



Preparing the Layout

**Product Version 23.1
September 2023**

© 2024 Cadence Design Systems, Inc.
Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

1		6
Preparing the Layout Overview		6
2		9
The Layout Cross Section		9
Planning the Cross Section Editor		9
Working with Cross-section Layers		13
Default Cross-section Values		13
Editing Cross Section Materials		14
The Default Materials File		14
The Local Materials File		14
The Materials Editor		14
Related Topics		15
APD: Importing the Layer Stackup of the Substrate		16
APD: Die-Stack Editor		16
Top and Side Views of the Design		17
Spacers		18
Interposers		21
Die-Stack Editor Report		25
3		26
Working with Graphic Design Elements		26
Adding Elements to a Design		27
Lines		27
Rectangles		27
Filled Rectangles		27
Circles		28
Arcs		29
Chamfers		31
Fillet		31
Shapes		32
Editing Elements in a Design		33
Copying Elements in a Design		34
Copying and Pasting Elements in a Design		35

Moving Elements	41
Changing Element Characteristics	48
Moving Elements to Other Classes	49
	50
Changing Line Fonts of Elements	50
Deleting Graphic Elements from a Design	52
Creating Mirror Images of Graphic Elements with the Standard Mirror Option	52
Editing Vertices	57
Editing Shapes	57
Editing Properties	58
Composing Shapes	58
Decomposing Shapes	59
Cutting and Pasting Design Elements	61
Keepin and Keepout Areas	63
Route Keepins	63
Route Keepouts	64
Wire Keepouts	64
Via Keepouts	64
PCB Editor: Shape Keepouts	65
PCB Editor: Package Keepins	65
PCB Editor: Package Keepouts	66
APD: Component Keepin	67
PCB Editor: Probe Keepouts	67
Gloss Keepouts	67
Artwork Keepins	67
Setting Height Restrictions in Package Keepout Areas	68
	69
Layout Padstacks, Vias, and Etch/Conductor Shapes	69
Editing Layout Padstacks with the Padstack Designer	70
Editing Layout Pad Shapes	71
Creating Vias	73
Via Array Advantages	77
Working with ETCH/CONDUCTOR Shapes	78
Dynamic vs. Static Shapes	78
Working with Dynamic Fill Mode	79
Crosshatched Shapes	81

Unfilled Shapes	82
Frozen Dynamic Shapes	82
Related Topics	84
Freezing and Un-Freezing Shapes	85
Updating Frozen Shapes	86
Creating Shapes Using Shape Operations	87
Logical OR	88
Logical AND	89
Logical ANDNOT	90
XOR	91
Setting the Shape Parameters	93
Using ETCH/CONDUCTOR Shapes in Embedded Planes	107
Creating an Embedded Plane	107
Thermal Relief and Antipad Representation	107
Thermal Relief and Antipads on a Negative Plane Layer	110
Negative Plane Islands	111
Negative Plane Slivers	113
Pad Drawing Mechanism	114
ETCH/CONDUCTOR Shapes' Effect on Routing	116
5	118
Metal Usage Report	118
6	122
Thieving	122
Related Topics	123
7	124
SiP and APD: Degassing	124

Preparing the Layout Overview

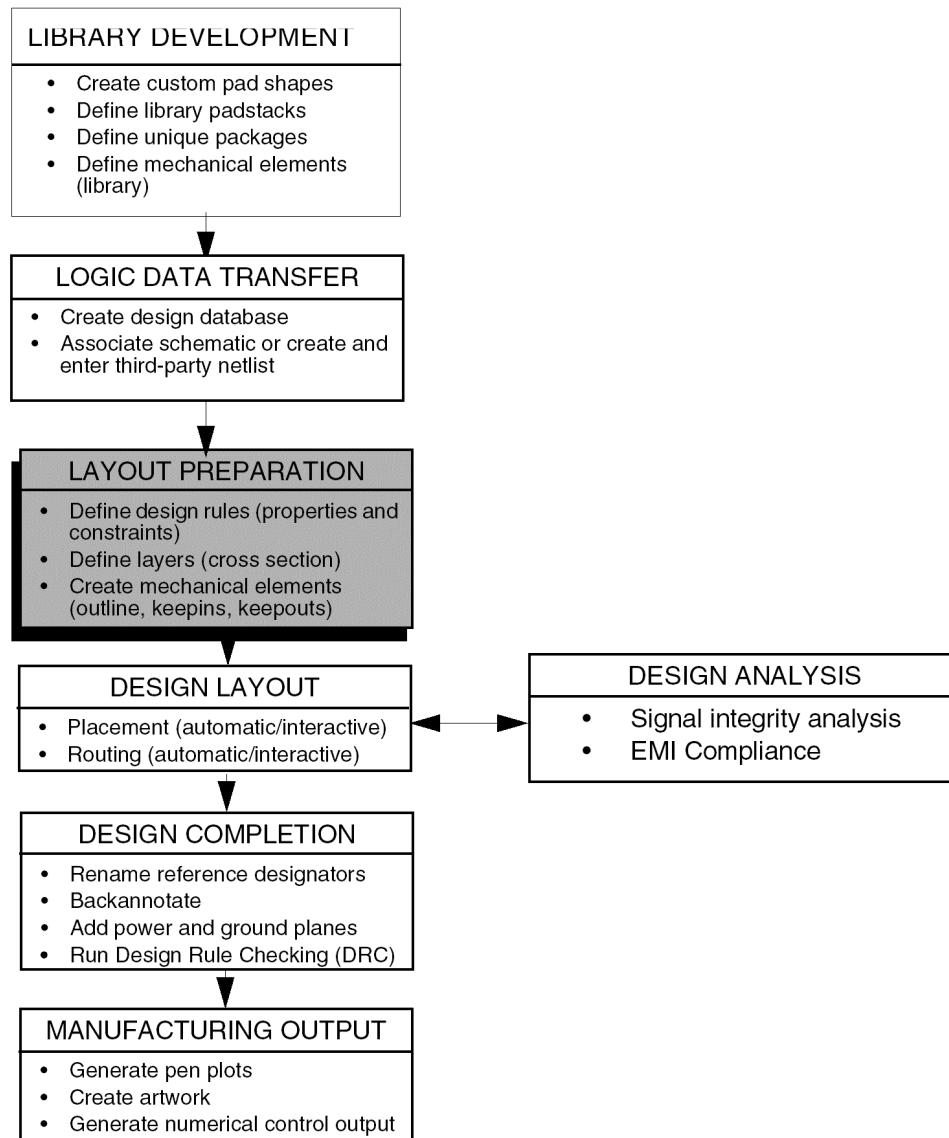
This section introduces the functions available to you during preparation of your layout:

- Defining the layout cross-section
- Adding graphic elements
- Editing layout padstacks and pad shapes
- Creating interactive blind and buried vias
- Creating and editing etch/conductor shapes
- Prepare the layout before placing components on your design.

 Many features are common to all three layout editors: Allegro X PCB Editor, Allegro Package Designer, and System-in-Package tools. When a feature is not common to all editors, it is noted in the heading. If an illustration shows only one of the editors, it is also noted.

The following figure shows layout preparation in the overall design flow.

PCB Editor: Layout Preparation in a Design Flow



For information on the APD flow, see the *Allegro Getting Started User Guide*. For information on the Cadence SiP flows, see the *System-in-Package Flow Guide*.

The Layout Cross Section

Before placement and routing, you normally define layers and their various characteristics in setting up a layout. During placement and routing, you may need to insert extra routing layers because the design is too dense to complete. You may also need to delete layers because of an ECO.

The layout cross section consists of the ordered layers of the layout, including the information about their type, thickness, electrical behavior, and shielding. You also specify whether to photoplot positively or negatively when you set up a cross section.

Cross-section layers to which you assign a name become subclasses of the ETCH/CONDUCTOR class. The layout tool performs DRC on all objects added to all ETCH/CONDUCTOR subclasses.

ETCH/CONDUCTOR layers have a subclass name field to the right of the material name. This shows the name of the ETCH/CONDUCTOR subclass that describes the routing of the layer. Each time the dialog box is opened, the ETCH/CONDUCTOR subclasses are scanned. If necessary, ETCH/CONDUCTOR layers are automatically added to the Cross Section dialog box (for example, when the dialog box is first opened and each time ETCH/CONDUCTOR subclasses are added to the drawing). There is always one ETCH/CONDUCTOR layer for every ETCH/CONDUCTOR subclass.

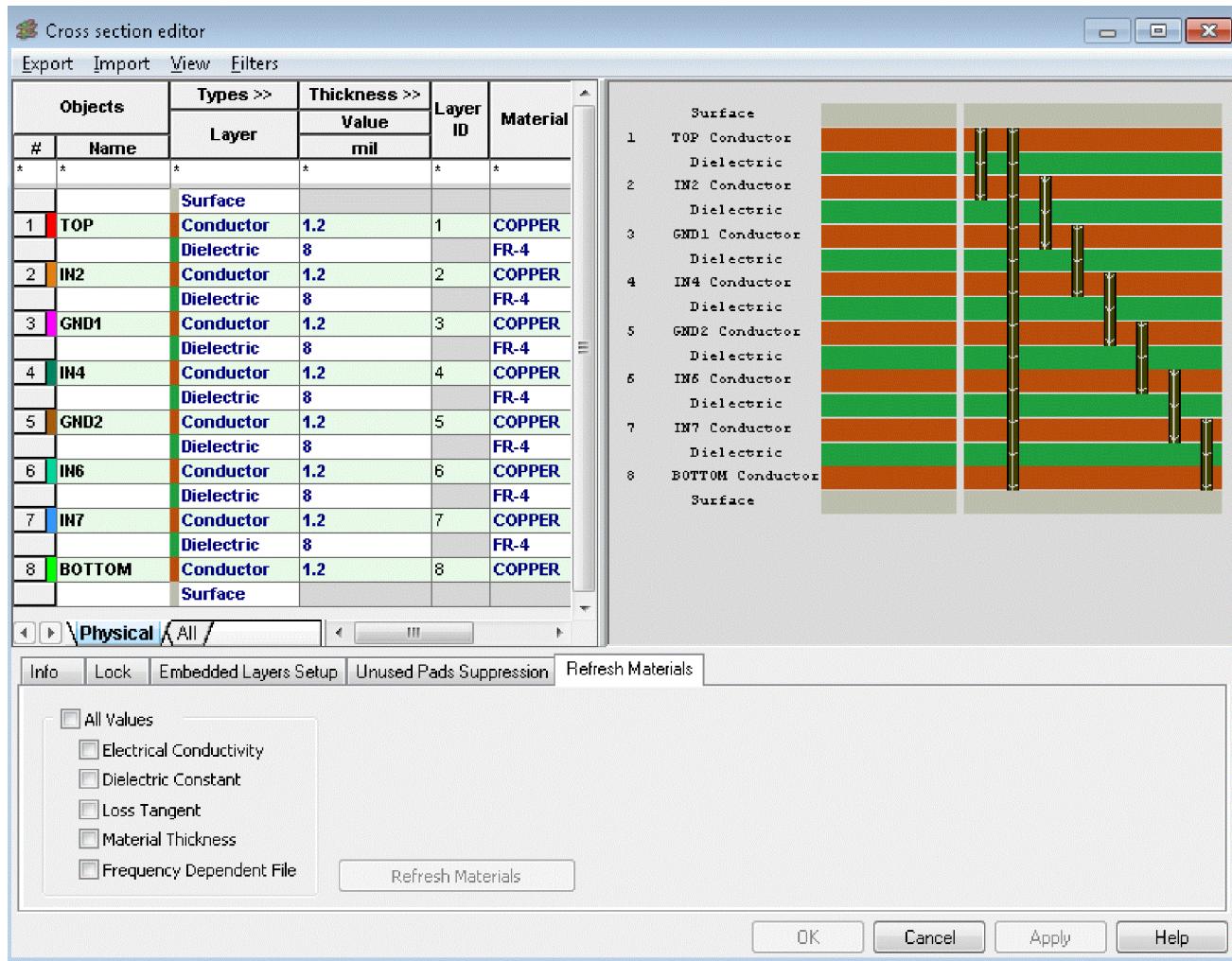
Adjacent ETCH/CONDUCTOR layers are always separated by one dielectric layer. These are added automatically as ETCH/CONDUCTOR layers are added. The layers above the top and below the base ETCH/CONDUCTOR layers are surface layers. The surface layers need not be solid materials. In fact, the default material for these layers is air. The presence of a conformal coating or thermal heat sink in these layers has an effect on both impedance and thermal calculations, so these layers are just as important as the others.

 Adding an ETCH/CONDUCTOR subclass (a layer) also adds the same subclass to the pins, vias, and DRC classes.

Planning the Cross Section Editor

The Cross Section Editor dialog box lets you insert, delete, and define characteristics of the layers of a layout. This dialog box displays one line for each layer of the layout cross section. You enter all this information in the Cross Section Editor dialog box when you choose *Setup – Cross-Section* (`xsection` command).

Cross Section Editor Dialog Box



The lines are in the physical order of the layers, from TOP to BOTTOM as they exist in the layout:
 For each layer, you can define the following:

- Layer name, if it is to be an ETCH/CONDUCTOR subclass (for example, TOP or VCC)
- Type of layer:
 - Conductor
 - Dielectric
 - Diestack
 - Plane

- Layer material

The default choices are:

- Air
- Conformal Coat
- Aluminum Heat Sink
- Tetrafunctional
- Polyimide
- BT Epoxy
- Cyanate Ester E
- Cyanate Ester S
- PTFE
- Polyimide Film
- Copper
- G-type or FR-type dielectric

- Photoplot Film Type

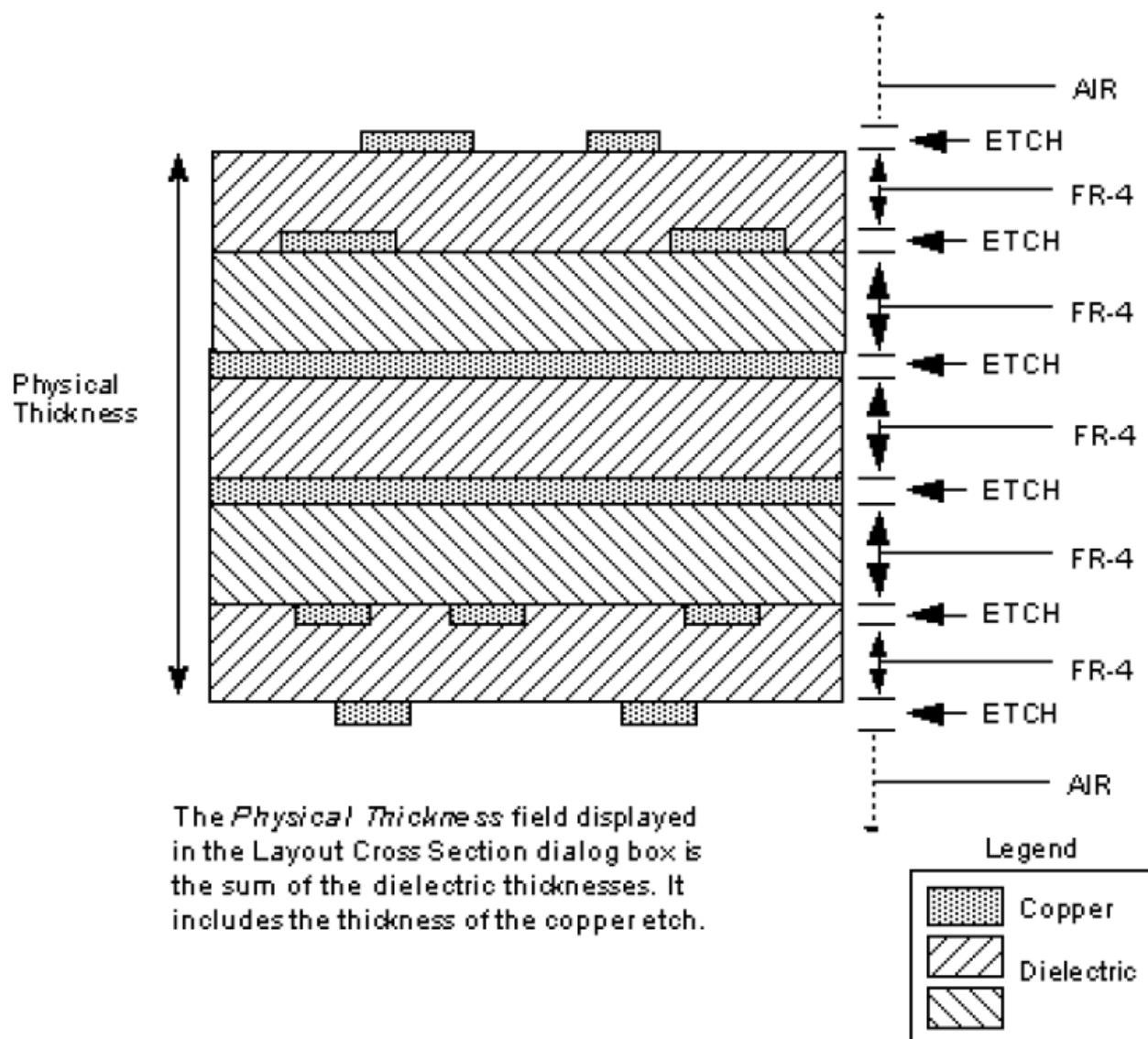
- Positive
- Negative

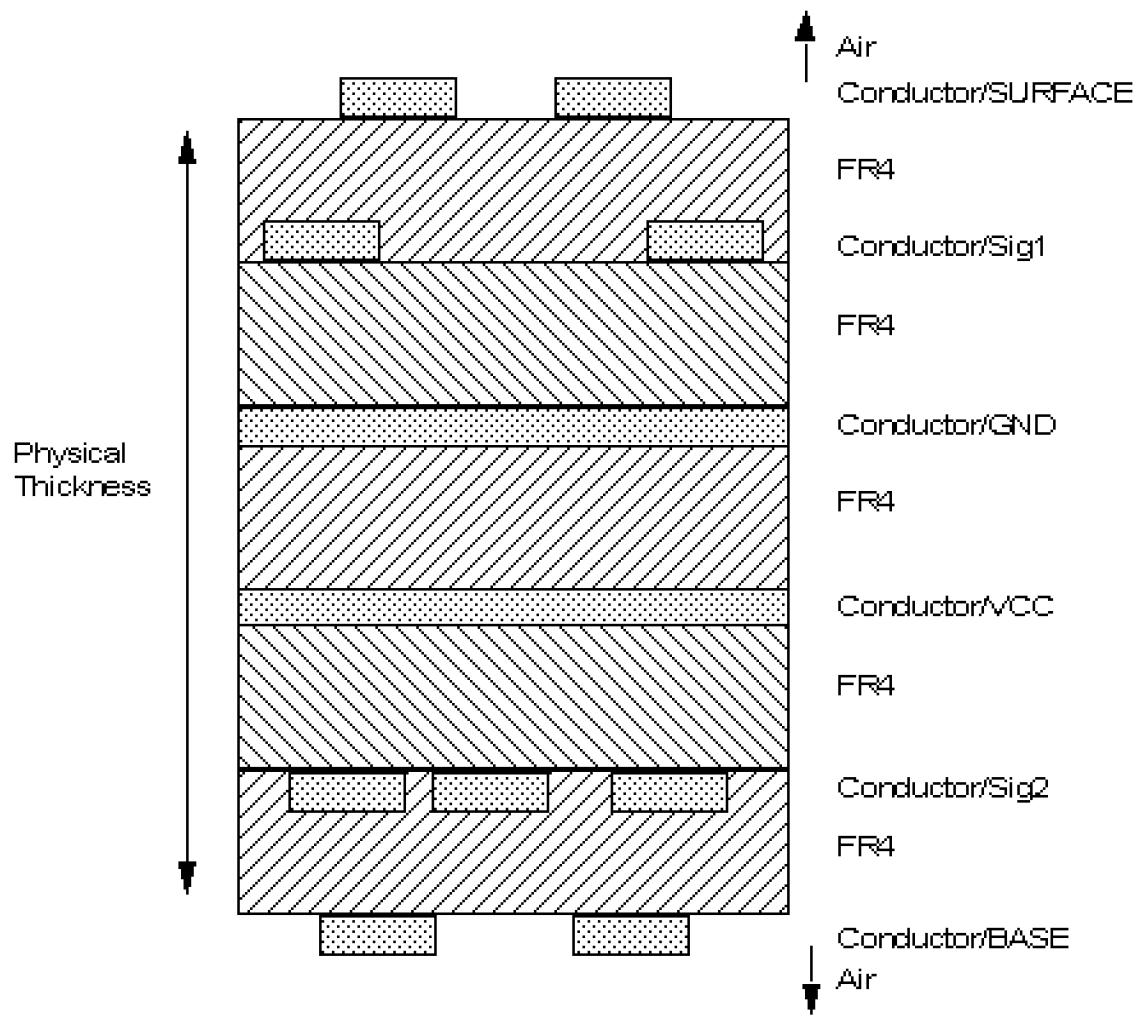
Once you define a layer, you can specify the following:

- Thickness
- Electrical Conductivity
- Whether or not it is a shield layer

You can predefine any number of layer material types in a text file called `materials.dat`, then add layers of that type to the cross section. A later part of this chapter describes the grammar of `materials.dat` file.

The following illustrations show sample cross sections of a layout.





Working with Cross-section Layers

Default Cross-section Values

The default settings for the Cross Section Editor dialog box are stored and maintained in the `materials.dat` file. This file is found using the `$MATERIALPATH` environment variable. It is accessed when the dialog box is opened after new subclasses have been added, and when you change materials for a layer. A message appears when you choose a material name not found in the file.

External conductor layer dielectric values of 1 are allowed and correct. Inner conductor dielectric values of 1 may impact Cross Section calculations and performance, processing of Electrical DRC checks, and extract (including Valor output).

Use the Materials Editor dialog box to view the materials.dat file.

Editing Cross Section Materials

The Default Materials File

The layout tool provides a default Materials file that contains typical industry fabrication materials such as COPPER and FR4. This file is read-only and its location is specified in the search path defined by the environment variable \$MATERIALPATH.

The Local Materials File

Using the Materials Editor, you can add, delete and modify the materials used in your default Materials file and then write this data as a local Materials file to your working directory. Once written, this local Materials file supersedes the default file due to the fact that it is found and loaded first as specified by the search path defined by \$MATERIALPATH.

The Materials Editor

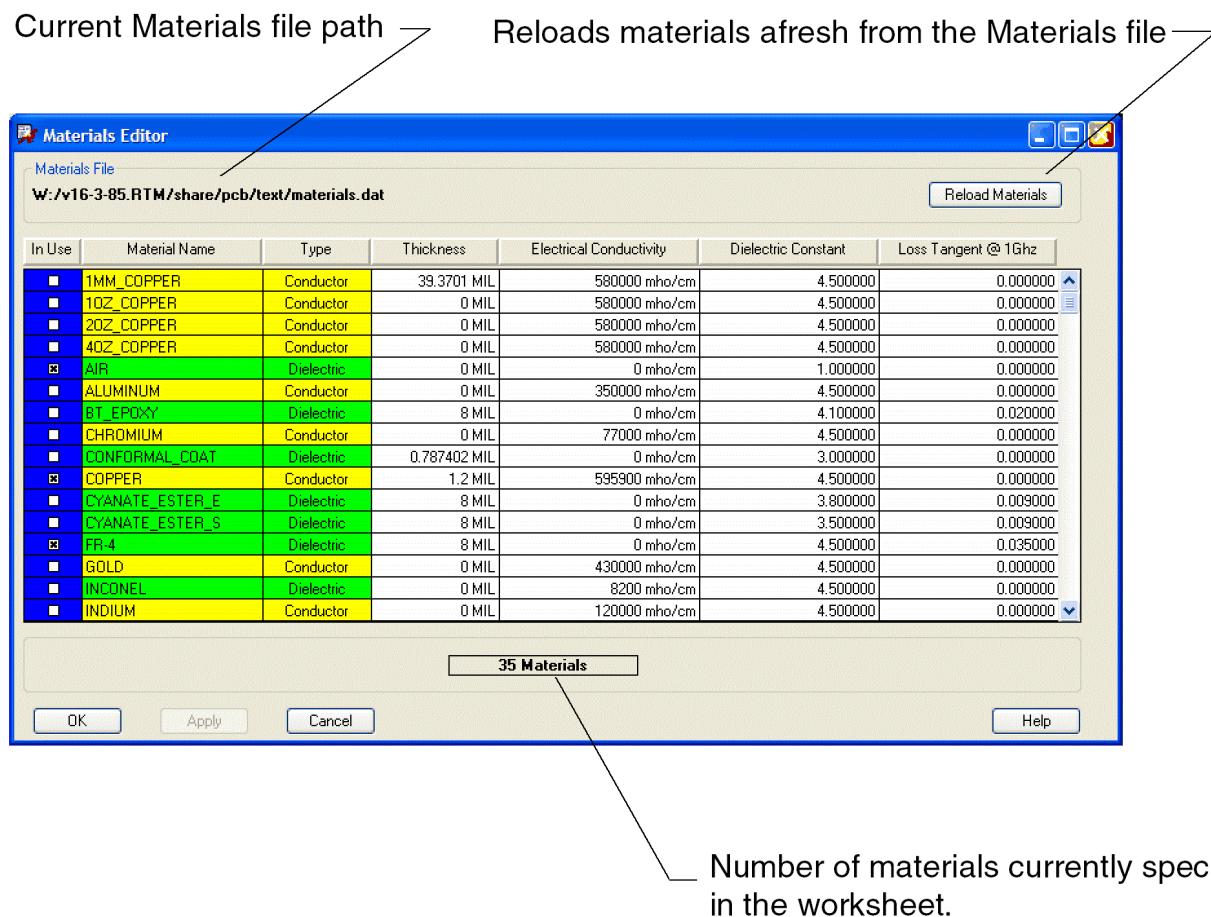
The Materials Editor dialog box shown in the following figure contains a worksheet that presents materials currently defined in your Materials file. Each row represents a single material with columns representing the various attributes of the material. You can resize the dialog box to fully display an extended range of materials available in the Materials file (the default size presents 20 materials).

You can modify material names and most other attribute values by entering a new value in the appropriate cell. Two exceptions are *In Use* and *Type* which cannot be changed.

For descriptions of the Material Editor options and controls.

For information about adding, deleting, and modifying materials using the Materials Editor.

Materials Editor Dialog Box Displaying a Local Materials File



Related Topics

- [xsection](#)
- [Materials Editor Dialog Box](#)
- [Adding a Material](#)
- [Editing a Material](#)
- [Removing a Material](#)

APD: Importing the Layer Stackup of the Substrate

Alternatively, you can import stackup information through a technology file (tech file). A tech file contains parameters, design-level constraint data and modes, including the cross-section, and user-defined properties. Stackup also encompasses information required to perform thermal and signal integrity analysis.

When you import the tech file, the tool overrides all values in the layout. If a constraint in the tech file does not exist in the layout, the tool adds it. If an error occurs in the tech file, the tool tries to continue reading the file, and writes warning and error messages, but does not write an updated layout. If you select the *Run DRC* and update Shapes check box in the *Tech File In* dialog box, the tool automatically recalculates the DRC errors and updates all the dynamic shapes.

Related Topics

- [techfile in](#)

APD: Die-Stack Editor

A die stack is a vertical stack of dies consisting of one or more dies, spacers, and interposers. With this feature, you can:

- Add a spacer from a library of spacers or create one in real time.
A spacer is a block of material (manufactured, molded, or deposited) providing clearance or adhesion, or both, between the spacer and other die-stack components. You can also change the dimensions of an existing spacer.
- Add an interposer, which is a substrate with a single conductor layer that is used in the manufacture of a die stack to support die connectivity.
- Specify thickness and material information for dies, spacers, and interposers.
- Move, rotate, swap, and delete dies, spacers, and interposers.
You can swap two members even when they are not part of the same die stack. You can swap dies and interposers with other dies and interposers because they are both placed on conductor layers. You can swap spacers only with other spacers since they are placed on named dielectric layers, not conductor layers.

- Change the layer of a diestack member.
- View the die stack from the side (2D elevation view), which reflects stack member ordering and Z-axis spatial relationships.
- Visualize and validate the integrity of a die stack in the 3D space with the Cadence 3D Design Viewer option.

 You can extract die-stack data from your design.

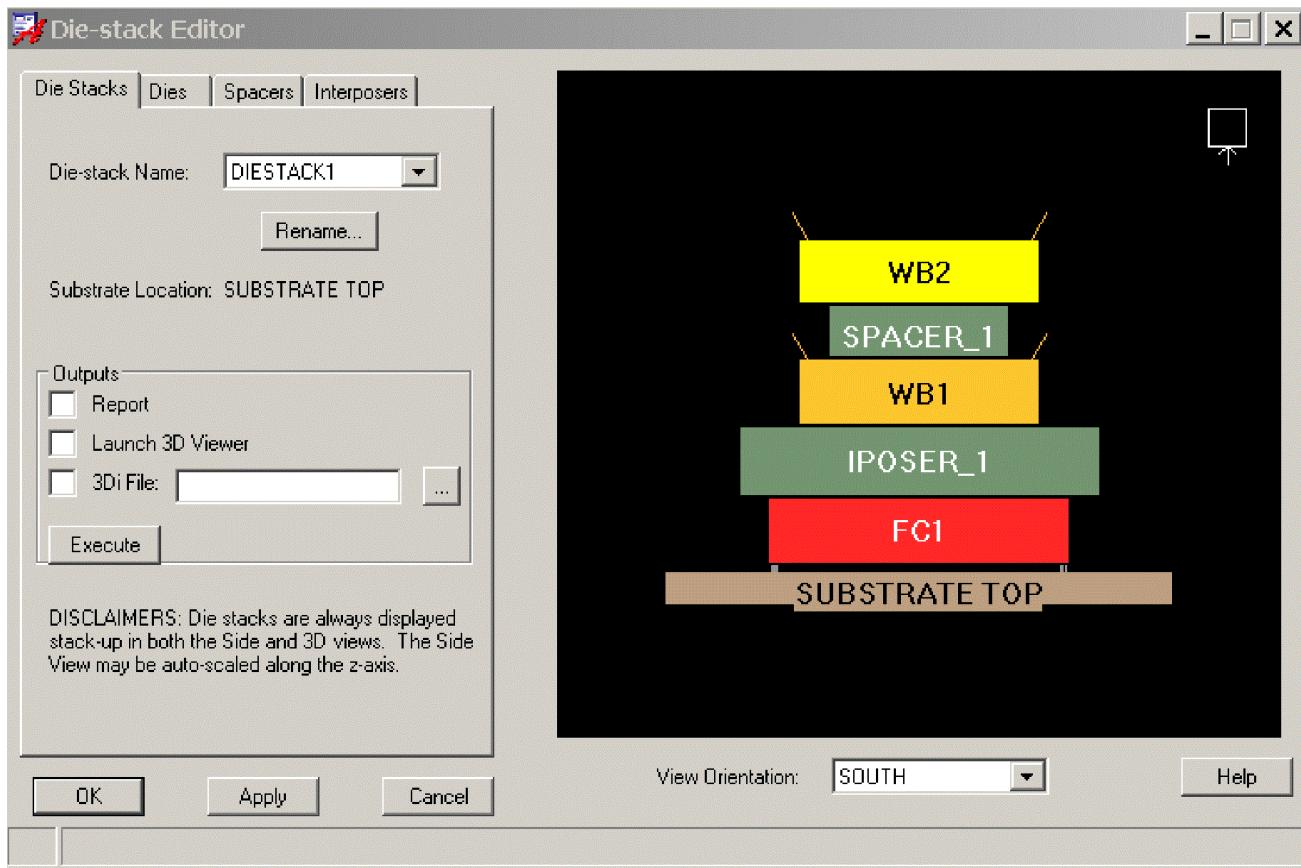
Top and Side Views of the Design

The Design Window provides a plan, or top, view of the design that allows you to look down on a design. You can see design entities and their relationships along the X- and Y-axes, but not along the Z-axis. Yet, you need to be able to accurately model Z-axis information for a die stack and its components as well as locate bond wire die-pad heights above the substrate surface. While the X and Y coordinate locations for die pads are available from a die's footprint, the height above the substrate surface is not. For flip-chip die bumps that are flattened during manufacturing, you need to know the solder bump dimensions to determine accurate die-stack vertical dimensions. The Z-axis information is also necessary for visualizing and performing DRC checks in the Cadence 3D Design Viewer and DRC checker.

An elevation or side view is necessary since visualization and ordering of a die stack is very difficult using the plan view. In the die-stack editor (see the following figure), a wire bond die height is the distance from the substrate surface to the wire bond die surface. A flip-chip die height is the distance from the substrate surface to the top surface of the die (die thickness plus bump height).

The following figure shows the die-stack editor.

Die-Stack Editor



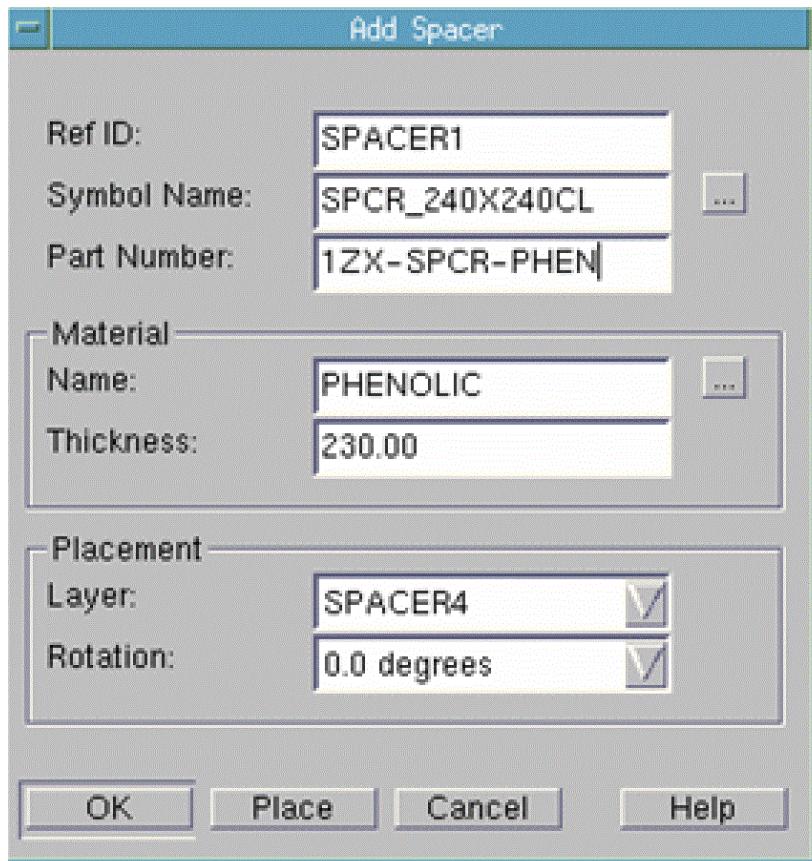
Spacers

A spacer is rectangular and provides the clearance or adhesion, or both, between dies or other die-stack objects that may be necessary to manufacture a die stack. You use the `add spacer` command to add spacers and capture the values for the material and other properties for these items that are used in both electrical and thermal analyses.

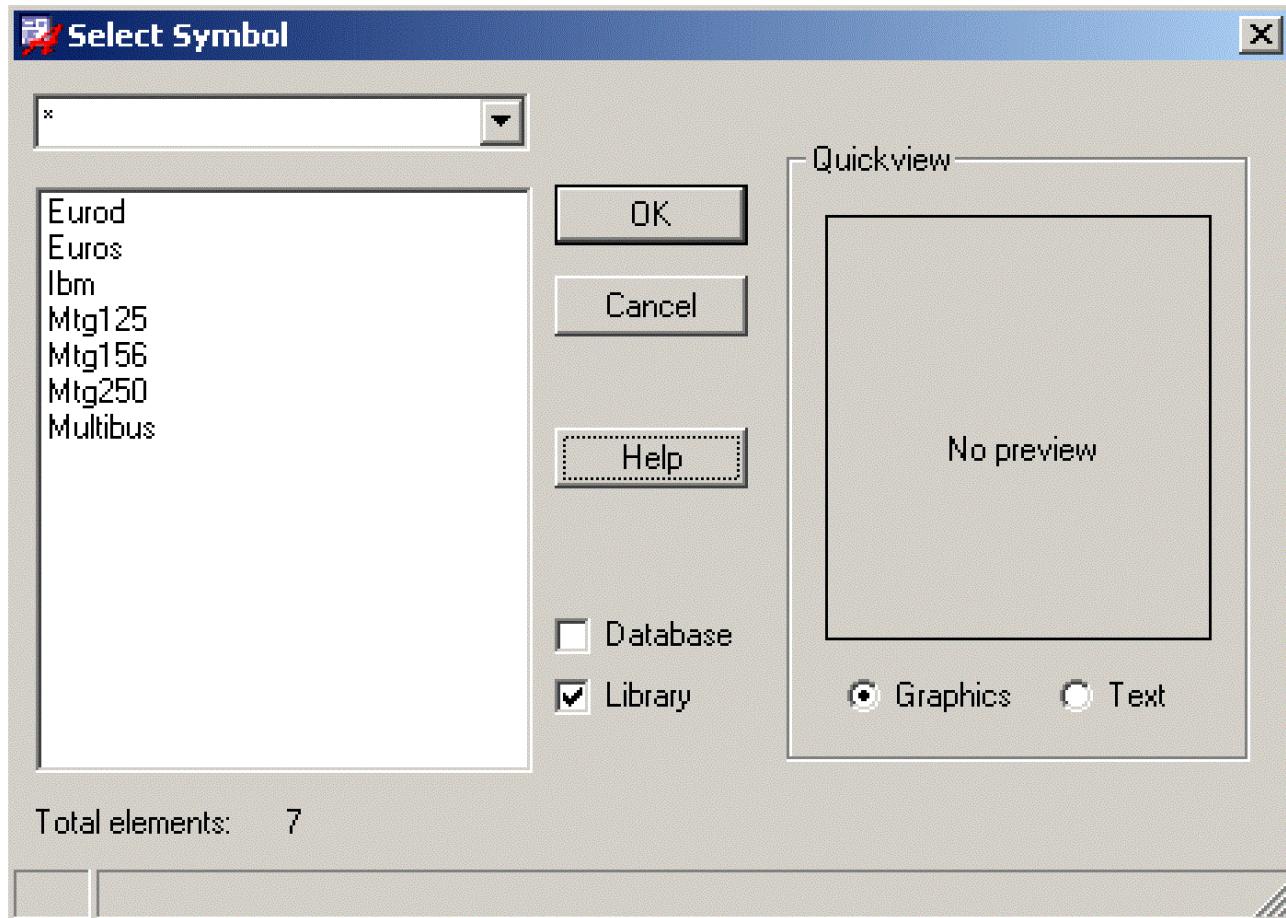
The layout tool automatically attaches the **LOCKED** property to a spacer so that you cannot accidentally edit (move, delete, rotate) the symbol children, for example, place-bounds, assembly-rectangles, and so on. Although you can edit this property, it is recommended that you do not as corruption can occur if symbol children are edited. Whenever you update the spacer, the layout tool automatically adds the property to the spacer if you have removed it.

The following figures show the dialog boxes that appear when you run the `add spacer` command.

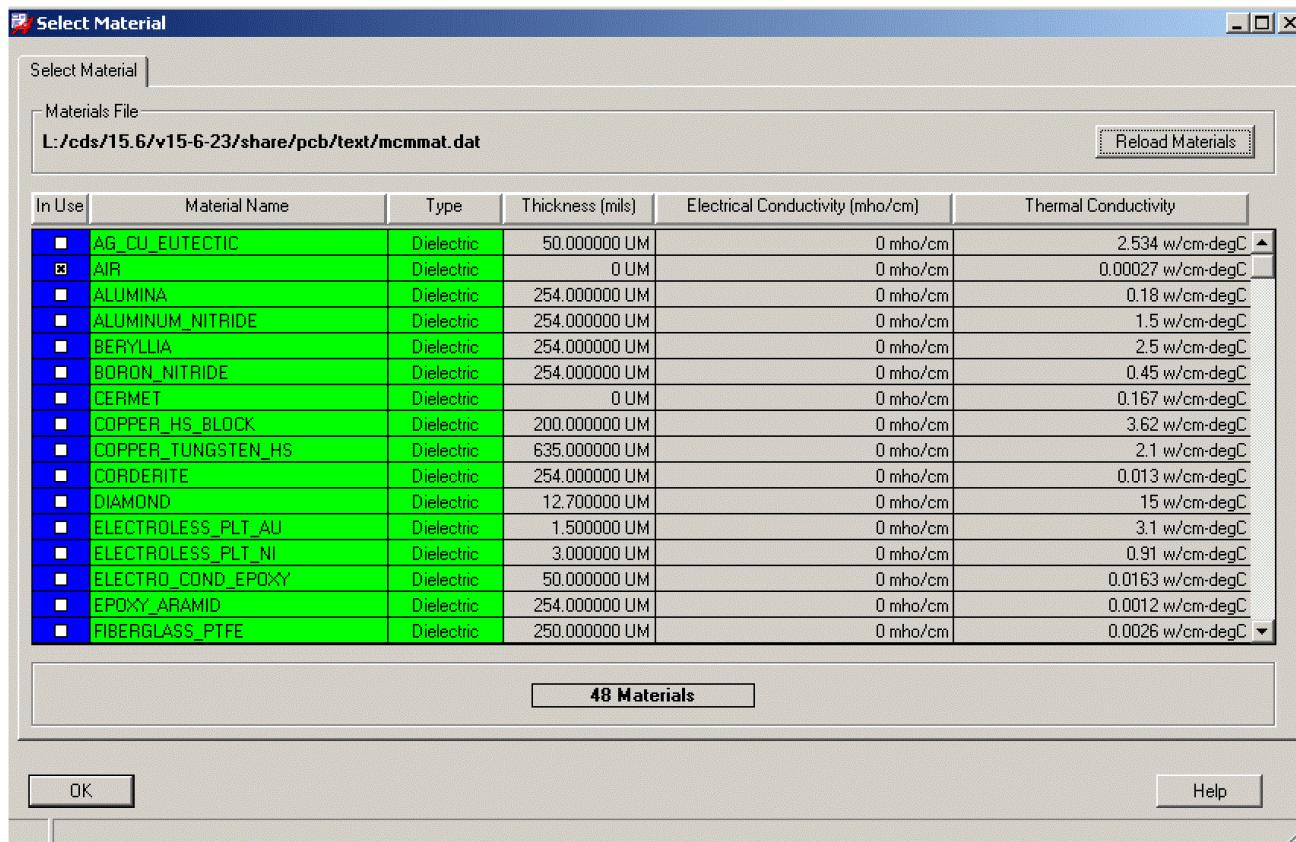
Add Spacer Dialog Box



Select Symbol Dialog Box



Select Material Dialog Box



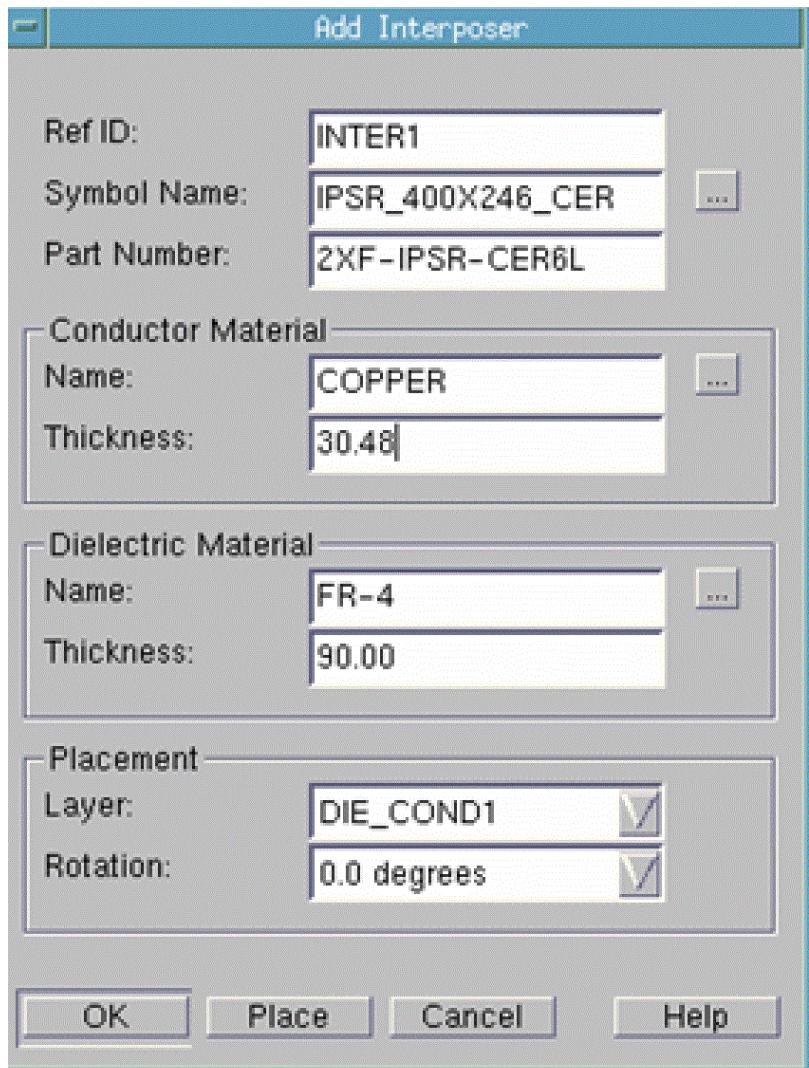
Interposers

Use an interposer with wire bond dies where the die-pad positions create wire bond lateral spans that are beyond the physical limits of a wire bonding machine. Use the `add interposer` command to add interposers and capture the values for the materials and other properties for these items that are used in both electrical and thermal analyses.

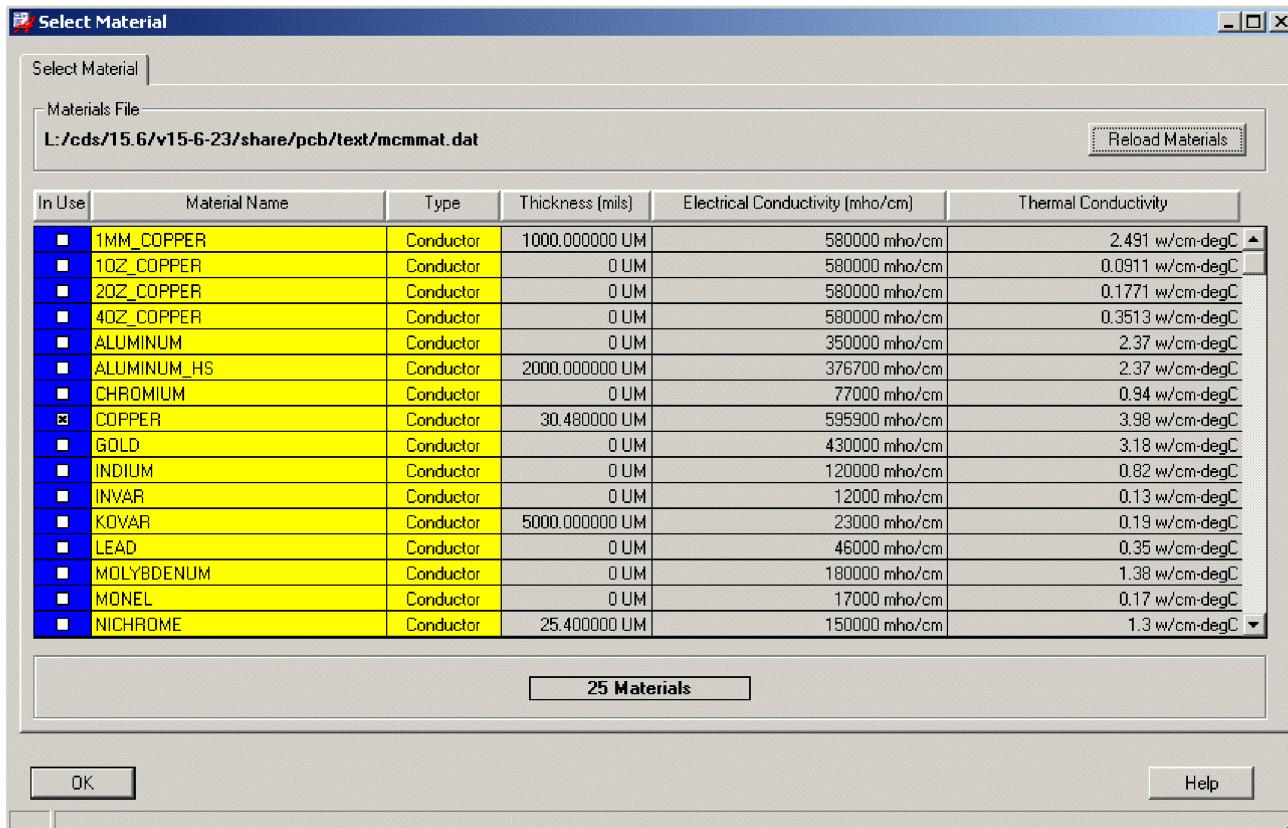
The layout tool automatically attaches the LOCKED property to an interposer so that you cannot accidentally edit (move, delete, rotate) the symbol children, for example, place-bounds, assy-rects, vias, conductor, and so on. Although you can edit this property, it is recommended that you do not as corruption can occur if symbol children are edited. Whenever you update the interposer, the layout tool automatically adds the property to the spacer if you have removed it.

The following figures show the dialog boxes that appear when you run the `add interposer` command.

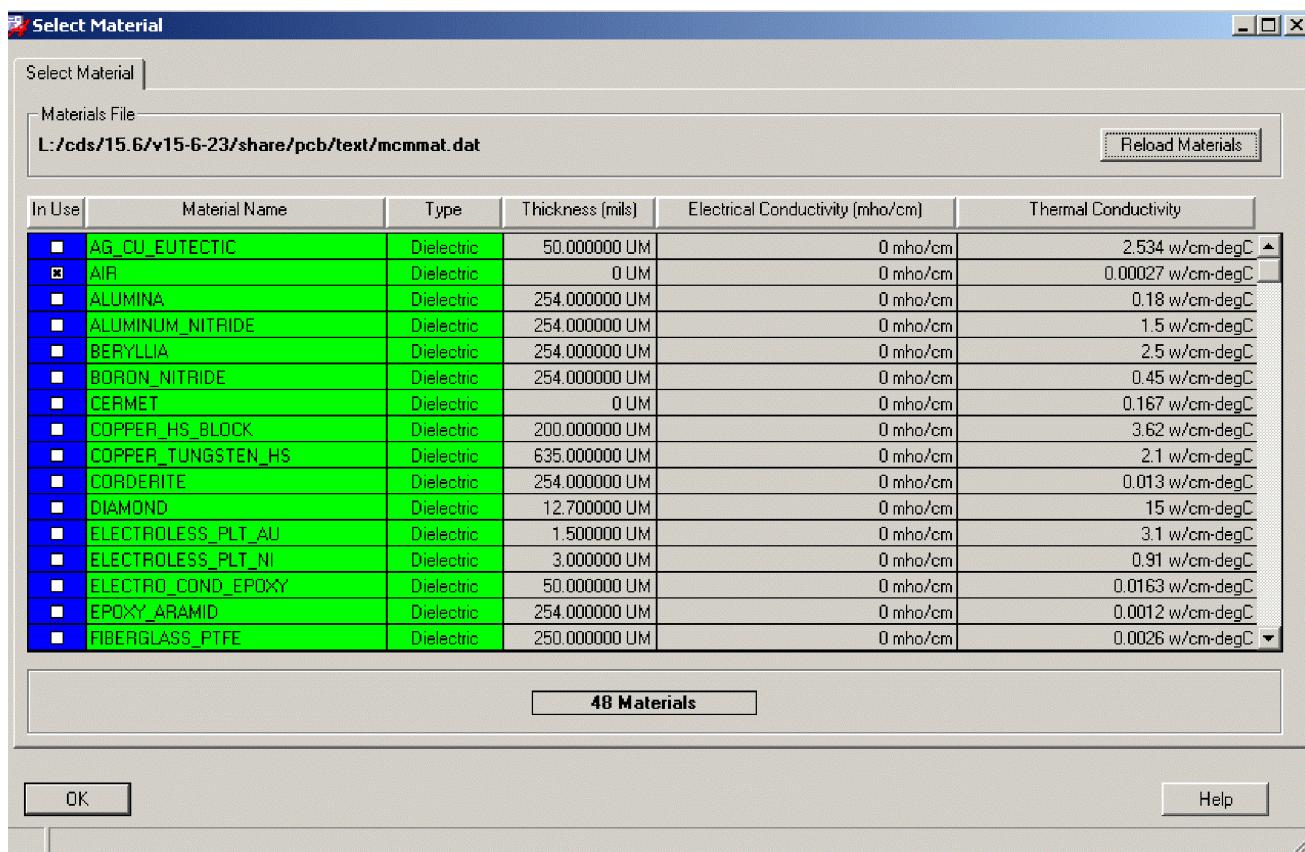
Add Interposer Dialog Box



Select Material Dialog Box - Conductor

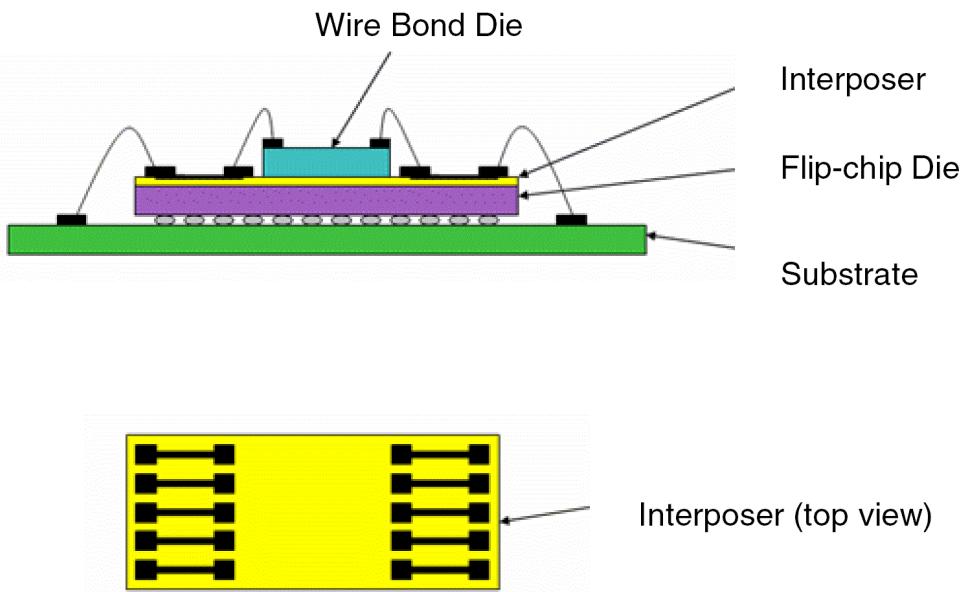


Select Material Dialog Box - Dielectric



The following figure shows an example of a die stack where an interposer is necessary. For this application, interposers have no logical components, that is, they are not part of the package design's netlist and have no reference designators or pins. An interposer's conductor paths connect through vias, clines, and shapes inheriting the net of the circuitry to which they attach. For applications where you need an interposer with a logical component, for example, reference designators and pins, you need to design and implement the interposer as a standard or co-design die.

Die Stack with Interposer



Die-Stack Editor Report

You can generate a report which includes the following information about the selected die stack:

- Component design name and date
- Stack name
- Number of objects in the stack (dies, spacers, and interposers) including the number of each object type in the stack
- Substrate surface and stack height above that substrate surface
- List of ordered stack objects
- Detailed description of each object in the stack

Related Topics

- [Die-stack Data Extraction](#)
- [add spacer](#)
- [add interposer](#)

Working with Graphic Design Elements

The layout tool provides several methods for adding elements to design drawings. Using arcs and lines, you can compose complex shapes, especially useful for laying out design-specific ground and power planes. In addition, the layout tool lets you create other shapes that act as boundaries. Each shape (or area constraint) is referred to as a keepin or a keepout.

You can add or edit graphic elements to a design at any point in the design flow, but typically occurs during the layout stage. After you have added graphic elements to a design, you often must edit one or more characteristics of some elements. You can copy, delete, or mirror an element.

 The functionality this chapter describes may not be available in all versions of Allegro X PCB Editor.

Adding Elements to a Design

You add graphic elements to layouts and symbols by using the `add` commands.

Lines

Choose *Add – Line* (`add line` command) to create outlines, irregular shapes, and other figures in a design. When you create a line, the tool displays a rubber band from the point you chose to the cursor. The rubberband line adheres to the 90- or 45-degree constraints specified in the *Line Lock Direction* field of the Options tab, and draws arcs or line segments as specified in the *Line Lock Mode* field.

Rectangles

You can add rectangles in drawings and define them as:

- Route keepins
- Package keepins
Keepins and keepouts can be any shape.
- All other non-etch (conductor) rectangles (for example, mechanical, package, and format symbols)

Choose *Add – Rect* (`add rect` command) to create outlines, shapes, and other figures in a design.

Filled Rectangles

You can add filled rectangles (rectangles) in drawings that you can define as:

- Etch/Conductor rectangles (with associated net name for voltage distribution)
- Masks
- Package keepouts
- Package placement boundaries
- Route keepouts
- Via keepouts

 Keepouts can be any shape.

Filled rectangles added to the ETCH/CONDUCTOR class represent etch/conductor on the design. Choose *File – Plot* (`plot` command) to write line-plot commands to the photoplot file to fill that area on that layer. Filled etch/conductor rectangles are frequently used to distribute a voltage over an area on a layer, so a net name (voltage) is associated with each such filled rectangle.

When you add a filled rectangle as etch/conductor, the tool displays a dialog box that prompts you for the name of the net with which the filled rectangle is to be associated. Thereafter, you can attach connect lines to the rectangle so it is physically attached to its net. When you choose *Route – Connect* (`add connect` command), you can make the connection because the rectangle is logically on that net.

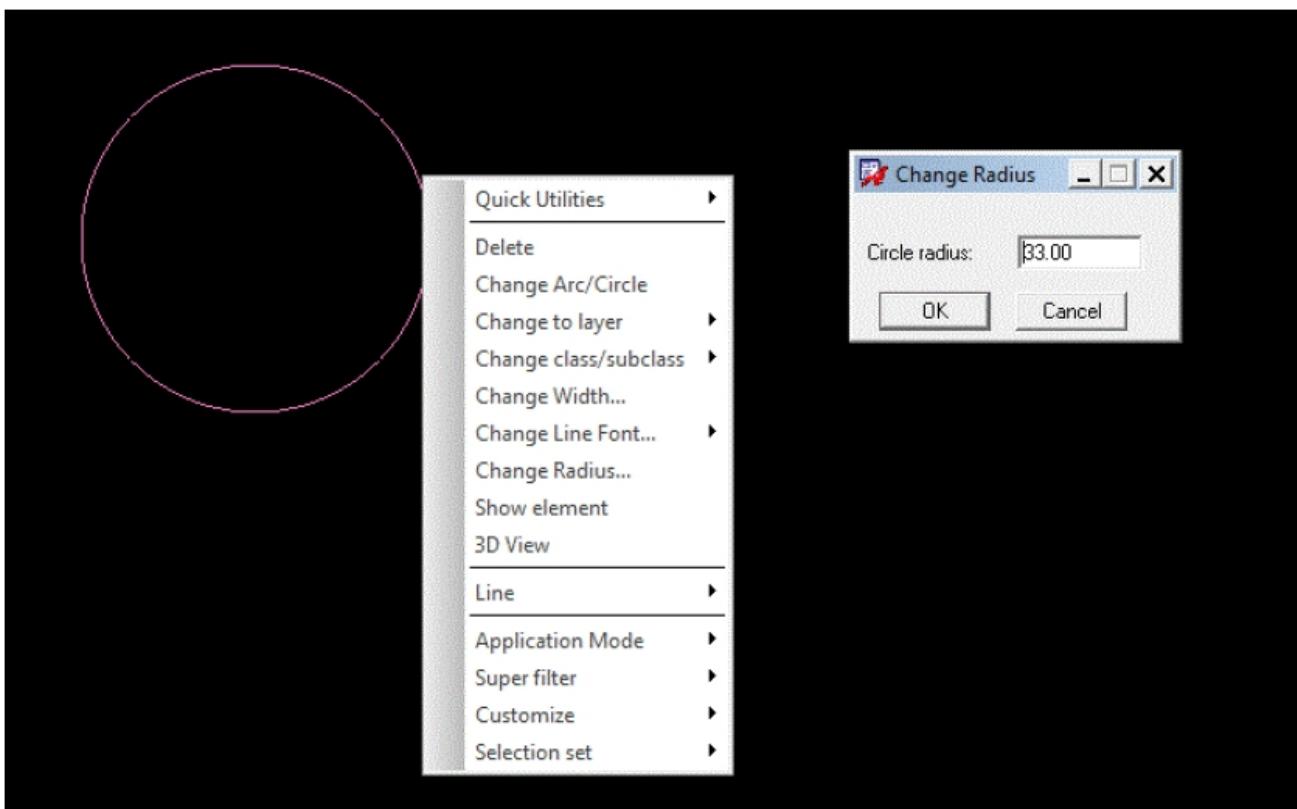
 You can verify the net of any etch/conductor rectangle by choosing *Display – Element* (`show element` command) on the rectangle.

Circles

You can add circles to drawings in the following classes:

- BOARD GEOMETRY
- ETCH/CONDUCTOR
- PACKAGE GEOMETRY

You can change the radius of the circle, using *Change Radius* (`change radius` command).



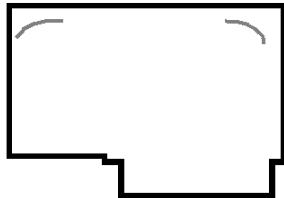
Arcs

You can include arcs in a drawing that you can use to round off edges in an outline or keepin or keepout area. The layout tool lets you add arcs to a drawing by either specifying the radius of the arc, or by picking the end points. The layout tool provides two menu commands for adding arcs:

- When you know the radius of the arc, choose *Add Arc w/Radius* (`add_rarc` command).
- When you know the end points of the arc, choose *Add – 3pt Arc* (`add_arc` command).

Specifying Arcs by Radius

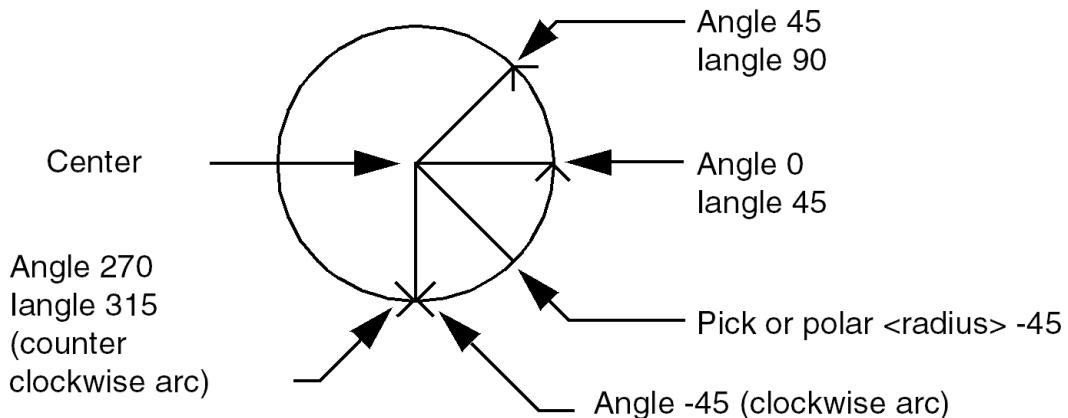
When you know the radius of the arc you are adding to the design, choose *Add Arc w/Radius* (`add_rarc` command). Locating the center point of an arc at a fixed reference is often important for mechanical specification of a design, particularly the outline. For example, to round off edges of an outline, you can create arcs, as shown in the following illustration:



Choosing *Add Arc w/Radius* (`add rarc` command) lets you specify the precise center point location or radius of the arc to be created and is typically used for the OUTLINE subclass, the default. However, you can specify another layer for the arc by picking the subclass field in the Options tab and then choosing from the pop-up list of subclasses that appears.

When adding arcs by specifying the radius, you must enter three points:

- Center point
- Start point, or enter a polar coordinate to establish the start point
- End point or enter an `angle` or `iangle` command as illustrated in the following example:



Specifying Arcs by Pick Points

Choose *Add – 3pt Arc* (`add arc` command) when you know the end points of the arc. Three points are required: a point to start the arc, an end point, and a third point to determine the radius of the arc.

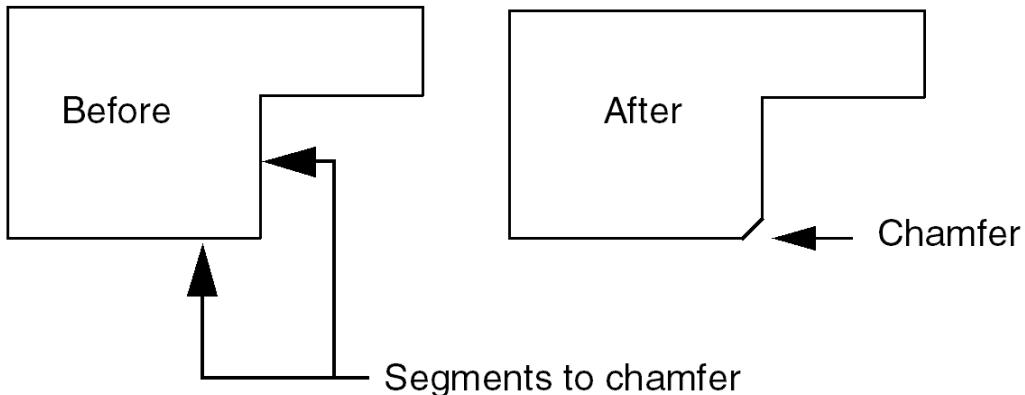
Chamfers

When you choose *Manufacture – Dimension/Draft – Chamfer* (`draft chamfer` command), the layout tool fits a line segment between two existing line segments according to parameters you define. If the chosen lines do not intersect, the tool projects the lines to their intersection and inserts the chamfer defined by the settings specified in the Options tab.

⚠ The `draft chamfer` command does not operate on connect lines.

When you choose two line segments for a chamfer, the tool merges all related segments, including the new chamfer, into one line.

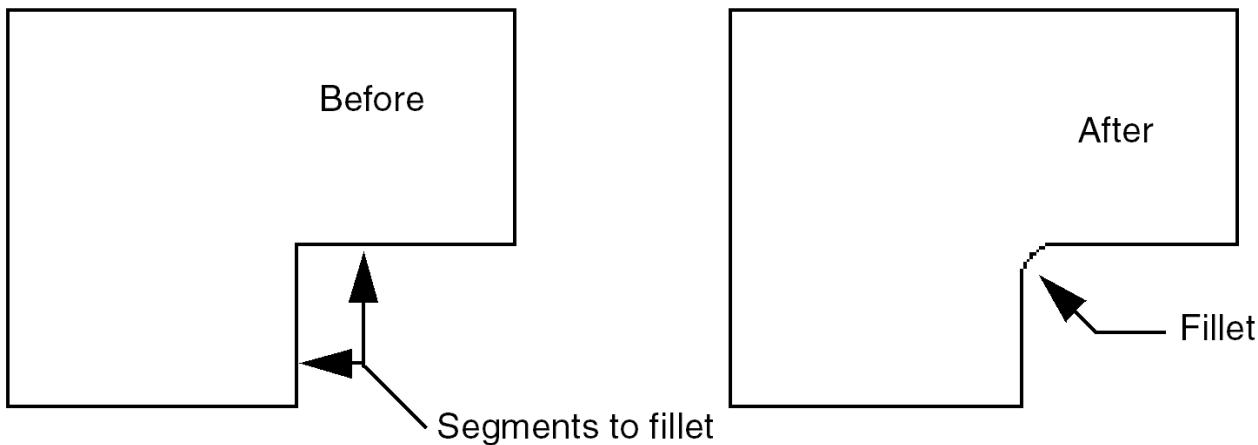
Specifying Lines for *Draft Chamfer*



Fillets

When you choose *Manufacture – Dimension/Draft – Fillet* (`draft fillet` command), the tool fits an arc between two existing line segments but does not operate on connect lines. When you choose two line segments for a fillet, the tool merges the related segments, including the new fillet, into one line.

Specifying Lines



Shapes

You can add shapes to layouts by choosing *Shape – Polygon* (`shape add` command), *Shape – Circular* (`shape add circle` command), or *Shape – Rectangular* (`shape add rect` command).

Related Topic

- [add line](#)
- [add rect](#)
- [plot](#)
- [add connect](#)
- [show element](#)
- [add frect](#)
- [add circle](#)
- [change radius](#)
- [add rarc](#)
- [add arc](#)
- [angle](#)
- [iangle](#)
- [draft chamfer](#)
- [draft fillet](#)
- [shape add](#)
- [Working with ETCH/CONDUCTOR Shapes](#)

Editing Elements in a Design

Use editing commands to alter an design object after you have added it to the design.

 By default, symbol pin editing is inhibited in the tool. This may prohibit you from performing certain editing functions. To perform unrestricted editing on symbol pins, attach the UNFIXED_PINS property to the symbol instance or to the layout drawing.

Copying Elements in a Design

You can duplicate one or more elements in a design as:

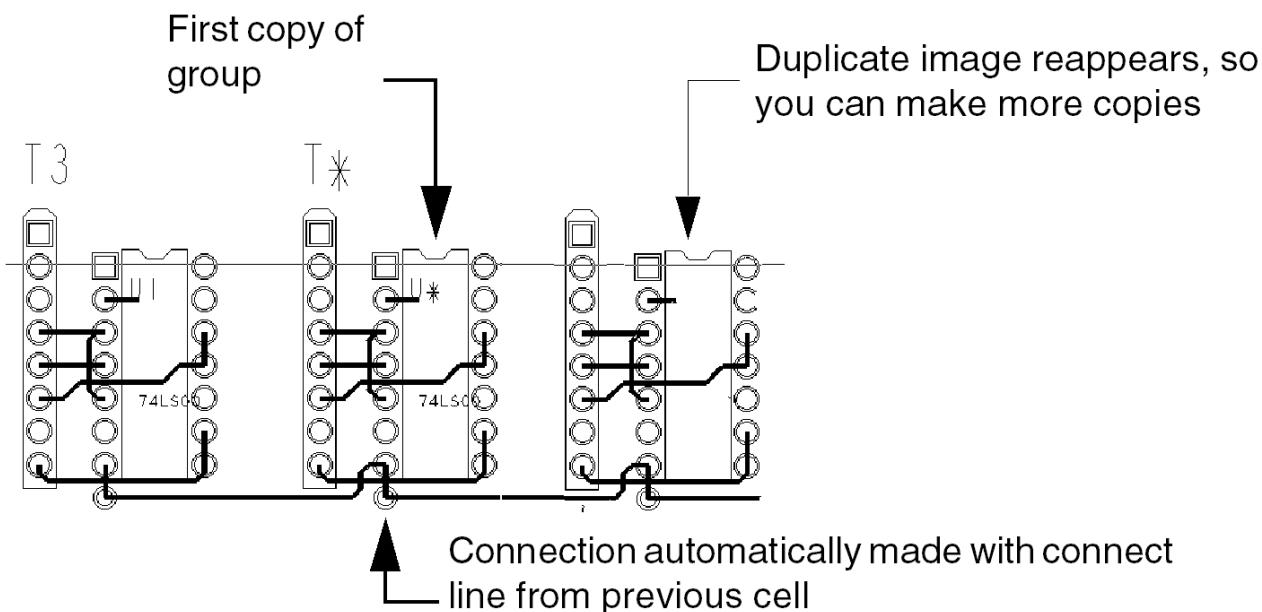
- Individual elements
- Step and repeat patterns on an X,Y (rectangular) grid
- Radial (polar) patterns around a user-defined point

Use the Options tab to control the copy method as *Rectangular* (step and repeat patterns) or *Polar* as described next.

When you choose *Edit – Copy* (`copy` command), you can choose one or more elements, including package symbols and their connecting etch/conductor and vias, and place copies of the elements, as a group, anywhere in a drawing. Reference designators of copied packages are changed to generic reference designators (U*, R*, C*, and so on) so they do not conflict with any existing reference designators in the drawing. Choose *Logic – Assign RefDes* (`assign refdes` command) to assign reference designators to the copied components.

You create arrays of packages and connecting etch/conductor lines by picking a "cell" of packages and connecting etch/conductor lines, and then copying the cell. It is particularly useful that `copy` automatically connects connecting etch/conductor lines with the pins they overlay during the copying. This means that all power, ground, bus, address, and data lines automatically connect to the pins of copied components.

In the following illustration, the copy appears at the location picked. Note that the reference designators have been changed to dummies. Each refdes has its leading characters, but has had its number changed to *.



Copying Elements in Rectangular Patterns

You can copy elements or add pins in a rectangular grid array when using the `copy` command.

Copying Elements in Radial Patterns

You can copy an element around a user-defined origin in angular increments. Radial patterned elements are useful for placing components in odd-angle circular patterns such as round test boards. Use the Options tab to control how the tool copies the element.

In symbol editing mode, polar copy is also available when you choose *Layout – Pins* (`add pin` command) to add pins for round connectors and switches with pins that are difficult to correctly orient.

Copying and Pasting Elements in a Design

When you copy an element it is stored in a buffer and available for pasting at multiple destinations later in the design cycle. You can select single or multiple objects during the `copy` command using any of the standard selection methods.

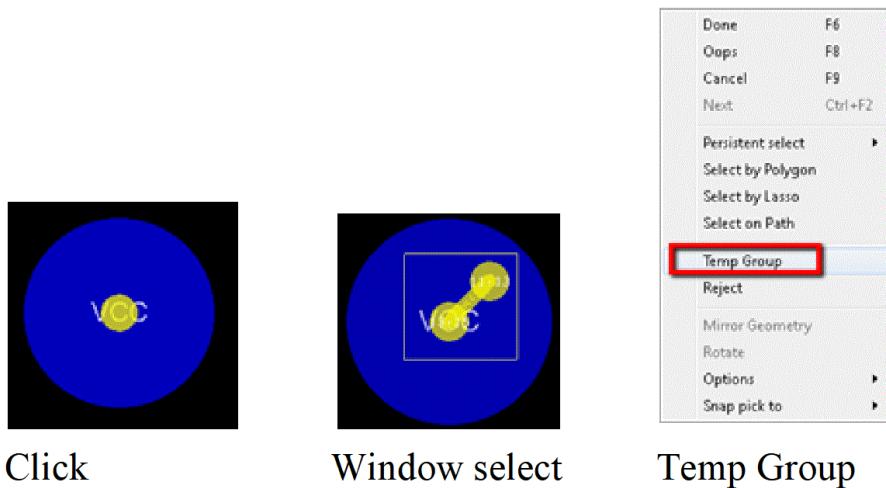


Figure 2-1 Methods for copying an element

For single element, the copy origin is set to the value specified in the *Options* tab. But when more than one objects are selected or the *Copy origin* is set to *User Pick* in the *Options* tab, you need to specify the copy origin. To select the origin either click in the canvas, or use pop-up menu command or specify the coordinates of the origin in the command window.

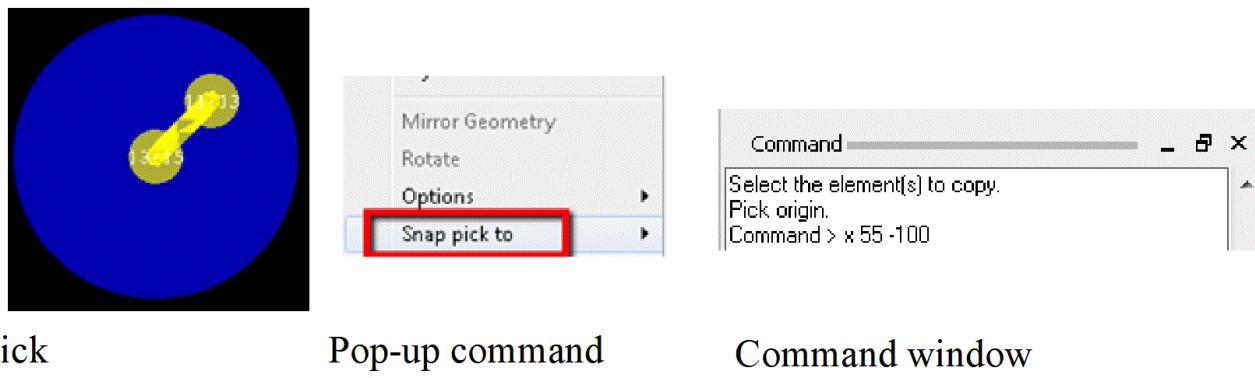


Figure 2-2 Methods for selecting Copy origin

Once an object is copied, the following message appears in the command window and the *paste* command invokes automatically.

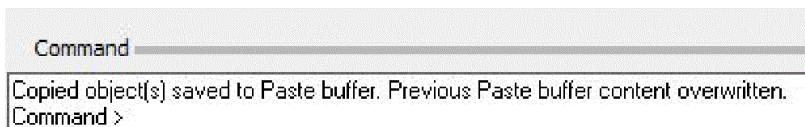
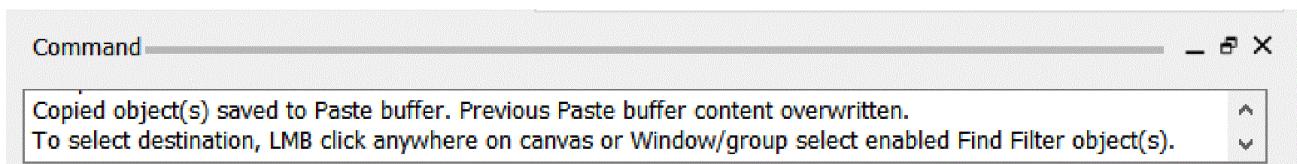


Figure 2-3 Message for successful copy action

The copied object remains in the buffer until it is overwritten and as long as the session is active. You can now paste the same object to different destinations any time after copying it.



The `paste` command is available only for vias, pins, fingers, clines, or combination of these objects. This command automatically snaps the origin of the copied objects to the center of pins, vias, fingers and dangling end of clines.

Pasting Objects in Pre-Select Mode

Use any of the standard selection methods to select the destination objects.

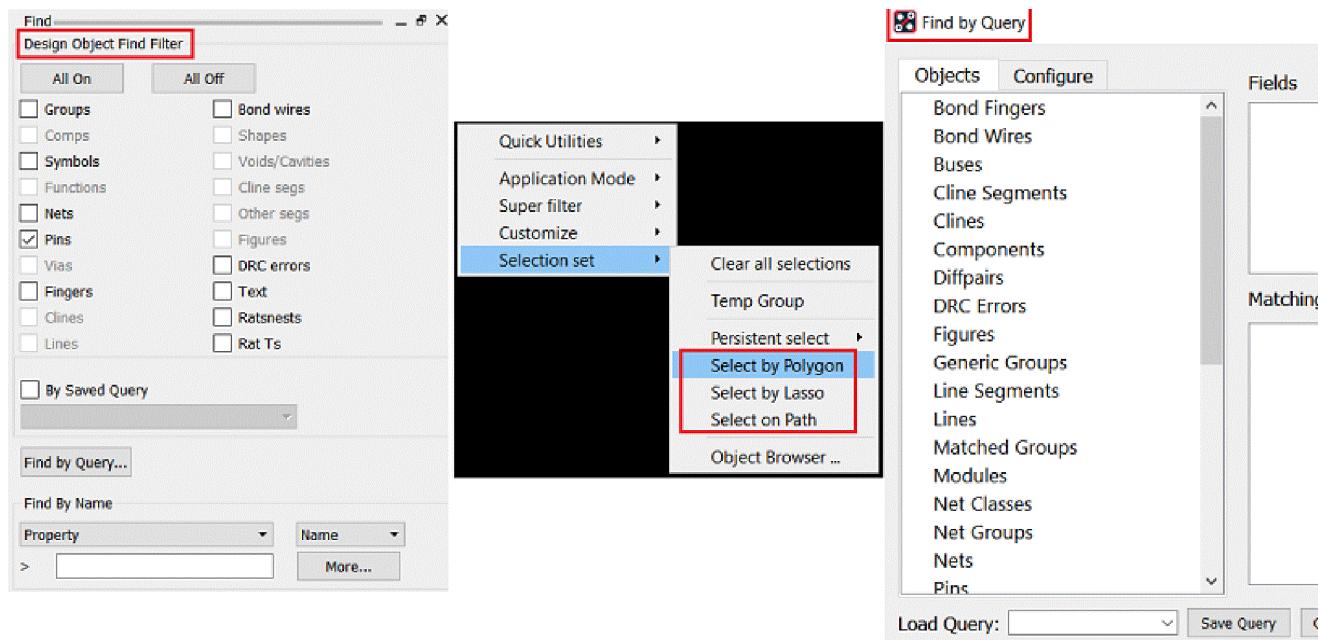


Figure 2-4 Methods for selecting destinations for paste

Hover the mouse over one of the destination objects and click to paste. You can use right-click options to rotate and mirror the objects before placing them.

The copied object is pasted on each of the selected objects and following message appears in the command window.

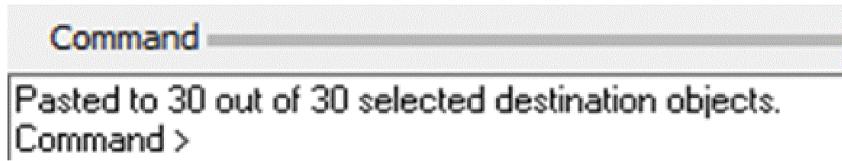
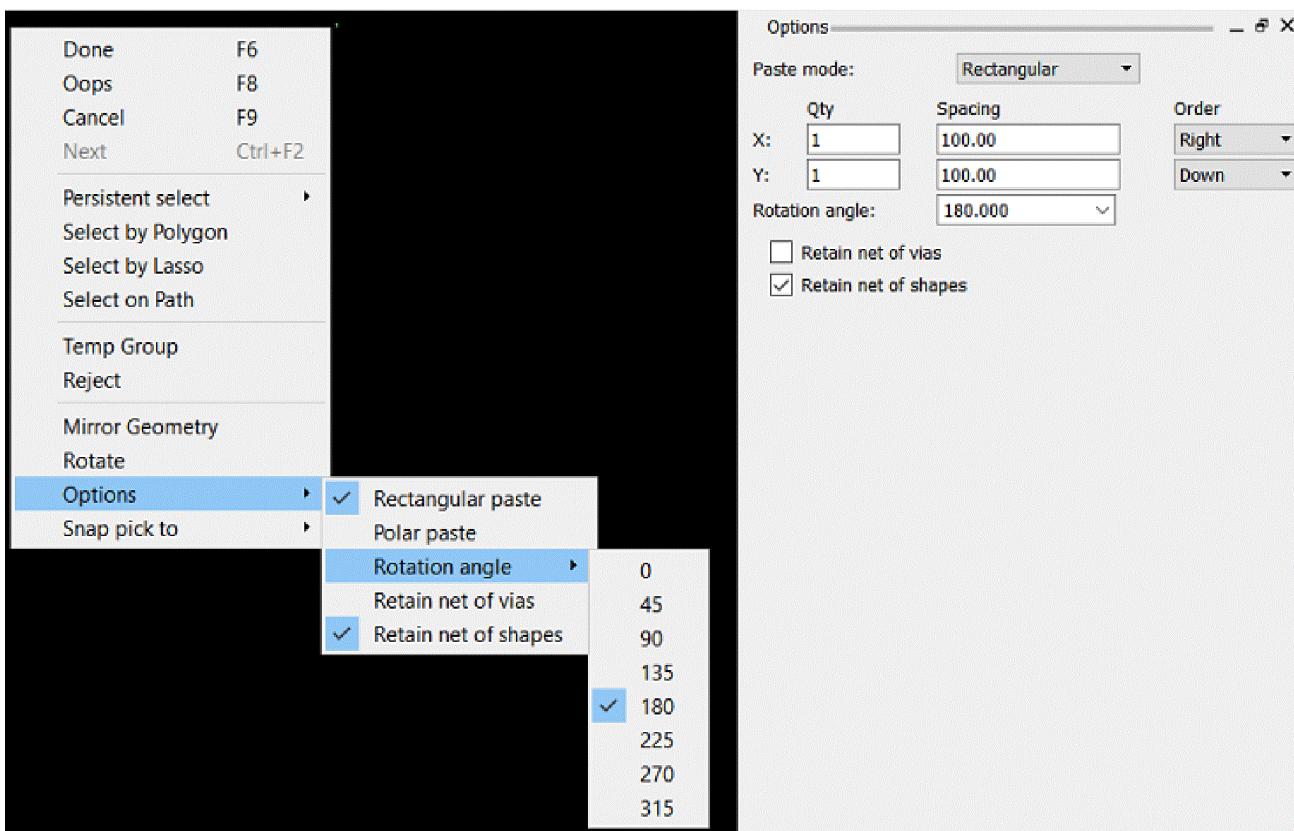


Figure 2-5 Message after successful paste action

Pasting Objects in Post-Select Mode

In the post-select mode, the `paste` command provides functionality to paste objects in a rectangular or radial pattern. In addition to that you can use pop-up menu options to rotate and mirror the objects before placing them to multiple destinations.

Choose *Edit – Paste* and specify the settings in the *Options* tab.



For pasting to the selected objects, first choose objects in the *Find* filter and then pick the destination objects using any of the selection method. The copied object attaches to each of the destination objects.

For object independent paste, click anywhere in the design or select a window or right-click and choose multi-destination paste pop-up options, such as *Select Poly/Lasso/Path*, *Temp group*, and *Snap pick to*. Depending on the selection method, the `paste` command snaps copied objects in the design.

- Blank canvas:
 - Click: Copied objects are snapped at the click location.
 - Window select: Copied objects are not placed and the following error is displayed.
 - No valid destination selected. Valid selections when enabled in find filter are pins, vias, fingers and/or clines with one dangling end.
- Click at pads or clines:
 - Click: Copied objects are snapped to the center of the pads or clines end.
 - Window select: If the destination object is selected in the *Find* filter, the `paste` command snaps copied objects to the center of the pads or clines end.

If no object is selected in *Find* filter, copied objects are not placed and the same error is displayed.

- Click objects other than pads or clines:
 - Click: Copied objects are snapped at the click location.
 - Window select: Copied objects are not placed and the same error is displayed.

Pasting Copied Objects to Clines

You can paste objects to clines if the cline has only one dangling end.

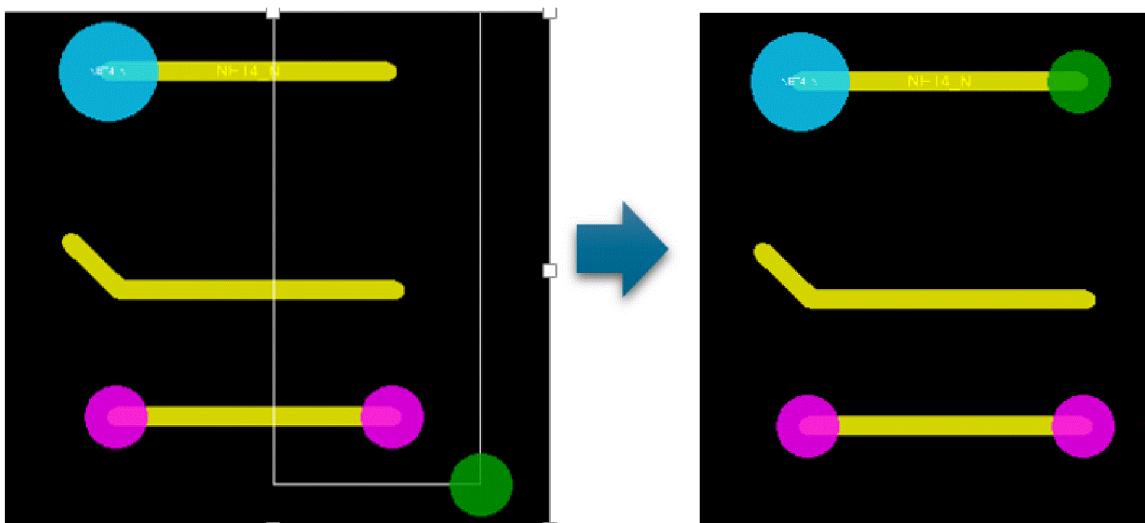


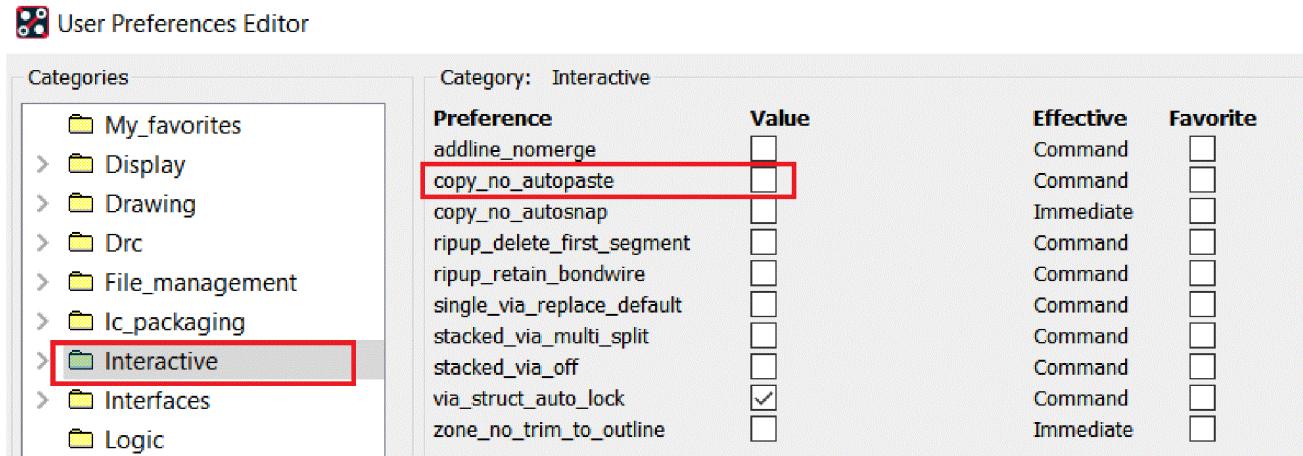
Figure 2-6 Pasting objects to clines

If there is no dangling end, no placement happens. The command checks for the dangling ends of clines and displays the following message in the command window:

```
Command
Pasted to 4 out of 6 selected destination objects. 2 of the clines selected do not have only one dangling end.
Command >
```

Pre-17.4 Copy and Paste Use Model

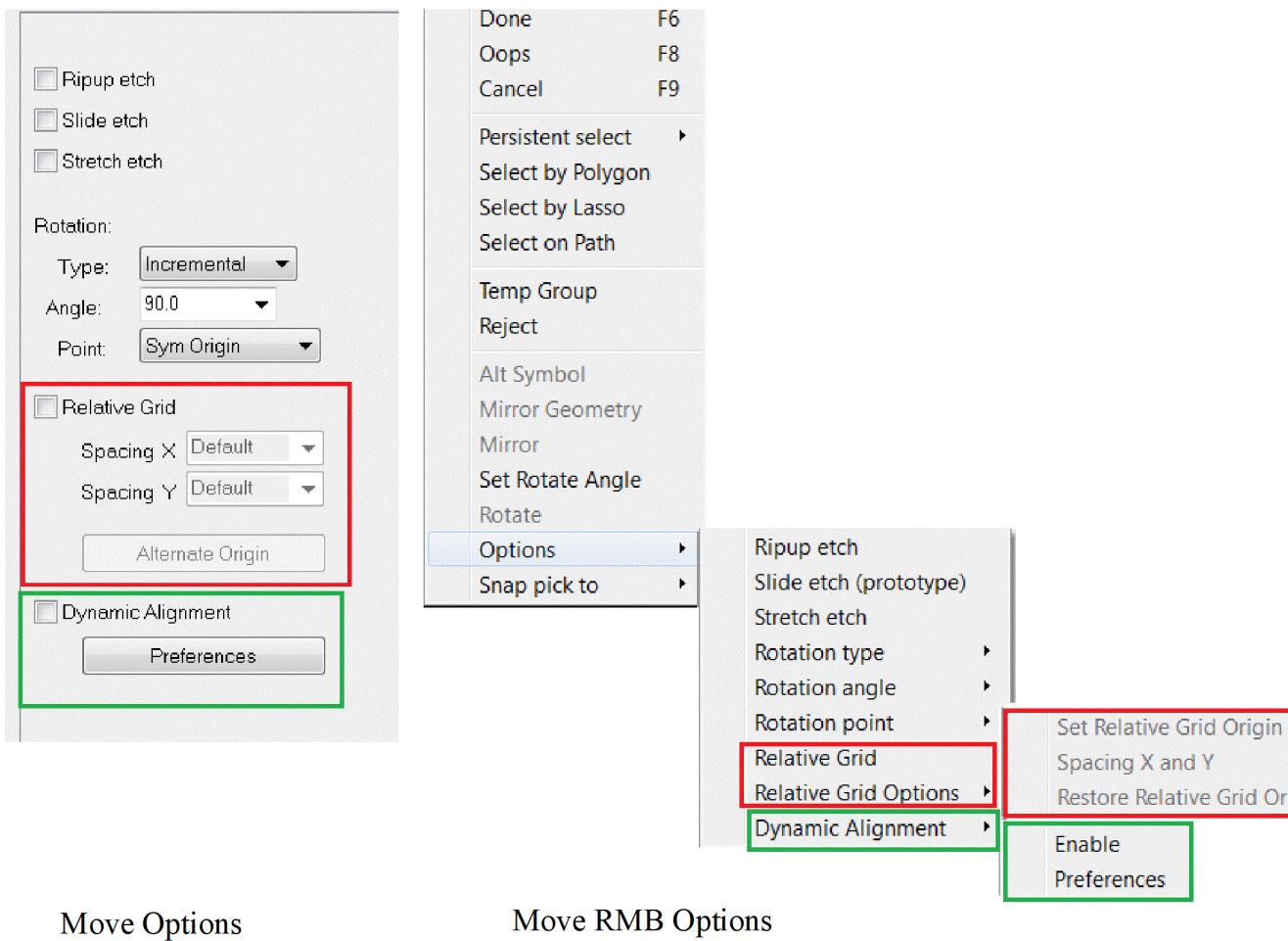
To revert `copy` command behavior to pre-17.4 release, set an environment variable `copy_no_autopaste`: If this variable is enabled, the `copy` command pastes the objects to a single destination location at a time.



Moving Elements

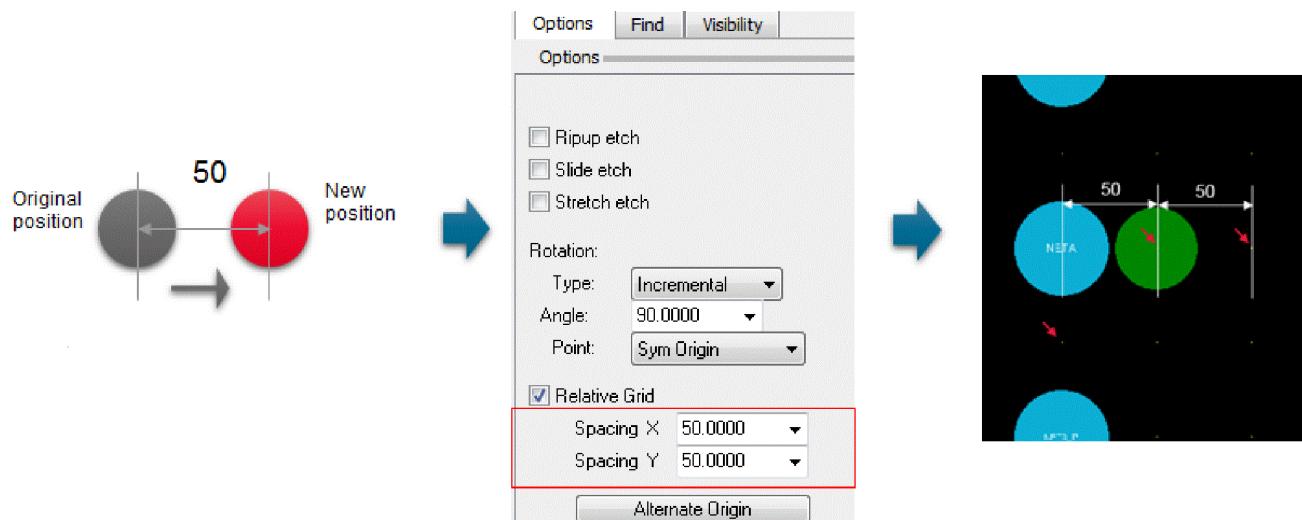
When you select *Edit – Move* (`move` command), you can choose one or more elements to rotate and move to a new location. The command also provides options:

- to move objects inline, or relative to other objects or location by enabling *Relative Grid* option
- to align objects with already placed objects dynamically by enabling *Dynamic Alignment* option



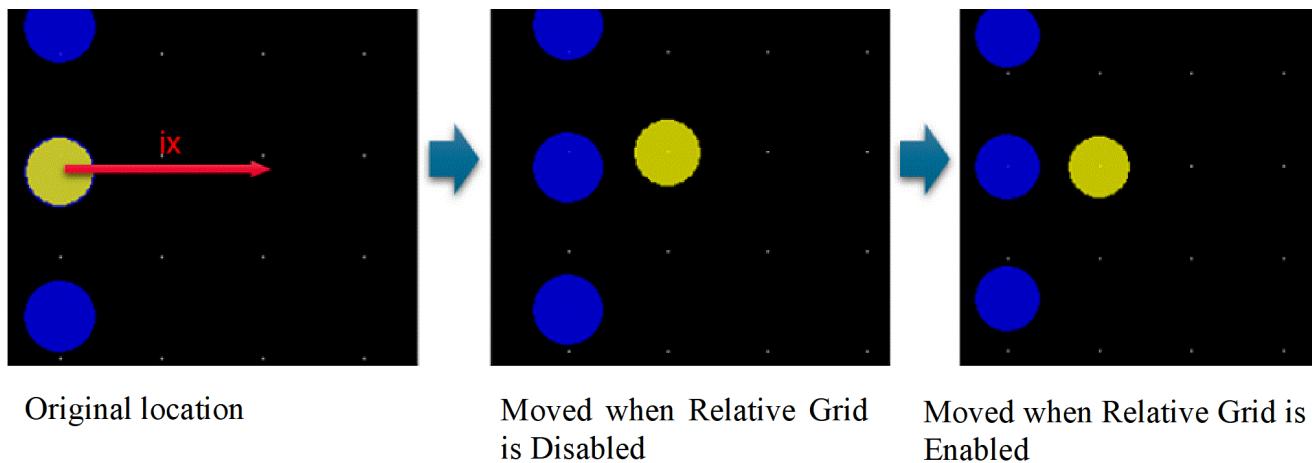
Move with Relative Grid ON

You can specify different grid values to move object inline or relative to an object at a certain distance. By default, relative grid spacings are set to the grid spacing values defined for the design.



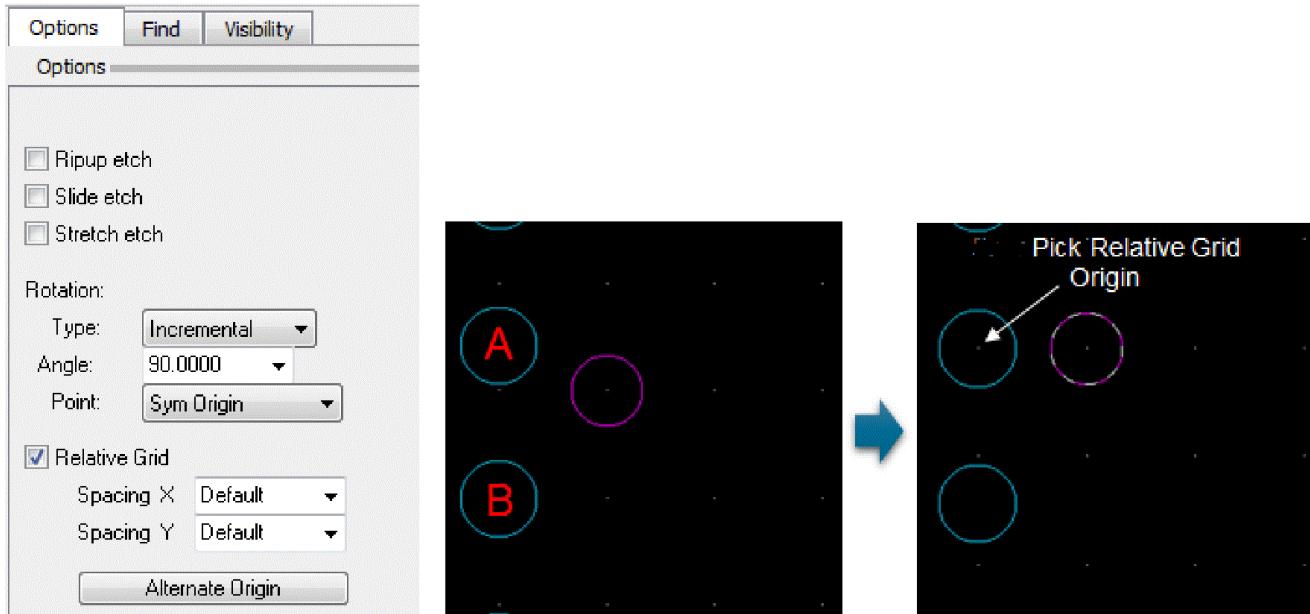
In-Line Incremental Move

For in-line incremental move, you set the relative grid origin to the origin of the selected objects. You can now move the selected objects from their original X/Y position.



Relative Incremental Move

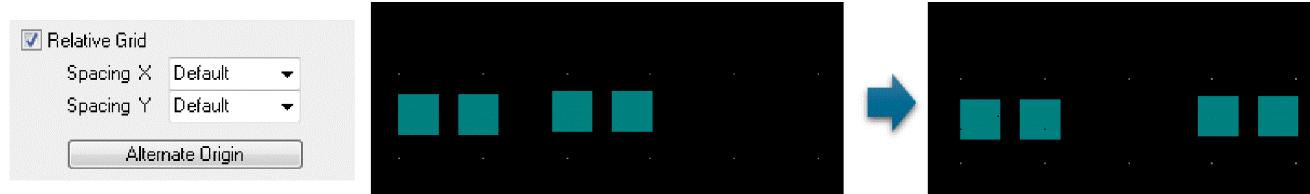
For relative incremental move, you can choose an alternate origin, such as that of an object or grid point. You can now move objects incrementally, relative to that object or point.



Examples

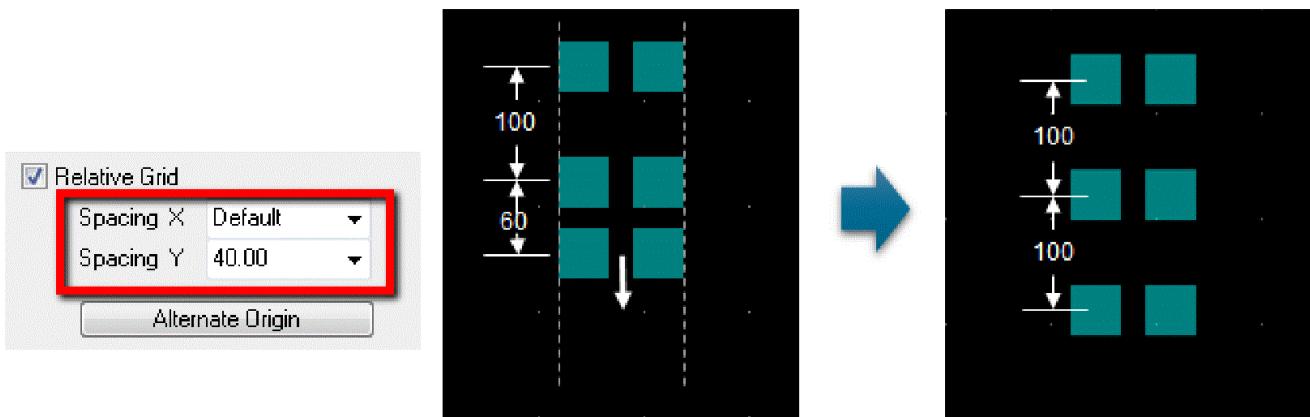
Moving Off-Grid Objects Inline with Default Grid Spacing

You can move an off-grid object incrementally in X/Y direction inline with original location. Enable *Relative Grid* with spacing set to default and select the object(s). Grid is temporarily changed relative to selected object(s) origin using default grid spacing.



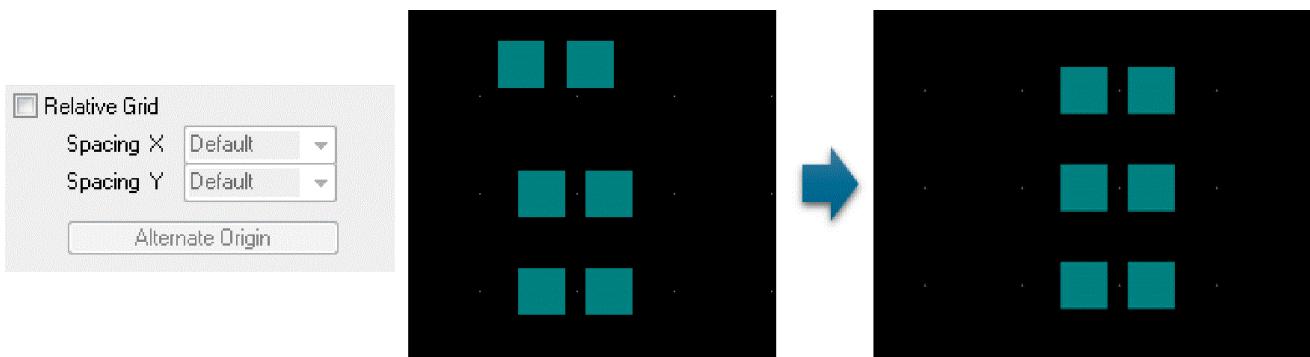
Moving Objects Inline with Relative Grid Spacing

You can move an off-grid/on-grid object incrementally in X/Y direction inline with original location with different grid spacing values. Enable *Relative Grid*, specify grid spacing X/Y values and select the object(s). Grid is temporarily changed relative to selected object(s) origin using specified grid spacing values.



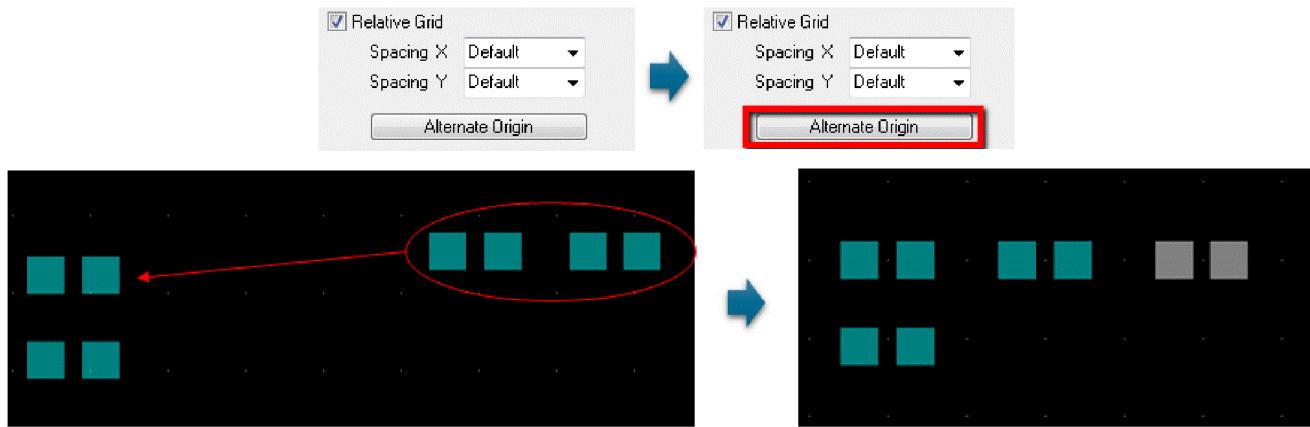
Moving an Off-Grid Object on the Grid (Default Move Operation)

You can move an object on-grid that is currently off-grid. Do not enable *Relative Grid*. The move command uses the drawing origin and default grid values.



Moving Object Relative to Another Object with Default Relative Grid Spacing

You can move an object relative to another object using default grid spacing values. Enable *Relative Grid* with spacing set to default and click *Alternate Origin*. For accurate selection of the new relative grid origin use *Snap pick to* option from the right-click pop-up menu. You can also click in the canvas to select the relative grid origin. Grid is temporarily changed to the new alternate origin and uses default grid spacing.



Move Object Relative to Another Object with Relative Grid Spacing

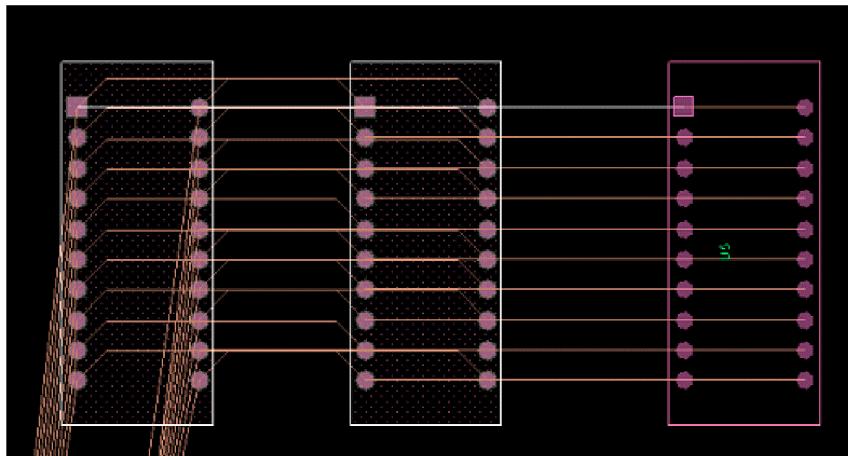
You can move an object relative to another object at a specified distance. Enable *Relative Grid* and set spacing X and/or Y value(s). Click *Alternate Origin* and define a new relative grid origin. Grid is temporarily changed to the new alternate origin and use specified grid values.



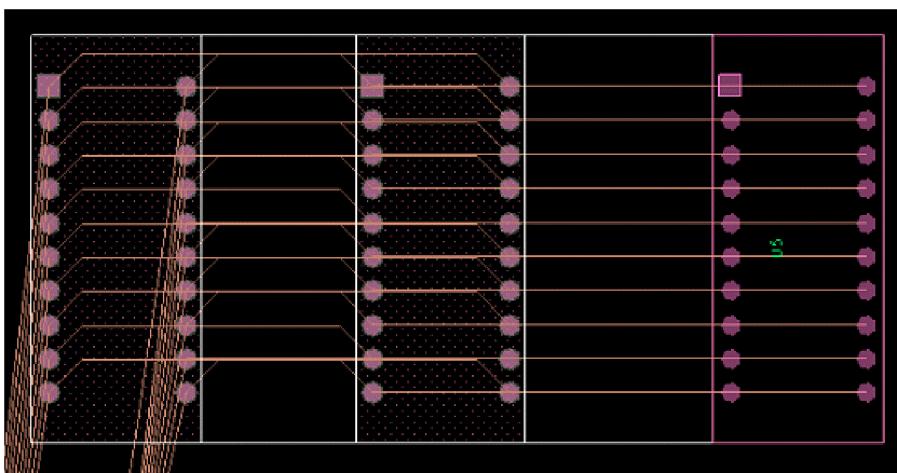
Move with Dynamic Alignment ON

You can enable the component alignment behavior during the move operation either from the *Options* tab or from the pop-up menu options.

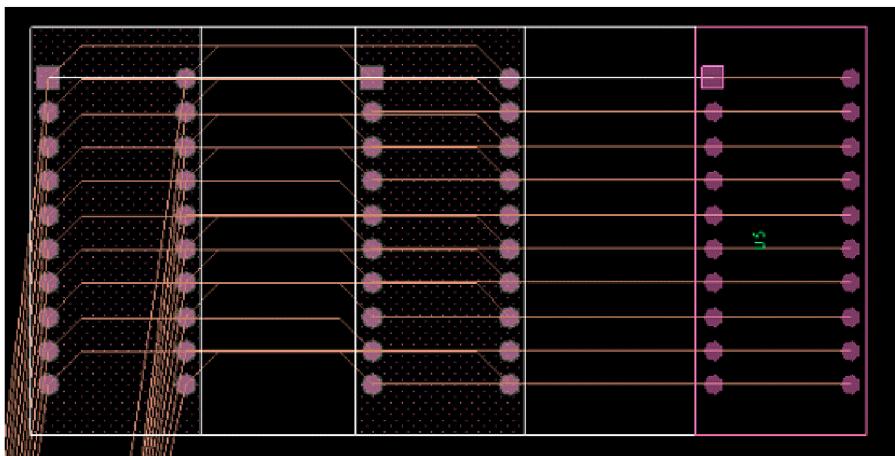
When enabled, align guides appears as you move the component that matches with already placed components. You can configure the align guides either by component origin or by place bound extent of the component, or by both.



Alignment by component origin



Alignment by component place bound extent

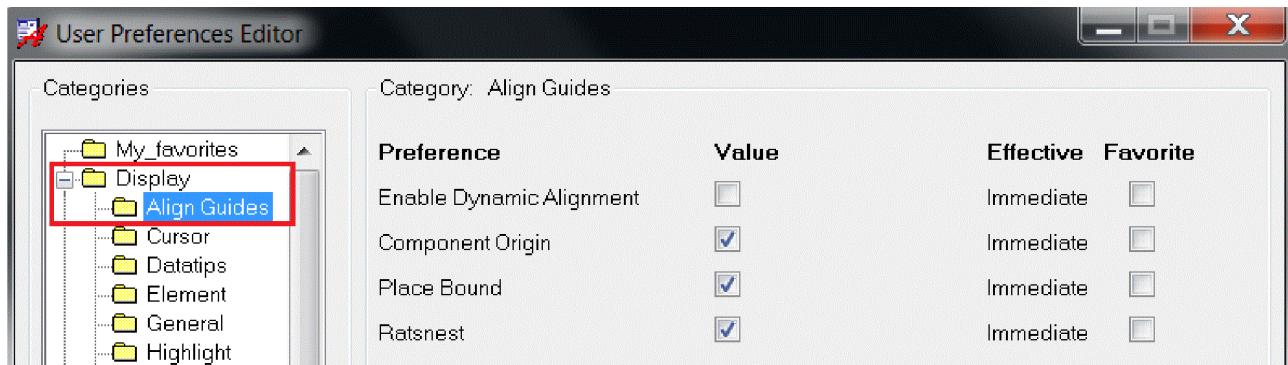


Alignment by both component origin and place bound extent

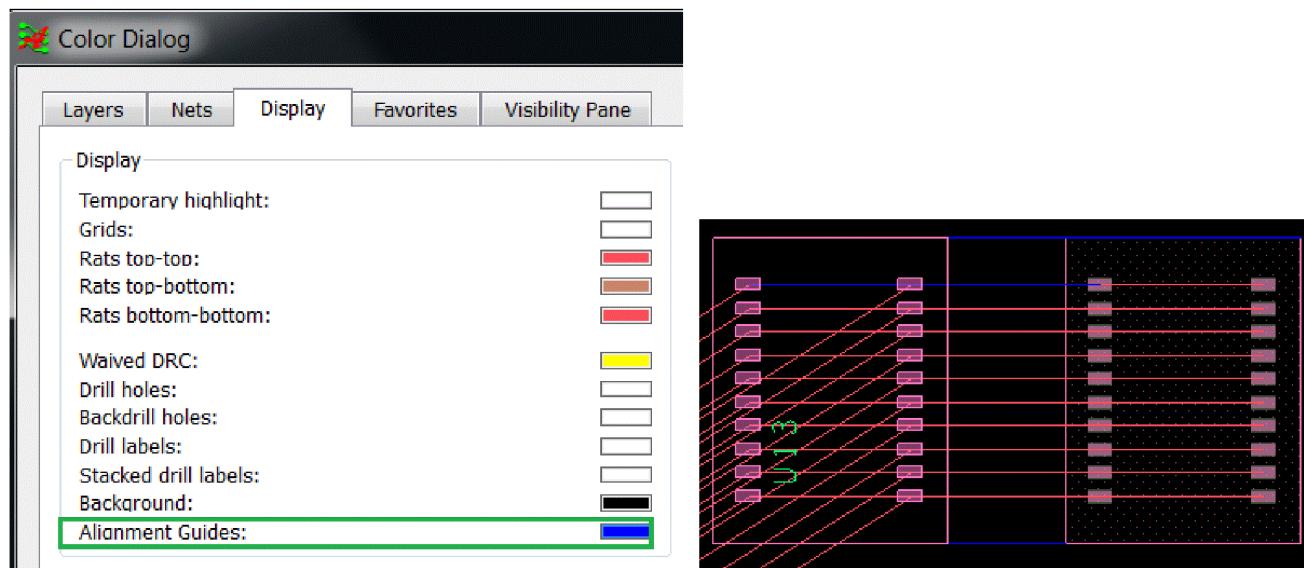
For non-orthogonally placed components, the align guidelines are not visible. The dynamic

alignment functionality does not work for such components.

The guidelines appear in both the vertical and the horizontal directions and snaps to align with both on-grid or off-grid components. The align guide settings can be changed any time during move operation using *Preferences* button. It opens the *Align Guides* section in the *Display* category of the *User Preferences Editor* dialog box.



You can also select the color of the align guides lines that indicate the available snap points. This setting is located in the *Display* tab of the *Color* dialog box.



Changing Element Characteristics

You can change the characteristics of graphic elements in a design by choosing the *Edit – Change* (change) command. Characteristics you can change are:

- Width of lines and connect lines—entire lines, segments, or cuts of segments or all connect

lines on a net.

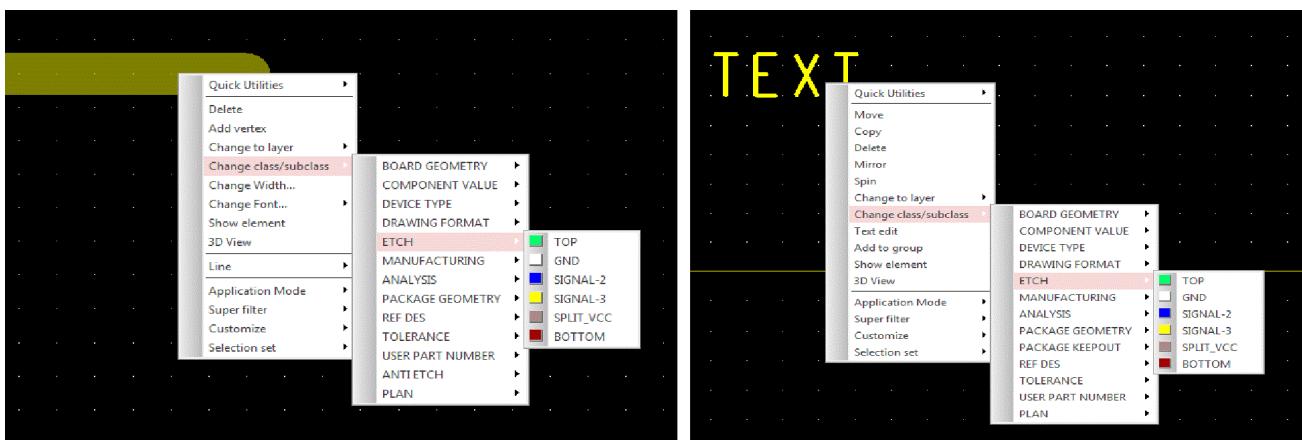
- Subclass (layer) of text, rectangles, filled rectangles, lines, and connect lines; in the case of lines and connect lines, element lines, segments, or cuts of segments or all connect lines or filled rectangles on a net.
- Text parameter block number.

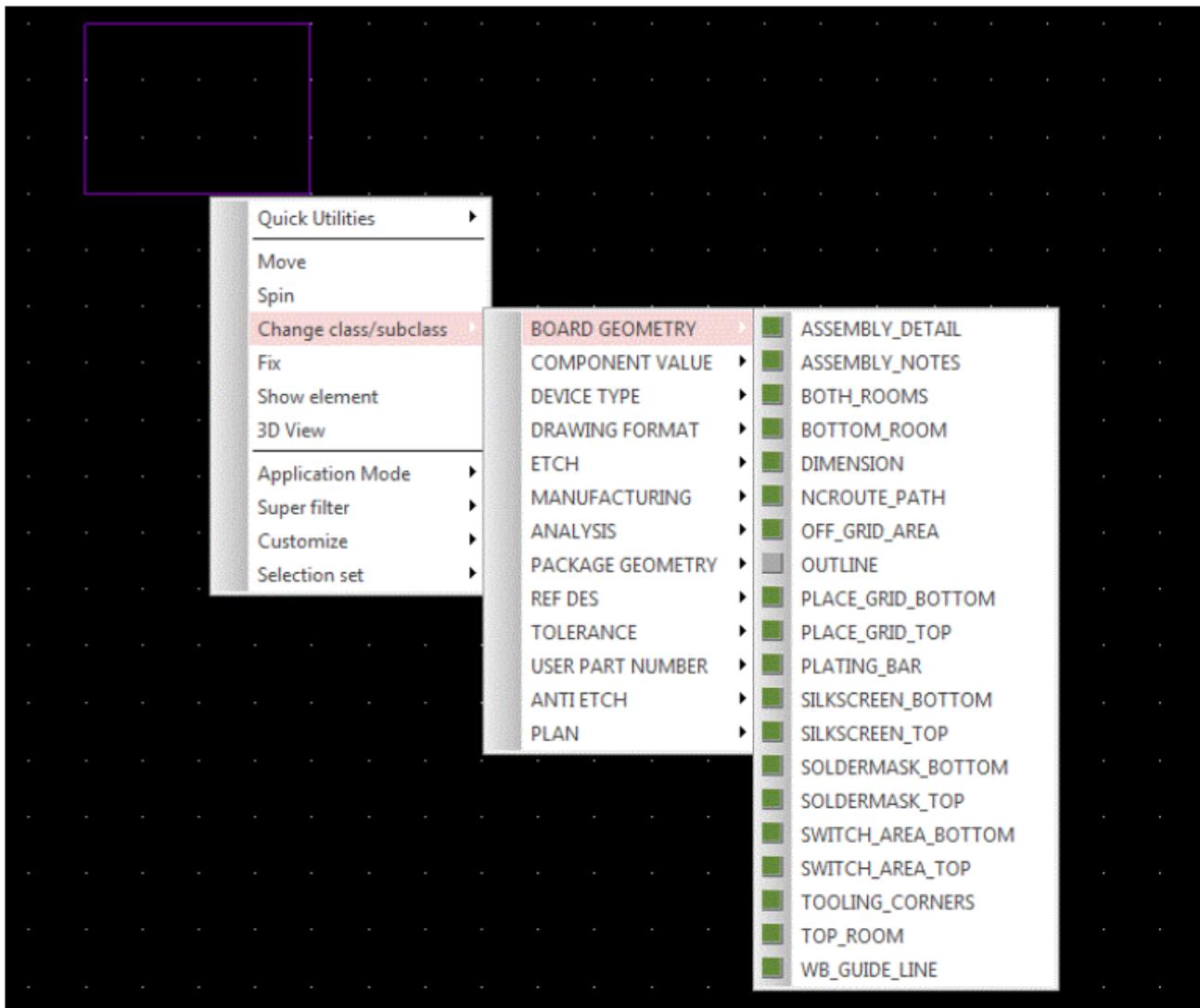
Moving Elements to Other Classes

You can move graphic elements in a design to other class and subclass by using *Change class/subclass* command. Valid elements are:

- Lines
- Line Segments
- Text
- Rectangles

Select the line or the text element or the rectangle and right-click to choose *Change class/subclass* command. Choose a new class and subclass from the list appears. You can select multiple elements with a single pick, window drag, Select by Polygon, and Temp Group selection modes.





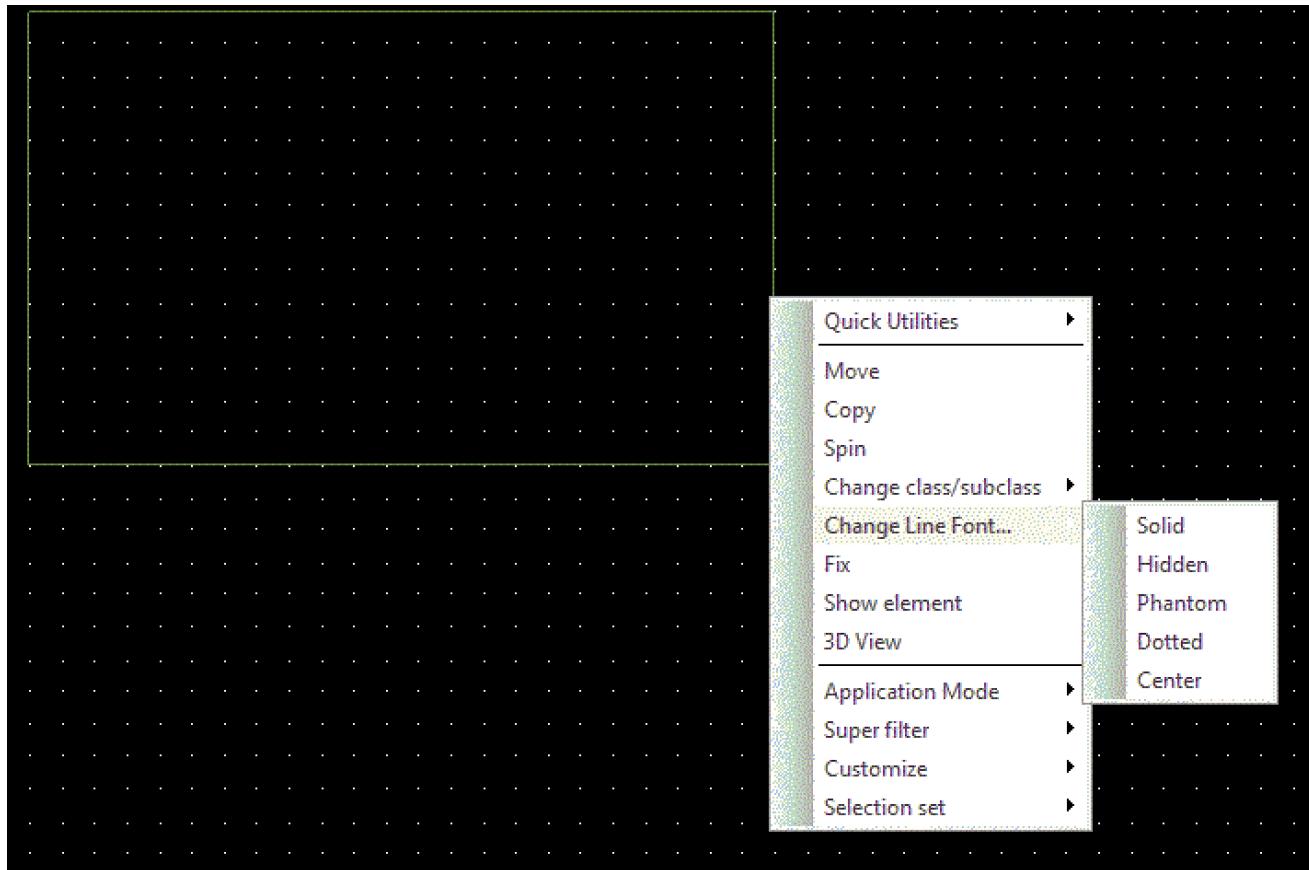
Changing Line Fonts of Elements

You can change the line pattern used in creating the graphic elements in a design by using *Change Line Font* command. Valid elements are:

- Lines
- Line Segments
- Arcs

- Circles
- Rectangles

Select the element and right-click to choose *Change Line Font* command from the pop-up menu. Choose a font from the list that appears. You can select multiple elements with a single pick, window drag, Select by Polygon, and Temp Group selection modes.



Line fonts, other than Solid, are allowed on the following Class/Subclasses for zero width segments:

- DRAWING FORMAT/All user defined subclasses
- MANUFACTURING/NCDRILL_LEGEND
- MANUFACTURING/All user defined subclasses
- PACKAGE GEOMETRY/ASSEMBLY_TOP
- PACKAGE GEOMETRY/ASSEMBLY_BOTTOM
- PACKAGE GEOMETRY/All user defined subclasses

- BOARD GEOMETRY/OUTLINE
- BOARD GEOMETRY/ASSEMBLY_NOTES
- BOARD GEOMETRY/ DIMENSIONS
- BOARD GEOMETRY/ASSEMBLY_DETAIL
- BOARD GEOMETRY/All user defined subclasses

The *Change Line Font* command is not available for the elements that are placed on the other Classes/Subclasses.

Also, if you change the font for the element with non-zero width following message is displayed in the Allegro command window:

E- (SPMHGE-271): Line fonts other than SOLID must remain at 0 width.

Deleting Graphic Elements from a Design

You can delete graphic elements from a design by choosing the *Edit – Delete* (*delete* command). You can also delete elements connected to the picked elements.

Creating Mirror Images of Graphic Elements with the Standard Mirror Option

You can change the layer of symbols and text from TOP to BOTTOM or BOTTOM to TOP. When you choose *Edit – Mirror* (*mirror* command) and *Standard Mirror* on the *Options* tab (or *Place – Mirror* in Allegro SI), you can create mirror images of elements and display them dynamically, similar to using *Edit – Move* (*move* command), so you can easily place them in their new orientation.

The geometry on subclass TOP (or that has a suffix of TOP) changes to subclass BOTTOM (or the subclass that has the BOTTOM suffix).

For example, SILKSCREEN_TOP exchanges values with SILKSCREEN_BOTTOM. The padstacks for each pin and via exchange values and now match etch or conductor subclasses in reverse order. If there is an odd number of layers, the middle layer exchanges values with itself. Any etch or conductor that was built with a package/part symbol exchanges values the same way.

With regard to the cross-section, pins, and vias, subclass names are not the determining factor for mirroring data. Data moves on any layer to the corresponding opposite layer based on its relative position in the cross-section. For example, the second layer from the top moves to second layer up from the bottom, regardless of the conductor layer names.

When mirroring, a swap occurs between user-defined NON-ETCH subclasses whose names are

suffixed with _TOP and that contain a matching subclass whose name is suffixed with _BOTTOM. If no match exists, the tool creates the matching subclass.

Besides mirroring the geometry of all elements in the symbols you choose, the tool exchanges certain subclasses of ETCH/CONDUCTOR and PACKAGE GEOMETRY, as if you had turned the element upside down, and placed it on the opposite side of the drawing.

The tool erases and displays the symbol as a duplicate image, mirrored about the origin of the component (or the origin of the group, if you use *Group*); you can move the mirrored symbol to a new location. If it is a package symbol with connected ratsnest lines, the ratsnest lines become dynamic rubberbands.

When you pick a destination point for the symbol, the tool redraws it, mirrored, at the new location. The tool creates mirror images of all etch/conductor associated with the symbol and all pin and via padstacks associated with the symbol: their TOP pads exchange with BOTTOM pads.

For procedures on creating mirror images of elements in a design, see *Edit – Mirror* (`mirror` command) in the *Allegro PCB and Package Physical Layout Command Reference*.

Mirroring Subclasses

Subclasses are exchanged as shown below.

Mirrored Subclasses

Class	Subclass	Subclass
PACKAGE GEOMETRY	XXX-TOP XXX-TOP	XXX-BOTTOM XXX-BOTTOM
REFDES	XXX-TOP	XXX-BOTTOM
ETCH/CONDUCTOR	XXX-TOP	XXX-BOTTOM

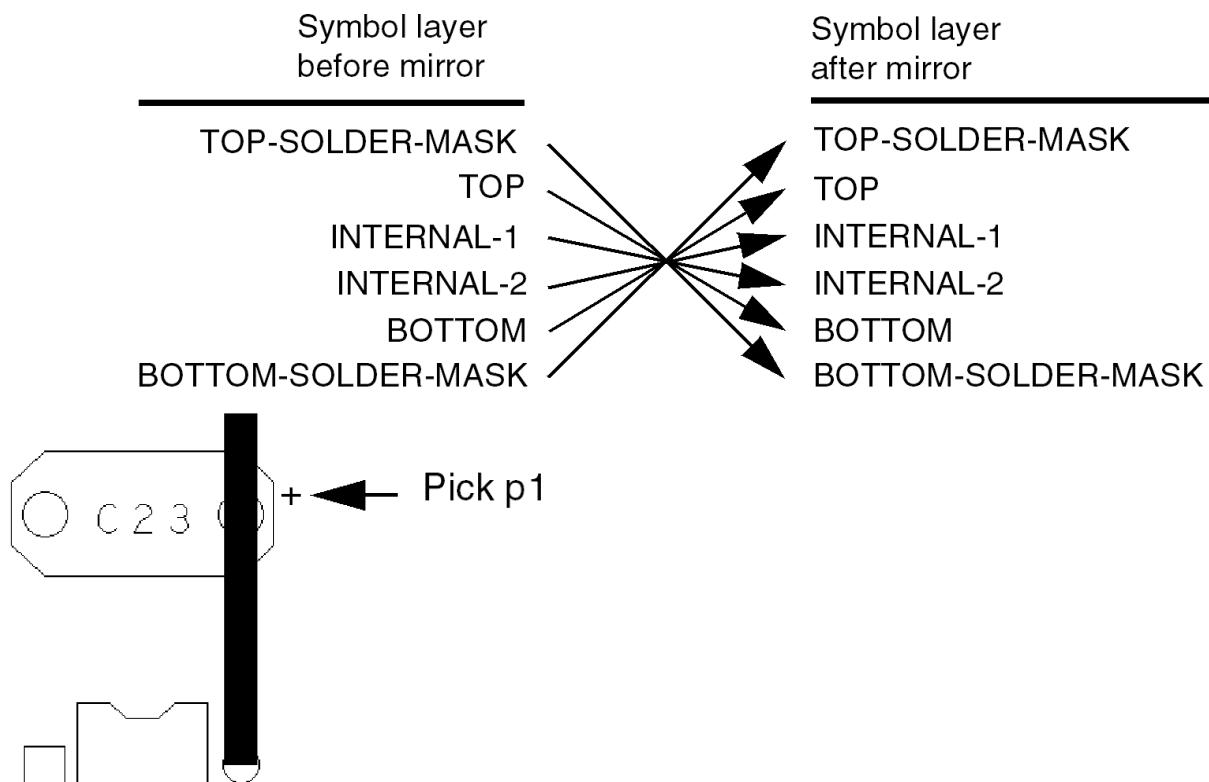
 Any user-defined subclass with the keywords TOP and BOTTOM is also mirrored.

Mirroring Symbol Etch/Conductor and Pads

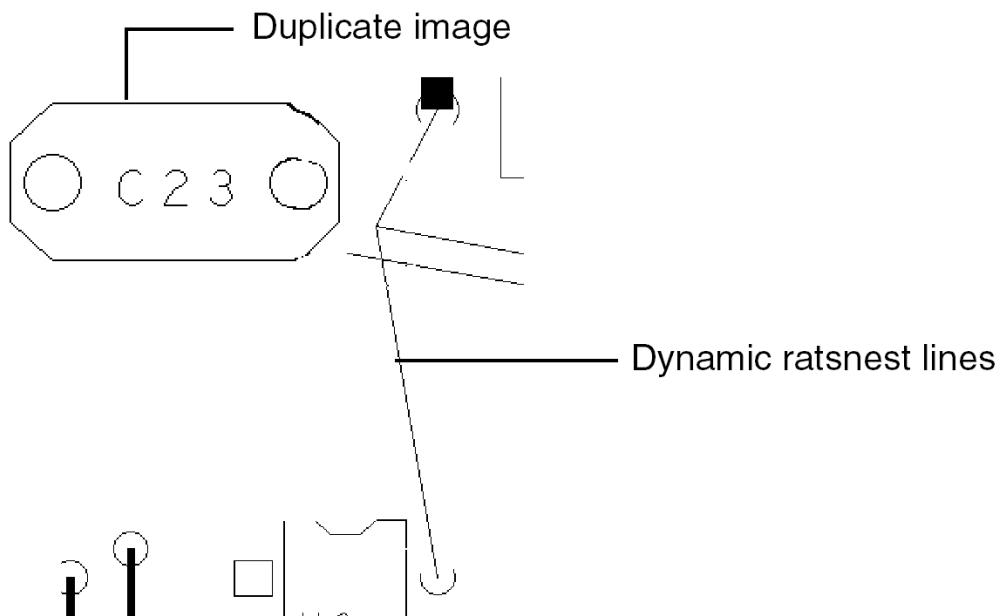
You can create symbols with predefined etch/conductor (connect lines, rectangles, shapes, and so on) and vias. The tool analyzes which etch/conductor lines and vias belong to a symbol and, in many cases, it mirrors all predefined etch/conductor and all pin and via pads when you mirror a symbol. It does not mirror shapes.

The etch/conductor via mirroring process works as follows:

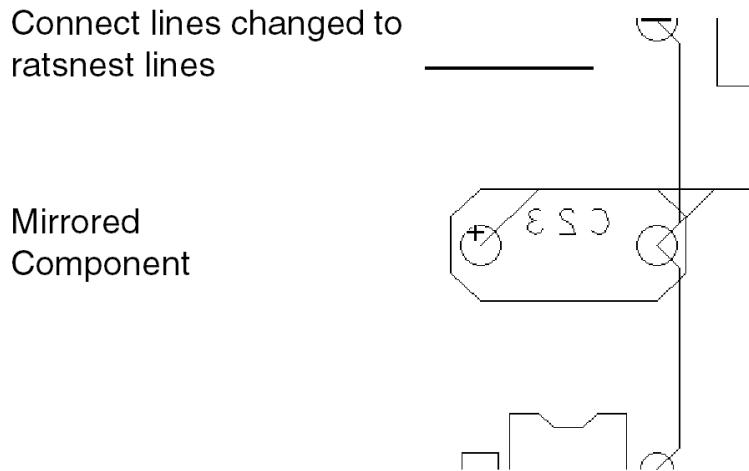
1. The geometry of each etch element and pad shape displays mirrored.
2. The ETCH/CONDUCTOR layer of each etch/conductor element and pad of each pin and via "pivots" from TOP to BOTTOM.
The TOP layer exchanges with BOTTOM and the next layer from the TOP exchanges with the next layer from the BOTTOM, until mirroring is complete.
3. If the design has an odd number of ETCH/CONDUCTOR layers, the middle layer exchanges with itself.



The next illustration shows the display after choosing the component. It displays as a duplicate image, mirrored in x about its origin, with the cursor attached to its origin. The connect lines attached to it have been deleted and replaced by dynamic ratsnest lines.



The next illustration shows the display after you picked the destination for the component. The connect lines have been deleted and replaced by ratsnest lines.



Using Different Pad Sizes on Different Layers of a Mirrored Symbol

When the symbol is mirrored and you edit padstacks, the padstacks may not appear on the correct layers unless you create a new padstack that is a mirrored version of your original padstack. Consider a four-layer design and a padstack for 60cir36d.pad defined as follows, for example:

Layer	Layer Name	Padstack Definition for 60cir36d.pad
-------	------------	--------------------------------------

1	Top	TOP PAD Circle 60
2	VCC	VCC PAD Circle 75
3	GND	GND PAD Circle 60
4	Bottom	BOT PAD Circle 60

The padstack editor counts the layers from the side of the design where the symbol is placed (either TOP or BOTTOM). In this case, the second layer for a mirrored component is the second layer from the BOTTOM, which is the GND layer, rather than the VCC layer. When you require different pad sizes on different layers of a mirrored symbol, consider the following example:

Unmirrored Pad: Padstack 60cir36d.pad	Mirrored Pad: Padstack M60cir36d.pad
TOP PAD Circle 60	TOP PAD Circle 60
VCC PAD Circle 75	VCC PAD Circle 60
GND PAD Circle 60	GND PAD Circle 75
BOT PAD Circle 60	BOT PAD Circle 60

Replacing padstacks on mirrored components with the M60cir36d.pad results in 75 mil pads on the VCC layer, given the inversion of the defined mirrored pad (due to mirroring the component). The 75 mil pad ends up one layer from the TOP on the VCC layer.

Mirroring Elements on the Same Subclass with the Mirror Geometry Option

Elements such as die symbols or those that are part of symbols may also be mirrored on the same subclass around the Y-coordinate of the copy origin, when you choose *Edit – Mirror* (`mirror` command) and *Mirror Geometry* on the *Options* tab.

When creating mirror images of elements for the same subclass, the tool mirrors elements around the Y coordinate of the copy origin. If you rotate the mirrored elements, the Y axis for the chosen group is also rotated with respect to the chosen elements. The following elements mirror:

- connect line and line segment vertices to new locations
- rectangle corners
- text

- shapes

Vias move to a new location, but their padstacks are not mirrored. Likewise, the origin of figures moves, but the figures are not mirrored.

You can also mirror elements on the same subclass using *Edit – Copy* (`copy` command), choosing *Rectangular* mode on *Options* tab, and right mouse clicking to use the *Mirror Geometry* command on the popup menu that displays. All graphic elements in a design eligible for copying can be mirrored onto the same subclass, except symbols and stand-alone pins. However, if you choose a symbol or stand-alone pin (or if a symbol or stand-alone pin is included in group or window of elements), the `mirror geometry` command is not available on the pop-up menu.

Editing Vertices

You can move and alter vertices on shapes, rectangles, filled rectangles, and line and arc segments.

Editing Shapes

You edit shapes using *Shape – Select Shape or Void* (`shape select` command). If you apply the `shape select` command to a filled rectangle, the filled rectangle changes to a shape. You cannot reverse this change.

You can then add and delete polygonal and circular voids inside existing shapes as required by using the following menu choices or commands:

Menu Choice	Console Window Command
<i>Shape – Manual Void – Add Polygon</i>	<code>shape void polygon</code>
<i>Shape – Manual Void – Add Circle</i>	<code>shape void circle</code>
<i>Shape – Manual Void – Add Rectangle</i>	<code>shape void rectangle</code>
<i>Shape – Manual Void – Element</i>	<code>shape void element</code>
<i>Shape – Manual Void – Move</i>	<code>shape void move</code>
<i>Shape – Manual Void – Delete</i>	<code>shape void delete</code>

Editing Properties

When you choose *Edit – Properties* (`property edit` command), you can identify and change the values of properties in a design. Use the Find Filter to choose the specified element. After you specify a category of elements, the tool displays the Edit Property dialog box.

Related Topics

- [copy](#)
- [assign refdes](#)
- [Copying Elements in Rectangular Patterns](#)
- [add pin](#)
- [change](#)
- [delete](#)
- [mirror](#)
- [move](#)
- [shape select](#)
- [shape void polygon](#)
- [shape void circle](#)
- [shape void rectangle](#)
- [shape void element](#)
- [shape void move](#)
- [shape void delete](#)
- [property edit](#)

Composing Shapes

When you choose *Shape – Compose Shape* (`compose shape` command), you can build complex shapes using arcs and lines. You compose shapes on a DRAWING FORMAT. Cadence recommends that you compose such shapes on a user-defined subclass; for example, a CONSTRUCTION subclass, using the following procedure:

1. Choose *Setup – Subclasses* (`define subclass` command) to display the Define Subclass dialog box.
2. Click on DRAWING FORMAT to display the Define Non-Conductor (or Define Non-Etch) Subclass dialog box.
3. Enter "USER-DEFINED" in the New Subclass field and press Enter.
4. Click *OK* in the Define Subclass dialog box.

⚠ While it is recommended that you create shapes on a layer that you create and define, the program creates a shape when data from any CLASS/SUBCLASS is chosen for compose shape.

To set up shape parameters, choose *Setup – Design Parameters* (`prmed` command) to access the Design Parameter Editor or right mouse button click whenever you are working in an application mode, then click the *Shapes* tab to edit global dynamic shape parameters, static shape parameters and split plane parameters.

Using a drawing tool on the subclass layer, you create the basic shape that you want. Then you pick the lines and arc that constitute the shape. Choose *Shape – Compose Shape* (`compose shape` command) to connect the ends of the lines (or trims crossing lines) to create a single solid shape. You can also create voids in the solid shape where they are needed using the same procedure you use to create the solid shape.

Related Topics

- [compose shape](#)
- [define subclass](#)
- [prmed](#)
- [Composing a Shape](#)

Decomposing Shapes

To decompose a previously composed shape, choose the shape and choose *Shape – Decompose Shape* (`decompose shape` command). Line and arc segments remain trimmed, chamfered, or

rounded, but each segment is detached from each other. Then you can modify the shape and reconnect the segments with the *Shape – Compose Shape* ([compose shape](#) command). See *Shape – Decompose Shape* ([decompose shape](#) command) in the *Allegro PCB and Package Physical Layout Command Reference*.

Cutting and Pasting Design Elements

You can copy and paste elements between designs or symbol drawings by using commands. The tool stores the elements that you copy in a clipboard (.clp) file. You can create a library of clipboard files, each containing frequently used or unique elements for future use. For details, see the *Defining and Developing Libraries* user guide in your documentation.

You can copy and paste elements from:

- Design to clipboard to design
- Design to clipboard to symbol drawing
- Symbol drawing to clipboard to design
- Symbol drawing to clipboard to symbol drawing

As an example of using a clipboard library, if you are creating multilayer ceramic modules, an element library lets you organize a design into a hierarchy of elements. You could create the following hierarchy, storing each element in a separate clipboard file:

1st Level:	Store a single rectangle as a pad
2nd Level:	Group multiple pads together to form a chip site
3rd Level:	Add pin escapes to the chip site to create a cell
4th Level:	Group several cells to form a larger cell

You can continue to build this element hierarchy until the entire module design is complete. At any level in the hierarchy, you can use the Find Filter (*Find by Property*) to identify all of the pieces that form that level. For example, you can show each chip site in a cell.

When you copy elements that contain user-defined properties into a clipboard, be sure that the properties are also defined in the destination drawing before pasting the elements into it. The tool pastes the object but does not attach user-defined properties that have not already been defined. When user units of elements in a clipboard file and a destination drawing are different, the tool converts the clipboard elements into the destination units. If the conversion might cause a loss of accuracy, the tool writes a message to the log file and continues processing. If the elements copied into a clipboard file have properties attached, the properties and their original values are transferred with the elements when the clipboard file is pasted into a drawing (if the property definition exists in the new drawing).

When you paste elements in a design, they retain the class and subclass information from the original design from which they were copied.

If elements have class and subclass information that does not exist in the destination drawing, the tool does not paste those specific elements, but writes a message to the log file and continues processing.

The tool changes assigned package symbols to unassigned package symbols and uses the default reference designator (refdes) of the symbol definition and places it at the refdes label location for the symbol. Similarly, any device type, component value, tolerance, and user part numbers assume the default values from the symbol definition.

When you paste elements, the tool can change element types. For example:

- The tool pastes lines on etch/conductor as connect lines.
- Symbol var pins are pasted as standalone pins when copied from designs and pasted to symbol drawings.
- When the tool changes elements types, properties might not be transferred to the destination design. The tool pastes the element, writes a message to the log file, and continues processing.

The net name associated with any element copied into a clipboard file is *not* transferred to the new design when the file is pasted.

You may encounter space limitation problems when attempting to paste clipboard files into your design. If you receive an error message saying there is no more space in the database, you may need to increase the amount of swap space on your system. Perform the following calculation to determine the amount of space the tool is using. Then increase swap space accordingly. All values are in megabytes.

40 + (design size x 2) + (clipboard file x 4) + baseline

where "baseline" is all other applications running before the tool is started (typically ::100 MB)

 The above may not apply if you are running HPUX with limit enabled.

Backward Compatibility

Clipboard files created in newer versions of the tool might not work with previous versions of Allegro X PCB Editor.

For instructions, see the *File – Export –Sub – Drawing* ([clpcopy](#) command) and *File – Import – Sub – Drawing* ([clppaste](#) command) in the *Allegro PCB and Package Physical Layout Command Reference*. Also included in the description of the `clppaste` command are notes about different types of design elements.

Keepin and Keepout Areas

The layout tool lets you create shapes that act as boundaries. Each shape (or area constraint) is referred to as a keepin or a keepout. The tool classifies area constraints into the following types:

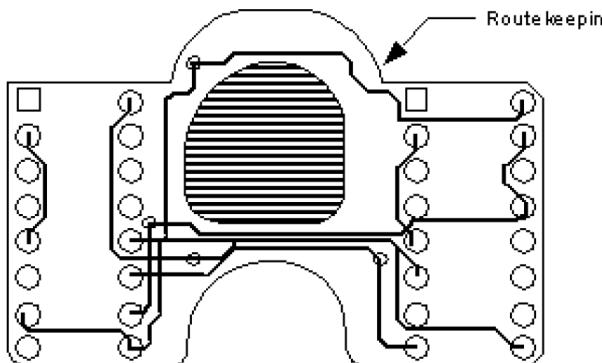
- Artwork keepin
- Gloss keepout
- PCB Editor: Package keepin
- PCB Editor: Package keepout
- PCB Editor: Probe keepout
- Route keepin
- Route keepout
- Wire keepout
- Via keepout
- APD Component keepin
- APD Component keepout
- PCB Editor: Shape keepout

You use the *Setup – Areas* (default menu selection) commands to define constraint areas. You define areas with a series of mouse clicks that connect line segments.

Route Keepins

A route keepin is an unfilled polygon that defines the area where etch/conductor is permitted in a layout. A layout can have only one route keepin. Etch/conductor is placed within DRC limits of the keepin. The automatic router ignores any connect points outside the route keepin. When running checks for route keepin, only etch elements are considered. Pins are excluded and do not generate DRCs. Connections may touch, but may not cross, the route keepin. The routers recognize a keepin of any shape and attempt to maximize the use of the space within the keepin.

Route Keepin



Route Keepouts

Route keepouts are filled shapes that you create to indicate areas of a design that may not contain etch/conductor objects. Etch/conductor may touch, but not enter a keepout area. The automatic router does not add etch/conductor inside a route keepout. You may add as many route keepouts to the design as you require. You can also add route keepouts to package and mechanical symbol drawings. Then when you add those symbols to a drawing, they bring their own route keepouts, defined relative to their own position, rotation, and mirroring.

Wire Keepouts

Wire keepouts are filled shapes that you create to indicate areas of a design that may not contain etch/conductor objects. Etch/conductor may touch, but not enter a keepout area. The automatic router does not add etch/conductor inside a route keepout. Wire keepouts also have the VIAS_ALLOWED property attached to them. This means that vias can drill through the shape, although the tool prevents the wires from routing through the shape. You may add as many wire keepouts to the design as you require.

Via Keepouts

Via keepouts are filled shapes that define where vias may not exist. Vias may touch, but not enter a keepout space. The automatic router does not add vias inside a via keepout. You may add as many via keepouts to a design as you require. You can also add via keepouts to package and mechanical symbol drawings. Then when you add those symbols to a drawing, they bring their own via keepouts, defined relative to their own position, rotation, and mirroring.

PCB Editor: Shape Keepouts

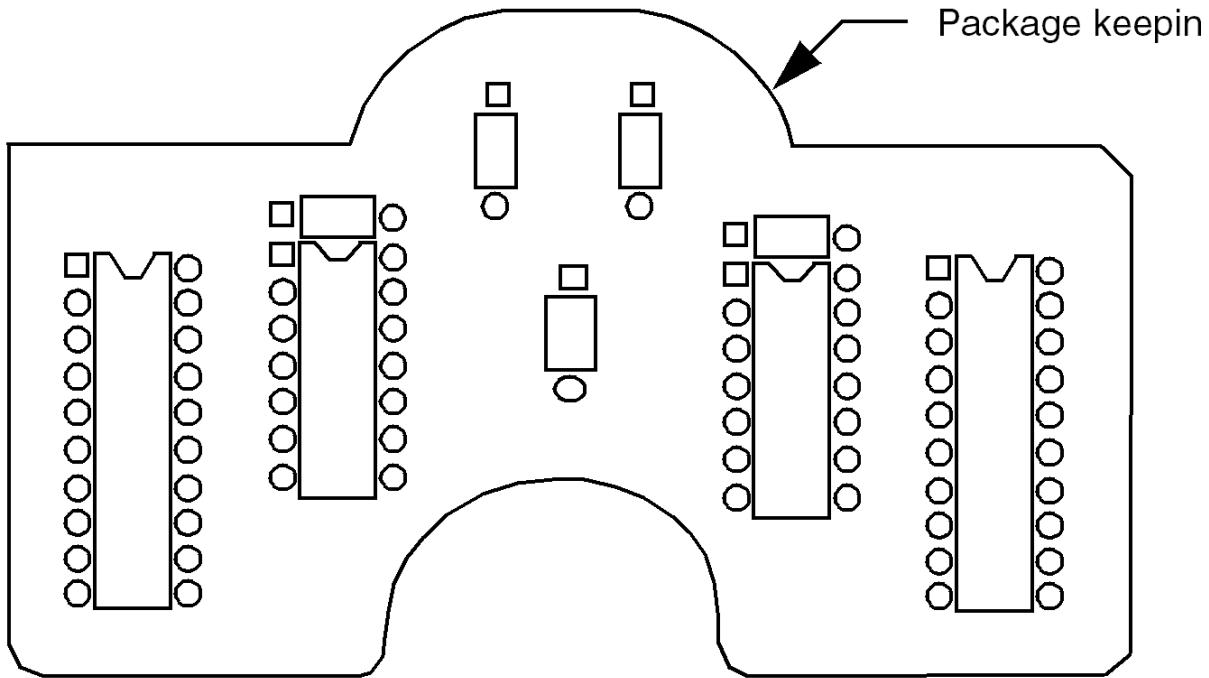
Shape keepouts are filled shape areas that you create to indicate portions of a design that may not contain conductive shapes. Shape keepouts have the VIAS_ALLOWED and ROUTES_ALLOWED properties attached to them. This means that vias can drill through the shape, and routing is permitted through it. You may add as many shape keepouts to the design as you require. You can also add shape keepouts to mechanical symbol drawings. Then when you add those symbols to a drawing, they bring their own shape keepouts, defined relative to their own position, rotation, and mirroring.

PCB Editor: Package Keepins

A package keepin is an unfilled polygon that defines the area where package symbols may be placed in a layout. The layouts permit one and only one package keepin. The area of a package symbol may touch but not cross the package keepin.

The tool's automatic placement recognizes a keepin of any shape and attempts to maximize use of the space within the keepin.

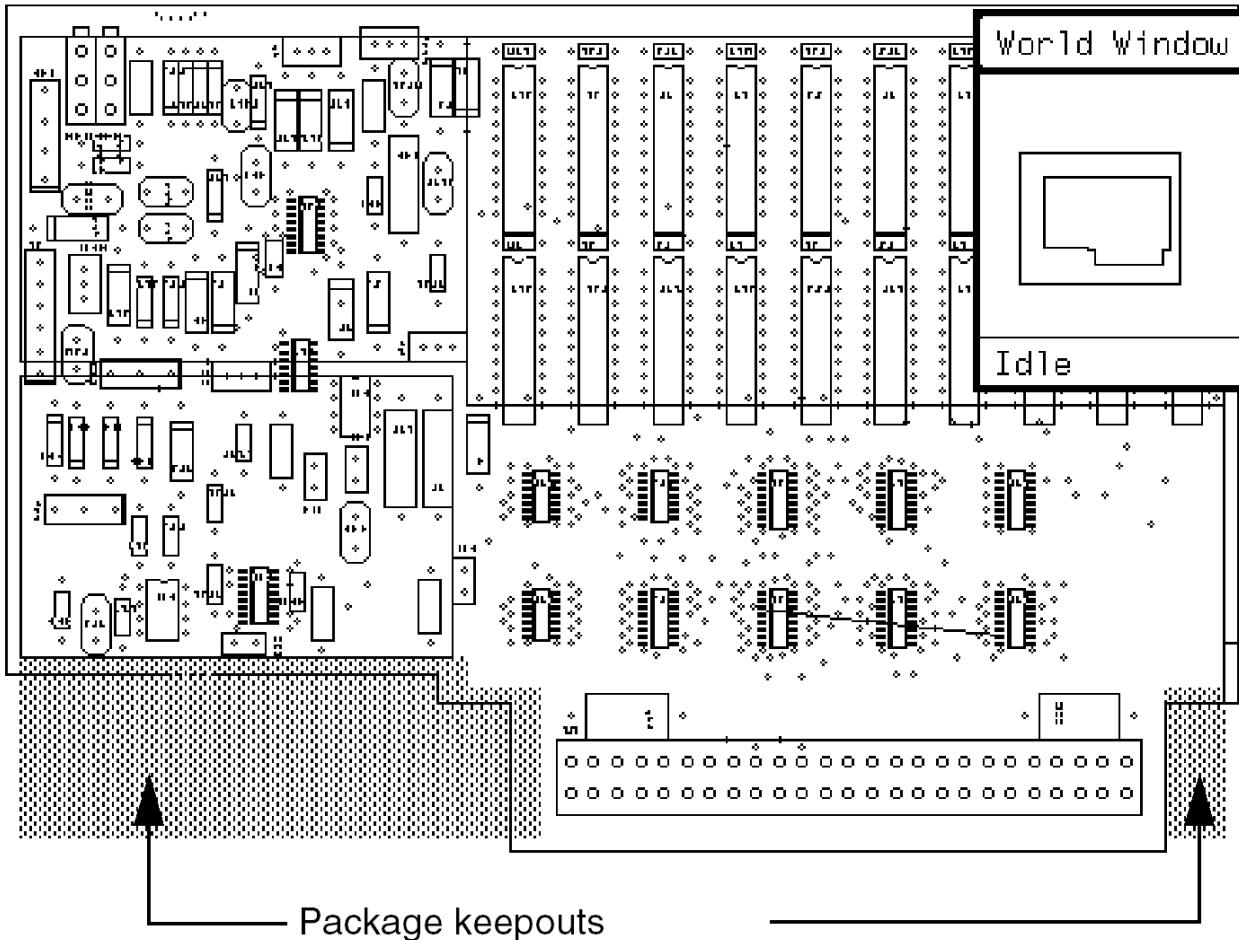
Package Keepin



PCB Editor: Package Keepouts

Package keepouts are filled shapes that you create to indicate areas of a design where you do not want package symbols. Automatic placement does not place package symbols in these areas.

PCB Editor: Package Keepout



Shapes in class PACKAGE KEEPOUT subclass ALL keep packages out of both TOP and BOTTOM.

You can add as many package keepouts to the design as you require. You can also add package keepouts to mechanical symbol drawings. When you add those mechanical symbols to a drawing, they bring their own package keepouts, defined relative to their own position, rotation, and mirroring.

APD: Component Keepin

A component keepin is an unfilled polygon that defines the area where component symbols may be placed in a design. The tool permits only one component keepin. The area of a component symbol may touch but not cross the component keepin.

The tool automatic placement recognizes a keepin of any shape and attempts to maximize use of the space within the keepin.

Shapes in class COMPONENT KEEPOUT, subclass ALL keep packages out of both TOP and BOTTOM, and SURFACE and BASE.

You can add as many component keepouts to the design as you require. You can also add component keepouts to mechanical symbol drawings. When you add those mechanical symbols to a drawing, they bring their own component keepouts, defined relative to their own position, rotation, and mirroring.

PCB Editor: Probe Keepouts

A probe keepout is a filled shape that defines an area in which test probes are restricted.

Gloss Keepouts

A gloss keepout is a filled shape that defines an area in which glossing is restricted.

Artwork Keepins

An artwork keepin is an unfilled shape used to contain artwork.

The `keepin photo` command lets you define the area of a drawing that you want on the photoplot film. When you choose *Setup – Areas – Photoplot Outline* (`keepin photo`), The tool places you in class: MANUFACTURING. This option specifically lets you create photoplot windows so the default subclass is PHOTOPLOT_OUTLINE. You can only have one photoplot window defined at a time. If you choose a second, it overwrites the first. This option is typically used at the manufacturing stage of design.

Drawing an Area Constraint Using Line Segments

You draw shapes for Route, Package, Via, and Probe area constraints by connecting a series of line segments.

Setting Height Restrictions in Package Keepout Areas

You can attach properties defining a height restriction to a package keepout, allowing package symbols whose height is below a minimum or above a maximum to be placed in that area. For example, if the height range of a keepout is 500 to infinity, package symbols whose height is less than or equal to 500 can be placed in it.

Related Topics

- [Using ETCH/CONDUCTOR Shapes in Embedded Planes](#)
- [keepin router](#)
- [Creating Keepin and Keepout Areas](#)
- [keepout package](#)
- [keepout via](#)
- [keepout shape](#)
- [keepin package](#)
- [keepout probe](#)
- [keepout gloss](#)
- [keepin photo](#)
- [Drawing an Area Using Line Segments](#)
- [Adding Height Specifications to Component Keepout Areas](#)

Layout Padstacks, Vias, and Etch/Conductor Shapes

During layout preparation, you can:

- Edit layout padstacks
- Edit layout pad shapes
- Create through-hole or blind/buried vias interactively
- Create and edit dynamic or static etch/conductor shapes
- Generate via arrays

The tool differentiates padstacks into two categories, *library* padstacks and *layout* padstacks: A library padstack is a padstack definition contained in the tool's library or a user-defined library directory. A layout padstack is a padstack definition associated with a pin or via in a design. See the *Defining and Developing Libraries* user guide in your documentation set for a description of library padstack functions.

 The functionality this chapter describes may not be available in all versions of Allegro X PCB Editor.

Editing Layout Padstacks with the Padstack Designer

Once a design contains padstack definitions, those padstacks are considered layout padstacks. Choose *Tools – Padstack – Modify Design Padstack* (`pateditdb` command) to define different sets of pad data for internal layers in layout padstacks. Pins share layout padstack definitions—all pins with the same padstack name refer to the same padstack definition stored in the layout.

Use the `pateditdb` command to display Padstack Editor, which lets you:

- Modify or copy the padstack definitions
- Modify or copy padstack instances
- Purge padstacks

Refer to the *Allegro PCB and Package Physical Layout Command Reference* for specific information on *Tools – Padstack – Modify Design Padstack* (`pateditdb` command) and associated procedures.

- Ensure that the custom pad shape has been previously created. Define a pad for at least one layer if you want to save a padstack file. Also, you can save the padstack without completing every field.
- Do not specify null pad definitions on internal layers of through-hole padstacks.
- Define a pad size for every internal layer so that DRC checks line-to-pad spacing. It is the pad definition that alerts DRC. If there are no pad definitions on internal layers, the tool does not perform design rule checking on line-to-pad spacing. This may cause a situation where etch/conductor can be placed in the path of a drill hole without being flagged by DRC.
- Define the pad size smaller than the drill hole to drill the pad out during manufacturing and DRC verifies the spacing. As long as a pad is defined, DRC uses the larger of the drill hole or pad to check spacing.
- Ensure the overall array of the drill holes fits within all of the individual pads for multiple drill padstacks.

Related Topics

- [pateditdb](#)
- [Padstack Editor](#)
- [Custom Pad Shape Symbols](#)

Editing Layout Pad Shapes

When you edit layout pad shapes, you can:

- Trim or enlarge a pad shape
- Edit a pad shape
- Display derived padstack names
- Restore a pad to its original pad shape

Some designs may have component or via pads that need to be either enlarged or trimmed. For example, you might need to trim a portion of a single pad, perhaps due to the density of the design or for packaging reasons.

To enlarge or trim pads on placed components, the tool lets you do the following:

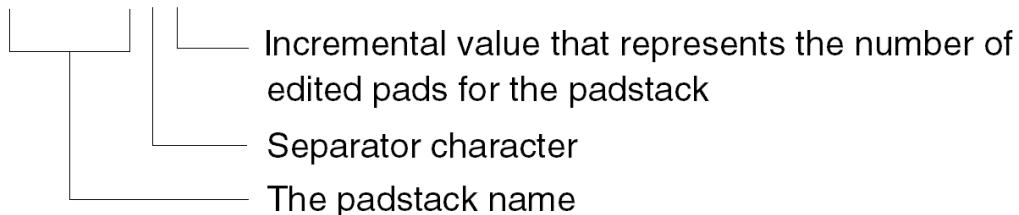
- Change the geometry for a pad, yet still maintain permanent association between the pad and the package symbol
- Restore edited pad shapes to their original state if you do not want to keep the edited pads (for example, if constraints change)

You trim or enlarge pads using the same technique as that used for editing pad shapes. You can move, rotate, and mirror the symbol without losing the edited pad shape. As part of the pad shape editing process, you can set the specified grid value in the *Options* tab.

The layout tool identifies the edited pad with a name derived from the original pad and padstack name. The tool incrementally adds a numeric value to the end of the pad shape and padstack names.

For example, if you edited a pin for a padstack called SMD50, the resulting, "derived" padstack name is:

SMD50 - 1



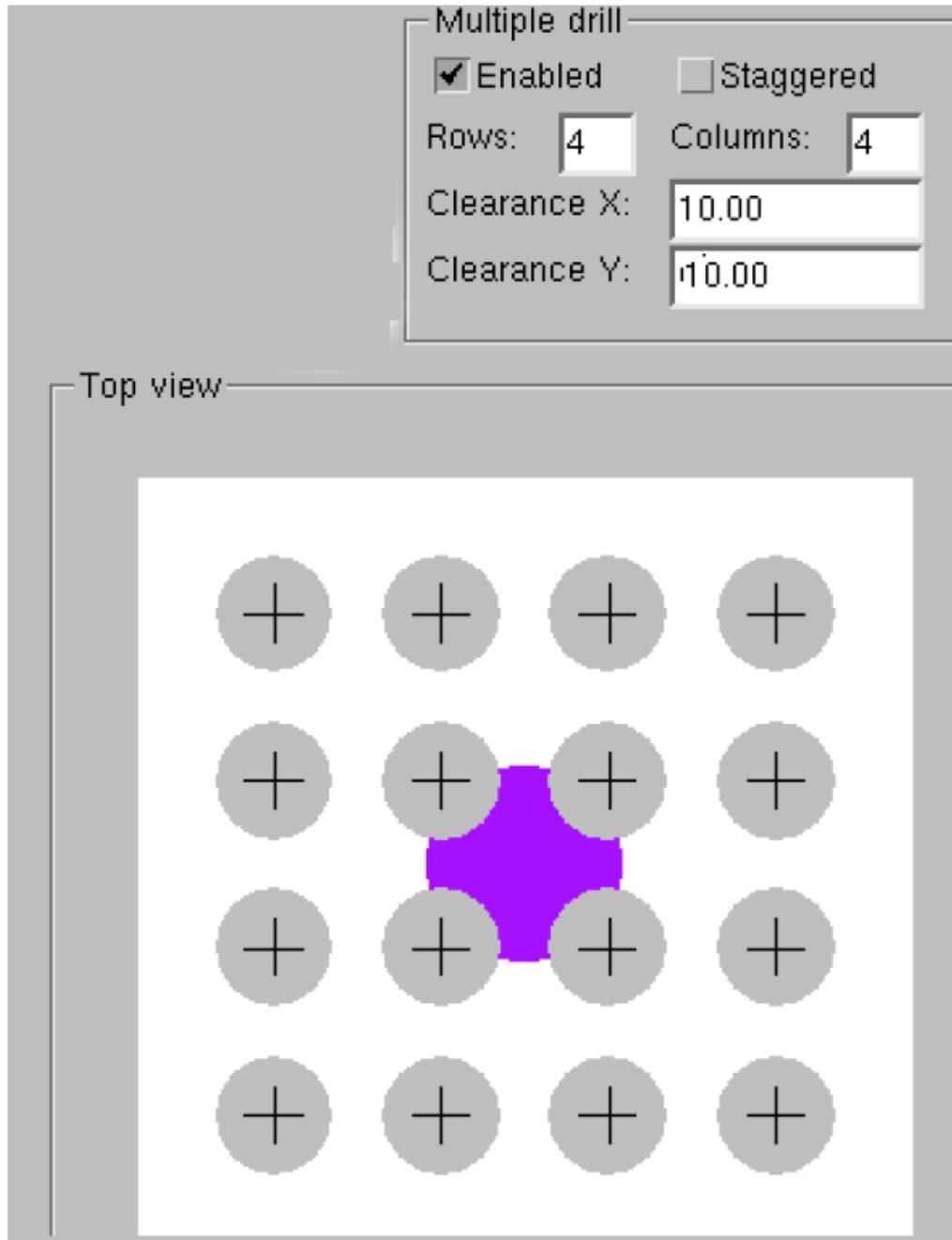
For edited padstacks, you can use the replace padstack feature to update that padstack in a design with your edited padstacks. The replace padstack feature also lets you replace single vias when you choose the single via replace option.

Creating Vias

You can create either a through-hole or a blind/buried (BBVia) via as part of a connection or as a stand-alone via.

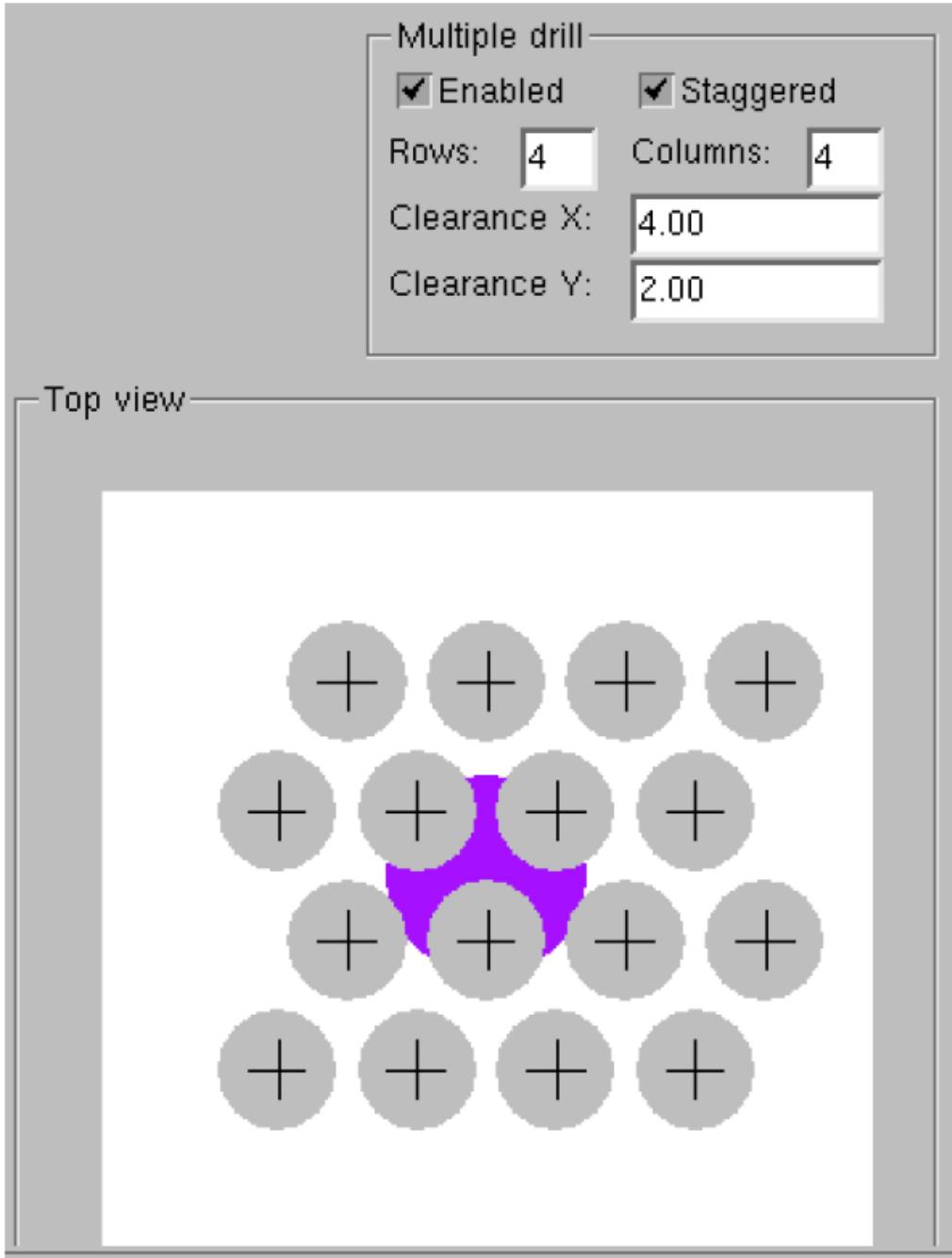
Vias in a design display their overall pad boundaries as they are configured in the Padstack Designer, as either single or multiple hole. The following figure illustrates a via configured in Padstack Designer as multiple-drilled. If the check box is disabled, the via is single-drilled.

Multiple Drill Section of Padstack Designer



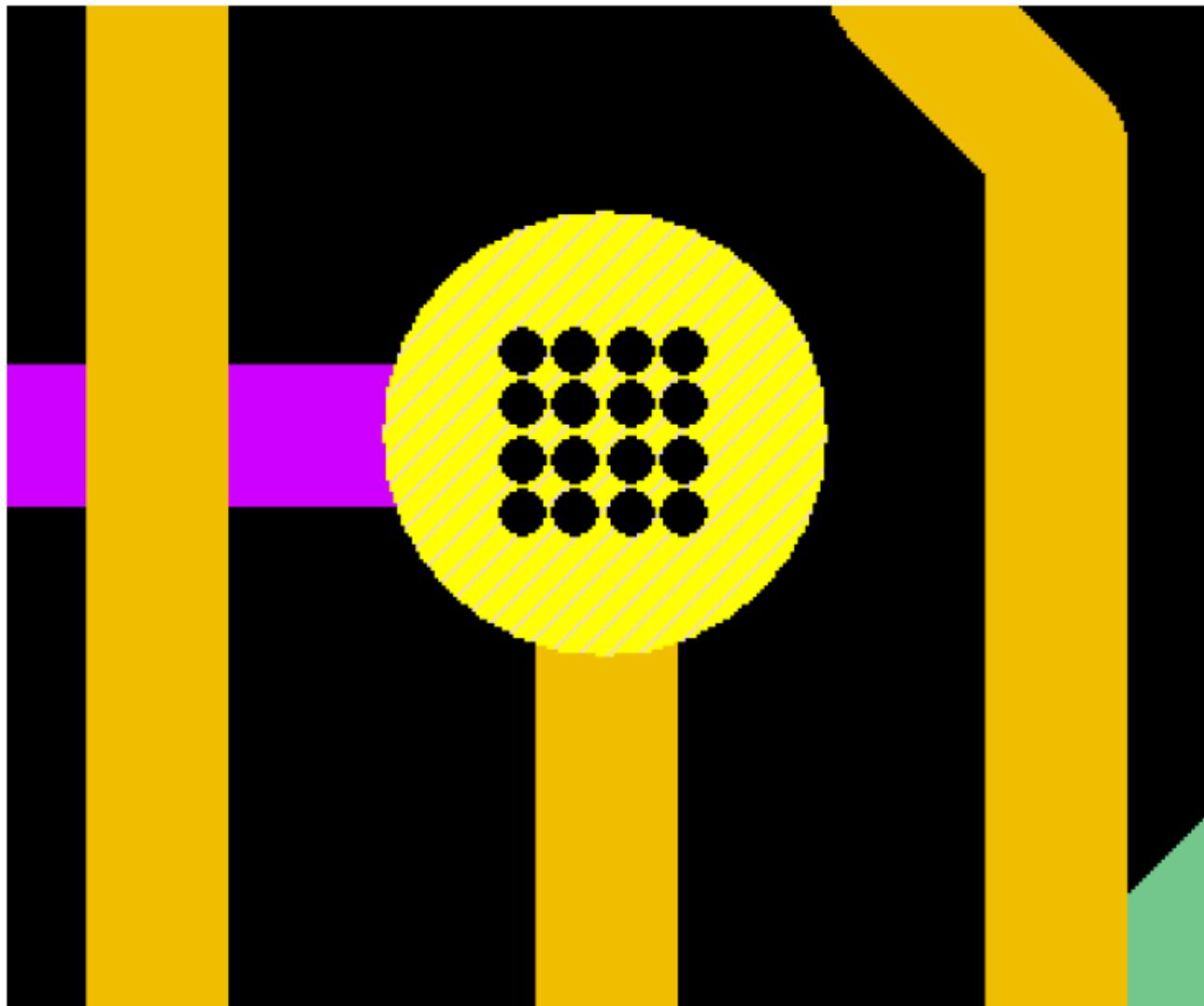
The following figure illustrates a via configured in Padstack Designer using the *Staggered* option in conjunction with separate values in the *Clearance X/Y* fields.

Staggered Option with Separate X/Y Clearance Values



If you display drill holes by choosing *Display Plated Holes* or *Display Non-Plated Holes* in the *Display* tab of the Design Parameter Editor, available by choosing *Setup – Design Parameters* (`prmed` command), they appear as single-drilled holes or in an array of rows and columns. If you choose *Filled pads* as well, the drill holes of the vias display as filled circles. The layer priority for the particular layer determines the color in which they display. The following figure illustrates the result on a multiple-drill via in a design display.

Via Display in Design



⚠ If the tool is running on Windows, it temporarily turns off the display drill hole options during plotting. The options are automatically turned on after plotting is complete.

Via Array Advantages

Using via arrays in your design provides the following advantages:

- Mitigates electromagnetic interference (EMI)
- Controls crosstalk
- Suppresses resonance
- Improves power integrity

Working with ETCH/CONDUCTOR Shapes

In the layout tools, shapes comprise bounded areas of conductor on etch/conductor layers that are solid-filled or crosshatched with conducting etch/conductor (usually copper). You use etch/conductor shapes as shielding around components, in coupons, as pads that are not one of the regular shapes (circle, rectangle, oblong), and to fill entire layers with conductor as voltage distribution (embedded) planes. You can add and edit positive shapes at any time in the design process, controlling when and how each shape's fill is updated and voided. It is suggested not to use two shapes on top of each other on the same net.

Dynamic vs. Static Shapes

A dynamic shape is one whose fill is automatically updated to execute connectivity, generate voids, and run design rule checking to produce artwork quality output. Its *Dynamic Copper Fill* mode is defined as *Smooth* on the *Global Dynamic Shape Parameters* dialog box. This means that no additional postprocessing is required on the shape. Use dynamic positive shapes as ground shielding on outer layers and as inner layer planes if current designs use positive static planes, and performance is acceptable.

Dynamic shapes can only be added to etch/conductor layers. When you add a dynamic etch/conductor shape that crosses the route keepin, or modify the route keepin boundary, the dynamic shape boundary is clipped to the route keepin by default. When you edit or move a dynamic shape that crosses a keepin, the tool does not clip the shape at the keepin by default.

To preserve any user-defined dynamic shape boundary and re-clip it to the route keepin during a dynamic shape update, enable the `shape_rki_autoclip` environment variable in the *User Preferences* dialog box, available by choosing *Setup – User Preferences* (`enved` command).

For example, mechanical engineering changes a board outline and route keepin, thereby generating a new Intermediate Data Format (IDF) file. Upon reading the IDF file into the layout editor, any dynamic shapes currently clipped by the route keepin automatically update to the new route keepin location specified by the IDF file.

If you add a dynamic shape that is completely outside the route keepin, the tool ignores the route keepin when voiding.

Use static (manual) solid or crosshatched positive shapes for critical handcrafted etch/conductor that you do not want modified automatically. Static and dynamic shapes each have a unique graphic pattern. Although drawn in the same color, the stencil pattern associated with dynamic shapes is drawn more densely than that for static.

You can define a default group of global parameters that apply to all dynamic shapes you create

when you choose *Shape – Global Dynamic Parameters* (`shape global param` command) and the Global Dynamic Shape Parameters dialog box. These default global parameter settings can be changed, and the modifications then propagate to all dynamic shapes.

Choose *Tools – Reports* (`report` command) and choose the *Dynamic Shapes* report for information on shape settings; void generation results, including number of dynamic etch/conductor shapes and their areas; shape fill type; thermal relief connects; void controls; and clearances.

Dynamic shape voiding and healing ensure ECOs can be accommodated easily, as voiding occurs on the fly. Whenever an object is placed on top of a shape on a different net, the shape is immediately voided automatically. These objects can include the placement of a component or test pad or routing a cline or via. Choose *Route – Connect* (`add connect` command), *Route – Slide* (`slide` command), and *Edit – Vertex* (`vertex` command) to add or edit etch/conductor on top of a positive shape without causing DRC errors, and any item shoved into the dynamic shape behaves as if the shape is not there, as the shape is immediately voided around the etch/conductor. Routing a cline through a shape is called plowing into the shape. When routing within a dynamic shape, voiding occurs after each cursor pick but not continuously with each cursor movement. Only after the cursor pick is the cline committed to the database.

Working with Dynamic Fill Mode

To speed performance, you can set the *Dynamic Copper Fill* mode to *Rough* on the *Global Dynamic Shape Parameters* dialog box, which lets you see connectivity without full edge smoothing and thermal hookups in a fast fill mode to obtain true clearances around objects and resolve intersections with other voids. Artwork quality results and artwork are not created.

You can work with the dynamic copper fill mode disabled and then enable and update the shapes as a batch process. You may also want to disable the dynamic updates to increase throughput when shapes require significant editing:

- Changing spacing rules
- Changing global shape parameters
- Refreshing, replacing, or editing padstacks
- Manipulating large pin count devices

Deferring Dynamic Fill

To defer dynamic filling of shapes on a global basis, selecting a *Dynamic Copper Fill* mode of *Disabled* in the *Global Dynamic Shape Parameters* dialog box allows you to edit etch/conductor for medium to large ECOs; manual ECOs; or run batch processes such as `netin`, `gloss`, and adding/replacing vias during testprep, for example, without impacting performance. However, shapes created under this global setting are not voided, nor does DRC run. They are marked out-of-date to be filled later. Artwork cannot be produced. With the dynamic copper fill mode disabled, the tool stores the original outline and all parameter settings without any information loss.

To defer dynamic filling of a particular shape, you can enable *Defer Performing Dynamic Fill* on the *Options* tab while retaining shape boundaries. After you create the initial shape boundary, you often refine it to meet the final intent. It may be advantageous to enable this option while editing the boundary to maximize efficiency. The chosen shape becomes temporarily unfilled until the option is disabled, or you right click to display the popup menu and choose *Done*.

In the SiP and APD tools, once you have finalized your copper and metal pours, you can then perform thieving and degassing as required. For additional information, see Thieving and Degassing.

Cancelling Dynamic Fill

To cancel dynamic filling of complex shapes for a large design, you can use the `Esc` key to stop the process, which leaves the shapes out of date. If several shapes are in the midst of dynamically filling when you invoke the `Esc` key:

- Shapes already dynamically filled remain completed.
- Shapes in the process of dynamically filling remain unfilled and marked out of date.
- Shapes whose dynamic fill is yet to be updated remain filled but marked out of date.

Updating Out-of-Date Dynamic Shapes

To view the status of all dynamic shapes in the tool, you can use the *Out of date Shapes* field on the *Status* tab on the *Status* dialog box to verify the current state of dynamic shapes. To generate a report showing the status of each dynamic shape in the design, click the *Out of date Shapes* color box. The report, sorted by layer, provides details as follows:

- Smooth: ready for artwork
- Out of date: update required

- No Etch/conductor: shape has no etch/conductor, possibly due to a route keepout. Delete the dynamic shape or add etch/conductor to produce artwork.

You can update dynamic shapes if they are out of date using the following methods:

- With the *Update to Smooth* button on the Status tab.
- Set the *Dynamic Copper Fill* mode on the *Global Dynamic Shape Parameters* dialog box to *Smooth* and click *Apply*

An out-of-date dynamic shape is one for which the *Dynamic Copper Fill* mode has been set to *Rough* or *Disabled* on the *Global Dynamic Shape Parameters* dialog box (non-*Smooth* *Dynamic Copper Fill* mode). Out-of-date (non-*Smooth*) dynamic shapes prevent you from running the batch commands `artwork` and *Manufacture – Stream Out* (`stream_out` command) and creating artwork when dynamic copper fill shapes are out-of-date.

When you update out-of-date dynamic shapes with the *Update to Smooth* button, the tool automatically voids and runs DRC, as if the *Dynamic Copper Fill* mode were set to *Smooth* in the *Global Dynamic Shape Parameters* dialog box.

To defer dynamically filling a particular dynamic shape instance, you can choose the *Defer performing dynamic fill* option on the *Options* tab during shape creation or editing.

To apply custom parameters to an individual dynamic shape, use the `shape param` command and the Shape Instance Parameters dialog box. If you choose an individual static shape and run the `shape param` command, the *Static Shape Parameters* dialog box appears. Custom parameter settings override the default global parameter settings on the *Global Dynamic Shape Parameters* dialog box. Alternatively, choose the shape, right click, and choose *Parameters* from the popup menu.

Copying Dynamic Shape

When copying dynamic shapes across layers the custom parameters are retained. For both, shape copy to layers and z-copy shape any override instance parameters (such as via oversize values) are retained.

Crosshatched Shapes

Crosshatched shapes' boundaries and fill patterns display at the actual width you specify for them. To crosshatch fill a shape, choose the following on the *Shape fill* tab of the *Global Dynamic Shape Parameters* dialog box:

- Crosshatch fill type, either single or double crosshatch

- Line width, spacing, and angle between the two sets of crosshatch lines
- Crosshatch line origin

The tool automatically performs design rule checking on all items in the crosshatched shape, flagging any clearance errors on the shape. Thermal-relief connections extend from the thermal relief to the centerline of a hatch segment, shape outline, or void outline.

Connect lines, pins, and vias on the same net as the shape can touch the shape without causing DRC errors. The artwork generator automatically flashes thermal-relief pads for pads of that net, and antipads for non-member pads.

Unfilled Shapes

You can also create an unfilled shape on a non-etch/conductor layer, useful for drawing objects such as card outlines and legends. You must specify a non-etch/conductor subclass before you draw the shape.

Related Topics

- [enved](#)
- [shape global param](#)
- [add connect](#)
- [slide](#)
- [vertex](#)
- [Status Dialog Box](#)
- [artwork](#)
- [stream_out Batch Command](#)
- [shape param](#)

Frozen Dynamic Shapes

Frozen shapes are supported in OrCAD X, Allegro X, Allegro, and APD.

The existing Allegro shape model supports fixed and unfixed dynamic shapes. A fixed dynamic

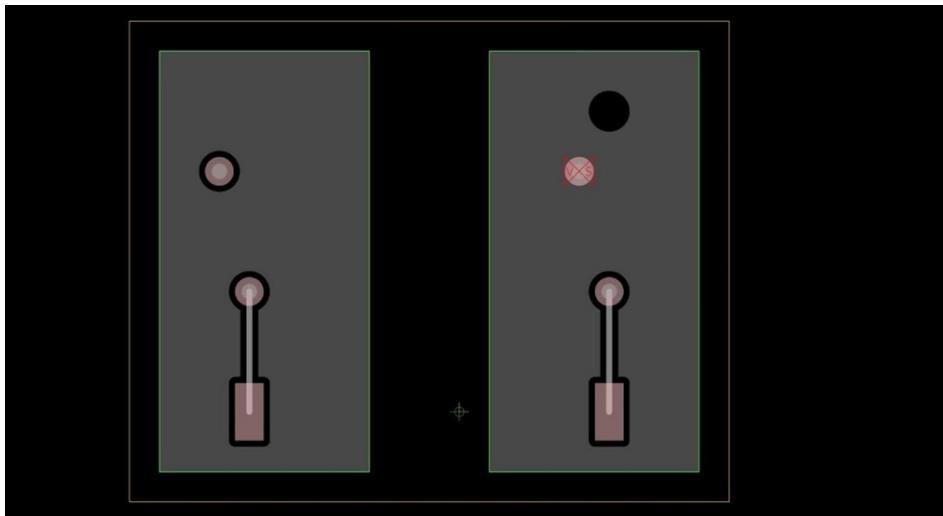
The dynamic shape will support moves and changes by dynamic shapes and dynamic voids. A dynamic shape will not allow any user-controlled behaviors (that is, a fixed shape cannot be moved, deleted, edited, or updated). The fixed shapes will respond to system-controlled behavior (that is, allow for dynamic voiding in response to design changes).

Conversely, frozen dynamic shapes enable only user-controlled behaviors and do not allow (or suspend) system-controlled behaviors. Frozen dynamic shapes offer control—and therefore can be updated only with user-driven change requests—and will be protected from any unwanted auto-voiding.

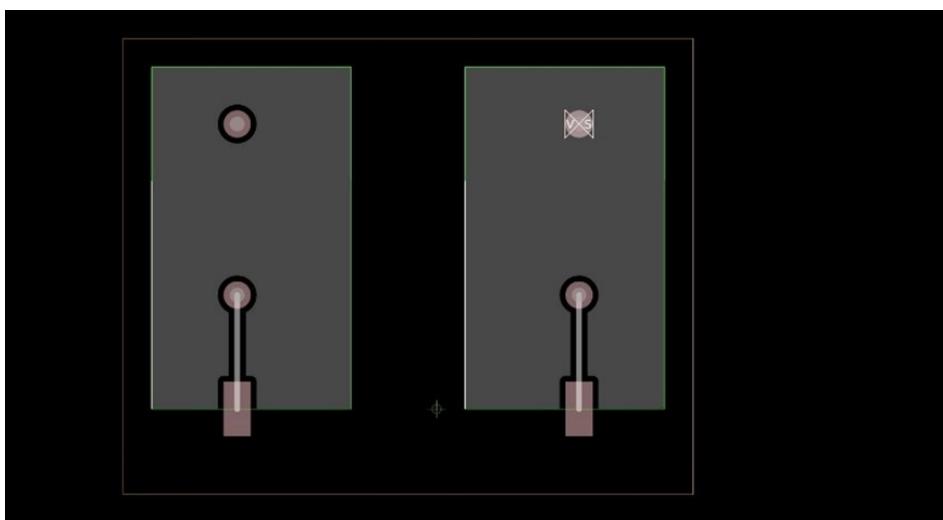
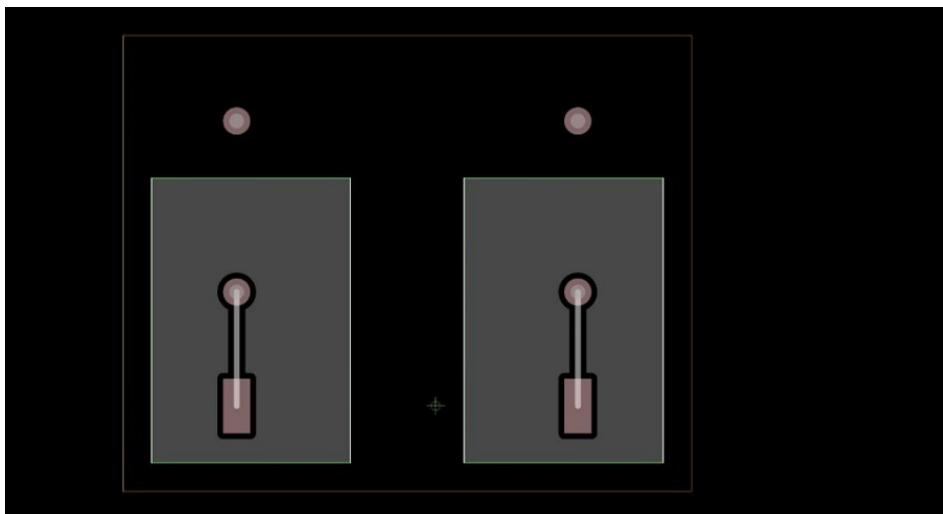
Frozen dynamic shapes can also be fixed, which essentially protects the dynamic shape from any type of update unless the dynamic shape is either unfrozen or unfixed.

Frozen shapes are always considered artwork-ready even if the shape visibly displays possible issues. The possible issues will display a drc marker if they are a real issue.

The following shows a dynamic shape on the left and frozen dynamic shape on the right. Each had the upper via moved. The left shape updated as it should. The right shape did not update because it is frozen. Note the void was not removed and a drc marker added showing an issue where the via is now located.



Dynamic shapes update whenever the shape boundary changes or whenever an object changes that is close to or inside the shape. The method of updating depends on what mode you are using (smooth or fast), as frozen shapes do not respond to requests to be updated when objects close to or inside the shape change. They do respond to self-shape boundary changes by removing voids in order to maintain a legal shape.



The images immediately above show before (top) and after (bottom) boundary edits. The shape on the left of each is a dynamic shape and the shape on the right is a dynamic frozen shape. Note the boundary edit at the *lower* part of the shape merged the void into the boundary in both frozen and unfrozen shapes; while the boundary edit at the *upper* part of the shape added a void to the non-frozen shape, but did not add a void to the frozen shape. A drc was added to the frozen shape.

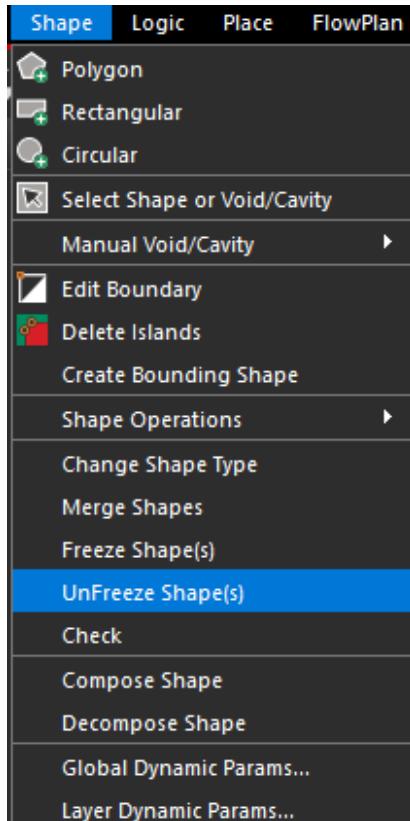
- (i) Currently, you cannot void shapes updated on Allegro. For this reason, Allegro and OrCADX cannot coexist in a Symphony session.

Related Topics

- [Updating Frozen Shapes](#)
- [Freezing and Un-Freezing Shapes](#)

Freezing and Un-Freezing Shapes

You can freeze or un-freeze shapes by choosing either *Shape – Freeze Shape* or *Shape(s) – Unfreeze Shape(s)*, respectively, from the main menu bar drop-down.



You can change a frozen dynamic shape to a static shape (without changing the voiding) by right-clicking and choosing *Change Shape Type* to *Static*. If you then choose *Change Shape Type* and set it to *Dynamic*, it will *not* be a frozen shape.

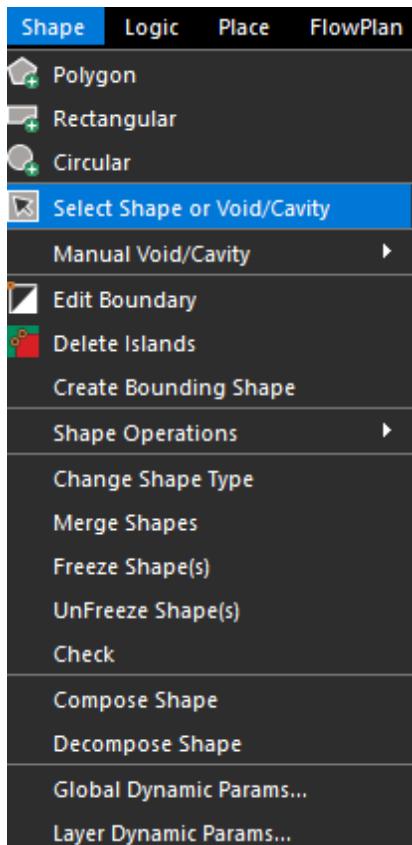
- ⓘ Shapes that are NOT frozen can be edited and the fill is deferred until all edits are finished, so it happens only once. With frozen shapes, fills are updated instantaneously, in real time.
You cannot set parameters of frozen shapes.

You can also freeze and un-freeze shapes by choosing a single dynamic shape and right-clicking. If the selected shape is frozen, *UnFreeze Shape* appears in the menu, otherwise *Freeze Shape* appears.

- [Overview of Frozen Dynamic Shapes](#)
- [Updating Frozen Shapes](#)

Updating Frozen Shapes

To update a frozen shape, choose a single shape using the *Shape* menu – *Select Shape or Void/Cavity* and right-click to choose update.



Or in shape edit mode, choose a shape(s) and right-click to choose update.

- Frozen shapes do not get updated when you choose *Update All* or *Update to Smooth* from the *Global Dynamic Shapes Parameters* form.
- Frozen shapes do not get updated when you choose *Update All* or *Update to Smooth* from the *Status* form.

You can also un-freeze and update frozen shapes using any of the dynamic shape updating commands.

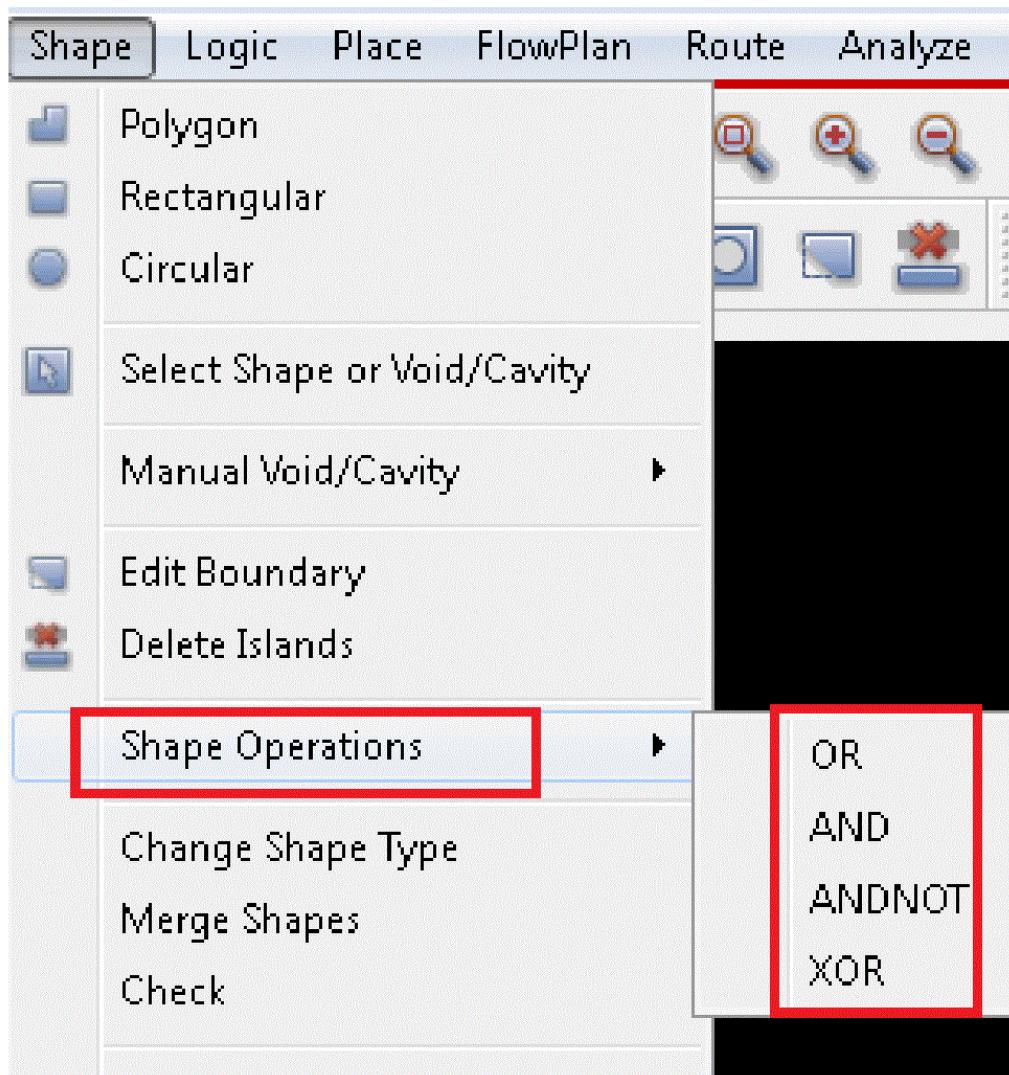
- [Frozen Dynamic Shapes](#)
- [Freezing and Un-Freezing Shapes](#)

Creating Shapes Using Shape Operations

Creating irregular shapes by merging two or more shapes is easier than creating polygon. Use of logical operations on shapes provides a way to create such shapes.

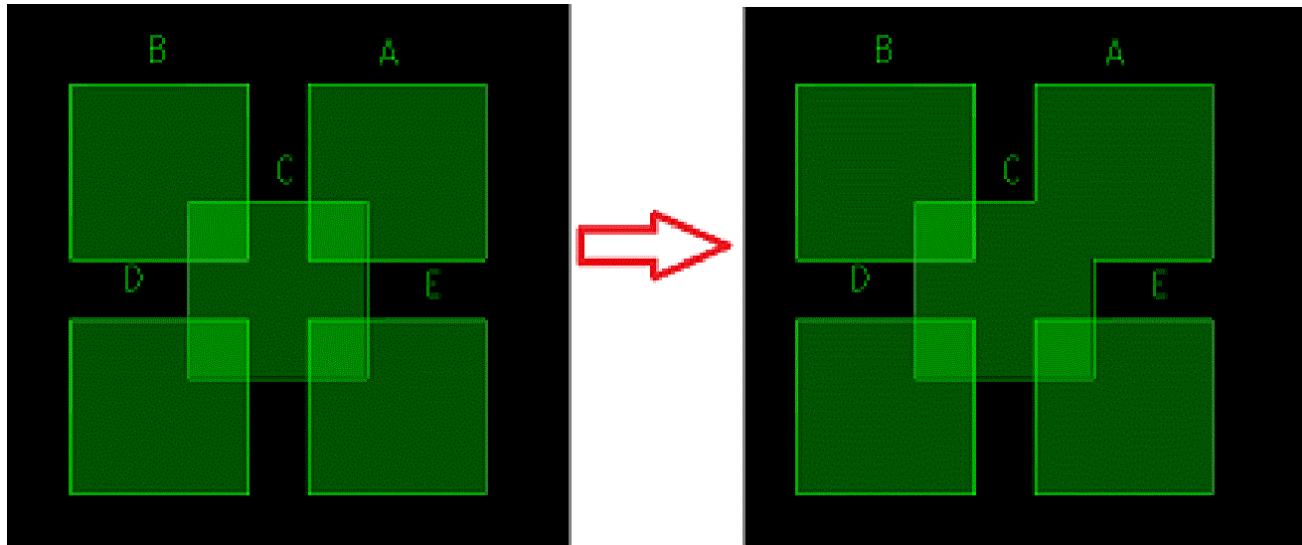
Two or more overlapping shapes that are on same class/subclass and connected to same logical net can be logically operated. The final shape can be the union of two shapes or the difference of two shapes depending on the operation performed on them.

Four logical operators: OR, AND, ANDNOT, and XOR are available to create shapes.



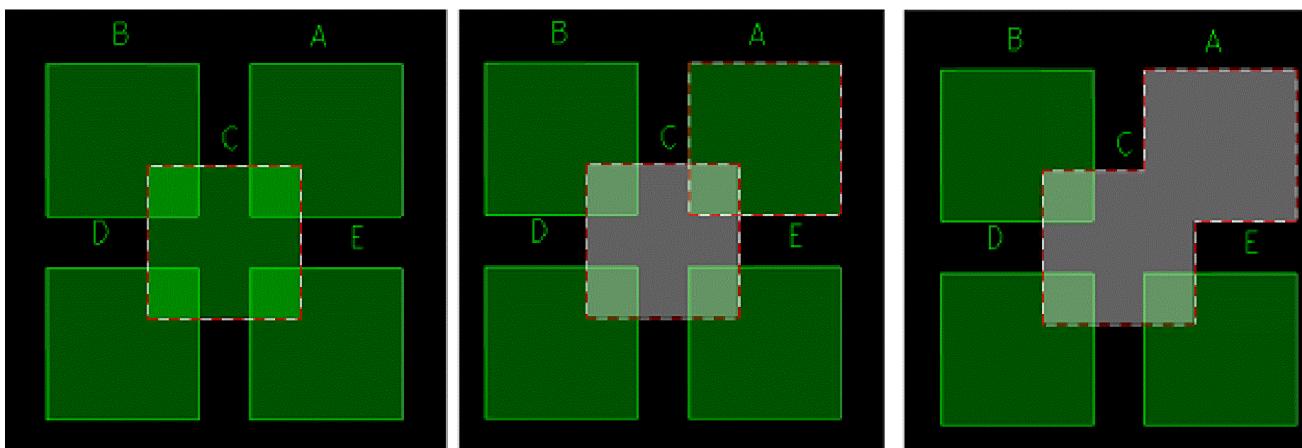
Logical OR

Logical OR operator combines selected shapes into a single shape. In the following image, logical OR operation is performed on shapes A and C.



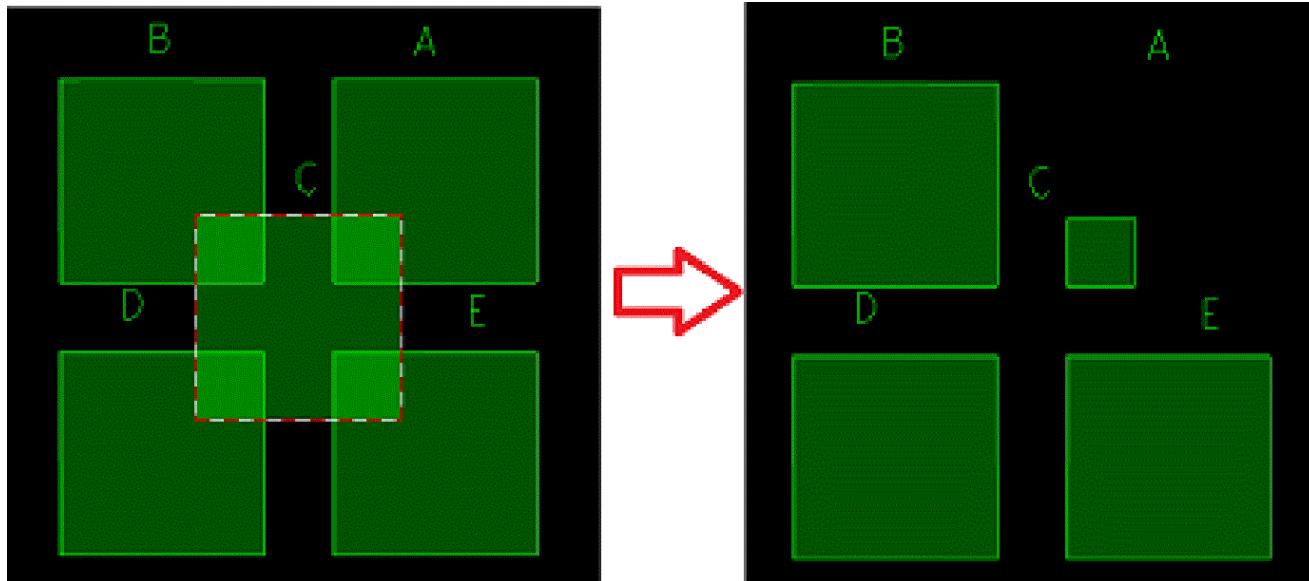
Example

In this example, five shapes A, B, C, D, and E are placed so that shape C overlaps with other shapes. Choose shape C and A for shape operation. The final shape is the union of shapes of A and C.



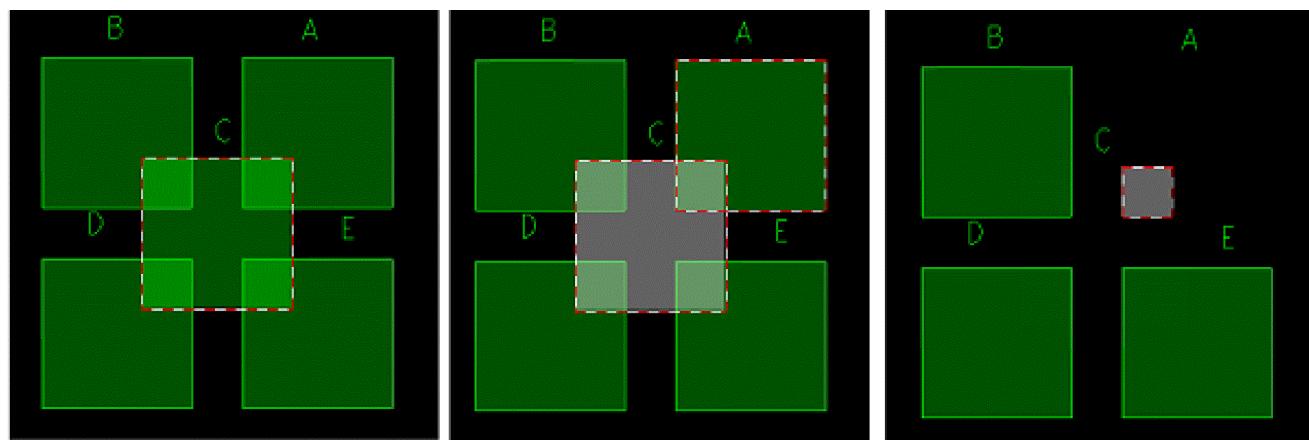
Logical AND

Logical AND operator creates a shape that is common to the selected shapes. In the following image, logical AND operation is performed on shapes A and C.



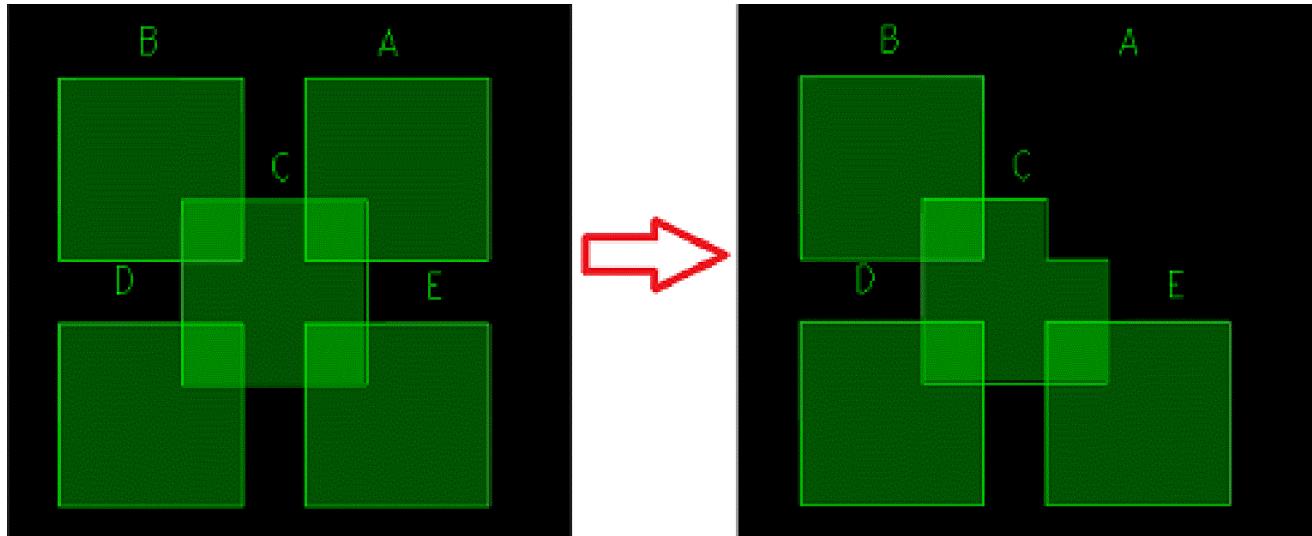
Example

In this example, five shapes A, B, C, D, and E are placed so that shape C overlaps with other shapes. Choose shape C and A for shape operation. The final shape is the intersection of the shapes A and C.



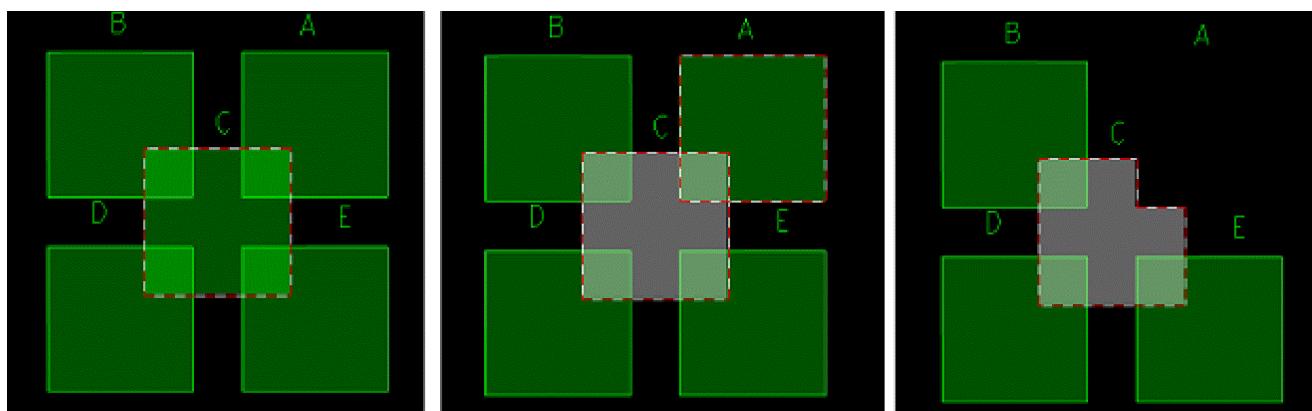
Logical ANDNOT

Logical ANDNOT operator creates a shape that is void of area that is overlapping with other shapes. In the following image, logical ANDNOT operation is performed on shapes A and C.



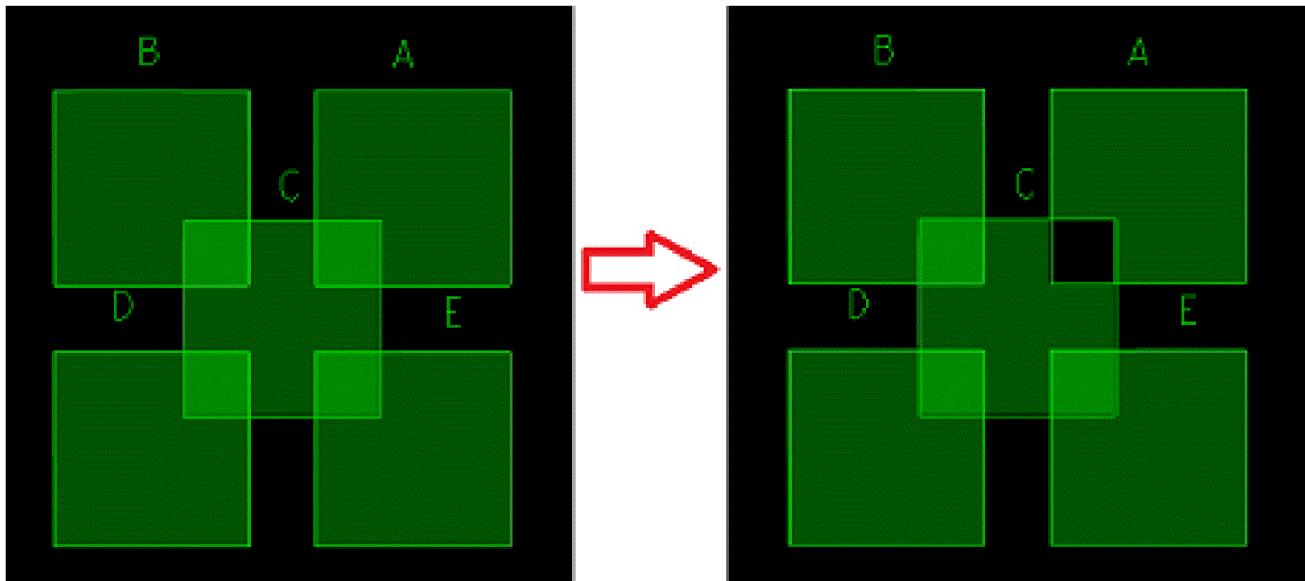
Example

In this example, five shapes A, B, C, D, and E are placed so that shape C overlaps with other shapes. Choose shape C and A for shape operation. The final shape does not include area common to shape A and C.



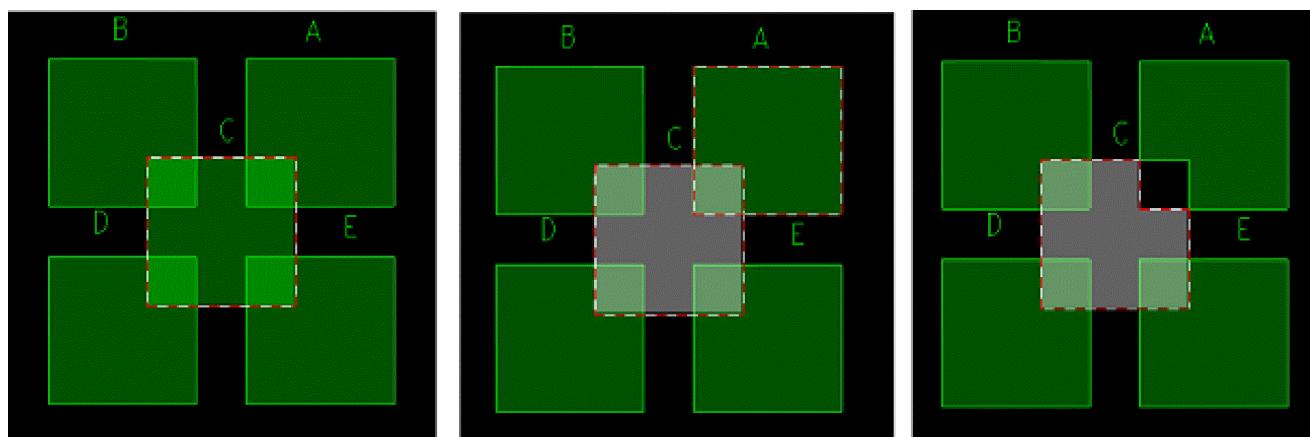
XOR

Logical XOR operator creates a void in the area where shapes are overlapping. In the following image, logical XOR operation is performed on shapes A and C.



Example

In this example, five shapes A, B, C, D, and E are placed so that shape C overlaps with other shapes. Choose shape C and A for shape operation. The final shape has a void in the overlapping area.



Property Handling During Shape Operations

If the properties assigned to shapes are common with same values they are passed to the final shape as it is.

If the properties assigned to shapes are common with different values then the final shape inherits properties from:

- first-selected (base) shape
- other shapes (non-base) shapes with highest layer priority

Setting the Shape Parameters

In creating a shape outline, the tool supports three levels of shape parameters to control the result of dynamic fill and voiding. These shape parameters include those available on the *Shape fill*, *Void controls*, *Clearances*, and *Thermal relief Connects* tabs and can be set at the global, shape instance, and object levels.

- ✓ You can set shape parameters in the *Shapes* tab of the Design Parameter Editor. Use *Setup – Design Parameters* (`prmed` command) to access the Design Parameter Editor or right mouse button click whenever you are working in an application mode.

- Global Parameters:

Set only in the *Global Dynamic Shape Parameters* dialog box and apply to all dynamic copper fill shapes.

- *Dynamic fill* on *Shape fill* tab
- *Artwork formation* on *Void controls* tab
- *Acute angle trim control* on *Void controls* tab (if raster artwork specified)
- *Minimum aperture for gap width* on *Void controls* tab
- *Snap voids to hatch grid* on *Void controls* tab (for xhatch dynamic shapes)

- Shape Instance Parameters:

Set in the *Shape Instance Parameter* dialog box for a specific dynamic copper fill shape and apply only to that shape, overriding global settings. Override values display in a bold blue typeface. You can override any of the fields on the *Shape Fill*, *Void controls*, *Clearances*, and *Thermal relief connects* tabs.

- Static Shape Parameters

Set in the Static Shape Parameter dialog box for static shapes. You can override any of the fields on the *Shape Fill*, *Void controls*, *Clearances*, and *Thermal relief connects* tabs.

- Object Parameters:

Set the following properties using the `property edit` command on a specific pin, cline, or via. Object-based parameters take priority over both shape-instance and global parameters when the object is voided.

- Type Of Thermal Connection (DYN_THERMAL_CON_TYPE)
- Allow Best Fit (DYN_THERMAL_BEST_FIT)

- Minimum Number Of Thermals (DYN_MIN_THERMAL_CONN)
- Maximum Number Of Thermals (DYN_MAX_THERMAL_CONN)
- Clearance Type (DYN_CLEARANCE_TYPE)
- Oversize Clearance (DYN_CLEARANCE_OVERSIZE)
- Oversize Thermal Line Width (DYN_OVERSIZE_THERM_WIDTH)

Set the following properties on dynamic shapes, and clines (lines, shape, and fret on ETCH/CONDUCTOR classes):

- Oversize Clearance (DYN_CLEARANCE_OVERSIZE)
- Do Not Void (DYN_DO_NOT_VOID)

For more information on these properties, see the [Allegro Properties Reference](#).

Precedence of Parameter Overrides

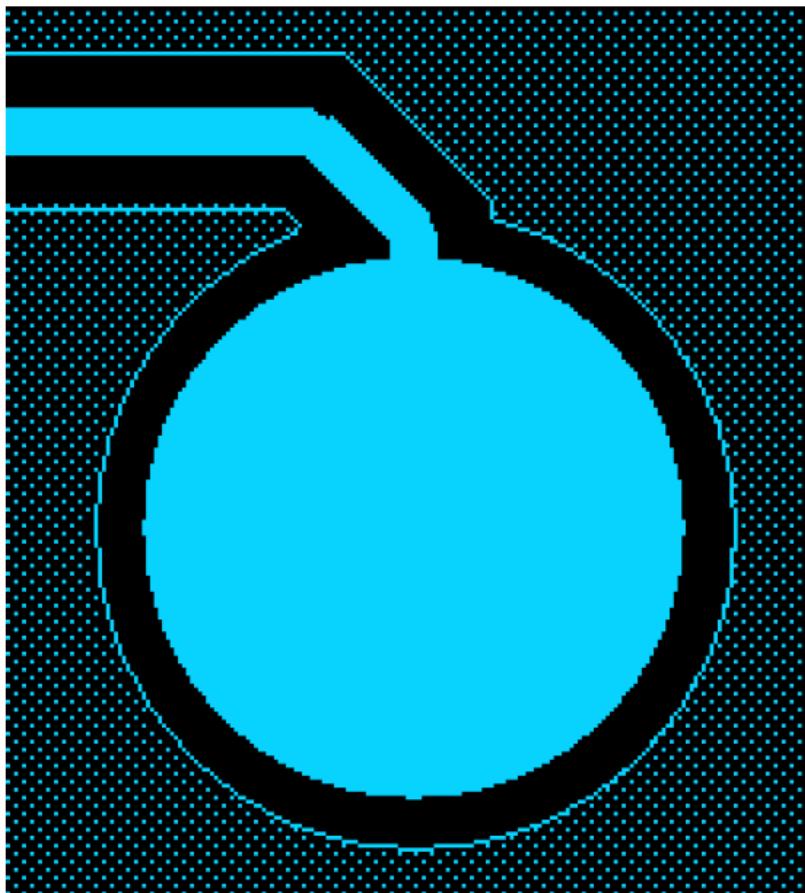
When a parameter has an override at the object level, a change to the global setting has no impact on that dynamic shape. These properties can be managed by choosing *Edit – Properties* ([property edit](#) command). You can also apply them to package symbols at the library level. Object-level overrides apply to all dynamic shapes that the object impacts.

The object-level overrides have the highest precedence. In the case of thermal oversize where multiple levels of object override exist, the tool's property inheritance feature establishes precedence.

See the [shape param](#) command or *Shape – Global Dynamic Parameters* ([shape global param](#) command).

For positive ETCH/CONDUCTOR layers (those for which the *DRC as Photo Film Type* field in the Layout Cross Section dialog box is set to Positive) with shapes whose *Dynamic Copper Fill* mode is *Smooth*, the tool automatically generates voids around any elements inside the shapes, such as connect lines, pins, and vias (based on clearance settings in the Global Dynamic Shape Parameters dialog box). The tool creates the voids using clearance values and void controls that you specify in the Global Dynamic Shape Parameters dialog box.

Sample Void before Smoothing

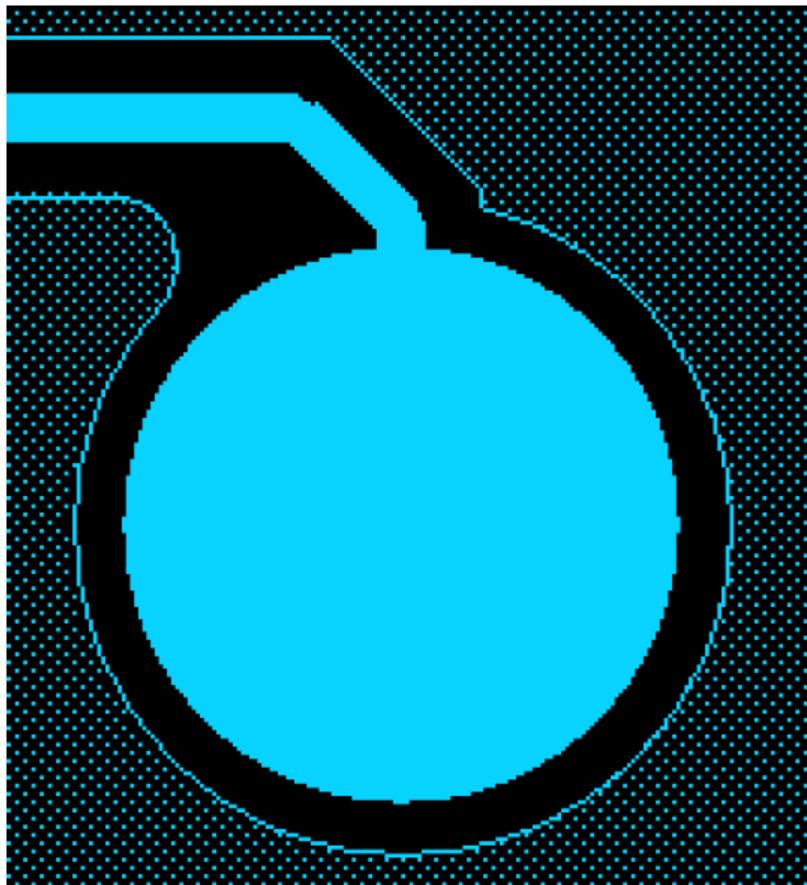


For negative ETCH/CONDUCTOR layers (those for which the *DRC as Photo Film Type* field in the Layout Cross Section dialog box is set to Negative) with shapes whose *Dynamic Copper Fill* mode is *Smooth*, the tool will not generate voids within the shape boundary. Negative shapes use padstack information to photoplot thermal and antipads, rather than DRC spacing values to calculate, display, and check the pads within the shape. Using negative shapes improves performance dramatically because this checking is not performed.

- ❗ Using negative layers limits pads to one antipad size regardless of the DRC spacing rules defined. Therefore, padstacks should be defined with the largest spacing needed as the antipad size. If complicated spacing rules are required, positive shapes should be used.

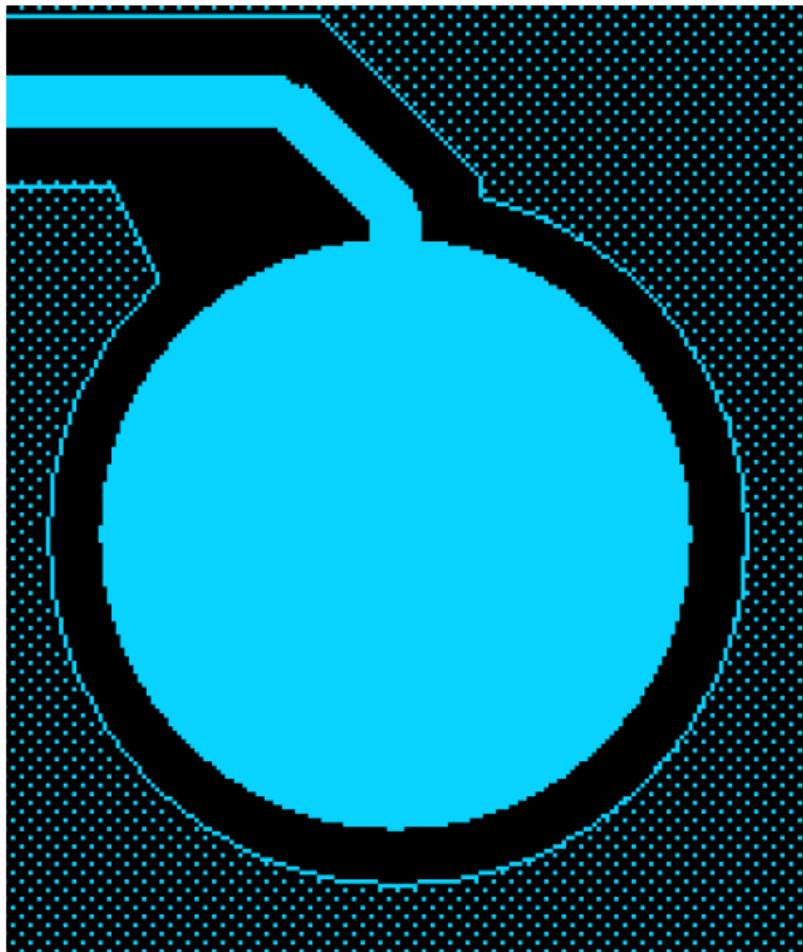
During dynamic voiding, the tool also smooths shapes to eliminate potential artwork problems, as shown below. If you specify a raster *Artwork Format* and an *Acute Angle Trim Control* of *round* on the *Void Controls* tab of the *Global Dynamic Shape Parameters* dialog box, a curved edge is created during voiding as follows.

Sample Void for Raster Artwork with Acute Angle Trim Control Set to Round



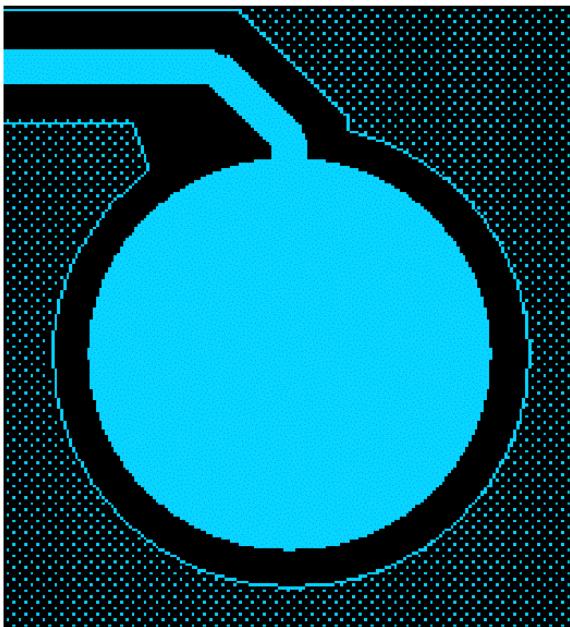
If you specify a raster *Artwork Format* and an *Acute Angle Trim Control* of *chamfer* on the *Void Controls* tab of the *Global Dynamic Shape Parameters* dialog box, the tool creates a flat edge during voiding as follows.

Sample Void for Raster Artwork with Acute Angle Trim Control Set to Chamfer

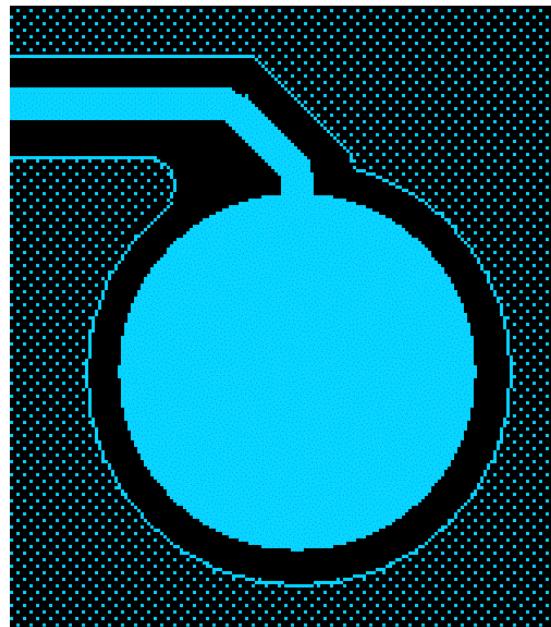


In the following figure, voiding chamfers all 90 degree corners, generating rounded corners all over the void if you move the shape in place.

Sample Void for Vector Artwork with Acute Angle Trim Control Set to Chamfer



Before moving the shape



After moving the shape

Voiding with Static (Manual) Shapes

After you create a static solid, or crosshatch-filled conductor shape, you can add user-defined voids, which are non-conductor (copper) polygonal areas or circles, to the shape interactively with these commands that are described in the *Allegro PCB and Package Physical Layout Command Reference*:

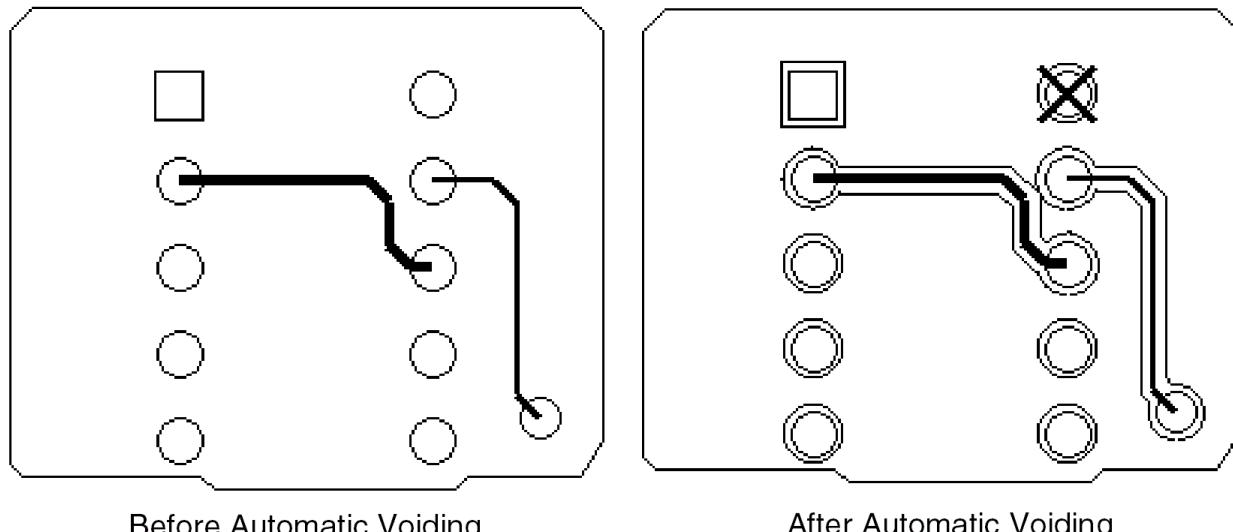
- *Shape – Manual Void – Polygon* ([shape void polygon](#) command)
- *Shape – Manual Void – Circular* ([shape void circle](#) command)
- *Shape – Manual Void – Rectangular* ([shape void rectangle](#) command)

You can interactively edit user-defined voids using these commands:

- *Shape – Manual Void – Element* ([shape void element](#) command)
- *Shape – Manual Void – Move* ([shape void move](#) command)
- *Shape – Manual Void – Delete* ([shape void delete](#) command).

The following example of Void Around Connect Line shows a shape surrounding a DIP with two sets of pins connected by connect lines, before and after void generation.

Example of Void around Connect Line



A DIP with two sets of interconnected pins

Voids are added around both the pads and the connect lines, with thermal-relief connect lines to the pin associated with the same net as the shape

For irregularly shaped objects that require voids, such as connect lines, the tool contours the void around the object for greater accuracy as the example of Crosshatched Shape with Voids and a Thermal Relief shows.

For crosshatched shapes, the tool identifies small or narrow areas that might cause problems and flags them with circles, called shape problems.

⚠ If the shape is on a negative ETCH/CONDUCTOR layer, do not generate voids automatically. On negative layers, the [artwork](#) command adds pad flashes to create the required antipads (voids) and thermal reliefs in the photoplot command file it generates.

Creating Artwork with Dynamic Shapes

When dynamic shapes are out-of-date, the tool displays a *Dynamic Shapes Need Updating...* button on the dialog boxes that appear when you choose *Manufacture – Artwork* ([film param](#) command), *File – Export – ODB ++ inside* ([odb_out](#) command), or *Manufacture – Stream Out* ([stream out](#) command).

If you attempt to use the *Create Artwork* button on the Artwork Control Form dialog box, an error message appears:

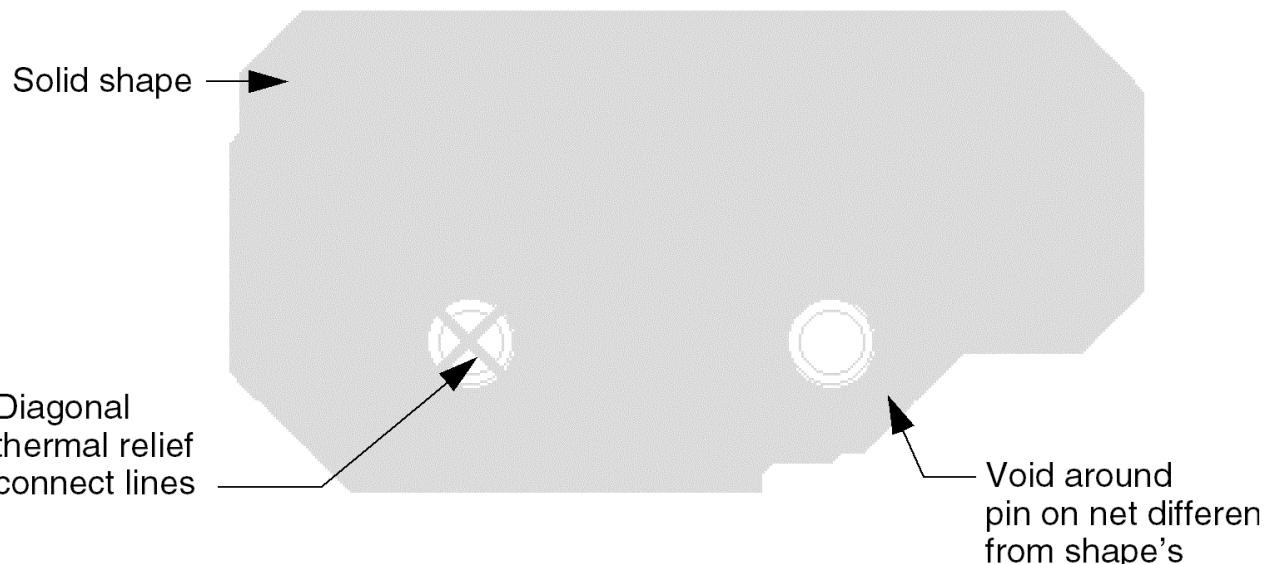
Dynamic Shapes are out of date, please update them.

Click *Dynamic Shapes Need Updating...* to open the *Status* tab of the Status dialog box, which becomes active, blocking any use of the Artwork Control Form dialog box until you update dynamic shapes and/or DRC before proceeding with artwork.

For pins on the same net as an etch/conductor shape on a positive layer, the tool automatically adds connect lines between pins and vias to form thermal reliefs, since you cannot flash negative pads on a positive layer.

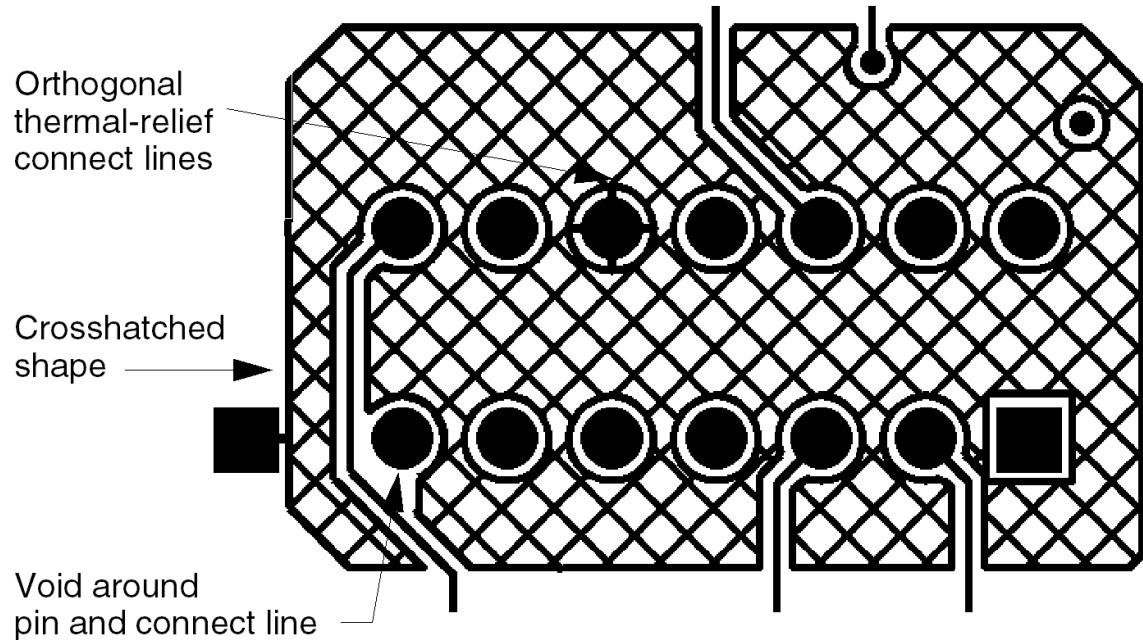
The layout tool offers full-contact, thermal-relief connection. Depending on certain settings in the Global Dynamic Shape Parameters or Shape Instance Parameters dialog boxes, the tool generates no void around pins on the same net as the shape. This ensures a connection to the plane all around the pin.

Example of Solid Shape with a Void and Thermal Relief



⚠ Illustrations used in the documentation represent thermal relief symbols as circles with cross-hairs. If you are creating new designs in post-13.6 versions of the layout editors, thermal relief symbols display as they appear in artwork.

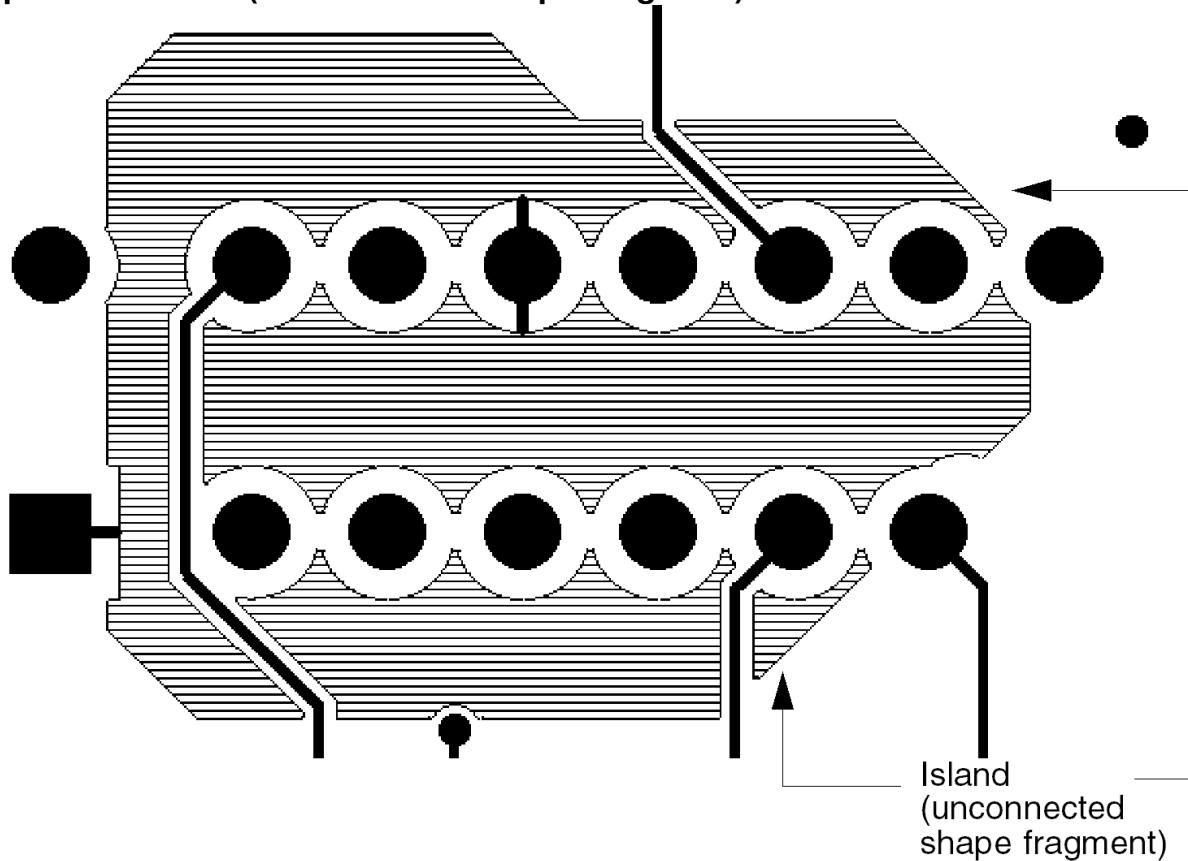
Example of Crosshatched Shape with Voids and a Thermal Relief



Deleting Islands

When the tool updates a dynamic shape, it generates automatic voids, and it may split the shape into multiple shapes or "islands," which are unconnected shape fragments, as the following figure shows. Choose *Shape – Delete Islands* (`island_delete` command) to remove islands.

Example of an Island (Unconnected Shape Fragment)



Inline vs. Individual Voids

You can control whether the tool voids certain patterns of pins in one unit, called inline voids, instead of individually.

This program creates voids around any elements inside the shape, such as connect lines, pins, and vias (based on antipad definitions in the padstacks for pins and vias). The tool smooths shapes to eliminate potential artwork problems and checks for small or narrow areas that might cause problems. These problem areas are identified with circles.

The clearances adhere to the parameters set in the Shape Instance Parameters form. If the shape is on a negative etch/conductor layer, do not generate voids automatically.

Examples of Individual and Inline Voids

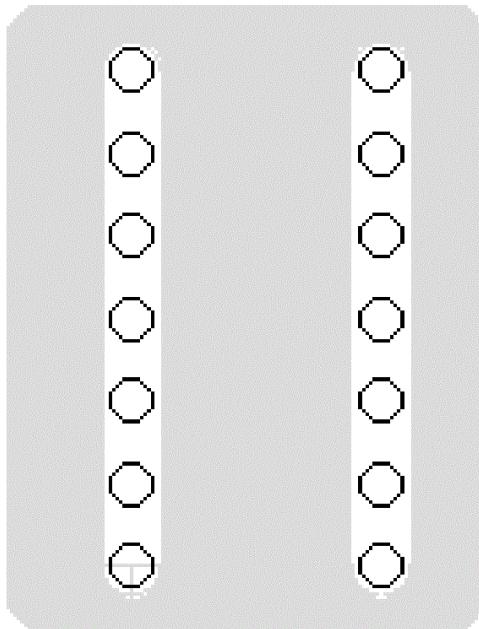
You control the type of pin voids through the *Create Pin Voids* field in the Global Dynamic Shape Parameters or Shape Instance Parameters dialog box. The tool considers the pins to be a pattern if they are:

- All of equal size
- No more than 100 mils apart
- Line up either horizontally or vertically

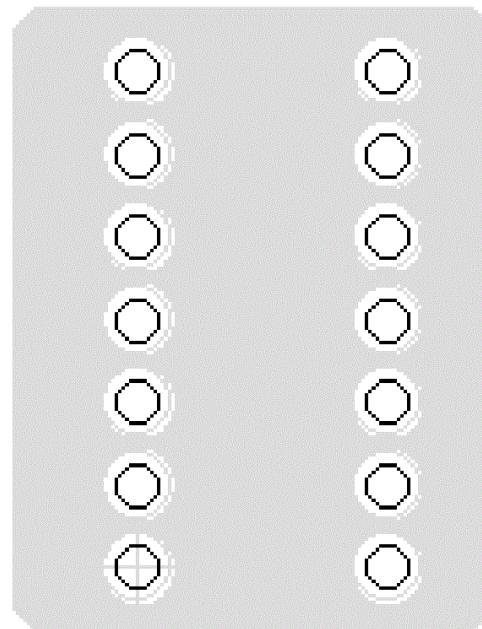
Inline voids generated around these patterns of pins are usually rounded at the ends. If the pin at the end of the pattern is square, so is the endcap of the void. Using inline voids generally speeds automatic void creation and DRC processing time. Shapes with inline voids are also easier and faster to fill when artwork is done.

The tool creates voids around the pins in one of two ways, as the following example shows:

Example of DIP14 with Inline and Individual Pin Voids



Inline Voids (voids between pins are merged)



Individual Voids (each void is treated as a separate object)

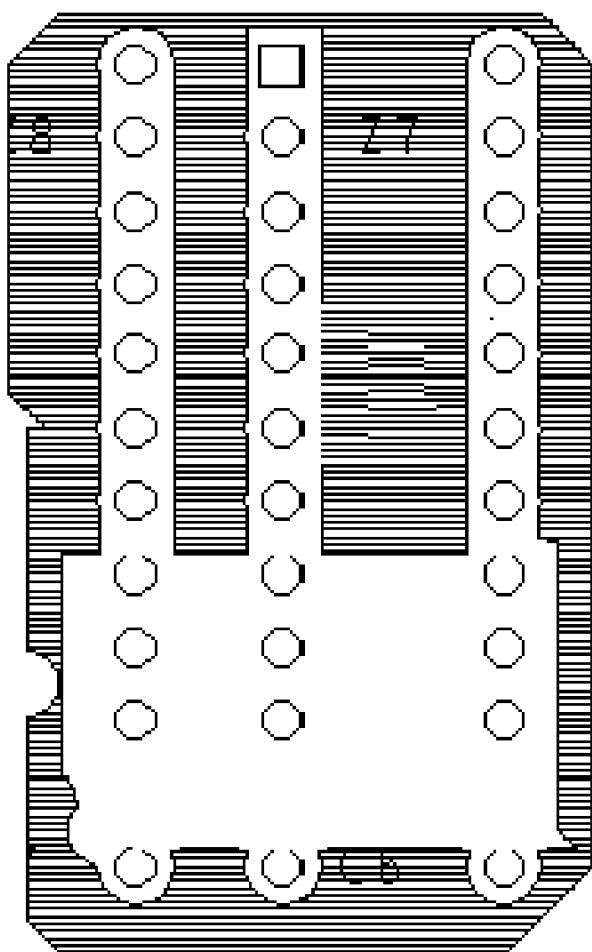
Merging Overlapping Shapes

When the tool generates automatic voids during dynamic copper fill in *Smooth* mode, or when you create voids interactively, some voids might touch or overlap. During dynamic copper fill, the tool merges the voids whenever possible.

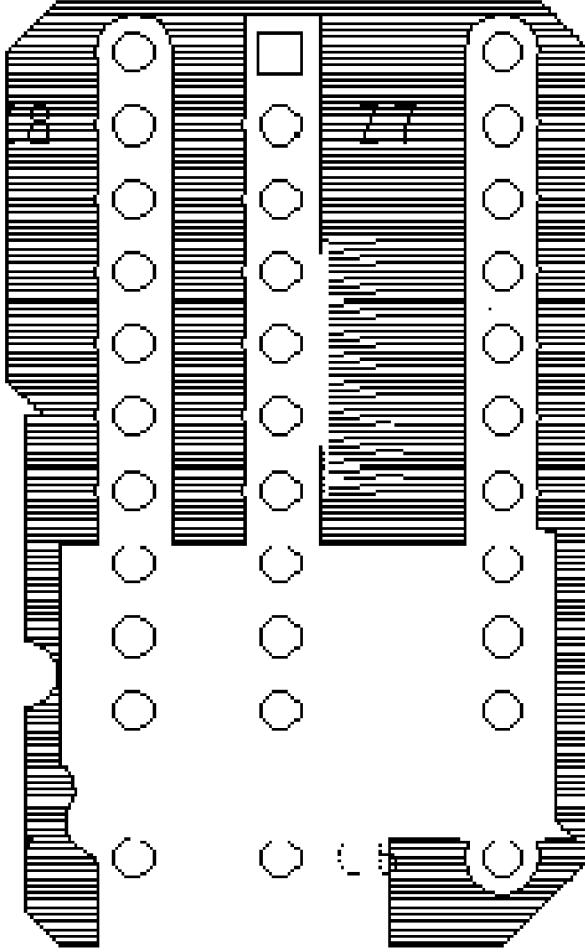
When you edit a static (manual) shape, you can choose to enable the *Merge Overlapping Shapes on Same Net* field on the *Static Shape Parameters* dialog box to fill the shape with any overlapping shapes on the same etch/conductor layer on the same net. Overlapping shapes on a different net results in DRC errors. The tool warns you when a static shape must split or merge.

The tool checks each void to see if it intersects another void or the edge of its parent shape. If the void does intersect, it merges with the other void or parent shape, as follows.

Examples of Merged Voids



Merged Voids



Creating Voids
Near Shape Edge

The tool merges voids in logical patterns. When voids are created near the edge of a shape, the tool cuts away only the part within the shape. When a void is created completely inside another void or outside the shape, the tool immediately deletes the interior void.

In addition to merging voids, the tool also smooths some shapes. This smoothing makes better quality shapes and makes it easier and faster for artwork to fill the shapes. Smoothing only happens on edges that form an angle of less than 90 degrees.

Thermal-Relief Pads

Full-contact, thermal-relief connection is an option you can choose on the Global Dynamic Shape Parameters or Shape Instance Parameters dialog box, if you choose Thermal/Antipad as a pin clearance choice and choose Full Contact as the thermal-relief connect type for pins or vias. With these selections, the tool does not generate voids around pins on the same net as the shape. This ensures a connection to the plane all around the pin.

Multiple-Drill Vias

For multiple-drill vias that connect to shapes, voids and thermal relief lines are not created when dynamic copper fill automatically generates voids. Rather, multiple drill holes are treated in the same fashion as full-contact vias.

Related Topics

- [Property Inheritance](#)
- [Creating Flash Symbols](#)

Using ETCH/CONDUCTOR Shapes in Embedded Planes

An embedded plane is an etch/conductor layer in a design composed primarily of etch/conductor that you use to distribute voltage for power and ground nets in a design.

Creating an Embedded Plane

Developing an embedded plane entails creating an ETCH/CONDUCTOR subclass (a layout cross section layer that has an ETCH/CONDUCTOR subclass name and layer type of plane) and then adding a shape that fills the layer. Generally, the shape boundaries correspond closely to those of the route keepin with a clearance determined by your own design standards.

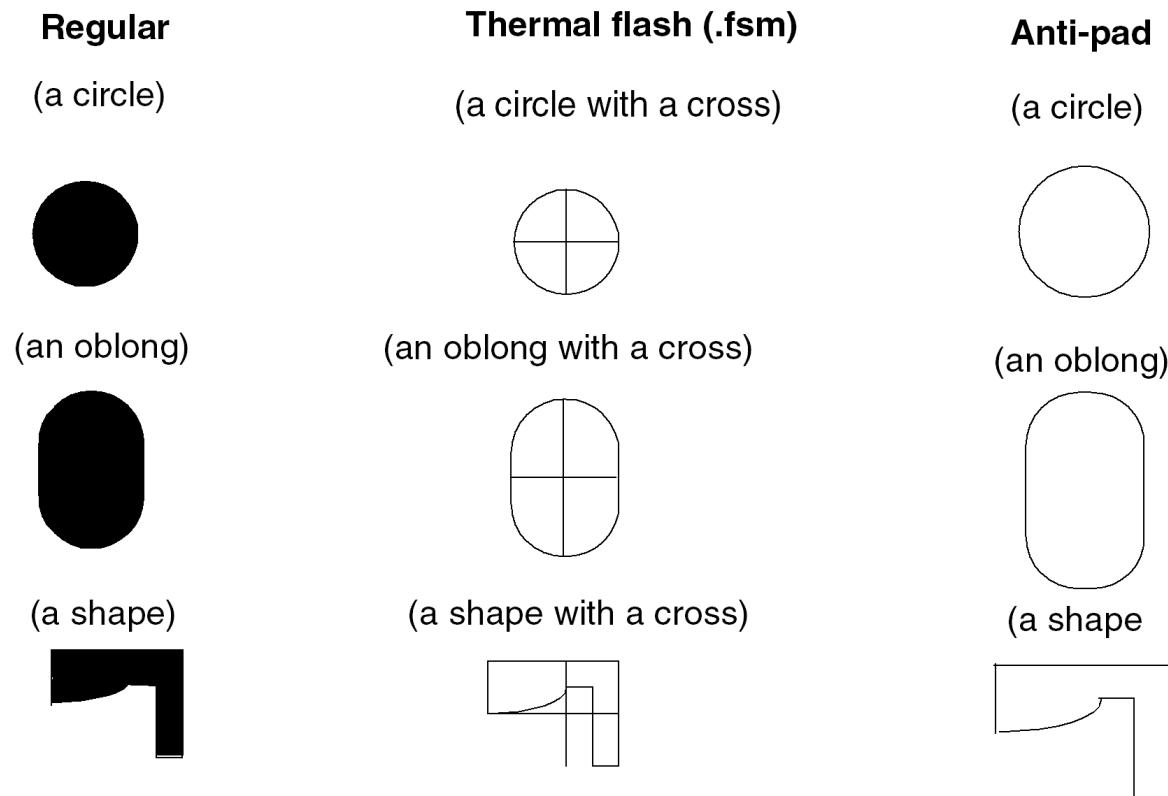
Thermal Relief and Antipad Representation

Thermal relief and antipad representation are determined by entries in corresponding fields of the padstack and by the mode in which you are working. Thermal reliefs are represented by their actual shape (Smooth mode) or as a circle enclosing cross-hairs, as shown in the following figure. Regular pads and antipads display as defined in the padstack.

Preparing the Layout

Layout Padstacks, Vias, and Etch/Conductor Shapes--Using ETCH/CONDUCTOR Shapes in Embedded Planes

Pad Representation



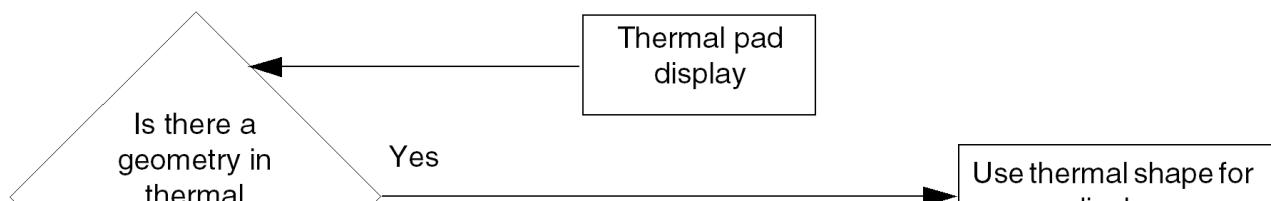
Embedded flash



The cross size for thermals, if displayed as old style flash symbols, is determined by the size of the pad as it is defined in the padstack. Since extents determine shape size, the cross may extend beyond the boundaries of the shape.

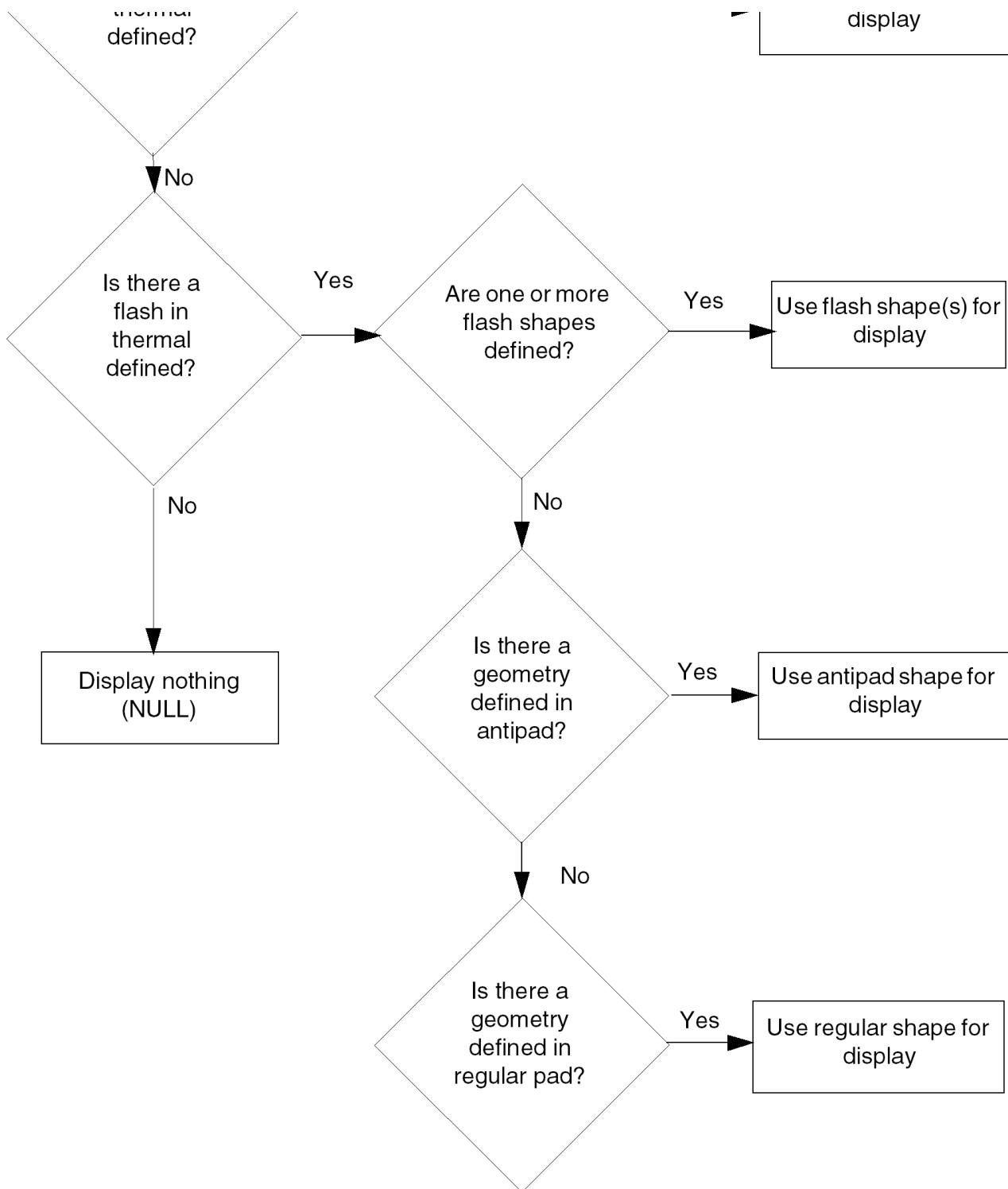
Currently if a flash is set for a pad, the other padstack fields are typically blank. The Thermal Pads function needs the geometry and size fields to be filled in. If not, the following thermal pad display determination algorithm is used, as shown in the following figure.

Thermal Pad Determination Algorithm



Preparing the Layout

Layout Padstacks, Vias, and Etch/Conductor Shapes--Using ETCH/CONDUCTOR Shapes in Embedded Planes

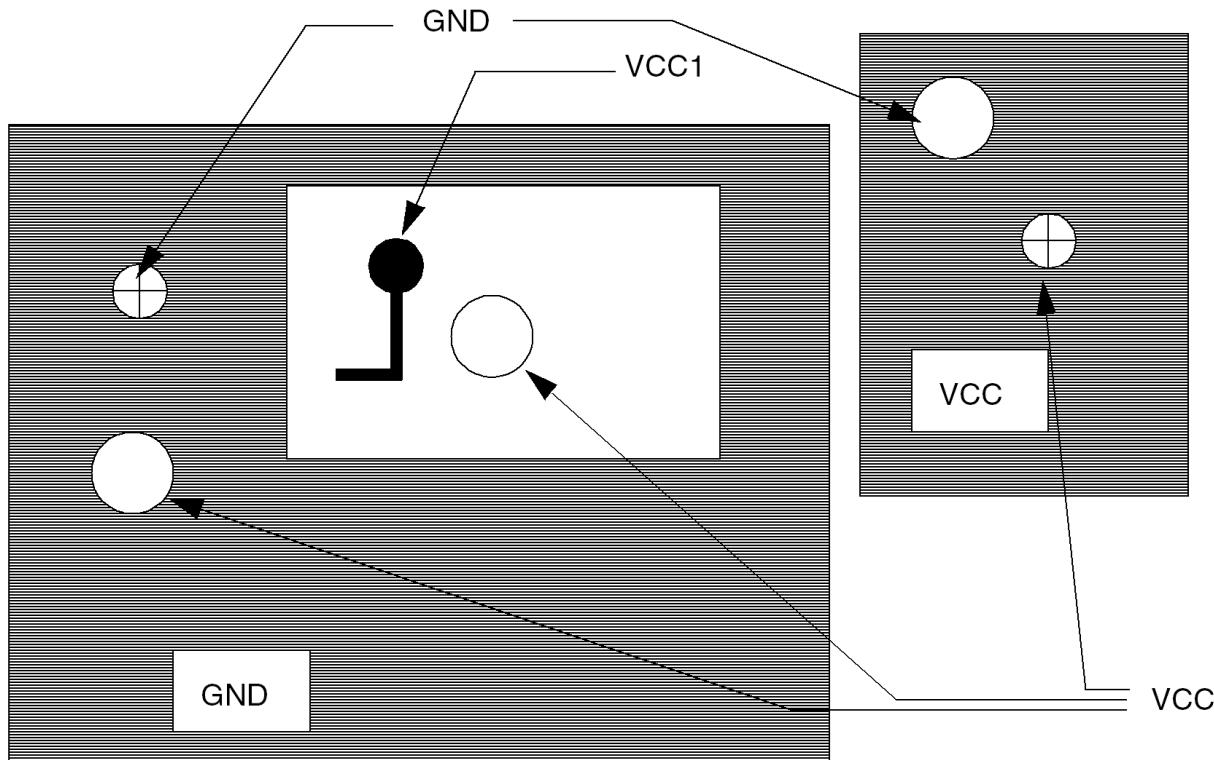


The antipad representation determination algorithm is similar to that of the thermal. It checks geometry in antipads first, thermal second, and regular pads last.

Thermal Relief and Antipads on a Negative Plane Layer

To differentiate between thermal relief and antipads on a negative plane layer on your display and penplot, enable the Thermal Pads function in *Display* tab of the Design Parameter Editor, available by choosing *Setup – Design Parameters* (`prmed` command). The following figure shows a negative plane layer with the Thermal Pads functions enabled.

PCB Editor Negative Plane Layer with Thermal Pads Enabled

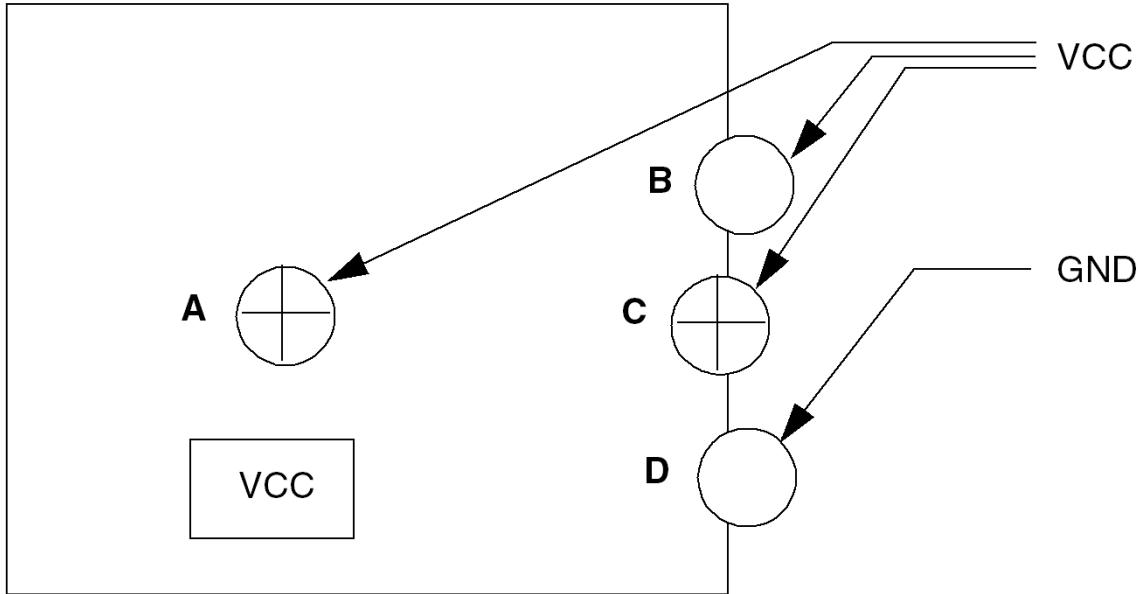


⚠ In the figure, thermal pads are represented as circles with cross-hairs.

Handling Thermal Relief and Antipads on a Shape Edge

The following figure shows how the thermal relief (A, shown with cross-hairs) and antipads (B,C,D) are handled in unusual situations (at the edge of the shape).

Thermal Relief and Antipads on a Shape Edge



The tool connectivity model determines whether or not the pad connects to the plane. In the figure, the pad (A) entirely inside the shape connects to it. It is therefore deemed thermal.

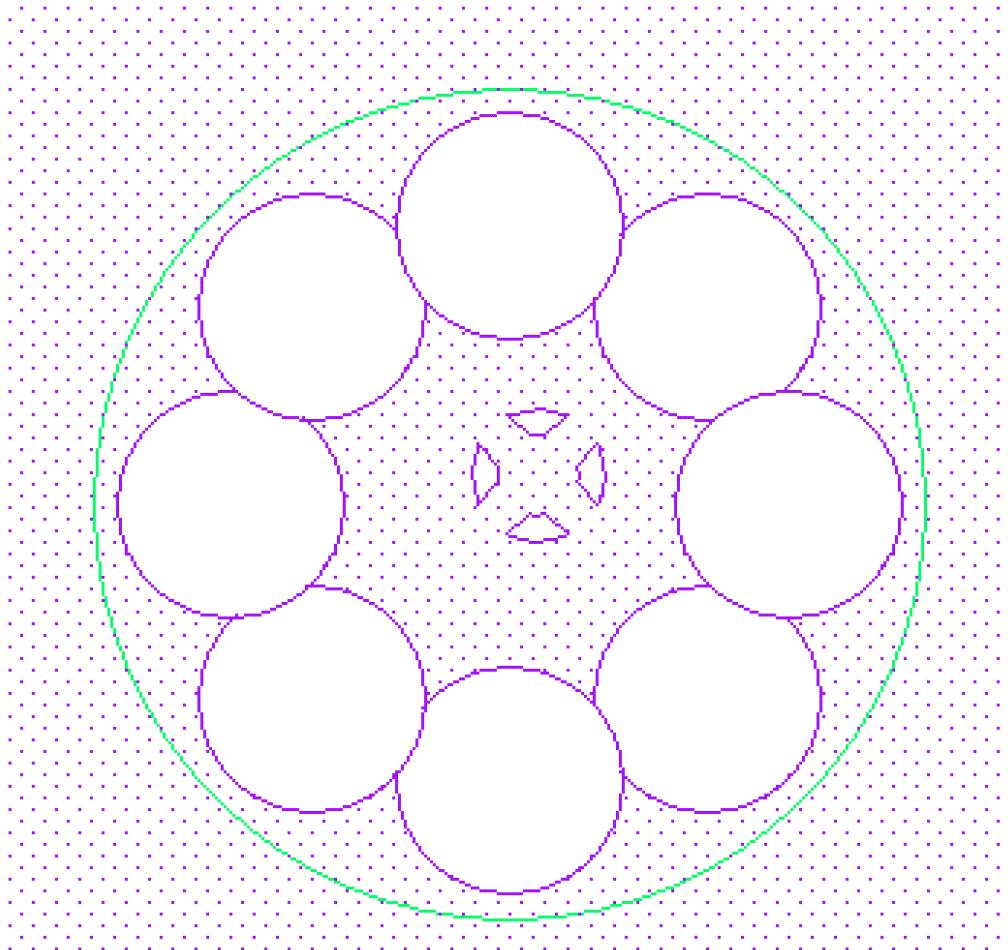
The top and bottom pads (B, D) on the shape's outer edge center outside the shape and therefore display as antipads even though they belong to the same net as the shape.

The middle pad (C) on the shape edge has its origin on the shape and therefore appears as a thermal relief. Note that both thermal-relief pads are displayed by the thermal pad representations (shape with a cross through it).

Negative Plane Islands

If you place anti-pads or thermal reliefs very close together on a negative layer, electrical opens, or "islands" may result, as illustrated in the following figure.

Negative Plane Island

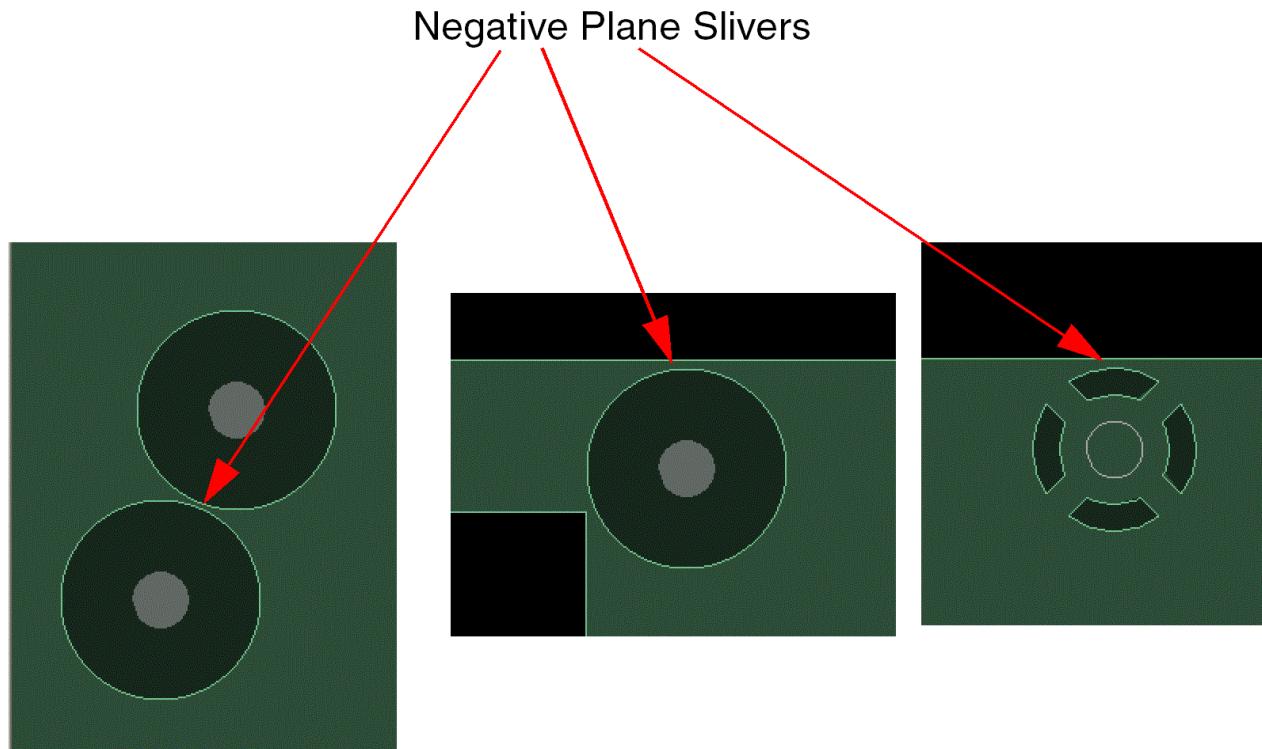


Thermal reliefs within a plane island will not connect to the main plane because the island is electrically isolated. To identify islands in a design, you can activate DRC for them by choosing *Setup – Constraints – Modes*, then click the *Design Constraints* tab and turn on the *Negative plane islands* option. When you run DRC, a bow tie DRC error marker appears with the D - I (Design-Island) error code at the location of a connection point of a member pin.

The pin or via elements that form an island define one aspect of a DRC record, and the shape in which the island is formed defines the other aspect. Islands can exist as either shape fragments or as shape islands.

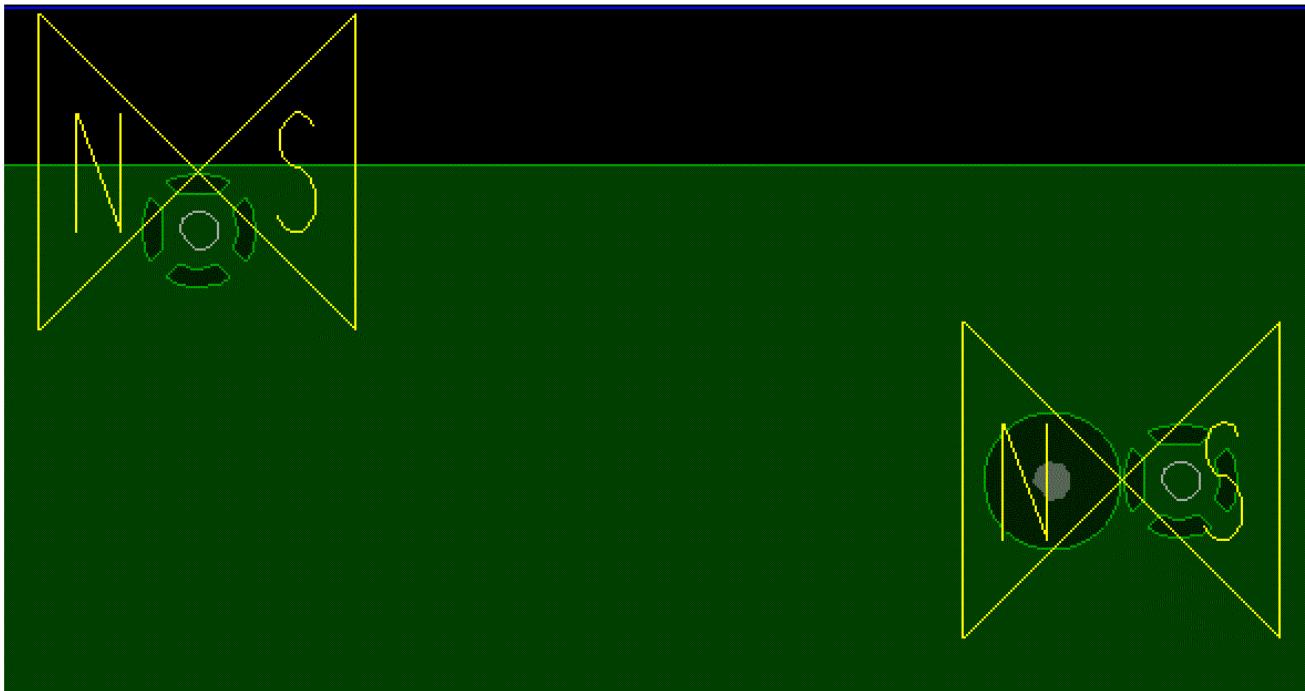
Negative Plane Slivers

Conducting slivers are created during the etching process and often become detached and may produce shorts on circuit boards. Slivers that do not get detached, can still cause electrical problems, such as limiting electrical current. While potentially existing on any type of conducting layer, slivers are most often produced on negative plane layers and are caused by antipads and thermal pads of pins and vias spaced too close either to other padstack referencing items or to the conducting plane boundary.



To identify negative plane slivers in a design, you can activate DRC for them by choosing *Setup – Constraints – Modes*, then click the *Design Constraints* tab and turn on the *Negative plane slivers* option. When you run DRC, a bow tie DRC error marker appears with the N - S (Negative-Sliver) error code at the location of the sliver.

Negative Plane Sliver DRC Markers



Pad Drawing Mechanism

The following mechanism displays thermal relief and antipads.

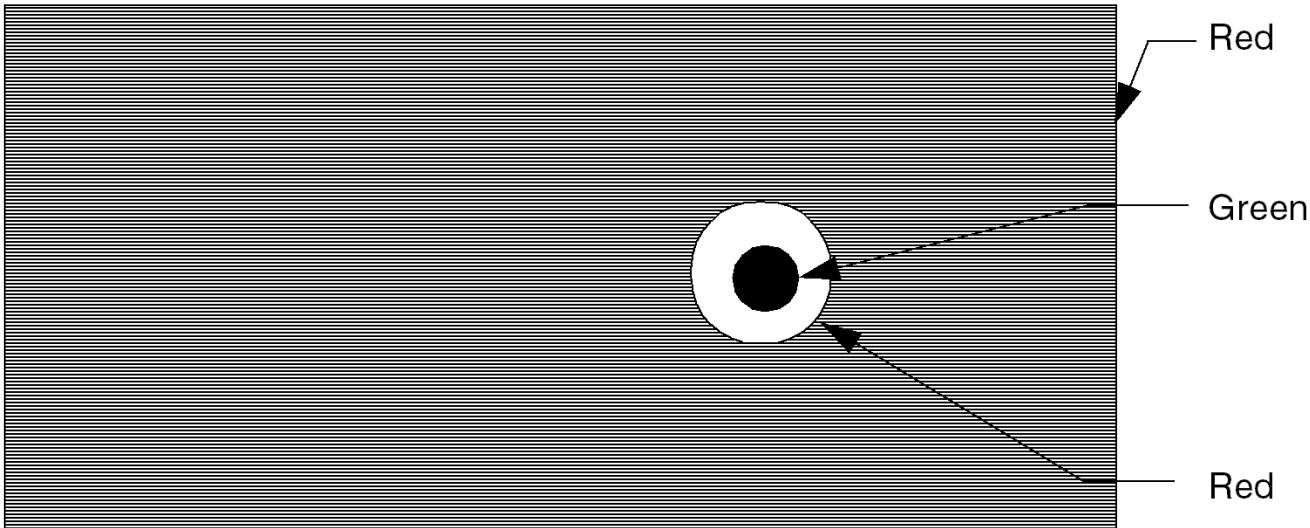
You cannot create and draw shapes with voids on the fly. To achieve the appearance of the shape with voids (antipads), pads are actually drawn in the background color. Instead of drawing a single filled pad (regular), as the tool does with the Thermal Pads function off, the following is drawn:

- Pad outline in chosen pad color (green, for example).
- Pad in background color.
- If the pad is thermal, a cross is drawn in color of the pad.

Layer priority takes precedence in any drawing. Consider the following two cases:

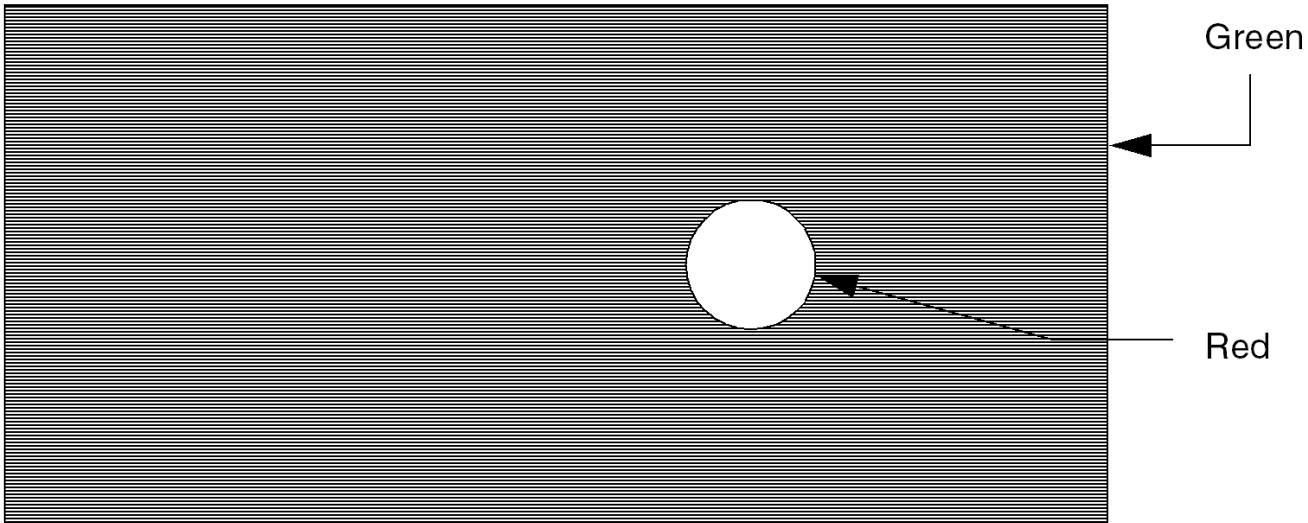
Preparing the Layout

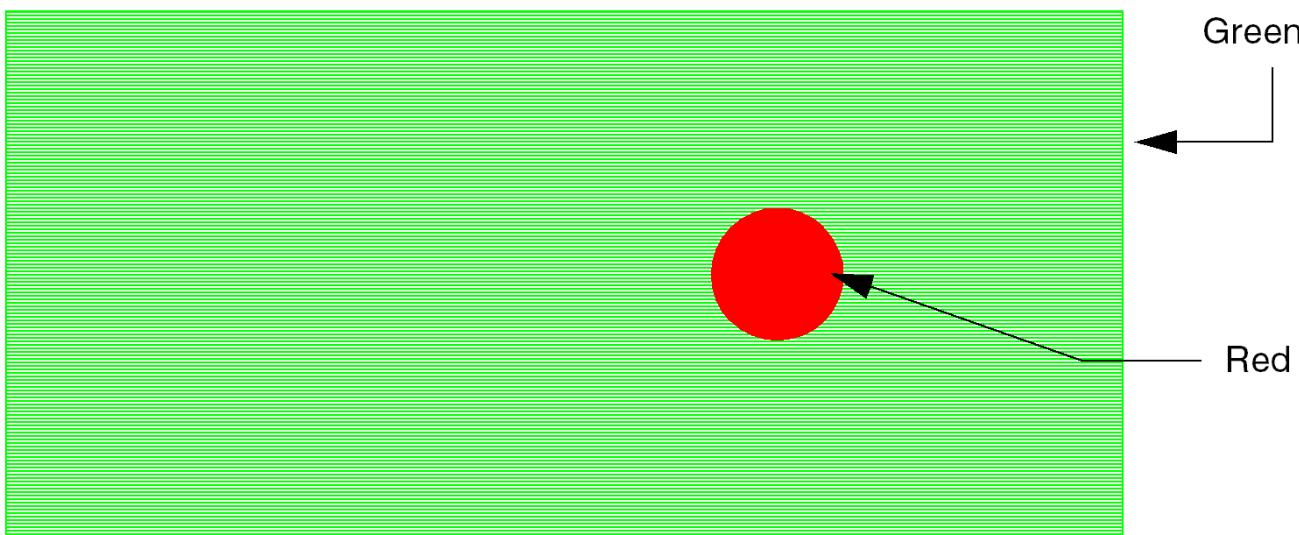
Layout Padstacks, Vias, and Etch/Conductor Shapes--Using ETCH/CONDUCTOR Shapes in Embedded Planes



In the first example, red has a higher layer priority. The shape is drawn first (Green); then the pad outline and the background filler. Note that both Thermal Pads and Enhance display are enabled. In the bottom case, red has a lower layer priority. The red pad is drawn first, background filler is drawn right after it, and shape is drawn last. Because the shape is drawn last, it covers the pad. It is important to set up the Layer Priority table correctly.

You can also display pads from other layers simultaneously. Consider the following illustration:





In the illustration, both shape and pad are red. The internal pad, displayed on a different layer, is green. Green has a higher Layer Priority than Red. If Green had a lower priority, the internal pad would not be displayed. This would happen because the antipad is not hollow. It is a filled pad, drawn with background color. The antipad would cover the internal green pad, because the background filler is drawn at the same time as the red outline pad, with red having a higher priority.

Consider the following:

- Use only one ETCH/CONDUCTOR (negative) and no more than one Pin and Via layer (the same layer) displayed at a time.
For example, if you wish to check the +5VP layer, turn off all layers, turn +5VP ON for ETCH/CONDUCTOR, Pin and Via (or just Pin Or Via).
- Keep the same color for Pin, Via and ETCH/CONDUCTOR for each layer.
If ETCH/CONDUCTOR, Pin and Via layers have the same color, Layer Priority is not important.

ETCH/CONDUCTOR Shapes' Effect on Routing

Whether shapes are defined as negative or positive affects routing results. You make this choice in the *Film Type* column of the Define ETCH/CONDUCTOR Subclass dialog box as you add each ETCH/CONDUCTOR subclass.

- Click `define subclass`, then click ETCH/CONDUCTOR in the Define Subclass dialog box.

Before specifying negative or positive, consider the following:

- The positive or negative selection you choose in the Layout Cross Section dialog box is for DRC checking only.

You choose Plot mode Positive or Negative in the Film Record dialog box for each film you set up for the `artwork` command to determine plotting for any layers in that film record.

- Define embedded as negative.

Add them before you run the router, especially if the design has surface mount components. This allows the Pin Escape router to define pads properly on those layers and keeps the routing tools from trying to connect power and ground pins that connect through the embedded planes.

- Define any shape sharing a layer with other etch/conductor as positive, and use *Shape – Manual Void – Element* (`shape void element` command) to create clearance voids. After all routing and glossing finishes, you can designate a layer as negative, then change it to positive when you void shapes.

Related Topics

- [Creating Flash Symbols](#)
- [prmed](#)
- [artwork](#)
- [shape void element](#)

Metal Usage Report

Often, manufacturing a package or a PCB substrate involves a process where layers are built up or created one on top of another. If the density of metal in one area of a layer is significantly higher than in other areas of a layer, the substrate can develop a bump in that area. To achieve improved manufacturing yields, it is essential to balance the percentage metal coverage over the area.

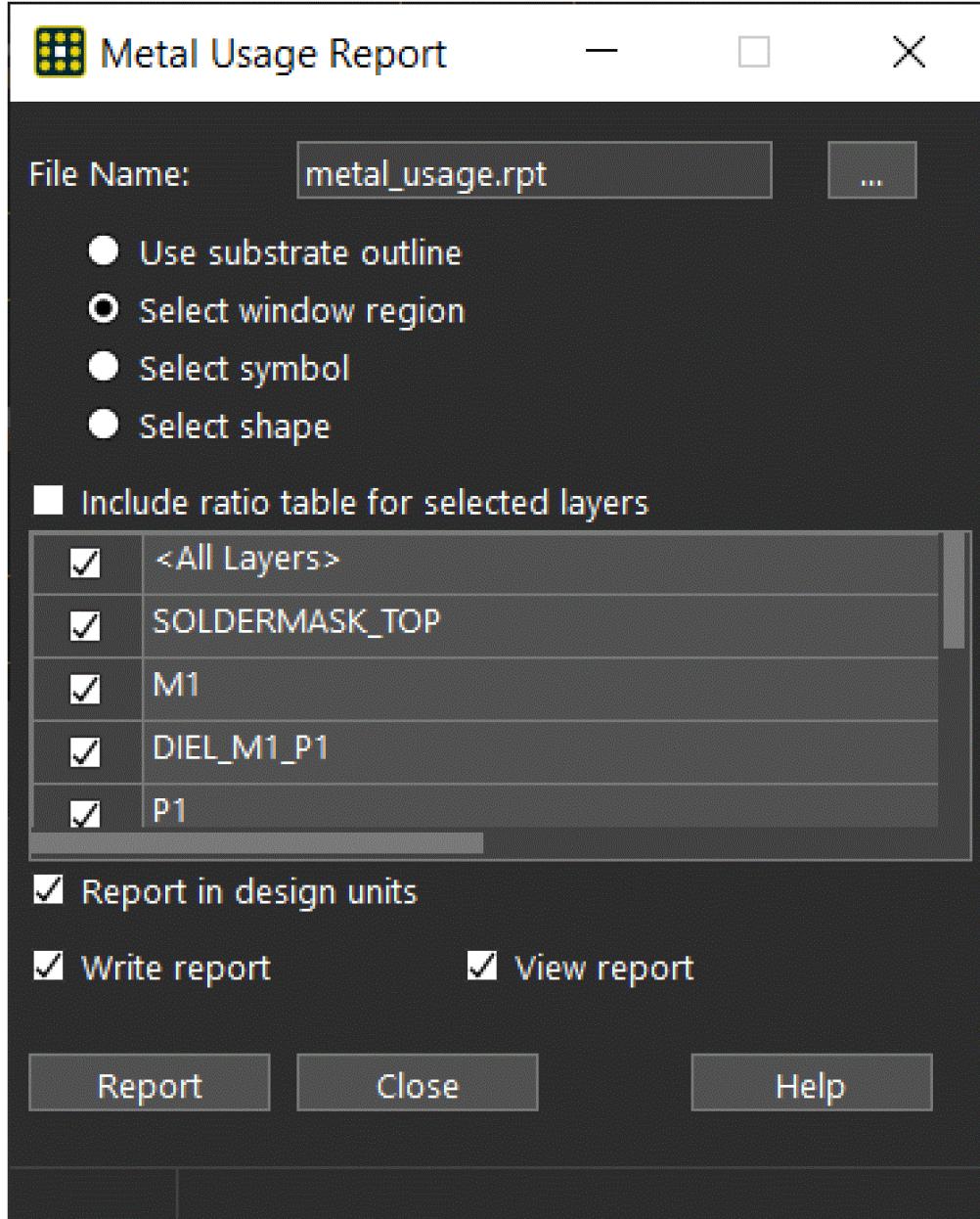
To accomplish this balanced distribution of metal, you need to know the percentage of metal coverage in specific areas of the design. For example, you may want to know the total metal coverage for a layer across the entire substrate, the coverage under a specific component (generally a die for packages), or the metal coverage that is under a large plane shape.

If the specified region has more metal density than desired, you may need to re-route traces or perforate plane shapes with a series of small voids placed at regular intervals. This lowers the overall metal density for the shape, and also provides a means for gasses to escape during the manufacturing degassing process. If the metal density is too low, you may have to create a set of shapes in the area to add extra metal until the desired percentage coverage is achieved, a process known as thieving.

The Metal Usage Report accurately assesses the percentage of metal in a specified region of the design. This information guides you in improving the overall layout to maximize yields. This report complements the existing Film Area Report, by offering interactive options for selecting the region and layers for the report.

The following figure shows the dialog box that appears when you choose the *Reports - Metal Usage Report* ([metal usage report](#)) command.

Metal Usage Report Dialog Box



The following figure shows a sample Metal Usage Report describing the metal usage under a shape on two layers.

Sample File

The screenshot shows a window titled "Metal Usage Report". The window has a toolbar with icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Help) and search functionality. The main content area displays the following information:

Design Name: SUB00060X1_180nm_WLCSP_For_extraction_330um_Sq_UBM_20170503
Report Generated: Mar 4 17:53:56 2022

Search Region: Lower Left: (-2077.4151 2441.9605)
Upper Right: (4078.4614 -2694.4096)

Layer	Total Area (sq um)	Metal Area (sq um)	Metal Percentage %
SOLDERMASK_TOP	31618859.9939	21404858.9939	67.6965
UBM	Failed	Failed	Failed
PBO2	31618859.9939	1255109.2500	3.9695
RDL	31618859.9939	5461088.4398	17.2716
PBO1	31618859.9939	1692128.0932	5.3516
PASSIVATION	31618859.9939	307289.7000	0.9719
SOLDERMASK_BOTTOM	31618859.9939	31618859.9939	100.0000

** Special Notes:
Some layers failed metal usage report computations.
If you believe the metal usage reported on a layer to be inaccurate, ensure that your cross-section has the embedded component status set accurately for these layers.
Any layer identified as supporting cavities will disregard any metal inside a cavity cutout as no metal may reside in this area after manufacturing of the substrate.

For additional information, see the [metal usage report](#) command.

Thieving

The `thieving` command enables you to add a pattern of non-conductive, single-layer figures to areas on the outer layers of a physical design that do not contain copper. You generate the thieving pattern to balance the plating distribution, placing it to avoid interference with the signal quality of adjacent circuits. Use thieving near the end of the design process, prior to artwork generation.

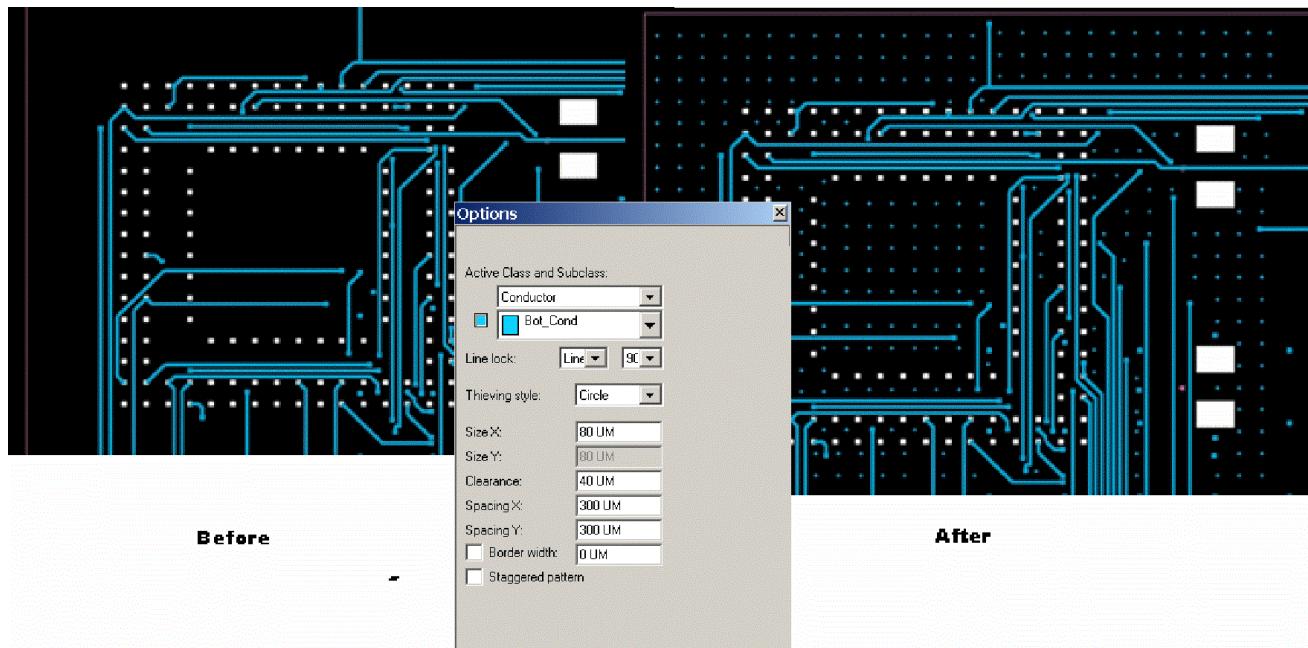
Once you generate a thieving pattern, the results appear in the *Padstack Usage Report*, available by choosing the *Reports – Reports (report)* command.

Thieving patterns adhere to the parameters you specify in the *Options* window pane, regardless of DRC rules. The parameters remain in effect until you change them.

Once you generate a thieving pattern, the tool handles the pattern as a group.

The example in the following figure shows the parameters set in the *Options* window pane to generate a staggered thieving pattern of 35 micron circles, 100 microns spacing, and 125 micron clearance. The thieving pattern adheres to all via and keepout boundaries that exist within the outlined area.

Example of a Staggered Thieving Pattern



Related Topics

- [report](#)
- [Working with Groups](#)

SiP and APD: Degassing

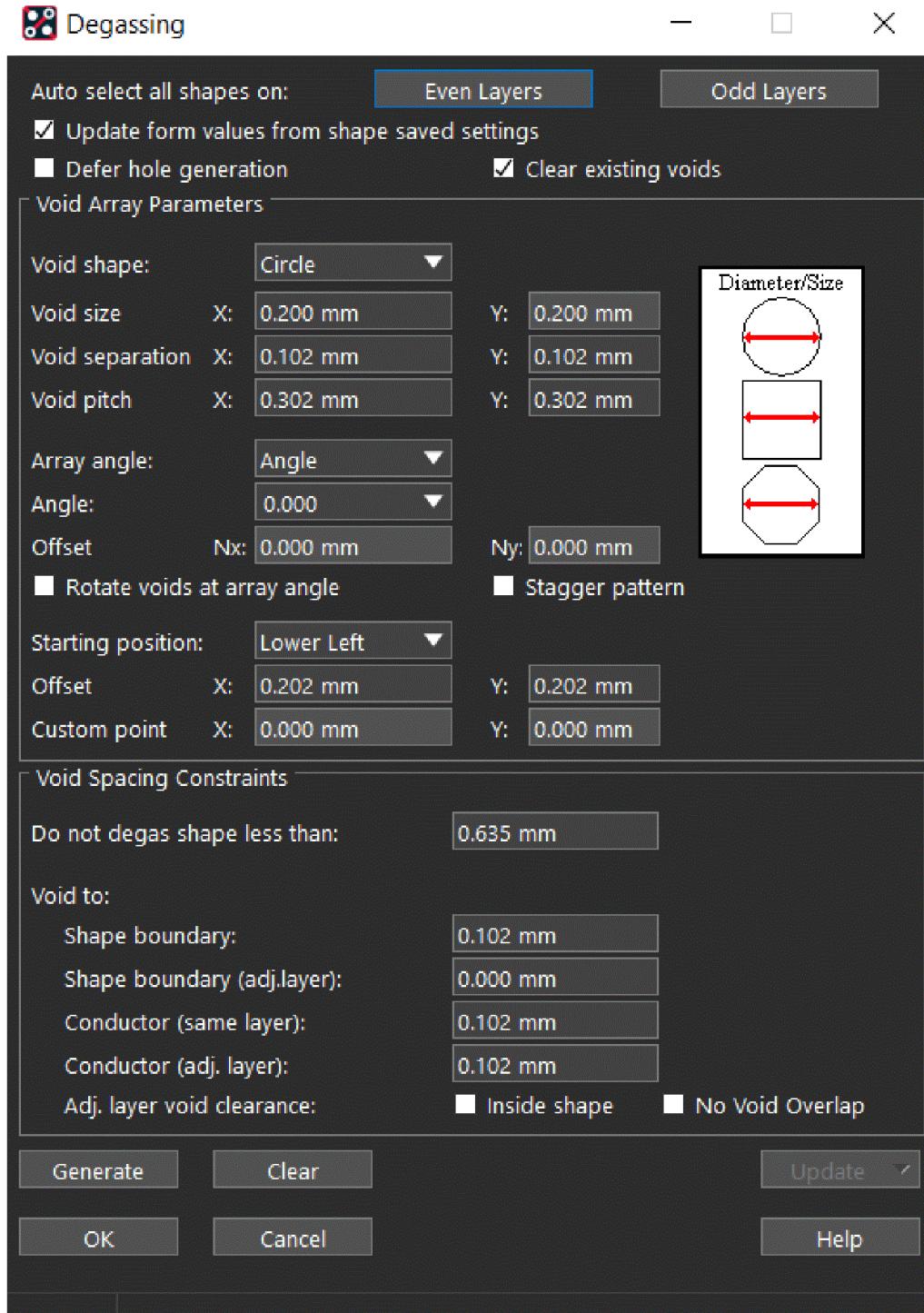
Degassing is a process in package design where you perforate power planes, voltage planes, and filled shapes in your design. The degassing holes allow gas to escape from beneath the metal during the manufacturing of the substrate. Failure to perform this task results in gas bubbles forming under the metal, which may cause the surface of the metal to become uneven.

The perforations for degassing are small, having a specified size and shape, and are spaced regularly across the surface of the plane. Due to the small size (relative to the shape or plane) of both the openings and their spacing, it may be necessary to create a large number of openings, which is a time-consuming manual task for a large shape. With the degassing feature, you specify the details of the perforation array pattern and rely on the system to automate the task of generating the exact placement of the voids across the entire shape. Using an automated approach minimizes the risk of creating an opening, which violates any of the spacing or manufacturing requirements for degassing.

Typically, you degas a design near the end of the design process while preparing the design for manufacture. Using the degassing feature improves the vacuum lamination and pattern plating manufacturing processes. After you degas your design, it is recommended that you perform final electrical verification.

The following figure shows the Degassing dialog box where you set your degassing parameters.

Degassing Dialog Box



For information on setting the fields in this dialog box, see the [degas](#) command in the *Allegro PCB and Package Physical Layout Command Reference*.

Also, see the [thieving](#) command in the *Allegro PCB and Package Physical Layout Command*

Reference. This command lets you add a pattern of non-conductive, single-layer figures to areas on the outer layers of a package substrate that do not contain copper. You generate the thieving pattern to balance the plating distribution, placing it to avoid interference with the signal quality of adjacent circuits.

