We don’t recommend you try to bring an old DXF file into a new release of PADS Layout.

Table 17-7. PADS Legacy Layer Mapping

PADS Perform	PowerPCB v1.1 - v1.3	PowerPCB v1.5 - current
TXTnn	TEXTnn	TEXT_nn

Table 17-7. PADS Legacy Layer Mapping

PADS Perform	PowerPCB v1.1 - v1.3	PowerPCB v1.5 - current
LINnn	2D_LINEnn	2D_LINE_nn
COPnn	COPPERnn	COPPER_nn
CCOnn	COPPER_CUTOUTnn	COPPER_CUTOUT_nn
CPRnn	POUR_UNFLOOD	POUR_VOID_nn
CPOnn	POUR_FLOOD	POUR_HEADER_nn
HOLnn	HATCH_OUTLINEnn	POUR_OULINE_nn
HVOnn	HATCH_VOIDnn	POUR_VOID_nn
THVnn	VIA_THERMALnn	POUR_VIATHERM_nn
THRnn	PAD_THERMALnn	POUR_PADTHERM_nn
BRD00	BOARD_OUTLINE	BOARD_OUTLINE_00
CONnn	CONNECTIONnn	LINK_00
PADXnn	PADS_INNER	PADS_INNER
PADXBnn	PADS_BOTTOMnn	PADS_BOT
PADSTnn	PADS_TOPnn	PADS_TOP
PNM00	PART_NAME_TOP and PART_NAME_BOTTOM	PART_NAME_TOP and PART_NAME_BOT
CMB00	PART_TOP	PART_TYPE_TOP
COM01	PART_BOTTOM	PART_TYPE_BOT
BDYnn	BODYnn	
DRLTPnn	DRIL_PLTE_THRUnn	DRIL_PLTE_THRU_nn
DRLTN	DRIL_NO_PLTE_THRUnn	DRIL_NPLTE_THRU_nn
DRLNP	DRIL_PLTE_PRTLnn-nn	DRIL_PLTE_PRTL_nn
DRLNN	DRIL_NPLTE_PRTLnn-nn	DRIL_NPLTE_PRTL_nn
TRKnn	TRACKSnn	TRACE_nn
VIAi	VIAi	VIA
PKGnn	PARTINFO	PART_INFO
PCB30	PADS_PARAMETERS	PCB_PARAMS
KEY00	DRILL_SYMBOL	DRIL_SYMBOL
CTBnn	COMPONENT_TEXTnn	PART_TOP_TEXT_nn and PART_BOT_TEXT_nn

Intermediate Data Format

If you have the IDF Interface option, you can import and export Intermediate Data Format (IDF) files. IDF is a format that allows data exchange between PADS Layout and a mechanical design system, such as Pro/ENGINEER. PADS Layout supports IDF 2.0 and IDF 3.0.

The Intermediate Data Format specification file IDF30.pdf is located at C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Documentation\<*language*>\Layout

Tip: PADS Layout can exchange IDF files with mechanical design systems that support IDF. While the following sections refer to Pro/ENGINEER, the information in those sections may also apply to other mechanical design systems.

Exporting IDF Files

You can, at any stage of your design, export the design data to IDF format, which Pro/ENGINEER can import to edit and analyze the mechanical data. When exporting IDF files, you create two files:

- A .emn board file
- A .emp library file

When exporting to IDF, you can export the following information about parts: part height, drilled holes, and part outlines. In addition, settings in another file, a .map file, allow Pro/ENGINEER to model parts from PADS Layout in 3D.

Exporting 3D Data

When you import IDF files into Pro/ENGINEER, it uses the 2D data exported from PADS Layout to create the appropriate part outline. The exported part outline and the height data are used to construct a protrusion that is the 3D component outline.

If you want to use geometrically accurate 3D models instead, you can replace simplified PADS Layout models with Pro/ENGINEER models using a Pro/ENGINEER library and the ecad_hint.map file. To replace PADS Layout models, you must create a Pro/ENGINEER 3D part library containing all your parts and then set up the ecad_hint.map file to map the 3D replacements.

The ecad_hint.map file is a text file that you create, edit, and maintain. This file replaces approximated parts from PADS Layout with geometrically accurate components, previously modeled in Pro/ENGINEER. This file must exist in either the current working directory or in the Pro/ENGINEER software loadpoint/text directory.

For more information on the ecad_hint.map file and using Pro/ENGINEER libraries, see the *Pro/ENGINEER Interface Guide* or the Pro/ECAD page of PTC's Web site at http://www.ptc.com.

Part Height Information

To export part heights to IDF, assign the Geometry.Height attribute at the Decal or Part Type hierarchy level. The IDF library file format requires height information for each part type and decal pair in the design. You should add height information to decals because of their geometric nature.

If the height of a part is less than the minimum height specified on the IDF Export dialog box, then the part is not exported.

During export, the file is searched for part height information in the following order:

1. Search for Geometry.Height attribute on part type or decal
2. Search for the text string ZHEIGHT=height in the decal
3. Search for the text string $… in the decal on layer 30 (in default layer mode) or 130 (in increased layer mode)

If the search does not find part heights, the Missing Height dialog box appears where you can enter part height information. See the Specifying Missing Heights During Export to IDF topic.

Exporting Holes

To model holes in Pro/ENGINEER, export from PADS Layout a single pin component with a nonzero drill diameter. For Pro/ENGINEER to recognize a part as a drilled hole, one of the following conditions must be true:

- Its part type has a HOLE attribute (for PowerPCB 3.0 and later)
- Its decal has a HOLE attribute
- Its decal contains the text string HOLE

For more information, see the To Add Drill Hole Information to IDF Files topic.

These specially-marked components are exported into the Drilled Holes section of the IDF file, and the mechanical design system interprets these as actual holes in the board. Drill oversize information is not included for plated holes, and oversize information does not apply to nonplated holes.

Tip: If you use the same decal for holes and nonholes, but use different part types, set the HOLE attribute in the part type.

Part Outline Information

In PADS Layout, a part outline is a closed line element, with perhaps some arcs drawn along the way, starting with a polygon or a completed polygon with segments stretched into arcs. Disjointed elements are not accepted.

The IDF library file format requires that PADS Layout exports outline information for each part type and decal pair in the design. Add this information to the PADS Layout library decal using the PCB Decal Editor; otherwise, the outline is approximated by a bounding rectangle, capturing the extents in a box.

Use the Shape Layer list to select the decal layer where you entered outline information. You must have only one, single, closed outline on the selected layer. Additional segments or more than one closed outline are not recognized as an outline; geometry is approximated by a bounding rectangle.

PADS Layout searches for the outline in the following order:

1. Selected layer
2. Layer 0
3. Layer 1
4. Bounding rectangle. If PADS Layout cannot find a valid outline on any of the above layers, PADS Layout use a bounding rectangle. The bounding rectangle is the smallest rectangle that encloses all nontext graphics on all layers.

Tip: Even though layer 0 is treated specially in PADS Layout because graphics on layer 0 appear on all layers, this does not affect how PADS Layout finds an outline for IDF export. For example, if you have outlines on layer 0 and layer 5 and you click layer 5 in the Shape Layer list, the layer 5 outline is exported.

Part outlines are not imported to PADS Layout.

IDF Export Messages

The following are possible IDF Export messages:

- Wrong Layer Specified for Cluster Keepout.

  Inner was specified as the layer for a cluster keepout. IDF does not support the Inner attribute for layers. Keepouts assigned to Inner in PADS Layout are not exported to IDF.

- Can't write the component outline for <reference designator>. The outline is self-intersecting. Continue with export?

  The outline for this component is self-intersecting. Pro/ENGINEER does not support outlines with this geometry. An outline is not passed for this part, and as a result, Pro/ENGINEER cannot import the IDF file properly.

- A part <part type> has a HOLE attribute, but has more than 1 pin. Continue with export?

  A part type was found with the attribute HOLE. Because this part type has more than one pin, the part can’t be passed to Pro/ENGINEER as a drilled hole. Click OK to ignore the warning and save the part type as a normal electrical part. Click Cancel to stop exporting and then change the part type to one pin in PADS Layout. Export can save the one pin part type to the IDF file as a drilled hole.

- A decal <decal> containing the text string HOLE has more than 1 pin. Continue with export?

  A decal was found with the text string HOLE. Because this decal has more than one pin, the decal can’t be passed to Pro/ENGINEER as a drilled hole. Click OK to ignore this warning and save the decal as a normal electrical part. Click Cancel to stop exporting and then change the decal to one pin in PADS Layout. Export can save the one pin decal to the IDF file as a drilled hole.

- A decal <decal> containing the text string HOLE has drill diameter = 0. Continue with export?

  A decal was found with the text string HOLE. Because this decal has a drill diameter of 0, the decal can’t be passed to Pro/ENGINEER as a drilled hole. Click OK to ignore this warning and save the decal as a normal electrical part. Click Cancel to stop exporting and then change the decal to a drill diameter greater than 0. Export can save the revised decal as a drilled hole.

- Via keepout containing point (x,y) on layer <layer> is not written to IDF file. IDF supports only through via keepouts.

  IDF recognizes only via keepouts that are defined for all layers. Click OK to continue exporting. Via keepouts that are not defined as through via keepouts are ignored.

- Trace keepout containing point (x,y) on layer <layer> is not written to IDF file. IDF does not support trace keepouts for individual inner layers.

A trace keepout was found for an individual inner layer. IDF supports trace keepouts only on TOP, BOTTOM, BOTH, INNER, and ALL. Click OK to continue exporting. Trace keepouts defined for individual inner layers are not exported.

- Warning: Keepouts of the following types are found in design, and are not written to IDF file. IDF does not support these keepout types. See the report file for a list of the keepouts found.

  Keepouts of a type that IDF does not support were found in the design. IDF supports keepouts for placement, routing, and vias only. Click OK to continue exporting. Keepouts that are not supported are not exported.

IDF Export Status File

When you export to IDF, PADS Layout creates a status log file named idfexport.sts in \PADS Projects. This file contains a log of the locations of height and outline information used during exporting for each part type and decal combination, as well as any name changes made to avoid illegal characters in Pro/ENGINEER.

For example, for part type CAP\CK05 using decal CAP\MR05, you might see the following four lines in the idfexport.sts file:

```
Warning: Decal illegal character, CAP\MR05 is now CAP_MR05
Warning: Part Type illegal character, CAP\CK05 is now CAP_CK05
Note: Part type CAP\CK05 using decal CAP\MR05: Layer 26 height 300 used.
Note: Part type CAP\CK05 using decal CAP\MR05: Shape found on layer 0.
```

Height Messages

The following are possible height messages:

- Note: Part type MTHOLE6 using decal MTHOLE6: Attribute height 1 2 used.

  The height was found in the value of a ZHEIGHT attribute found on the part type.

- Note: Part type MTHOLE6 using decal MTHOLE6: Dialog height 250.0 used.

  You entered a height in the Missing Height dialog box.

- Note: Part type MTHOLE6 using decal MTHOLE6: Layer 26 height 100 used.

  A ZHEIGHT text string was found on layer 26 of the decal.

- Warning: Part type MTHOLE6 using decal MTHOLE6: Dialog height 0.0 reused.

  You entered a height for a previous part type and decal pair in the Missing Height dialog box and selected the For All Parts check box.

- Warning: Part type MTHOLE6 using decal MTHOLE6: Dxf height $35.1 used.

  The DXF convention was used.

- Warning: Part type <part type> using decal <decal>: Not exported, has height <xxx> less than minimum <xxx> <units>.

  A part was not exported because it is less than the minimum height set in the IDF Export dialog box.

Outline Messages

The following are possible outline messages:

- Note: Part type TO-92 using decal TO-92: Shape found on layer 1.

  A valid outline was found on layer 1 and layer 1 was selected in the Shape Layer list in the IDF Export dialog box.

- Warning: Part type TO-92 using decal TO-92: Pick box shape used.

  A valid outline was not found on the layer selected in the Shape Layer list in the IDF Export dialog box.

Keepout Messages in the IDF Export Status File

- Warning: <Test Point | Copper Pour and Plane Area | Component Drill | Decal > keepout containing point <x,y> found and not written to IDF file.

  Keepouts were found in the design that IDF does not support. They are not written to the IDF file.

Importing IDF Files

When you import IDF files created in Pro/ENGINEER, PADS Layout imports the board outline and part placement that you changed in Pro/ENGINEER. PADS Layout also imports drilled holes, allowing a mechanical engineer to send hole placement to the layout designer. You do not need library parts for these holes in your PADS Layout library, they will be created as needed. For more information see “Importing Holes” on page 325.

You can also import components, allowing a mechanical engineer to send some part placement to the layout designer. These parts must have corresponding part types and decals in the PADS Layout library. Route data remains unchanged, except for unroutes which move with new pin locations. For more information see “Adding Components During Importing” on page 326.

If you import an IDF file that contains a scaled arc with a radius that is too big for the database, a dialog box appears asking for an arc approximation error. The arc approximation error determines how the arc will be broken into line segments. For more information, see the Drafting Properties Dialog Box topic.

Because importing an IDF file can move and delete parts and keepouts, the translator checks to see whether route and reuse protection is violated, and reports any violations. For more information, see "IDF Import Status File" on page 328.

Importing Holes

You can import drilled holes in the IDF file. Part type and decal names are created for all holes added to the design in the form DH<drill diameter><units><plated status>, where Table 17-8 shows the field definitions.

Table 17-8. Field Definitions for Imported Holes

Field	Description
DH	Drill hole notation
<drill diameter>	Size of the diameter, less than or equal to 11 characters, including the decimal point
<units>	Units: MM (millimeters) or ML (Mils)
<plated status>	Plating status: P (plated) or N (nonplated)

The entire drill hole part type and decal name must be less than or equal to 16 characters. For example, a 123.456 mil plated hole has a part type and decal name of DH123_456MLP, where DH = Drill Hole, 123_456 = drill diameter, ML = units, and P = plated status.

Parts are added to the design in the following order:

1. If the part type or decal with the assigned name already exists in the design, the existing part is used to represent the hole.
2. If the part type or decal with the assigned name does not exist, the library is searched.
3. If a match is found in the library, the part in the library is used.
4. If a match is not found in the library, a new part type and decal are created in the design using the logic family UND.

The new decal has a HOLE attribute (if importing IDF 3.0 format) or the text string HOLE (if importing IDF 2.0 format), one pin, and the proper drill diameter. These components receive the reference designators assigned to the logic family to which they belong. Holes are returned to PADS Layout with ECO registration turned off for the part and attribute.

Holes that existed in the same locations prior to the import keep their original reference designators. Holes that do not exist in the imported IDF file are deleted.

Adding Components During Importing

You can add components to your design when importing an IDF file. This lets mechanical designers place mechanical components, such as connectors or components with restrictions on placement (for instance, a processor with a large heat sink). Components are added to the design as follows:

1. If the correct part type and decal already exist in the design, the existing component is used.
2. If the part type or decal does not exist, the library is searched.
3. If a match is found in the library, the component in the library is used.
4. If a match is not found in the library, an error message appears.

Added components use the reference designator specified in the IDF file.

You usually need to edit the IDF board file when you add components in Pro/ENGINEER to resolve two incompatibilities that exist between Pro/ENGINEER and PADS Layout.

Part Name Incompatibility

When exporting the IDF board file, Pro/ENGINEER exports only the decal name. PADS Layout, however, exports both the part type and the decal. While Pro/ENGINEER imports this information correctly, PADS Layout can't import the information from Pro/ENGINEER. For example, PADS Layout exports the following:

<decal name>	<part name>	<refdes>
DIP14	7400	U1

But Pro/ENGINEER exports:

DIP14	DIP14	U1

Typically in PADS Layout, decal and part names are different. To import the IDF board file into PADS Layout, you must restore the part name field by editing the .emn file:

DIP14	7400	U1

Illegal Character Incompatibility

Many characters that are legal in PADS Layout names are not legal in Pro/ENGINEER. When PADS Layout exports part or decal names to the IDF files, an underscore (_) is substituted for the following characters, which are not legal in Pro/ENGINEER:

Forward slash (/), backward slash (\), period (.), exclamation point (!), at sign (@), pound sign (#), dollar sign ($), percent sign (%), circumflex (^), ampersand (&), asterisk (*), parentheses ( ), plus sign (+), equal sign (=), pipe (|), comma (,), question mark (?), colon (:), semicolon (;), apostrophe ('), and quotation mark (")

For example, PADS Layout writes the following to the IDF board (.emn) file for part type +5VREG with decal TO-220-UP:

```
<decal name>    <part name>     <refdes>
TO-220-UP_      5VREG           U5
```

PADS Layout, however, will not find this part in the library when it tries to add it because of the name change. You must edit the IDF board (.emn) file to match the following to correctly add the part.

```
TO-220-UP    +5REG     U5
```

IDF Import Messages

The following are possible IDF import messages:

- Component not found.

  The IDF data does not match the PADS Layout data for part type, decal name, or reference designator. The data is ignored. Choose whether to continue importing.

- Fatal error - Not a valid coordinate.

  The coordinates of one of the parts are outside of the work area. Return to Pro/ENGINEER and correct the part placement. Then export the IDF file from Pro/ENGINEER.

- Fatal error - Self-intersecting polygon.

  The polygon that defines either a cluster keepout or the board outline intersects itself. Return to Pro/ENGINEER and correct the self-intersecting polygon. Then export the IDF file from Pro/ENGINEER.

- Mechanical components found and ignored.

  PADS Layout found mechanical components in the IDF file which are not imported into PADS Layout.

- This is an IDF library file. You can only import IDF board files.

To ensure the integrity of your PADS Layout library you cannot import .emp library files. Import the .emn board file instead.

- This is not an IDF File.

  You are trying to import a file that is not in IDF. You can only import files with an .emn file extension using the IDF option.

- Warning: Board cutout containing point <x,y> intersects existing board cutout containing point <x,y>. Original board outline restored.

  Click OK to continue importing and restore the original board outline.

- Warning: Board cutout containing point <x,y> intersects the board outline. Original board outline restored.

  Click OK to continue importing and restore the original board outline.

- Warning: Keepouts are imported. Existing keepouts are found of the following types, and will be deleted.

  When you import an IDF file into PADS Layout, keepouts (placement, routing, or via) on the design are deleted and replaced with the same keepouts from the imported IDF file.

IDF Import Status File

As you import IDF data, PADS Layout performs several checks to ensure the validity of the data and that current PADS Layout database limits are not exceeded. PADS Layout creates an import status file named idfimport.sts in \PADS Projects. This file contains a log of the imported hole and component locations, including drill diameter, plated status, part type, and decal. This file also identifies whether the board outline was modified.

Since importing an IDF file can move and delete parts and keepouts, the translator checks to see whether route and reuse protection is violated, and reports the violations.

When the import status file is created, the message *IDF import finished with xx errors and xx warnings. Show report file?* appears.

Drilled Hole Messages

The following are possible drilled hole messages:

- Warning: (xxx,yyy) <drill diameter> mil <plated/nonplated> hole changed to <drill diameter> mil <plated | nonplated> hole.

  A hole exists at this location in both the design and the imported IDF file. Either the drill size or the plating status of the hole was changed. Drilled hole size and plating are not usually changed in Pro/ENGINEER. Ignore the warning if this was your intention.

- Warning: (xxx,yyy) <drill diameter> mil <plated/nonplated> hole added.

  A drilled hole exists at this location only in the imported IDF file, no corresponding hole exists in the design. This hole was added to the design. Drilled holes are not usually added in Pro/ENGINEER. Ignore the warning if this was your intention.

- Warning: (xxx,yyy) <drill diameter> mil <plated/nonplated> hole deleted.

  A hole exists at this location only in the design, no corresponding hole exists in the imported IDF file. This hole was deleted from the design. Drilled holes are not usually deleted in Pro/ENGINEER. Ignore the warning if this was your intention.

- Note: (xxx,yyy) <drill diameter> mil <plated/nonplated> hole unchanged.

  A hole exists at this location in both the design and the imported IDF file. The drill size and plating status of the hole are unchanged.

Component Messages

The following are possible component messages:

- Error: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> xxx Imported part type is <parttype>.

  A component with this reference designator exists in both the design and the imported IDF file, and the part type is different. You cannot change the part type with IDF import. Correct the design or the IDF file.

- Error: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> xxx Imported decal is <decal>.

  A component with this reference designator exists in both the design and the imported IDF file, and the decal is different. You cannot change the decal with IDF import. Correct the design or the IDF file.

- Error: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> xxx Attempted to add this component, but part type <part type> is not found in design or library.

PADS Layout could not find the part type from the IDF file in either the design or the PADS Layout library. The component cannot be added. Correct the IDF import file or add the part type to the library.

- Error: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> xxx Attempted to add this component, but decal <decal> is not found in design or library.

  PADS Layout could not find the decal from the IDF file in either the design or the PADS Layout library. The component cannot be added. Correct the IDF import file or add the decal to the library.

- Warning: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> <rotation> Added by IDF import.

  A component with this reference designator exists in the IDF file, but not in the design. The component was added to the design. Components are not usually added in Pro/ENGINEER. If this was your intention, ignore the warning.

- Warning: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> <rotation> Missing.

  A component with this reference designator exists in the design, but not in the IDF file. No change is made to the design. The component is not deleted. Components are not usually deleted in Pro/ENGINEER. If this was your intention, ignore the warning.

- Note: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> <rotation> Moved to (xxx,yyy) on layer <Top/Bottom> with rotation <rotation>.

  A component with this reference designator exists in both the design and the IDF file. The location, side, or rotation of the component is different. The component in the design was updated with the information from the IDF file.

- Note: <Ref Des> <parttype> <decal> (xxx,yyy) <Top/Bottom> <rotation> Unchanged.

  A component with this reference designator exists in both the design and the imported IDF file. The location, side, and rotation of the component are all the same.

Keepout Messages

The following are possible keepout messages:

- Warning: <Placement><Trace and copper><Via and jumper> keepout containing point <x,y> deleted from design.

  Keepouts in PADS Layout have been deleted and replaced with keepouts of the same type.

Cutout Messages

The following are possible cutout messages:

- Warning: Board cutout containing point <x,y> intersects existing board cutout containing point <x,y>. Original board outline restored.

  Cutouts in Pro/ENGINEER have been deleted and replaced with the original PADS Layout cutouts.

- Warning: Board cutout containing point <x,y> intersects the board outline. Original board outline restored.

  The board outline in Pro/ENGINEER was replaced with the original PADS Layout board outline.

Protected Trace Messages

The following are possible protected trace messages:

- Warning: Component <name> is attached to protected traces. Protection is ignored, and component is moved.

  The imported IDF file specifies moving a component regardless of it having protected traces attached to it.

- Warning: Component <name> which represents a drilled hole has a signal attached, so it can't be deleted.

  The imported IDF file specifies deleting a component. Importing cannot delete the component. Delete the component under ECO in PADS Layout.

Physical Design Reuse Messages

The following are possible physical design reuse messages:

- Warning: Component <name> is member of reuse <reuse name>. Reuse is dissolved, and component is moved.

  The imported IDF file specifies moving a component regardless of whether the component is part of a physical design reuse.

- Warning: Hole <name> is member of reuse <reuse Name>. Reuse is dissolved, and hole is deleted.

  The imported IDF file specifies deleting a component regardless of whether the component is part of a physical design reuse. This only occurs for a part representing a drilled hole; no other parts are deleted by IDF import.

- Warning: Keepout containing point (x,y) on layer <layer> is a member of reuse <reuse name>. Reuse is dissolved, and keepout is dissolved.

The imported IDF file specifies deleting keepouts of certain types regardless of whether they are part of a physical design reuse.

IDF File Format

This topic describes the IDF format and board file (.emn) details. All sections apply to both the IDF 2.0 and IDF 3.0 formats, unless otherwise stated. The IDF 3.0 specification, IDF30.pdf, is available at:
C:\MentorGraphics\<*latest_release*>PADS\SDD_HOME\Documentation\<*language*>\Layout

Panel Files

IDF 3.0 supports an optional, panel file. PADS Layout and Pro/ENGINEER do not support the panel file.

Board Outline

The IDF board outline consists of a single board outline. All objects are closed polygons made of arcs and lines.

The board outline describes the X and Y dimensions of the board. The Z dimension refers to the thickness of the board, not of 2D lines. To determine the thickness of the board, add the thickness of all substrate layers. View the thickness of layers by clicking the substrate layer in the Layer List on the Layer Thickness dialog box, and then viewing the Thickness text box in the Substrate/Prepreg area.

PADS Layout supports cutouts in boards. Board cutouts are imported from and exported to IDF. The board outline and all dimensions are imported from and exported to IDF.

Board outlines and keepouts can be owned by Pro/ENGINEER, PADS Layout, or neither. IDF 3.0 ignores this information on import and always sets it to UNOWNED on export.

Other Outline

This section defines outlines that specify heatsinks or board cores. PADS Layout does not support other outlines; this information is not imported from or exported to IDF.

Routing Outline

This section defines outlines that designate areas for routing. PADS Layout does not support routing outlines; this information is not imported from or exported to IDF.

Placement Outline

Each placement outline designates an area for placing components, in other words, a keepin. PADS Layout does not support placement outlines; this information is not imported from or exported to IDF.

Units

PADS Layout imports and exports design units to and from IDF. Units are converted to IDF units as shown in Table 17-9 and Table 17-10.

Table 17-9. Export to IDF Unit Conversion

PADS Layout Unit	Conversion Multiplier	IDF Unit
Mils	1.0 THOU/mil	THOU
Inches	1.0e3 THOU/inch	THOU
Metric (mm)	1.0 MM/mm	MM

Table 17-10. Import from IDF Unit Conversion

IDF Unit	Conversion Multiplier	PADS Layout Unit
THOU	1.0 mils/THOU	Mils
TNM (Ten nanometers) 10e-8 meters	10e-5 mm/TNM	Metric (mm)
MM	1.0 mm/MM	Metric (mm)

When importing THOU units, set the Design Units option in the Global tab of the Options dialog box to Mils. When importing TNM or MM units, set the Design Units option in the Global Options dialog box to Metric.

Routing Keepout

This section defines polygon route keepouts. PADS Layout supports routing keepouts; this information is imported from and exported to IDF.

Via Keepout

This section defines polygon via keepouts. Board level via keepouts specify regions on the board where you cannot place vias, but can route. Through vias are supported; blind and buried vias are not supported.

PADS Layout supports via keepouts; this information is imported from and exported to IDF.

Placement Keepout

In IDF, board-level placement keepouts specify regions on the board where you cannot place components. A keepout can apply to all components or to only those components above or below a height you specify. Placement keepouts can exist on the top layer, bottom layer, or both layers. Each keepout consists of a simple closed curve made of arcs and lines along with minimum and maximum height restrictions.

PADS Layout supports importing and exporting placement keepouts. Restrictions on exporting and importing placement keepouts include:

- You cannot export multiple, overlapping keepouts.
- You cannot export keepouts with the Inner setting.
- PADS Layout converts keepouts that have the Both setting (in Pro/ENGINEER) to All during importing.
- PADS Layout ignores the minimum and maximum height settings during importing.

Placement Group Area

This section defines polygon areas intended for placement of a group of similar components. PADS Layout does not support placement group areas; this information is not imported from or exported to IDF.

Drilled Holes

Drilled holes are specific mounting or tooling holes distinguishable from circular cutouts in the board outline because they are always drilled, instead of punched or routed, and you can plate them.

You can model holes using a single pin component with a nonzero drill diameter and either a HOLE text string on the decal or a HOLE attribute on the part type. These specially marked components are exported into the Drilled Holes Section of the IDF file, and Pro/ENGINEER sees these as actual holes in the board. Drill oversize information is not included for plated holes, and oversize information does not apply to nonplated holes.

Component Placement

This section specifies the locations of all components on the board. A location consists of an X,Y coordinate relative to the board origin, a rotation about the component origin, and a side of the board. Components are either placed or unplaced. You can fix the location of placed components to prevent changes in Pro/ENGINEER. If a component is glued in PADS Layout, it is considered fixed. If a component is not glued in PADS Layout, it is considered placed. If an

IDF part has the Unplaced attribute, PADS Layout considers the part unglued and places it at 0,0.

All component information is exported to IDF. Only partial component information is imported from IDF. Refer to the following table for information on IDF attributes and their PADS Layout equivalents.

PADS Layout does not import changes to part type, reference designator, or decal name. The data is ignored.

IDF 3.0 includes a field in the placement section that represents the mounting offset for a component. PADS Layout does not use the offset field. The height passed for the component should include any mounting offset as shown in Table 17-11.

Table 17-11. Component Placement

IDF Property	PADS Layout Equivalent	PADS Layout Export	Modify	PADS Layout Import
Package Name	Decal Name	Yes	No	No
Part Number	Part Type	Yes	No	No
Reference Designator	Ref Des	Yes	No	Check
X location	X location	Yes	Yes	Update
Y location	Y location	Yes	Yes	Update
Rotation Angle	Rotation Angle	Yes	Yes	Update
Side (Top or Bottom)	Side (Top or Bottom)	Yes	Yes	Update
Fixed	Glued	Yes	Yes	Update
Unplaced	No equivalent	No	Yes	Convert
Placed	Any unglued component	Yes	Yes	Update

Library File Details

PADS Layout exports the IDF library file (.emp), which describes the geometric information for each part type and decal combination on the board. You cannot import into PADS Layout any changes you make to a library file in Pro/ENGINEER.

Refer to Table 17-12 for information about IDF terminology describing elements of the library file and their PADS Layout equivalents.

Table 17-12. IDF Terminology for Library File Elements

IDF Terminology	PADS Layout Equivalent
Geometry Name	Decal Name
Part Number	Part Type
Units MM THOU TNM	Units Metric THOU No equivalent. IDF 3.0 does not support units of TNM (ten nanometers).
Component Height	The attribute Geometry.Height or the Decal text string ZHEIGHT=height is used if present. Otherwise, the value of Part Type attribute ZHeight is used. If there is still no height, the Missing Height dialog box appears.
Mechanical Component	No equivalent, mechanical components are modeled with single pin electrical components.
Component Outline	Level 20 Outline. If the outline is not present, the bounding box is used.

Part Outlines

Part outlines and keepouts can be owned by Pro/ENGINEER, PADS Layout, or neither. IDF 3.0 ignores this information during importing and always sets it to UNOWNED during exporting. Part outlines are never imported to PADS Layout.

IDF 3.0 Enhancements

The following information describes the enhancements to the IDF 3.0 format related to PADS Layout importing and exporting.

- IDF 3.0 recognizes 360-degree circular cutouts.
- Multiple-pin parts and via drills are supported.
- A field representing the mounting offset for a component was added to the placement section. PADS Layout does not use the offset field. The height passed for the component should include any mounting offset.
- A new optional panel file was added. PADS Layout and Pro/ENGINEER do not support panels.

- Board and part outlines and keepouts can be owned by Pro/ENGINEER, PADS Layout, or neither. IDF 3.0 ignores this information on import and always sets it to UNOWNED on export. Part outlines are never imported to PADS Layout.
- Units of TNM (ten nanometers) are not supported.
- Since importing an IDF file can move and delete parts and keepouts, violations to route and reuse protection are checked and reported.
- Routing keepouts and via keepouts are supported.

The IPC-D-356 Netlist

The IPC-D-356 netlist format describes the netlist in terms of X, Y locations which is the perfect format required for testing bare boards. Prior to the IPC-D-356 netlist, there was only the ASCII file output which describes connectivity between pins by their reference designators - machines need coordinates. And CAM outputs only consisted of artwork files. Connectivity could be discovered based on the layer stackup of the artwork, but this could never be checked against the original design.

The IPC-D-356 format is an ASCII format in machine readable form which contains the X,Y locations of points of connectivity of nets, ensuring that the resulting printed circuit board is not only manufactured correctly, but is correct to the original design.

The netlist includes net names, reference designators, and pin numbers. This is not required to test connectivity between coordinates, but it produces results which are easier to understand.

You can also use the updated IPC-D-356A format, which is an updated revision to the D-356 netlist format. The revision A format has more features and is an improvement on the original D-356 format.

Index

— B —

— C —

— D —

— E —

— F —

— N —

— O —

— P —

— Q —

— R —

— S —

— T —

— U —

— V —

— W —

— X —

— Z —

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