cadence®

Creating Design Rules

Product Version 23.1
September 2023

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

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 outlined 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 the use of such information.

Cadence is committed to using respectful language in our code and communications. We are also active in the removal and/or 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

preface

Preface

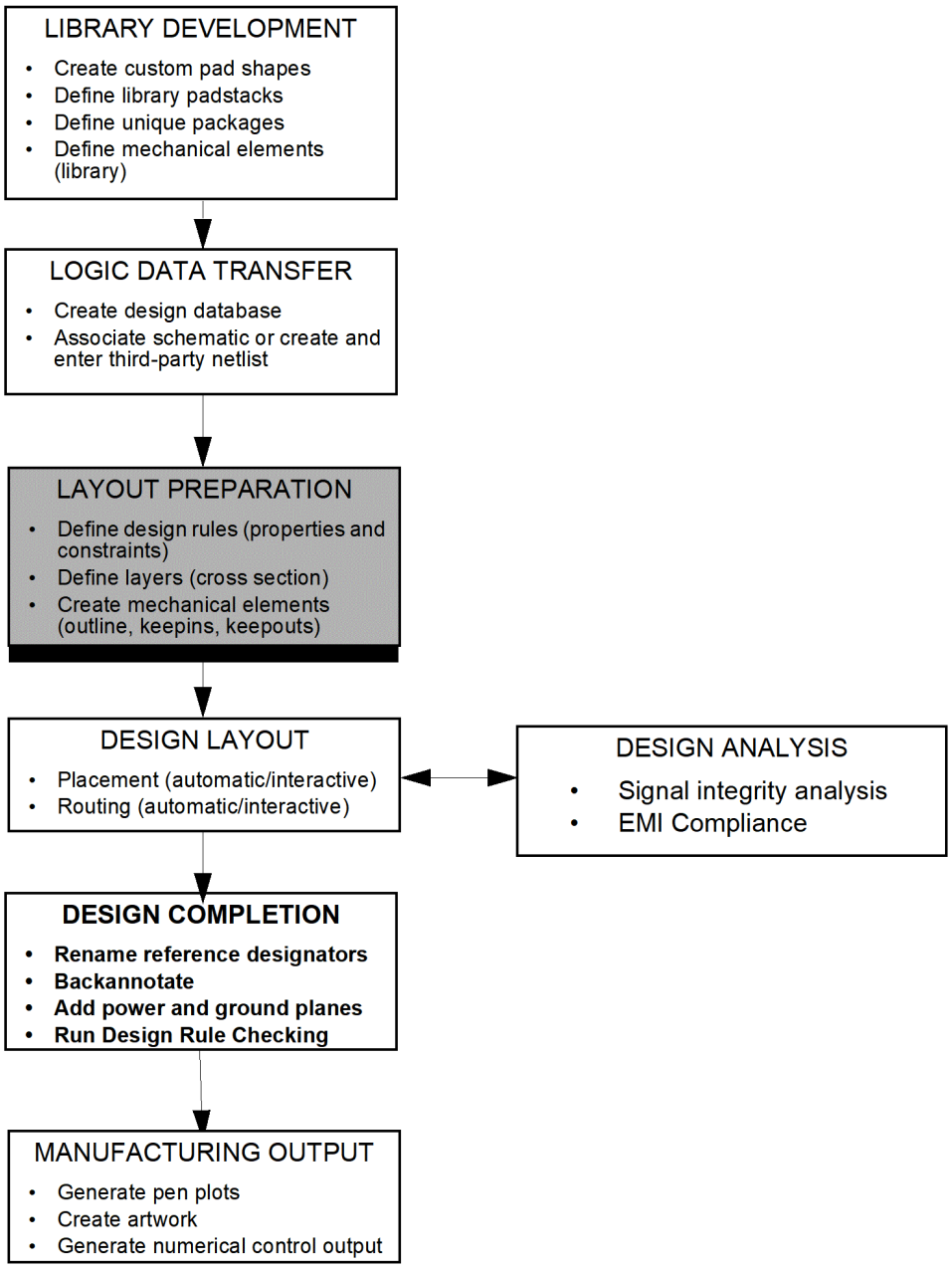
This document describes how to create design rules and prepare for creating a design. These topics are included:

- Establishing and modifying the rules that govern how operates on design elements, specifically:
  - Design Rule Checking (DRC)
  - Properties
  - Constraints
- Defining the layout cross-section

⚠ Many features are common to all three layout editors: Allegro 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.

You should set up design rules as part of your preparation for layout. The following figure illustrates where you would normally incorporate design rules in the design flow.

Design Rules in a PCB Editor Design Flow


LIBRARY DEVELOPMENT
• Create custom pad shapes
• Define library padstacks
• Define unique packages
• Define mechanical elements (library)
LOGIC DATA TRANSFER
• Create design database
• Associate schematic or create and enter third-party netlist
LAYOUT PREPARATION
• Define design rules (properties and constraints)
• Define layers (cross section)
• Create mechanical elements (outline, keepins, keepouts)
DESIGN LAYOUT
• Placement (automatic/interactive)
• Routing (automatic/interactive)
DESIGN ANALYSIS
• Signal integrity analysis
• EMI Compliance
DESIGN COMPLETION
• Rename reference designators
• Backannotate
• Add power and ground planes
• Run Design Rule Checking
MANUFACTURING OUTPUT
• Generate pen plots
• Create artwork
• Generate numerical control output

1

chap1

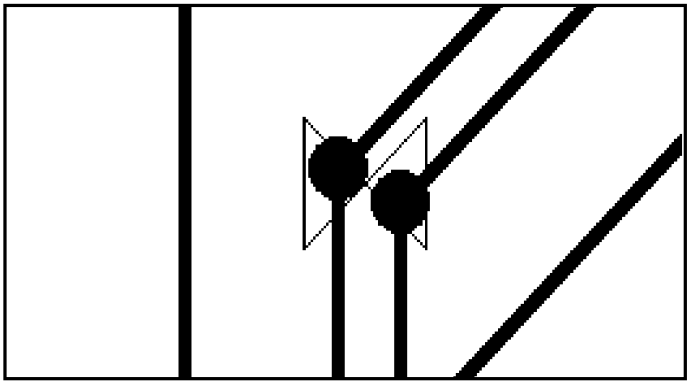
About Design Rule Checking

As part of preparation for layout, you should set up design rules. The layout editor performs design rule checking with Design Rule Check (DRC) to ensure that the design conforms to specified properties and constraints you attach to individual design elements or assign globally to the entire design. (Properties are described in Chapter 2, "Working with Properties," and constraints in Chapter 3, "Working with Constraints.") Design rule checking identifies violations of physical design rules whenever you add an element or make any change to the design.

You can check for violations in real-time as you design (called online DRC error), or in batch mode (called batch DRC error). You may prefer the instant feedback of online DRC, at the expense of system performance, or you may prefer to use batch mode to improve system performance and decide to resolve violations later in the design process.

When the layout editor detects a design rule violation, the offending design element is flagged with the appropriate design rule violation marker (bow tie), shown below.

Figure 1.1: Sample DRC Marker



DRC Modes

You control when DRC applies to each constraint by applying one of the following DRC modes:

On (or Always)	Run DRC for the constraint during interactive commands. This setting is for those checks you want to run while you modify design elements. The more constraints you check interactively, the slower your system performs during editing. The layout editor checks the constraint when you set DRC to run online in the Status dialog box or when you choose *Tools – Update DRC* (drc update command). The dialog box, menu item, and command are detailed in the *Allegro PCB and Package Physical Layout Command Reference*.
Off (or Never)	Do not perform DRC for the constraint. Use this setting for checks that your design process does not require. This enhances interactive performance. ⚠ Positive shapes (static or dynamic) void to a Cadence default of `13 mil` (or design unit equivalent) when the DRC mode is set to *Never*.

You can also globally disable online DRC in the Status dialog box or in the Electrical Modes tab of the Analysis Modes dialog box, available by choosing *Analyze – Analysis Modes* in Constraint Manager.

The DRC mode settings you choose depend on the design complexity, CPU performance, and available memory. In general, Cadence suggests that you use online DRC for these checks:

- All spacing, physical, and global package checks
- Electrical
    - Stub length
    - Net schedule
    - Max via count
    - Max exposed length
    - Propagation delay

A DRC mode cannot be different for specific constraint areas nor for a particular constraint in a constraint set. It can be different for various ETCH/CONDUCTOR subclasses. A single setting of the DRC mode applies to all instances of a constraint, such as Line to Line Spacing on an ETCH/CONDUCTOR subclass.

DRC Status

DRC checks may become out of date when you change a constraint or property value. The layout editor displays the status of the last DRC check in the Status dialog box, accessed by choosing *Display – Status* (`status` command).

DRC status can be:

Up To Date	All DRC checks (*On*) have been done and there have been no changes to any constraints or properties since the last full DRC update.
Out Of Date	One or more constraints set to the *On* DRC mode has been changed, properties have been changed, the DRC process was cancelled by a user command (`Control-C`, `Esc`, or the Stop button), or online DRC has been turned off globally in the Status dialog box or in the *Electrical Modes* tab of the Analysis Modes dialog box, available by choosing *Analyze – Analysis Modes* in Constraint Manager.

Making DRC Errors Visible

Before running design rule checking, ensure that any DRC violations are visible.

1. Choose *Display – Color/Visibility* (color192 command).

The Color dialog box appears.

1. Choose *Stack-Up*.
2. Check that the *DRC* box is chosen for *All* (all layers).
3. Click *OK*.

Running Online DRC

Online DRC provides immediate feedback when you violate a design rule.

1. Choose *Display – Status* (`status` command).

The Status dialog box appears.

1. Click the *On-Line DRC* button.
2. Click *OK*.

Running Batch DRC

Batch DRC allows you to do initial layout and then run DRC checking on the whole design at once. You can run batch design rule checking using any of the following:

- Within the layout editor, run *Tools – Database Check* (dbdoctor command) and enable the *Update all DRC (including batch)* option
- Launch the dbdoctor_ui command externally.

⚠ It is recommended that you run batch DRC only for island checks and parallel and cross talk checks if the design is complex.

Updating DRC

If you turned off online DRC in the Status dialog box, you can reactivate it and run DRC simultaneously. Choose *Tools – Update DRC* (drc update command), described in the *Allegro PCB and Package Physical Layout Command Reference.*

Controlling the Display of DRC Markers

The layout editor flags violations of design rules by displaying DRC marker(s), shaped like bow ties. DRC markers appear as non-filled outlines by default, as shown in Figure 1-1.

To display filled markers, do any of the following:

- Set the `display_drcfill` environment variable in the Display category of the User Preferences Editor choosing *Setup – User Preferences* (enved command), described in the *Allegro PCB and Package Physical Layout Command Reference.*
- Type `set display_drcfill` at the console window prompt.
- Enter `set display_drcfill` in your local environment file.

The default marker size is in user units (25 mils, for example). Change the size of the DRC marker in the in the *Display* tab of the *Design Parameter Editor*, available by choosing *Setup – Design Parameters* (prmed command).

Displaying Information About DRC Violations

DRC markers store the following information about a DRC:

- DRC class, subclass, and location
- Type of constraint set (spacing, physical, or electrical)
- Name of constraint set
- Constraint type being violated (for example, Line to Thru Pin Spacing)
- Data concerning first element in violation (type of element, location, refdes, if a package, and so on)
- Data concerning any second element in violation (type of element, location, refdes, if a package, and so on)

Viewing Information for a Specific DRC Marker

Before you can see information for a specific DRC marker, make sure that the visibility for the DRC layer class is turned on. For details, see Making DRC Errors Visible.

To see details about a DRC error:

1. Choose *Display – Element* (show element command).
2. In the Find Filter, make sure that *DRC errors* is checked.
3. Click on a DRC marker in the design.

The elements associated with the DRC highlight, and the Show Element dialog box displays information about the chosen marker.

Displaying the DRC Error Report

You can display a report listing all DRC errors in the design.

1. Choose *Tools – Reports* (`reports` command).
2. Choose *Design Rules Check* from the list in the Reports dialog box.
3. Click *Report*.

Waiving Design Rule Check Errors

Often you may need to set aside a design rule to meet design requirements. Waive DRC allows you to flag these violations as acceptable, and attach an explanatory comment to ensure that those working with the design later on understand the rationale concerning the existence of any given DRC violation.

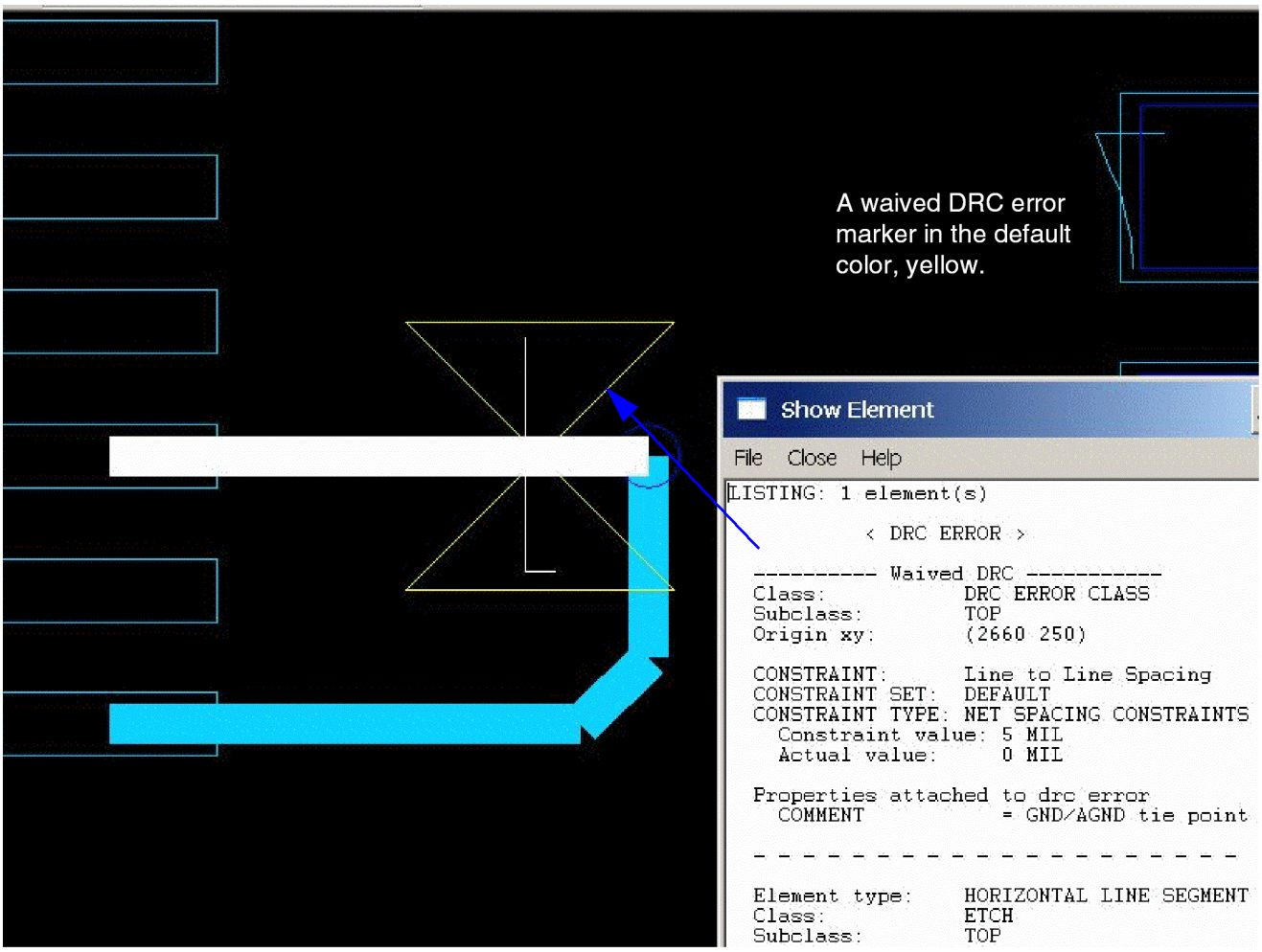
When you want to mark design rule violations as approved for the current design, the layout editor lets you:

- Waive DRC errors either globally or by individual pick (`waive drc` command)
- Choose whether to display waived DRC errors in the design (`show waived drcs` and `blank waived drcs` commands)
- Assign a color to waived DRC error markers that differs from active DRC error markers in the design using *Display – Color/Visibility* (color192 command)
- Restore waived DRCs errors to active status either globally or by individual pick (`restore waived drcs` and `restore waived drc` commands)
- Attach comments to waived DRC errors
- Generate a report that lists all waived DRC errors in the design

For more information about waiving DRC errors, see the waive drc command in the *Allegro PCB and Package Physical Layout Command Reference*.

Waiving a DRC Error Marker

When you waive a design rule violation, the layout editor flags the design element with a waived design rule error marker (rotated bow tie), as shown in Figure 1-2. Waiving a DRC error updates the count of DRC errors and waived DRC errors in the *Status* tab of the Status dialog box.
Figure 1.2: Waived DRC Error Marker



Making Waived DRC Errors Visible

ⓘ By default, *Waived DRCs* is disabled in the *Display* tab of the *Design Parameter Editor*, available by choosing *Setup – Design Parameters* (prmed command).

Before a waived DRC error marker is visible in the design, you must enable the *Waived DRCs* check box in the Design Parameter Editor and DRC visibility in the Color dialog box. If the latter is not enabled, waived DRC errors are invisible, regardless of whether you enable the *Waived DRCs* check box in the Design Parameter Editor.

The layout editor lets you show waived DRC errors when you:

- Enable the *Waived DRCs* check box in the Design Parameter Editor
- Use the show waived drcs command

For more information, see *Setup – Design Parameters* (prmed command) and *Display – Color/Visibility* (color192 command) in the *Allegro PCB and Package Physical Layout Command Reference*.

Waived DRC Error Behavior

Waive DRC suppresses design rule violations reported by Allegro. In certain cases, waived DRC markers may become stale. With a stale waived DRC marker, the underlying design rule violation no longer exists. Performing a DRC update with *Tools – Update DRC* (drc update command) removes any stale waived DRCs.

For example, if you waived a via-to-via spacing violation and then move one of the vias beyond the via-to-via spacing requirement, a stale waive DRC occurs. The waived DRC appears until you perform a DRC update.

For disabled DRC checks, any associated waive DRCs are deleted as well.

When waiving a differential pair DRC error, the DRC_ERROR_CLASS cline marker segments are hidden. External Waived DRC markers are never deleted.

Adding Comments to a Waived DRC Error

The layout editor lets you attach a comment to a waived DRC error. The comment then appears in the Show Element dialog box information for that waived DRC error.

Generating the Waived DRC Error Report

You can generate a report listing all waived DRC errors in a design.

1. Choose *Tools – Reports* (reports command).
2. Double-click *Waived Design Rules Check Report* from the list in the Reports dialog box.
3. Click *Report*.
   The report appears with information on all waived DRC errors present in the design.

DRC Suppression

Sometimes, you must override the rules in your design. The following Boolean properties disable DRC checking. Cadence recommends that you apply these properties at the end of the design cycle because they override DRC checks.

- NODRC_COMPONENT_BOARD_OVERLAP
- NODRC_ETCH_OUTSIDE_KEEPIN
- NODRC_VIAS_OUTSIDE_KEEPIN

2

chap2

Working with Properties

You attach properties to elements in a design to instruct the layout editor how to operate on those design elements. For example, the glossing router has no effect on a net with the NO_GLOSS property attached. Design Rule Check (DRC) adds markers to the design wherever violations occur. For details about DRC, see About Design Rule Checking. The layout editor has more than 180 predefined property types, described in the *Allegro Platform Properties Reference*.

Properties also override design elements with attached constraints. For detailed information on constraints, see Working with Constraints.

You can attach properties:

- Directly to a design element in the layout design.
- To an Allegro Design Entry HDL or System Connectivity Manager schematic on your UNIX workstation that you import into the layout editor. For additional information, see the *Transferring Logic Design Data* user guide in your documentation set.
- Through Allegro Design Entry CIS or a third-party netlist that you would import into the layout editor. For additional information, see the *Transferring Logic Design Data* user guide in your documentation set.

You can also define your own property types (user-defined properties) by assigning each type a new name and telling the layout editor what type of data the property represents (user units, integers, strings, or boolean) and to what types of elements the property can be attached (for example, nets, pins, or symbols).

Types of Values Used to Define Properties

Properties are defined by values of a type appropriate to their use. For example, the MIN_LINE_WIDTH property has a value giving the minimum line width expressed in user units. The NO_PIN_ESCAPE property has a boolean value—either true or false.

All properties have one of the following type of value:

STRING	LAYER_THICKNESS
INTEGER	IMPEDANCE
REAL	INDUCTANCE
DESIGN_UNITS	PROP_DELAY
BOOLEAN (true or false)	RESISTANCE
ALTITUDE	TEMPERATURE
CAPACITANCE	THERM_CONDUCTANCE
DISTANCE	THERM_CONDUCTIVITY
ELEC_CONDUCTIVITY	VOLTAGE

Types of Elements to Which You Can Attach Properties

CLINES (connect lines)	LINES
COMPONENT DEFINITIONS	NETS
COMPONENT INSTANCES	PIN DEFINITIONS
DEFINITIONS	PINS
DRCS	RECTANGLES
FIGURES	SHAPES
FRECTANGLES	SYMBOLS
FUNCTION	VIAS
FUNCTIONS	VOIDS
LAYOUT (a property attached to the entire design)	GROUPS

Extracting Property Values into a Text File

You may need data about the properties in your the layout design for processing with your own programs.The `extracta` command lets you extract property values from a design and place them into a text file. The command can extract both pre-defined and user-defined property values.

For example, you have defined a string property MY_PART_NUMBER, that can be attached to components. You would use `extracta` with the following command file to extract the reference designator and the MY_PART_NUMBER property value from each component in your layout.

```
#
#    This extract command file causes extraction of:
#    the user defined property MY_PART_NUMBER
#    and the reference designator
#    from each component in a layout.
#
COMPONENT
#
REFDES
MY_PART_NUMBER
#
END
```

If you had attached the property to a higher level element that owns the actual element (for example, the component definition of the component being extracted), `extracta` would still find the value and extract it.

For additional information, see the *Completing the Design* user guide in your documentation set.

Assigning Properties to Design Elements

Using *Edit – Properties* (property edit command), you can attach, edit, and delete properties on the following types of design elements:

Clines	Nets
Comps	Pins
Functions	Shapes
Groups	

Displaying Properties

Choose *Display – Property* (show property command) and use the Information tab on the Show Property dialog box to review properties attached to design elements or to show a property definition.

Use the *Graphics* tab on the Show Property dialog box to create text to visually identify one or more user-defined properties, for example, package height. Text location varies according to the type of property.

Property type	Text location
Component defined properties	Each placed symbol instance of that definition.
Lines, shapes, pins	Object.
Nets, XNets, diffpairs, buses	One of the pins.
Properties with or without values	Property name. For example, `<property_name>= <property_value>`

Property Inheritance

In the layout editor, the FIXED property is an inherited property that you attach to one object (a parent), which then becomes inherited by other objects (children). When you apply an inherited property to an object, you define the behavior of that object. That behavior then defines the behavior of each child object; those objects that exist beneath it in a logical hierarchy (see Figure 2-1).

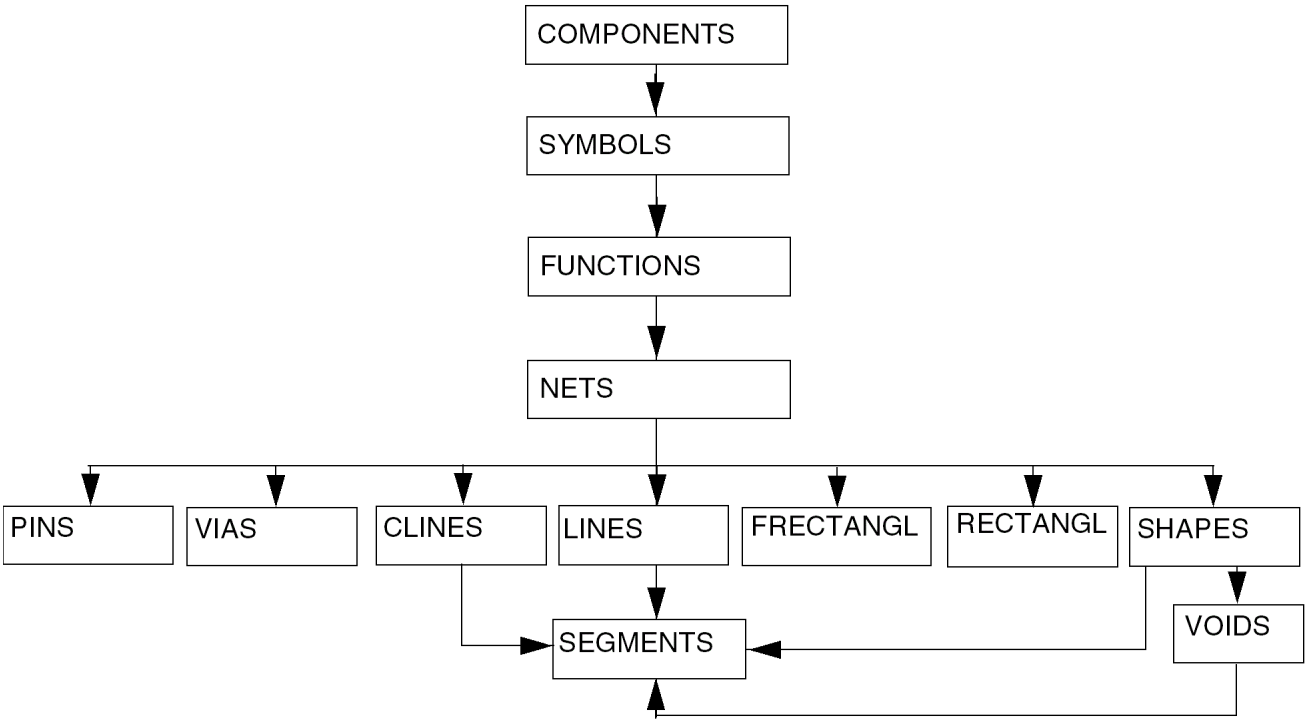
When you apply the FIXED property to a symbol, you prevent the movement or deletion of that symbol. Because it is an inherited property, you also prevent editing of the symbol children.

Design objects that can inherit the FIXED property are:

Connect lines	Nets	Symbols
Component instances	Rectangles	Symbol pins
Filled rectangles	Shapes	Vias
Lines		

The layout editor searches objects for inherited properties along a pre-defined search path, shown in Figure 2-1. Notice that the precedence mirrors the display of objects found in the the layout editor *Find* Filter.

Figure 2.1: Inherited Precedence



In Figure 2-1, you can see that a pin is part of a net, but is also part of a symbol. If you add the FIXED property to a net, you cannot move the symbol because the pins in the symbol inherit the FIXED property from the net.

Creating an Inherited Property

You create an inherited property with *Edit – Properties* (property edit command) in the same way that you attach any property to an object, using the procedure, Creating an Inherited Property, described in the *Allegro PCB and Package Physical Layout Command Reference*.

Displaying Inherited Properties on an Element

You can view inherited properties on a design element by choosing *Display – Element* (show element command), described in the *Allegro PCB and Package Physical Layout Command Reference*.

Changing Inherited Properties

You can modify inherited properties (add or remove) only at the parent object level. You cannot make changes at the child level. To make changes to an inherited property, choose *Edit – Properties* (property edit command), described in the *Allegro PCB and Package Physical Layout Command Reference*.

Creating and Editing User-Defined Properties

You can create and maintain properties of your own in the layout designs. Then you can use these properties for reporting and data extraction.

You can display user-defined properties by using the *Filter* button in the Show Property dialog box (`show property` command) or any other function or report that displays property assignments.

The layout editor recognizes a user-defined property assigned to an element during logic entry in Allegro Design Entry HDL or System Connectivity Manager if it has been defined in the layout editor before executing backannotation. The definition must be an exact match of the name and pass edit checks of element type, units, and range.

In all other instances, the layout editor recognizes user-defined properties only as information labels for your purposes. These properties do not affect automatic or interactive operations and do not generate DRC markers. You can create and use as many property definitions as you need without being concerned about their effect on the layout editor.

Defining User-Defined Properties

You can define a property in two ways:

- Interactively, choosing *Setup – Property Definitions* (`define property` command), described in the*Allegro PCB and Package Physical Layout Command Reference.*
- In a technology file, described in the *Defining and Developing Libraries* user guide of your documentation set.

Editing User Properties

You can change a property definition, delete a property, or copy a current definition to a new property definition by choosing *Setup – Property Definitions* (define property command), described in the *Allegro PCB and Package Physical Layout Command Reference*.

The layout editor lets you change and delete properties that are not currently in use and have not been defined through a technology file:

- This error message appears for a property that is currently assigned to elements:

  Property <name> is currently associated with elements, cannot modify/delete property definition.

- This error message appears for a property defined through a technology file:

  Property <name> defined through a technology file, cannot modify/delete property definition.

Storing Web Links as the Value of a Property

The layout editor supports the storing of web links as the value of a user-defined property. You can use this feature to access a specification for the component to which the property is attached.

1. Choose *Setup – Property Definitions* (define property command) to define the property, as described in the *Allegro PCB and Package Physical Layout Command Reference.*
2. Choose *Edit – Properties* (property edit command) to add set the value of the property. In this case, set the value to be the complete browser path to the destination.
3. Choose the component and use the *Display – Element* (show element command) to access the web link.
4. Click on the web link.

⚠ Ensure that you set the `allegro_html` environment variable in the User Preferences Editor. Use *Setup – User Preferences* (enved command) and click *UI*.

3

chap3

Working with Constraints

A constraint is a user-defined rule applied by Design Rule Check (DRC) to a physical element in a design. When you define and apply a constraint, the layout editor adheres to that constraint during automatic and interactive processing and flags violations with DRC markers.

Cadence recommends using Constraint Manager to specify a variety of constraints. Choose *Setup – Constraints – Constraint Manager* (cmgr command) to access Constraint Manager. Constraint Manager provides worksheets for specifying Electrical, Physical, and Spacing constraints, as well as general DRC values and Property assignments. Basic Spacing and Physical constraints are called standard or default values.

Constraint Manager provides the following functionality:

- Creating topology files to use with electrical constraint sets
- Importing electrical constraint sets
- Assigning electrical constraint sets to buses, differential pairs, and XNets

For additional information on Constraint Manager, see the *Allegro Constraint Manager User Guide*. For detailed descriptions of the constraints, see the Allegro Platform Constraints Reference.

You control when DRC runs for a constraint by setting a DRC mode on that constraint. See "DRC Modes" for definitions of each mode and tips for setting them.

Lower tiers of the layout editor restrict access to certain types of constraints. For example, if you set electrical DRCs in a high-tier version of the layout editor, and then open the design in a lower tier, the layout editor preserves the constraint data, but may disable DRC. When you again open the design in the higher tier, DRC functions, but the DRC status is out-of-date. You need to run a DRC update to have the DRC markers reflect the current design state.

Types of Constraint Sets

The layout editor organizes design rules, according to the behavior and type of element to which they apply, into the following predefined constraint sets

Spacing	Constraints governing the spacing between elements on different nets (for example, Line To Thru Pin Spacing).
Same Net Spacing	Constraints governing the spacing between elements on the same net
Physical	Constraints governing physical construction of a net (for example, Minimum Line Width and Allowed ETCH/CONDUCTOR Layers).
Electrical	Constraints governing electrical behavior of an entire net (for example, Minimum Propagation Delay) and differential pairs (for example, *Primary Gap*).
Design	Board-level constraints that are non-area and non-net in nature (for example, Negative Plane Islands). Constraints that flag a set of vias or pins with overlapping antipads or thermal pads that cut off a piece of a negative shape.

The layout editor designs begin with default constraint sets (named DEFAULT) for spacing and physical constraints. However, electrical constraint sets do not have a default.

The layout editor assigns nets without electrical constraint sets to the constraint set UPREV_DEFAULT in designs that you uprev to version 15.x.

You can edit the spacing and physical default constraint sets and specify where and to what elements each constraint applies. You can also assign height information to package symbol files (`.dra`) and to package keepin and package keepout areas of a board file (`.brd`) or substrate design file (`.mcm`).

⚠ In most cases, Class membership are different between Electrical and Physical/Spacing Domains. In other words, a physical class may contain more, less, or even different members than what is required for an electrical class member to drive specific rules. For this reason, Net Classes are different between the Physical, Spacing and Electrical worksheets. The only exception is a Net Class which can be created to exist in both the Physical and Spacing worksheets as in most cases the same members share the same Physical/Spacing rules. Creating a NetGroup will be available across the Physical, Spacing and Electrical worksheets so rules could be applied to the same group of nets. If class-based rules are required than the NetGroup can be added to a Net Class for those purposes.

Spacing Constraint Sets

A spacing constraint set (also called a spacing rule set) defines for each ETCH/CONDUCTOR subclass the spacing between pairs of elements on the subclass and the controls for same-net checking.

When checking design rules, a DRC does not check pins (thru pins, test pins, smd pins) against route keepins or keepouts.

⚠ Spacing Constraints are organized by net-to-net (*Spacing* domain) and same net (*Same Net Spacing* domain) in Constraint Manager.

For information on how to define a spacing constraint set, see the *Allegro Constraint Manager User Guide* and the Allegro Platform Constraints Reference.

Physical Constraint Sets

A physical constraint set defines rules that apply as you create the interconnections of an individual net. For example, physical constraints specify:

- Connect line (cline) widths to use in different areas and on different layers
- Allowable connections on a particular layer
- Acceptable vias to use with a particular net

This differs from the spacing constraint set, which specifies spacing rules between pairs of net elements.

The layout editor uses these constraints to construct and check clines for minimum line width, maximum line width, minimum neck width, and maximum neck length. During routing, the layout editor adds each cline segment using the minimum line width. When you choose Neck from the pop-up menu, the layout editor adds the next cline segment with minimum neck width. A DRC places an error marker on a cline segment for any of the following violations:

- Less than minimum neck width
- Less than minimum neck width and longer than maximum neck length
- Greater than the maximum line width

You can also attach properties to a net that correspond to physical constraint sets. Property values override values in the physical constraint set.

For information on how to define a physical constraint set, see the *Allegro Constraint Manager User Guide* and the Allegro Platform Constraints Reference.

Electrical Constraint Sets

Electrical constraint sets contain rules controlling the electrical behavior of a net, bus or differential pair, for example, timing and noise tolerance.

Electrical constraints apply to all parts of a net, regardless of ETCH/CONDUCTOR subclass or layout area. Therefore, constraint areas do not apply to electrical constraints, and there is no assignment table for them. Instead, when you assign an electrical constraint set to a net, the ELECTRICAL_CONSTRAINT_SET property attaches to the net that is set to the name of that constraint set.

You can also attach properties to nets that correspond to electrical constraints. Property values override values in the electrical constraint set.

⚠ Electrical constraints are not available in Allegro PCB Design L

For information on how to define an electrical constraint set, see the *Allegro Constraint Manager User Guide* and the Allegro Platform Constraints Reference.

Design Constraints

Design constraints comprise board-level constraints for which you can run a DRC. The layout editor considers placement keepins and keepouts design constraints, and as such they have a fixed spacing value of `0 mil`. A DRC considers ETCH/CONDUCTOR elements and package symbols in violation when they touch or are outside/inside a keepin/keepout. Similarly, the layout editor considers package to package placement a design constraint and has a fixed spacing value of `0 mil`.

Design constraints also let you identify potential alignment and spacing problems for soldermask openings within a symbol or pin or for pad soldermask to nearby pad or ETCH/CONDUCTOR soldermask. The soldermask constraints are not area-dependent.

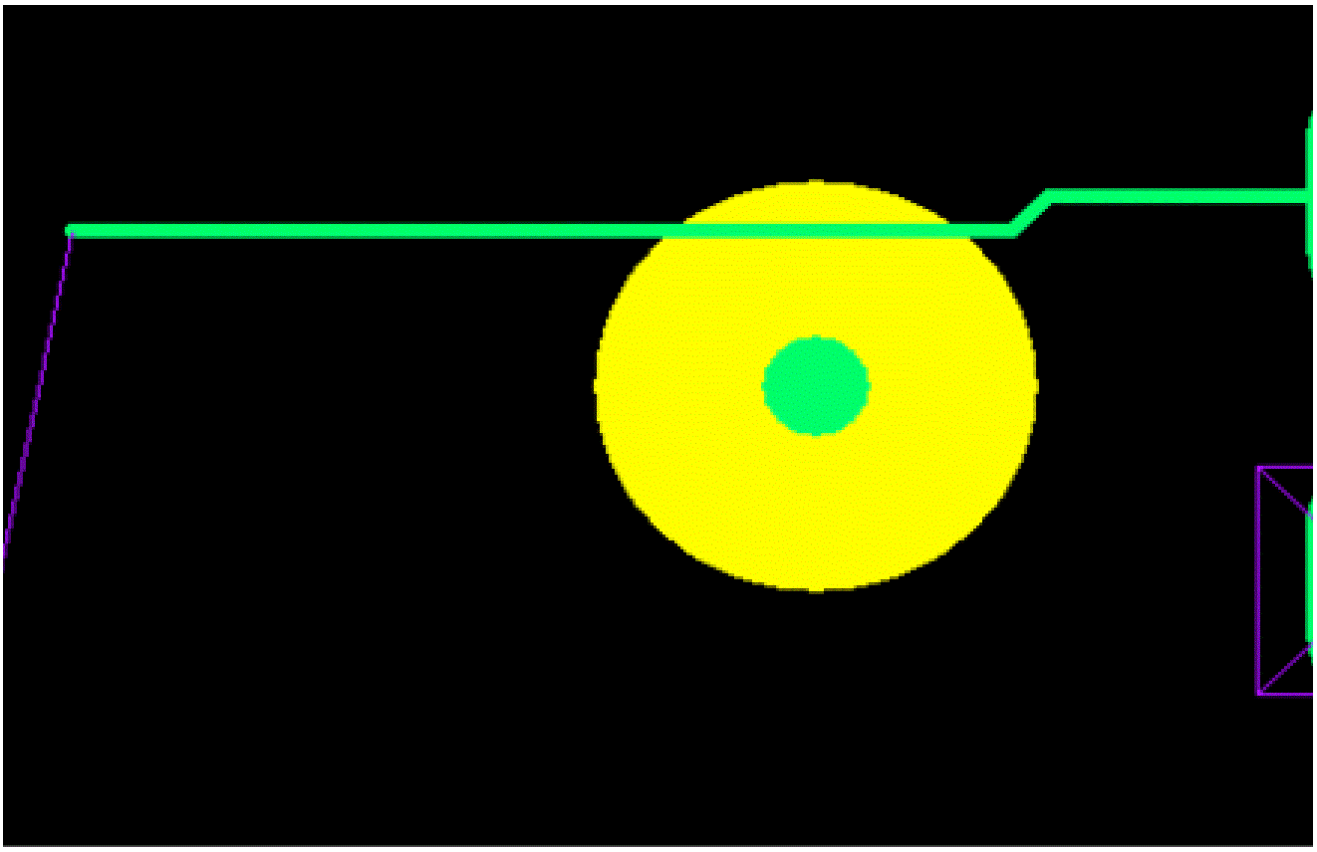
For information on how to define a design constraint set, see *Allegro Constraint Manager User Guide* and the Allegro Platform Constraints Reference.

The Soldermask Design Rule Check

External layer copper or etch/conductor that is not protected by a soldermask coating is considered exposed. Exposed copper can cause decay of the trace due to contamination buildup or acid cleaning as well as shorts in the design. Typically, using the symbol library, you build soldermask openings (on the soldermask layers) associated with the top- and bottom-side component pads. You can make these pads the same size as the surface pads or larger.

The Soldermask Design Rule Check reports exposed copper or etch/conductor on the outer layers with DRC errors. It checks the TOP_COND layer with the Soldermask_Top layer in the Substrate Geometry and the Pin and Via classes of the design. It also checks the BOT_COND layer with the Soldermask_Bottom layer in the Substrate Geometry and Pin and Via classes of the design. Any copper or etch/conductor that infringes on a soldermask opening generates a DRC error.

Example of Conductor Infringing on Soldermask Opening



Soldermask Design Rule Check Parameters

The Soldermask Design Rule Check supports these optional soldermask-to-copper or etch/conductor spacing parameters:

- Soldermask-to-pad and cline
- Soldermask-to-shape

To access the soldermask-to-copper or ETCH/CONDUCTOR spacing parameters, choose *Setup – Constraints – Modes* (`cns modes` command, or alternately `cns design` command) from the menu and click the *Design Constraints* tab. Refer to the cns design command in the *Allegro PCB and Package Physical Layout Command Reference* for the description of the soldermask-to-copper or etch/conductor parameters found in the Design Constraints dialog box.

Generation of Soldermask DRC Errors

The following table summarizes the situations when soldermask DRC errors are generated.

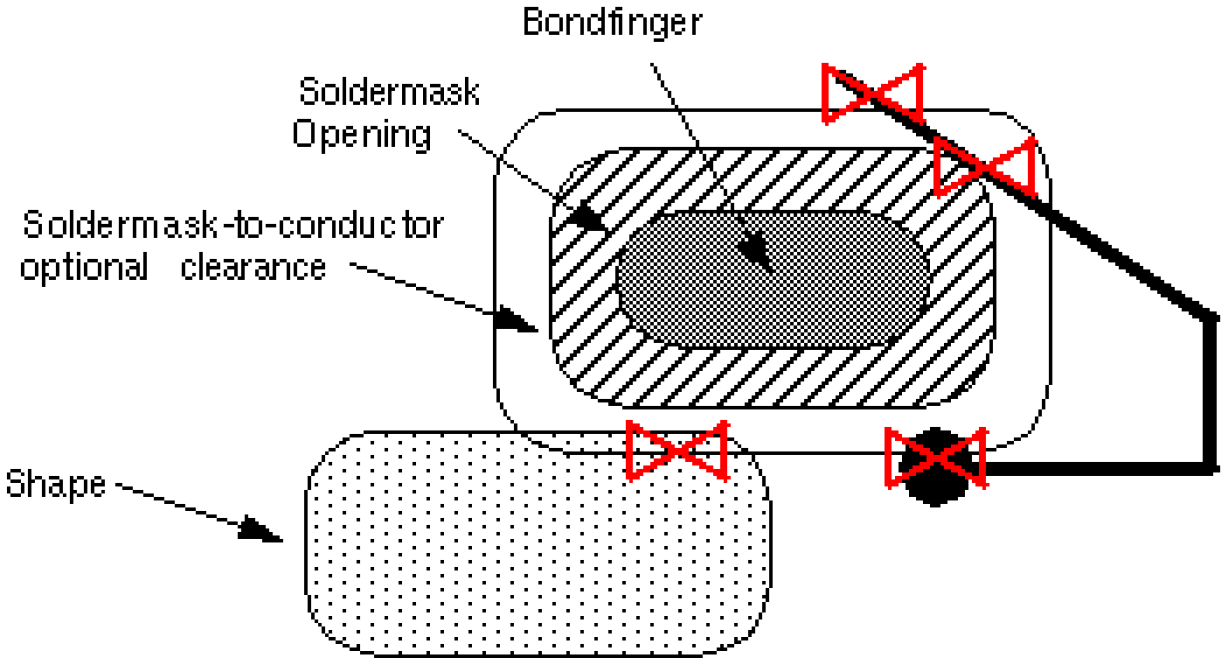
⚠ Constraint Manager displays the soldermask DRC errors in the DRC Design worksheet.

For....	A Soldermask DRC occurs when...
Pads	The pad is not entirely within the soldermask-to-pad and cline clearance. Overlap of the pad with the clearance results in a DRC error.
Clines	Any cline comes within the soldermask opening or the optional soldermask-to-pad and cline clearance. However, there are two exceptions. The layout tool suppresses soldermask design rule checks on clines: • When the cline connects to a pad that has a soldermask opening. • When the cline has the FILLET property attached.
Shapes	Any copper or etch/conductor shape comes within the soldermask opening or the optional soldermask-to-shape clearance. The layout tool suppresses soldermask design rule checks on clines when the shape connects to a pad that has a soldermask opening.

⚠ For Lines and text, no soldermask DRC error occurs.

The following figure shows the DRC errors generated during a soldermask design rule check.

APD+: Soldermask DRC Errors



Soldermask DRC Text Markers

The soldermask DRC marker appears on the DRC TOP_ETCH/COND or BOT_ETCH/COND subclass matching the copper or etch/conductor subclass object in violation. The layout tool generates a single error for each cline, shape, or pad. The following are soldermask DRC text markers:

- M-L: Soldermask to Cline
- M-V: Soldermask to Via
- M-P: Soldermask to Pin
- M-S: Soldermask to Shape

Differential Pairs

The layout editor supports routing physical and electrical DRCs for edge-coupled differential pairs, that is, a pair of nets or XNets routed side-by-side on the same layers of a board. You can set up differential pairs as an electrical constraint set, or if you require a more robust line and space gap control, you can use a combination of physical and electrical constraint sets.

The following sections describe the basic flow for defining differential pairs:

- Designating Nets as Differential Pairs
- Assigning Electrical Constraint Sets to Differential Pairs
- Enabling a DRC

Designating Nets as Differential Pairs

You can create differential pairs in the following ways:

- Assign nets you want routed as differential pairs by choosing *Logic – Assign Differential Pair* (diff pairs command). You can assign nets individually or in groups using the net naming conventions.
- Import differential pair logic from a schematic by choosing *File – Import – Logic* (netin command).
- Designate nets in a signal model as differential pairs. Refer to Model and Library Management in the *Allegro PCB SI User Guide* for details.

Assigning Electrical Constraint Sets to Differential Pairs

You can assign electrical constraint sets (ECSets) to differential pairs at the net or XNet level or the differential pair object level.

If you assign ECSets at both the net or XNet level and the differential pair object level, a DRC uses the differential pair constraints at the differential pair object level.

If you assign different ECSets to each of the member nets or XNets, a DRC uses the ECSet with the most conservative settings:

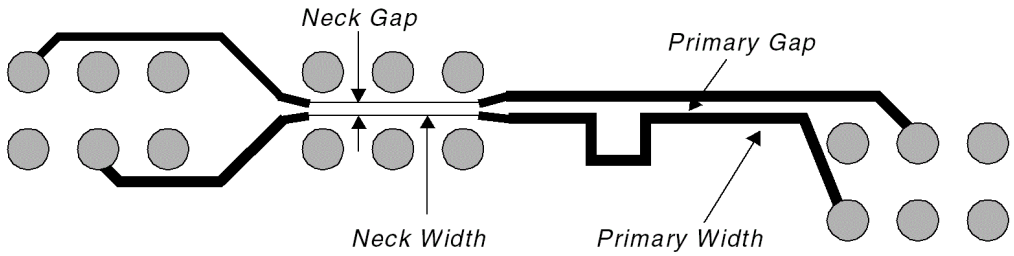
- Smallest phase tolerance
- Smallest gaps and gap tolerances
- *Gather control* set to *Include* takes precedence over Ignore

Note:

- Properties override ECSet settings.
- When you set differential pair properties on nets, nearly all of these properties rise up to the XNet and differential pair group levels. The differential pair properties begin with DIFFP_ and correspond to the constraints in the *DiffPair Values* tab of the Electrical Constraints dialog box. The MIN_LINE_WIDTH and MIN_NECK_WIDTH properties (*Primary Line Width* and *Neck Width* in the Electrical Constraints dialog box), also rise up to the XNet and differential pair group levels.

The following figure shows the various parameters used for controlling a differential pair:

Controlling a Differential Pair



+Tolerance and *-Tolerance* determine when lines are uncoupled

Enabling a DRC

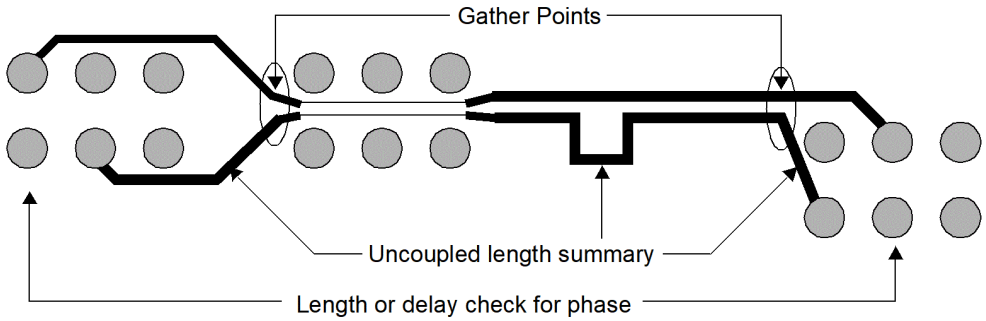
If not done in the previous step, enable a DRC for the differential pair constraints and update the DRC. See Setting DRC Modes for the Electrical Constraint Set in the *Allegro PCB and Package Physical Layout Command Reference*. Set *All Differential pair checks* to *On* and choose *Tools – Update DRC* (drc update command), so that the DRC markers reflect the current design state.

You can set and verify the following DRCs, shown in the following figure, based on the parameters in the figure above:

- Segment coupling (either primary gap or neck gap plus or minus tolerance, if defined). The length of uncoupled segments adds to the total of uncoupled length. This total is then compared to the *Max uncoupled length* constraint.
- Tolerance check comparison of the length or delay between corresponding

driver/receiver pairs of the two differential pair nets or XNets.

- Minimum segment-to-segment spacing.
  A minimum segment-to-segment spacing check is based on the specified differential pair minimum line spacing constraint or the *Line To Line* spacing in the Spacing rule set if a minimum spacing is not defined for the differential pair.
  Setting and Verifying DRCs



Defining Differential Pairs by Layer

The following describes the basic flow for defining differential pairs by layer:

- Designating Nets as Differential Pairs
- Assigning Electrical Constraint Sets for Differential Pairs
- Creating a Physical Constraint Set
- Enabling a DRC

Designating Nets as Differential Pairs

See Designating Nets as Differential Pairs for detailed information.

Assigning Electrical Constraint Sets for Differential Pairs

To define differential pair gap and neck width by layer, choose *Setup – Constraints – Physical* (`cmgr_phys` command). In the tree view, expand the *Physical Constraint Set* folder, then click the *All Layers* icon. Enter values under the *Differential Pair Gap - Primary* and *Differential Pair Gap - Neck* columns for each layer in the design.

ⓘ Differential Pair values that you enter in the Electrical worksheet of Constraint Manager (*Setup – Constraints – Electrical* or `cmgr_elec` command) will override the values you set in the Physical worksheet.

⚠ You do not need to perform the following steps if all the differential pairs are using the same *Primary gap* and *Line width* values from the default Physical Constraint Set.

Creating a Physical Constraint Set

Create a physical constraint set in the Physical worksheet of Constraint Manager by choosing *Setup – Constraints – Physical* (`cmgr_phys` command). Enter values for *Differential Pair Gap - Primary*, *Differential Pair Gap - Neck*, *Line Width - Min*, *Line Width - Max*, and *Neck - Min Width*.

Enabling a DRC

See Enabling a DRC for information.

Using Constraint Values in Routing and Checking Differential Pairs

When routing or checking differential pairs, the layout editor determines the following:

- Whether to use a primary gap or a neck gap
- The primary line width
- Neck line width

Primary Gap or Neck Gap

The layout editor uses *Neck gap* over *Primary gap* when the line width of the differential pair is less than the value of the *Primary line width*.

Special necking is implemented when the line width constraint equals the neck width constraint for nets, but the *Neck gap* is less than the *Primary gap*. A segment that comes within the neck gap plus the (+) tolerance to segments of its diff pair partner are considered necked for both the differential pair DRC check and the differential pair line width check.

- For the uncoupling check, the neck gap band is the defining coupled band.
- For the dynamic phase check, the neck gap band is the defining band for coupling/uncoupling events.

- For the neck length check, these segments are considered necked and must not exceed the neck length.

When using *Primary gap* during routing or checking, the layout editor looks for the following in sequential order:

1. DIFFP_PRIMARY_GAP property value on the differential pair object.
2. *Primary gap* value in the ECSet.
3. *DiffPair primary gap* value in the Physical Rule set.
   If you did not define a value in this rule set, the layout editor defaults to `0 mil`.

When using *Neck gap* during routing or checking, the layout editor looks for the following in sequential order:

1. DIFFP_NECK_GAP property value on the differential pair object.
2. *Neck gap value in the ECSet.*
3. *DiffPair neck gap value in the Physical Rule set.*

⚠ If the layout editor does not find any of these values, it uses the *Primary gap* value in the order described above to search for a primary gap value.

Primary Line Width

When determining the primary line width, the layout editor looks for the following in sequential order:

1. *Primary line width* assigned to the differential pair object. You set this only in the Constraint Manager, or you can set the MIN_LINE_WIDTH property on individual nets, which bubble up to the differential pair object.
2. Differential pair *Primary line width* in the ECSet.
3. *Min line width* in the Physical Rule set.

Neck Line Width

The layout editor determines the value of the neck line width in the following sequence:

1. *Neck width* assigned to the differential pair object. You set this only in the Constraint Manager, or you can set the MIN_NECK_WIDTH property on individual nets, which bubble up to the differential pair object.

2. Differential pair *Neck width* value in the ECSet.
3. *Min neck width* value in the Physical Rule set.

⚠ When you define the primary gap or neck gap in a physical constraint set (differential pair by gap layer), the layout editor always uses values from the constraint set that applies to the differential pair for points outside all constraint areas. The layout editor does not use values from constraint sets assigned to the differential pair for points inside a constraint area.

Viewing DRC Violations for Differential Pairs

In addition to flagging nets that violate differential pair constraints with DRC markers, the layout editor also marks the offending line segments with a highlighting color that is half the width of the line. To see these segments:

1. Choose *Display – Color/Visibility* (color192 command).
2. In the Color dialog box, choose *Stack-Up*.
3. In the *DRC* column, enable the ETCH/CONDUCTOR subclasses on which the differential pairs appear.
4. If necessary, modify the DRC subclass colors.
5. Click *OK*.
6. Choose *Display – Layer Priority* (layer priority command). See Assigning a Display Priority to Layers in the *Allegro PCB and Package Physical Layout Command Reference* for instructions.

Transferring Logic from Older Schematics

You can transfer logic from 14.x schematics into a Release 15.x design. The differential pair properties on nets in the logic file connect to the 15.x elements in the following ways:

14.x Properties	Connected to these 15.x Elements
DIFFERENTIAL_PAIR	differential pair group object
DIFFP_LENGTH_TOL	DIFFP_PHASE_TOL property
DIFFP_2ND_LENGTH	DIFFP_UNCOUPLED_LENGTH property

To transfer logic, choose *File – Import – Logic* (netin command) and the netrev command, described in the *Allegro PCB and Package Physical Layout Command Reference*.

Layer Sets

The layout editor lets you assign layer-set wiring rules to net-based objects to control impedance, shielding, or return path requirements. You place layer-set constraints on nets, XNets, differential pairs, or buses, ensuring adherence to wiring requirements by locking routes within the appropriate layer sets.

Topics covered in this section include:

- Defining Layer Sets
- Assigning Layer-Set Constraints
- Using Layer-Set Constraints in DRC Mode

The following definitions apply to layer sets:

Layer Set	Group of ETCH/CONDUCTOR layers (also referred to as subclasses) applied to nets, XNets, diff pairs, or buses. You can assign one or more layer sets to an object.
Pin Escape	Series of blind, buried, or through-hole vias and clines that defines a path from an outer layer to an inner target routing layer.
Short side of an XNet	Shortest two-pin net of a two-net XNet.

Defining Layer Sets

Before you can apply a layer-set constraint, you must define a layer set. Using the Layer Sets dialog box, accessed by running the `define layersets` command or through the Electrical Constraints dialog box (*Setup – Constraints – Electrical constraint sets)*, you choose available layers and group them in a layer set. There is no limit to the number of layers that can belong to a layer set.

Assigning Layer-Set Constraints

After you define the layer set, assign the layer-set constraint to applicable ETCH/CONDUCTOR objects using one of the following methods:

- Include a layer set in an ECSet.
  A layer set can belong to multiple ECSets. For additional information on assigning a layer set to an ECSet, see the Select Layer Sets Dialog BoxSelect Layer Sets Dialog Box in the *Allegro PCB and Package Physical Layout Command Reference*.
- Attach the LAYERSET_GROUP property.
  For more information about this property, see LAYERSET_GROUP in the *Allegro Platform Properties Reference*.
- Use Constraint Manager to assign a layer-set constraint to a net, XNet, differential pair, bus, or ECSet.
  Cadence recommends using the *Electrical Constraint Spreadsheet*, available by choosing *Setup – Electrical Constraint Spreadsheet*, to manage layer-set constraints. For information about layer sets in the *Electrical Constraint Spreadsheet,* see Layer Sets in the *Allegro Platform Constraints Reference*.

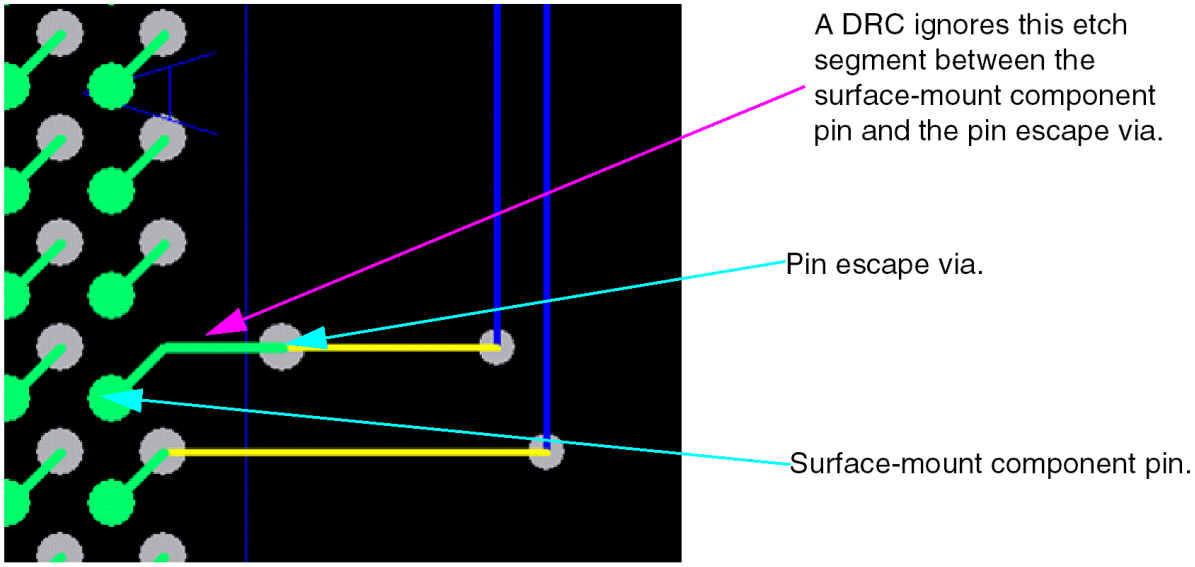
Using Layer-Set Constraints in DRC Mode

Enable design rule checking, so that the DRC markers reflect the current design state.

When checking layer-set constraints, the layout editor determines the following:

- All nets of an XNet route on the same layer set to avoid a DRC error. However, a DRC violation does not occur when the short-side of an XNet runs on a surface-layer between two pins.
- The DRC on a bus is independent from other nets on the bus.
- If the cline is on the path from a surface-mount pin to a legal routing layer, either through a via or a staggered path that progresses towards the legal routing layer, no DRC errors occur. This is considered a pin escape, as shown in the following figure.

PCB Editor: Example of Pin Escape Etch

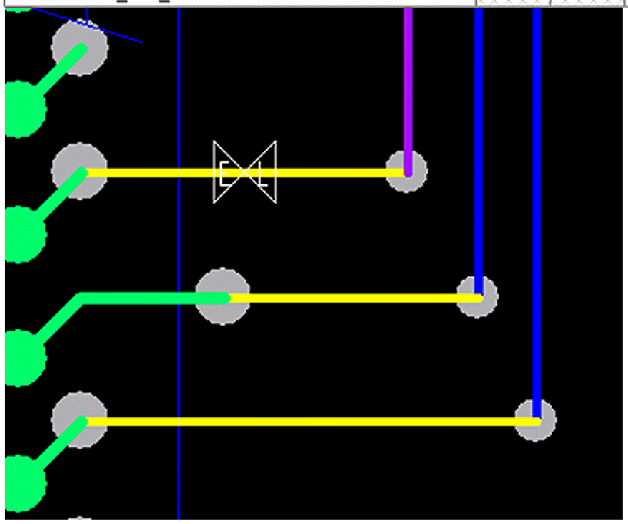


Viewing DRC Violations for Layer-Set Constraints

When a layer-set constraint violation occurs, a DRC marker appears directly on the cline and status information appears in the *Electrical Constraint Spreadsheet*, as follows.

Example of a DRC Violation in Constraint Manager and Allegro PCB Editor

Objects	sed Length		Parallel			Layer Sets			
	Actual	Margin	Max	Actual	Margin	Name	Actual	Margin	Length
	mil	mil	mil		mil				mil
System									
place									
AD_BUS						LS3-4:LS6-7			
H_PCI_AD<59>						LS3-4:LS6-7	PASS		92.36
H_PCI_AD<61>						LS3-4:LS6-7	PASS		35.36
H_PCI_AD<56>						LS3-4:LS6-7	FAIL		35.36



In this example, a violation occurred because you cannot interchange the two layer sets (defined as `LS3-4` and `LS6-7`), even though the purple cline exists on a subclass of the alternate layer set, `LS6-7`. To avoid this violation, you must route the purple cline on layer `Sig_4V` or route the yellow cline on one of the two layers of layer set `LS6-7`. Constraint Manager reports the actual results as a PASS/FAIL condition. The accumulated amount of etch length on non-layer set subclasses appears in the *Length* column of the Electrical Constraint Spreadsheet.

In addition to flagging nets that violate layer-set constraints with DRC markers, the layout editor also marks the offending cline with a highlighting color. See Assigning Colors to Subclasses in the *Allegro PCB and Package Physical Layout Command Reference* for instructions.

Using Pin Delay

You can include pin delay in DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY by assigning the PIN_ DELAY property to component instance or definition pins in the layout editor. The PIN_ DELAY property specifies the time delay or length from the internal package connection to the pin's mounting layer. Use the PIN_DELAY property to manage interchip delay or die-to-die timing paths across a printed circuit board and thereby factor inter-package delay into timing requirements.

When pin delay is measured in time units, it is multiplied by the *Pin Delay Propagation Velocity Factor,* which is a constant used to convert from time to ETCH/CONDUCTOR layer length units if you defined DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY in time units. To factor pin delay into these DRC calculations, choose *Setup – Constraints – Modes*, then click the Electrical Options tab. In the *Pin Delay* group box, enable *Include in all Propagation Delays and DiffPair Phase checks* and enter a value for *Propagation velocity factor*. For more information, see the *Allegro PCB and Package Physical Layout Command Reference*.

A schematic- or a board-driven flow supports pin-delay values.

Schematic-Driven Pin Delay Flow

In a schematic-driven flow, these pin-level delay values are defined as library properties that can be written to the Allegro Design Entry HDL, System Connectivity Manager, or Allegro Design Entry CIS library files sent to the layout editor. You can use Allegro PCB Librarian XL to manually assign the PIN_DELAY property and values to symbol pins or automatically import the PIN_DELAY values through its *Import Wizard*, which supports Comma Separated Value (`.csv`) and Excel (`.xls`) file format options.

Packager-XL packages the design into `pst*.dat` files. The `pstchip.dat` file contains the default values of the PIN_DELA Y property, subsequently imported into the layout editor using the latter's *File – Import – Logic* (netrev command).

You can use the PIN_DELAY property values in Constraint Manager, interactive routing in the layout editor or PCB Router, automatic routing in PCB Router, and DRC verification in both the layout editor and Allegro PCB Router. For more information on using pin delays in Allegro Constraint Manager and Allegro PCB Router, see *Analyze – Analysis Modes* in the *Allegro Constraint Manager Reference* and the *Allegro PCB Router User Guide*, respectively.

You can then optionally edit and override the PIN_DELAY values from the `pstchip.dat` file on component-instance pins. Only overrides are backannotated to the schematic.

Board-driven Pin Delay Flow

In a board-driven, pin-delay flow, you can export pin delay values from an external source using *File – Export – Pin Delay* (pin_delay out command) and then import them to another design and annotate them to component instance pins using *File – Import – Pin Delay* (pin_delay in command) in the layout editor. You can also use *Edit – Properties* (property edit command) to assign the PIN_DELAY property.

When you change the value of the PIN_DELAY property in the layout editor and use *File – Export – Logic* (feedback command) to export modifications to Allegro Design Entry HDL or System Connectivity Manager, both the layout editor-modified (instance) and Allegro Design Entry HDL or System Connectivity Manager-generated (definition) PIN_DELAY values come across in the `cmdbview.dat` file if you are using a Constraint-Manager-enabled flow. If you are using a flow without Constraint Manager, manual edits to the value of the PIN_DELAY property pass in the `pstxprt.dat` file.

Constraint Manager is an optional point to enter PIN_DELAY values and to edit those propagated from the `chip.prt` files. The Constraint Manager flow maintains any overrides of the PIN_DELAY property made in Constraint Manager or in the layout editor, but does not backannotate them to the Allegro Design Entry HDL or System Connectivity Manager schematic. For more information on using the PIN_DELAY property in Constraint Manager, see *Analyze – Analysis Modes* in the Allegro Constraint Manager Reference.

Using Z Axis Delay

To more accurately account for delay in your designs, you can include the conducting portion of a via/pin (also known as Z Axis Delay) in DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY. Z Axis Delay includes component pin and via depth values in a pin-pair path. The conducting portion of a via/pin comprises the thickness through the board from the placed symbol layer where a net enters a padstack, which may be a via or a pin, to the layer from which it exits.

All layer dielectric and conductor thickness lengths between the entry and exit layers are calculated for the conducting portion of a via/pin and are added to the overall net or pin pair length. The conductor thickness is included in the calculation only when the entry and exit layers are opposing z-axis directions. Surface mount vias, such as testpoints, and pin to pin connections on the same layer the component is placed have no effect on the Z Axis Delay calculations.

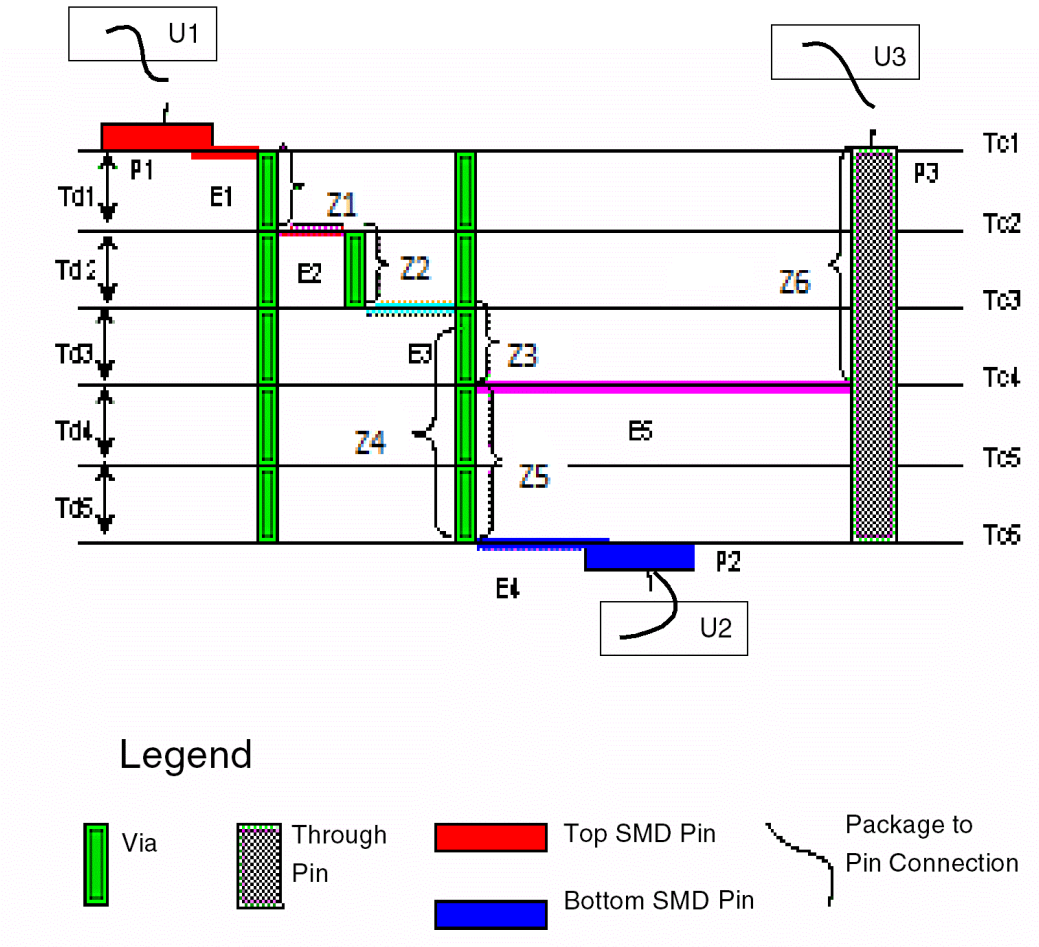
When the conducting portion of a via/pin is measured in time units, it is multiplied by the *Z Axis Delay Propagation Velocity Factor,* which is a constant used to convert from time to ETCH/CONDUCTOR layer length units if you defined DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY in time units.

To factor the conducting portion of a via/pin into these DRC calculations, choose *Setup – Constraints – Modes*, then click the Electrical Options tab. In the *Z Axis Delay* group box, enable

Include in all Propagation Delays and DiffPair Phase checks and enter a value for *Propagation velocity factor*. For more information, see the *Allegro PCB and Package Physical Layout Command Reference*.

Z Axis Delay Example

Assume the following stackup is defined in the *Layer Cross Section* dialog box, accessible by running *Setup – Cross-section* (xsectioncommand), described in the *Allegro PCB and Package Physical Layout Command Reference*:



Tdr = the thickness of the dielectric between r^{th} and r^{th}+1 layers

Tcr = the thickness of the conductor on r^{th} layer

Z* = Z- Axis delay portions of pin-pairs
E* = Horizontal displacement

Z-Axis delay for
U1 to U2 = Z1 + Z2 +Z4
= {TC1 + TD1} + {TC2 + TD2 + TC3} + {TD3 + TC4 + TD4 + TC5 +TD5 + TC6}
= 1.2 + 5.0 + 1.2 + 5.0 + 0.7 + 5.0 + 0.7 + 5.0 + 1.2 + 5.0 + 1.2
= 3.12 mils
U1 to U3 = Z1 + Z2 + Z3 + Z6
= {TC1 + TD1} + {TC2 + TD2 +TC3} + {TD3} + {TD3+ TC3 + TD2 + TC2 + TD1 + TC1}
= 1.2 + 5.0 + 1.2 + 5.0 + 0.7 + 5.0 + 5.0 + 0.7 + 5.0 + 1.2 + 5.0 + 1.2
= 36.2 mils
Total Length for
U1 to U2 = E1 + Z1 + E2 + Z2 + E2 + E3 + Z4 + E4
U1 to U3 = E1 + Z1 + E2 + Z2 + E3 + Z3 + E5 + Z6

Consider the following stackup and layer thicknesses, as outlined in the preceding figure, as an example.

Layer Type	ETCH/CONDUCTOR Subclass Name	Thickness
Conductor	Tc1	1.2 mils
Dielectric	Td1	5.0 mils
Plane	Tc2	1.2 mils
Dielectric	Td2	5.0 mils
Conductor	Tc3	0.7 mils
Dielectric	Td3	5.0 mils
Conductor	Tc4	0.7 mils
Dielectric	Td4	5.0 mils
Plane	Tc5	1.2 mils
Dielectric	Td5	5.0 mils
Conductor	Tc6	1.2 mils

Setting Nets to Check Themselves for Crosstalk and Parallelism

To enable the nets in your design to perform a design rule self-check for crosstalk and parallelism (in addition to the checks the net makes against all other nets), you can turn on the feature using the *Objects - Create - Electrical CSet i*n Constraint Manager.

Enabling this command creates same net crosstalk records in your design database. Because such data is not supported in releases prior to 15.5.1, you must perform a database down rev in Release 15.2 to remove these objects. For releases earlier than 15.5.1, attempts to open designs containing same net DRC data produce an error message and the design does not open.

Design for Assembly Legacy Checking

Dynamic Design For Assembly (DFA) checks in the layout editor support package-to-package clearance modeling. By implementing rules-driven design techniques, constraints identified during system design control the physical layout. Real-time design for assembly analysis occurs as you manually place components.

Using Dynamic Design for Assembly (DFA) Constraints

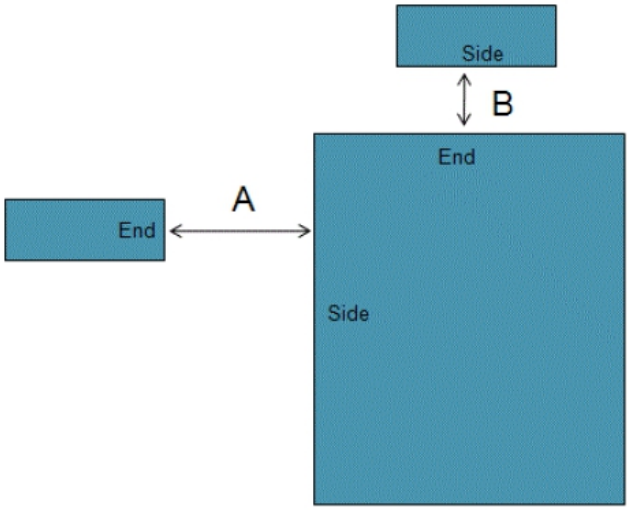
During interactive placement, designers can understand component-clearance requirements using the rules-driven DFA Constraints Dialog spreadsheet, whose values can be associated with a design.

You can apply an external design assembly checks table owned by manufacturing that represents corporate design assembly checks standards to a design, then edit or override the table values using the DFA Constraints Dialog spreadsheet, which defines component side-side (S-S), side-end (S-E), end-end (E-E), and end-side (E-S) spacing values in one view that supports top and bottom sides of the PCB separately.

In the figure below, A and B both represent a side-end (S-E) condition where different values are applicable:

- The symbol considered the Reference Symbol is located in the column of the DFA spreadsheet.
- If end-side (E-S) value is not present, DRC uses side-end (S-E) value for both the conditions (pre 16.5 behavior).
- When comparing two identical symbols, only the side-end (S-E) value is used and, end-side

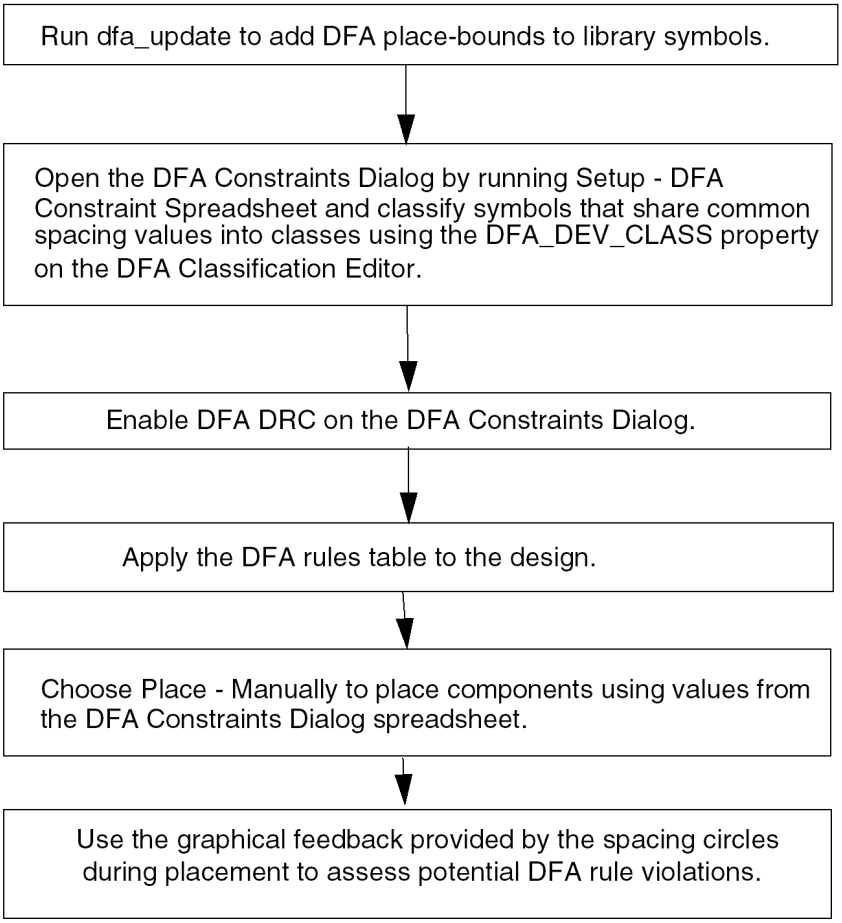
(E-S) is considered superfluous.



Conversely, use the *Read-only* field on the DFA Constraints Dialog spreadsheet to prevent users from overriding corporate design assembly checks standards. The DFA Constraints Dialog spreadsheet is available by choosing *Setup – Constraints – DFA Constraint Spreadsheet* (dfa_spreadsheet command).

These values in the design assembly checks rules table drive real-time DRC feedback during placement. For example, a DRC flags too-small part-to-part spacing during part placement and updates it in real time. All assembly checks occur in the same environment in which you designed the board, to preclude juggling between the layout editor and third-party tools for fabrication checks.

Real-Time Design Assembly Checks Process



Grouping Symbol Definitions with the DFA_DEV_CLASS Property

You can group symbol definitions that share the same clearance values by assigning the DFA_DEV_CLASS property to them. The DFA_DEV_CLASS property classifies components according to the Design for Assembly (DFA) package-to-package spacing values defined in the DFA Constraints Dialog spreadsheet, available by choosing *Setup – Constraints – DFA Constraint Spreadsheet* (dfa_spreadsheet command).

You use the DFA Classification Editor dialog box to add or remove symbol definitions from user-defined classes. (The DFA Classification Editor is available by clicking Show symbol classifications... on the DFA Constraints Dialog spreadsheet.) The layout editor treats these classes as components comprised of symbols to which the design assembly checks spacing values defined for the class default.

For example, fifty versions of an 0805 package symbol may exist, all complying to the same design

assembly checks set. A single class line entry in the spreadsheet assumes the rules for each instance of the 0805 class of package symbols.

Clicking *Update* on the DFA Classification Editor dialog box assigns the DFA_DEV_CLASS property to the symbol definitions in the classes you specified.

DRC for Design Assembly Checks

The DRC marker (D-C) for design assembly checks verifies design assembly checks spacing rules prior to and after symbol placement either interactively or in batch mode. When violations occur during manual placement, the D-C marker and design assembly checks extents appear in the color assigned to design assembly checks DRC in the Color dialog box. Design assembly checks DRCs appear in Constraint Manager in the DRC workbook under the Design worksheet. Design assembly checks DRC supports alternate symbols.

For example, a design assembly checks DRC error appears in error reports as:

```
DIP14-DIP14 violated 100:200:100 rule.
```

Design assembly checks DRC determines the correct rule set by checking:

- Defined design assembly checks in the DFA Constraints Dialog spreadsheet for the associated symbol definition
- The DFA_DEV_CLASS property assigned to symbol definitions and design assembly checks rules in the DFA tables

You can choose to enable or disable design assembly rules checking independently of existing package- to-package rules checking on the Design Constraints dialog box, accessed by running (cns design command) in the layout editor. Design assembly checks DRC occurs in conjunction with batch package-to-package DRCs if you:

- Enable design assembly checking on the DFA Constraints Dialog spreadsheet
- Create design for assembly place-bounds for the symbol, enable design assembly checking, and an entry exists in the DFA Constraints Dialog spreadsheet
- When one or both symbol definitions lack design assembly place-bounds, DRC uses the existing place-bound and generates a warning. You can generate design assembly place-bounds for legacy symbols automatically using `dfa_update`.
- Assign the HEIGHT property to either package

During manual placement, as a dynamic component approaches placed components' DFA place-bound extents, DRC runs based on the correct S-S, E-E, S-E, and E-S value in the DFA Constraints

Dialog spreadsheet. Spacing circles appear on screen between the components to show the actual spacing value visually, and to highlight potential design assembly checks DRC errors in the color you assigned for DRCs.

If a violation occurs, the layout editor calculates the design assembly extent on all placed components that come within the range of potential design assembly checks DRC errors, and a DRC marker displays there.

⚠ To remove the butterfly from appearing with the spacing circles, use *Setup – User Preferences (enved* command) and enable the `no_dfa_drc_marker` variable.

This DRC and design for assembly place-bound extents remain highlighted until you move the component from the violation extent area; the spacing circles then disappear as well. If you move components, DRCs become out of date for design assembly errors.

Using DFA_PAUSE_LEVEL property to remove design for assembly (DFA) DRC

To meet the minimum design for assembly clearance rule you can set the DFA_PAUSE_LEVEL property with value 3. This property pauses the active component during movement to meet the design for assembly clearance rule.

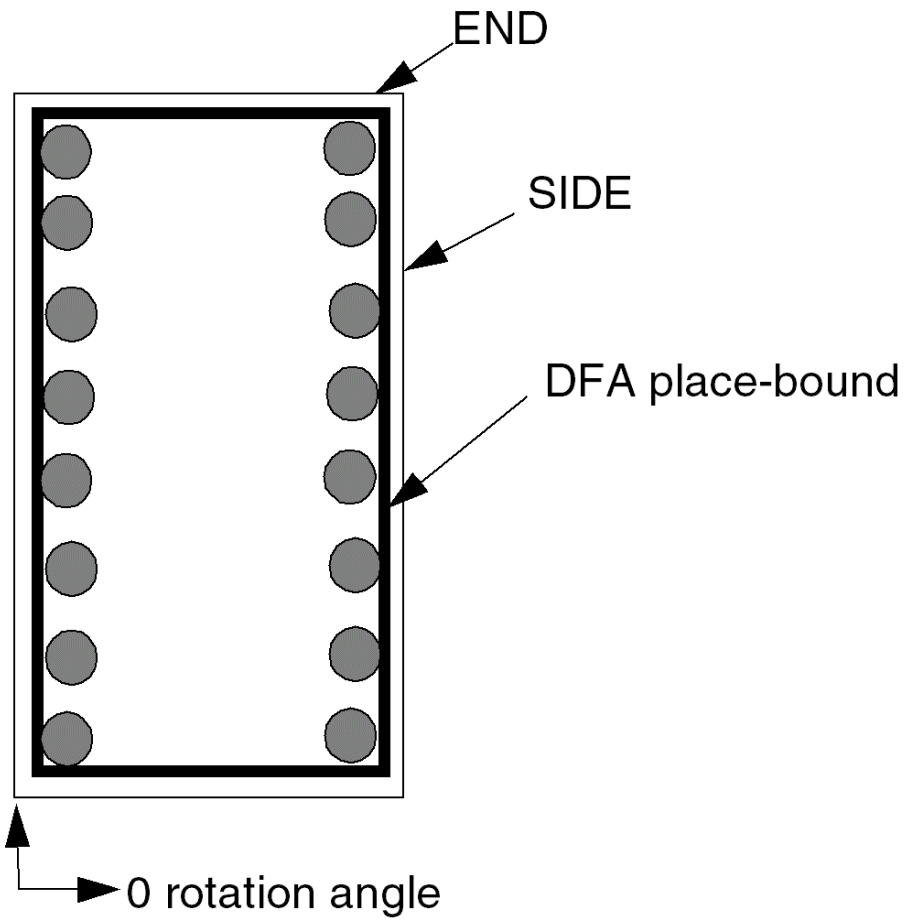
Design for Assembly Place-Bounds

You can create design for assembly place-bound shapes in Symbol mode as you do place-bound shapes using the package_bound command. Place-bound rectangles are filled rectangles that define the package boundary and govern placement restrictions. Placement tools use these rectangles for overlapping and mechanical restraints. DRC also uses them to check for violations of package-to-package keepin areas and keepout areas. For legacy symbols, you can add the design for assembly place-bound to the symbol definition, with dfa_update.

A rectangular-shaped design for assembly place-bound differs from a place-bound in that the former has differentiating edges and ends to comply with design for assembly spacing rules. For circles or polygons, design for assembly DRC uses the largest, or most conservative, value specified in the DFA Constraints Dialog spreadsheet for that symbol if three values exist. Unlike current place-bounds, which allow you to concatenate multiple shapes, each layer allows only one shape for a design for assembly place-bound.

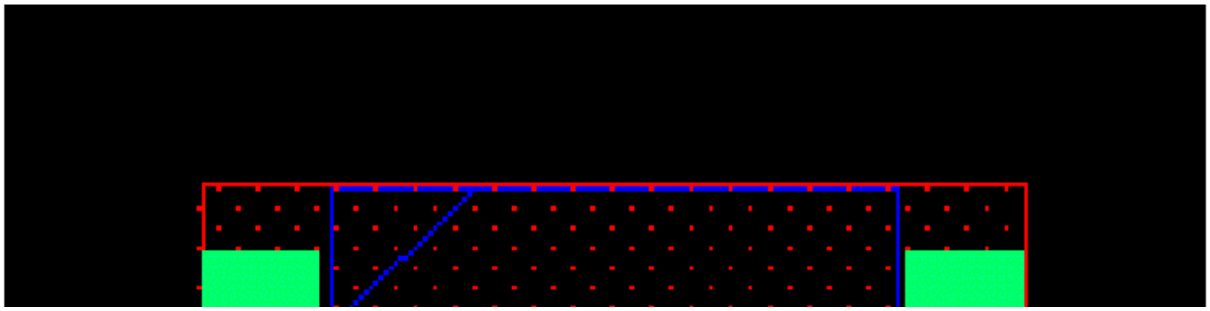
The layout editor determines the design for assembly place-bound end, which comprises the shorter sides of the extent at a zero degree rotation; side as the longer sides of the extent.

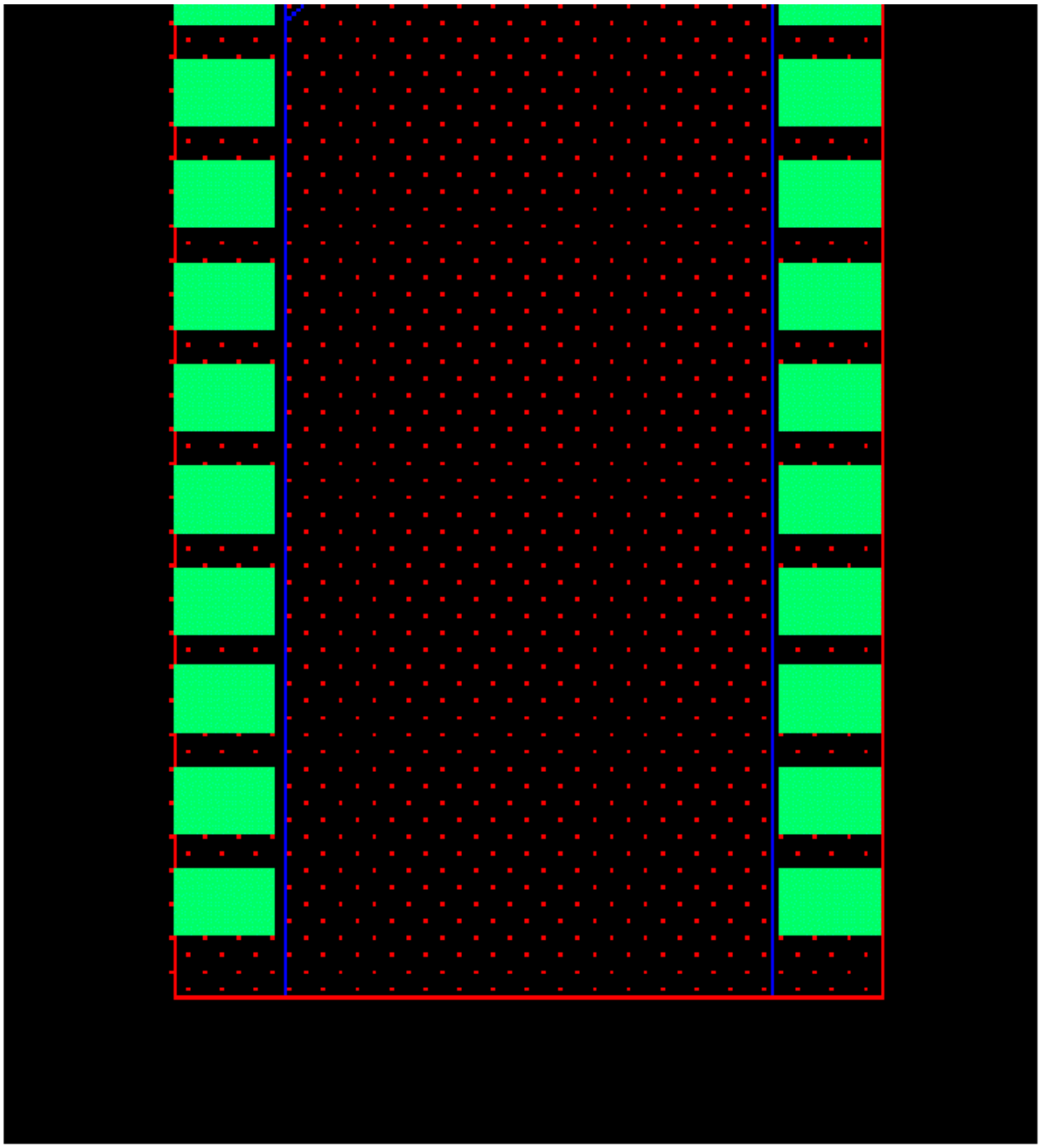
DFA Place-Bound Description



In the example below, the design for assembly place-bound surrounds the package pins, but with respect to the assembly outline in blue.

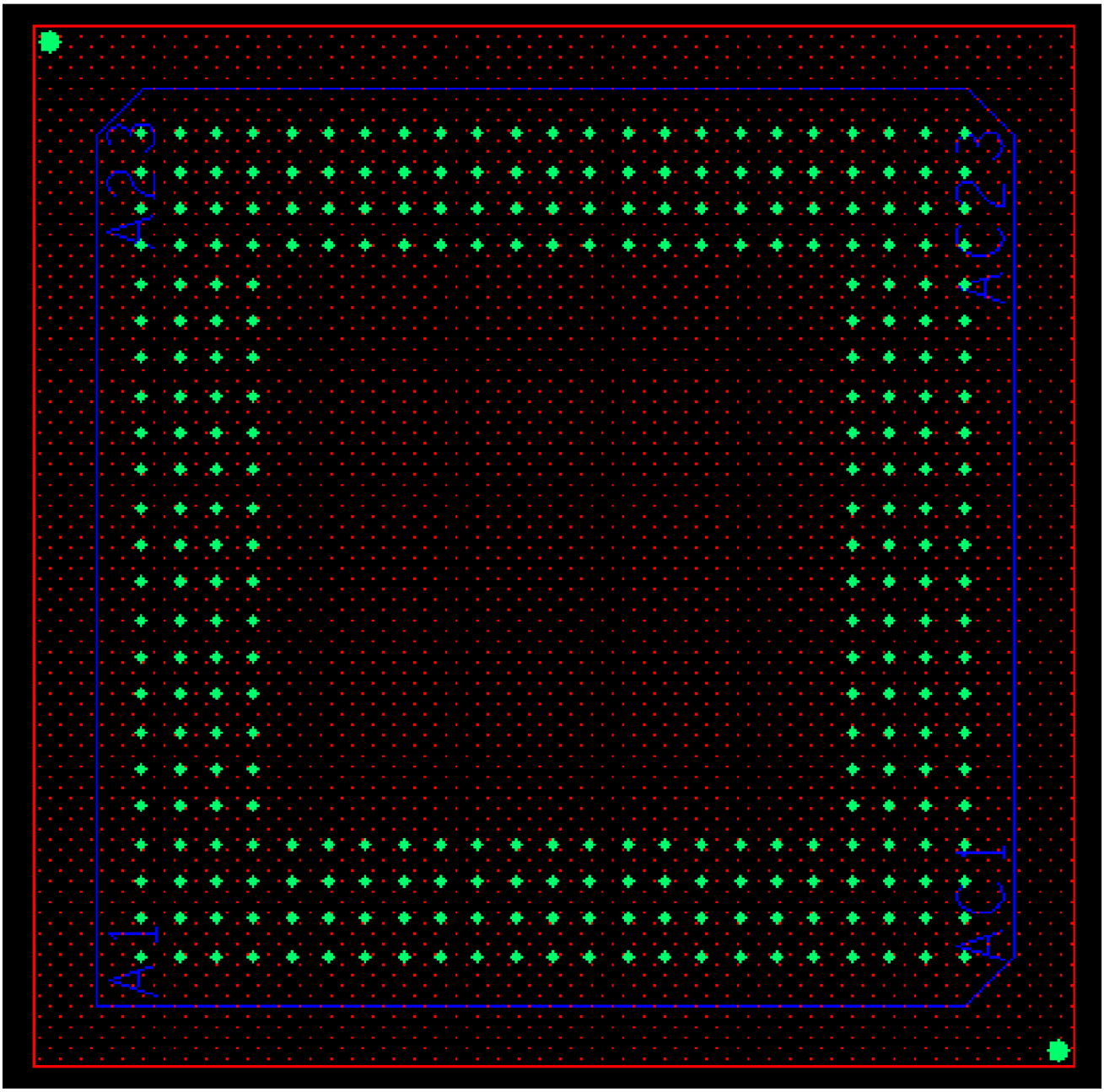
DFA Place-Bound: SOIC





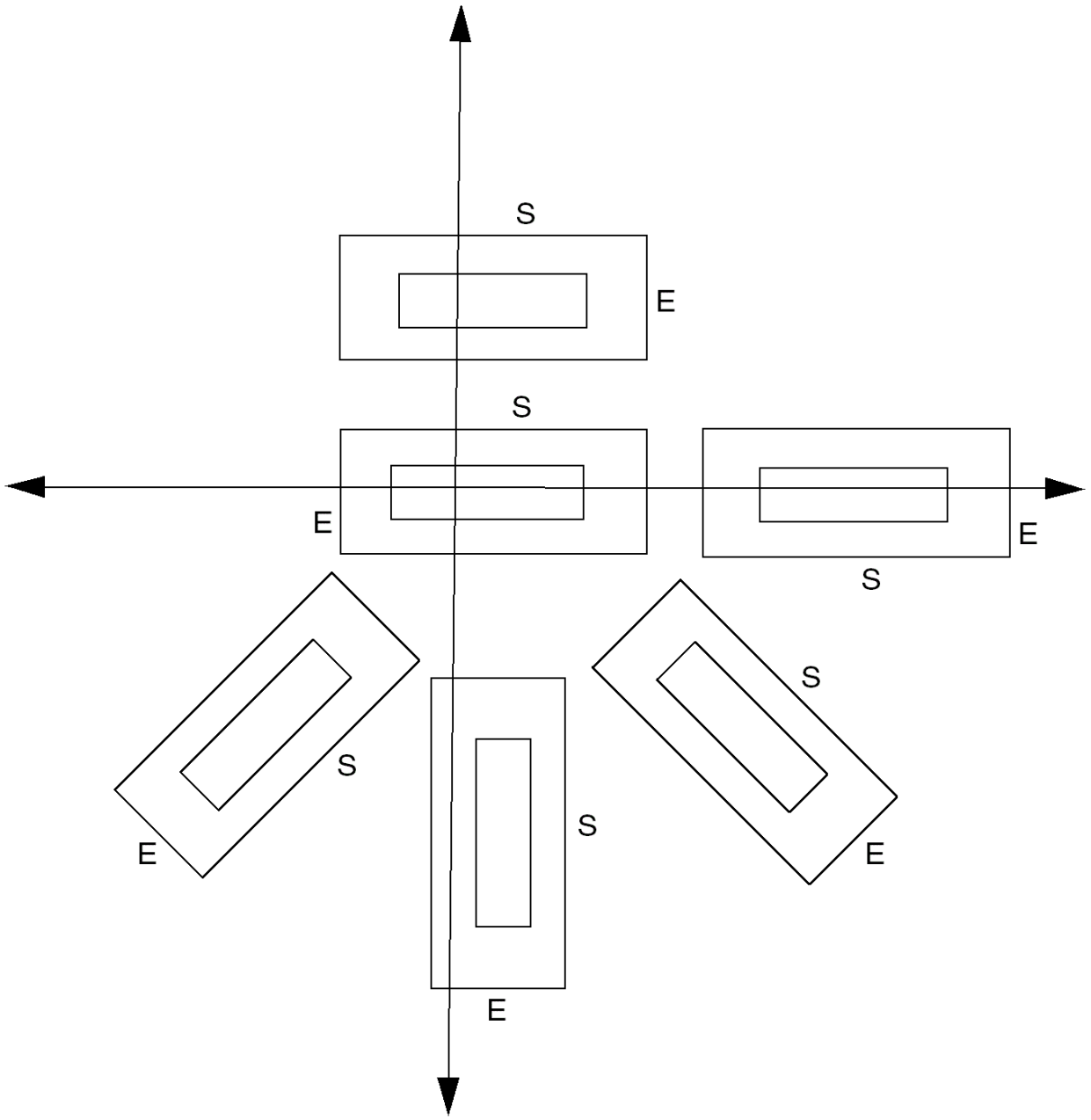
In the example below, the red outline surrounds the package pins, including fiducials.

DFA Place-Bound: BGA



Design for assembly place-bounds, created on DFA_BOUND_TOP and DFA_BOUND_BOTTOM layers of the PACKAGE GEOMETRY class, follow the symbol rotation as follows.

DFA Place-Bound Rotation Rules



Using Batch-Mode Design for Assembly (DFA) Analysis

The layout editor provides batch-mode design assembly checks analysis on a printed circuit board with a set of audits for which you can specify constraints, and define and prioritize them if necessary.

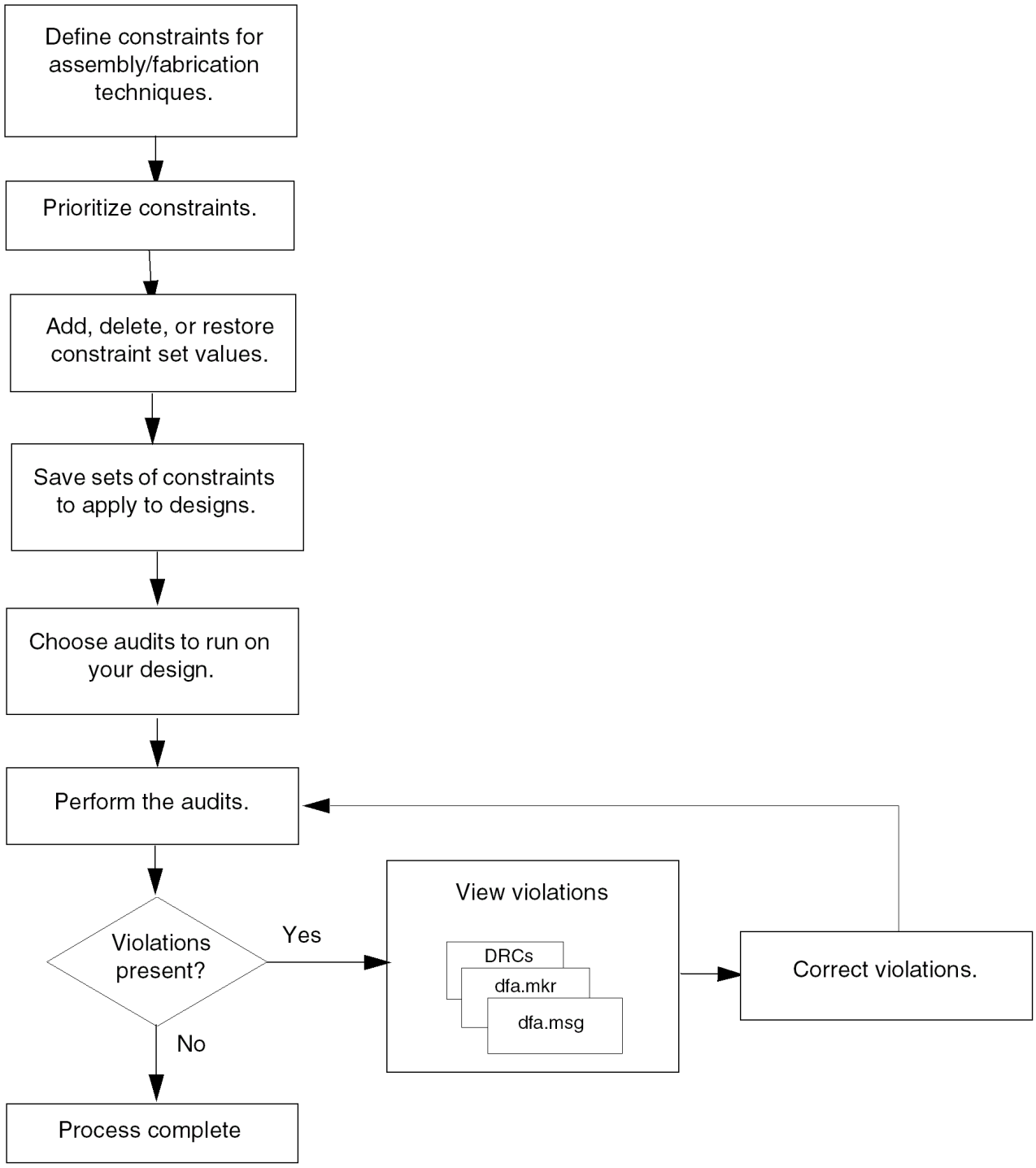
Design assembly checks allows you to run the following audits:

- Component clearance
- Component orientation and mounting layer
- Lead span
- Hole
- Testability
- Orphan via
- Hanging trace

Running these audits enables you to verify that your design adheres to a particular set of constraints. After performing the audits, you can view the violations written back to the design as DRCs; you can cross-probe and highlight the violations in the layout editor using the markers utility.

The following figure depicts the typical use model of batch-mode design assembly checks. Batch-mode design assembly checks can be performed anytime after placing components on your design.

Overview of Batch-Mode Design Assembly Checks Analysis



Defining Batch-Mode Design for Assembly (DFA) Constraints

Constraints are defined in the DFA Audit Setup dialog box, and constraint sets must be defined for each audit as the first step in the design assembly checks process. For details, see Creating Specialized Constraints for DFA Check in the *Allegro PCB and Package Physical Layout Command Reference*.

Prioritizing Constraints

The layout editor provides a default for the audits that support prioritization. Details are provided for all audits. Typically, you assign general—that is, non-specialized—constraint values to the default set (also known as a "child item"). In the case of the Component Assembly Clearance audit, for instance, the constraints written against the default child item apply to all components in your design. The default constraint set, therefore, represents the highest evaluated level of constraints within an audit. Additional child items that you create from the default (or current) child item should have more specialized constraints associated with them.

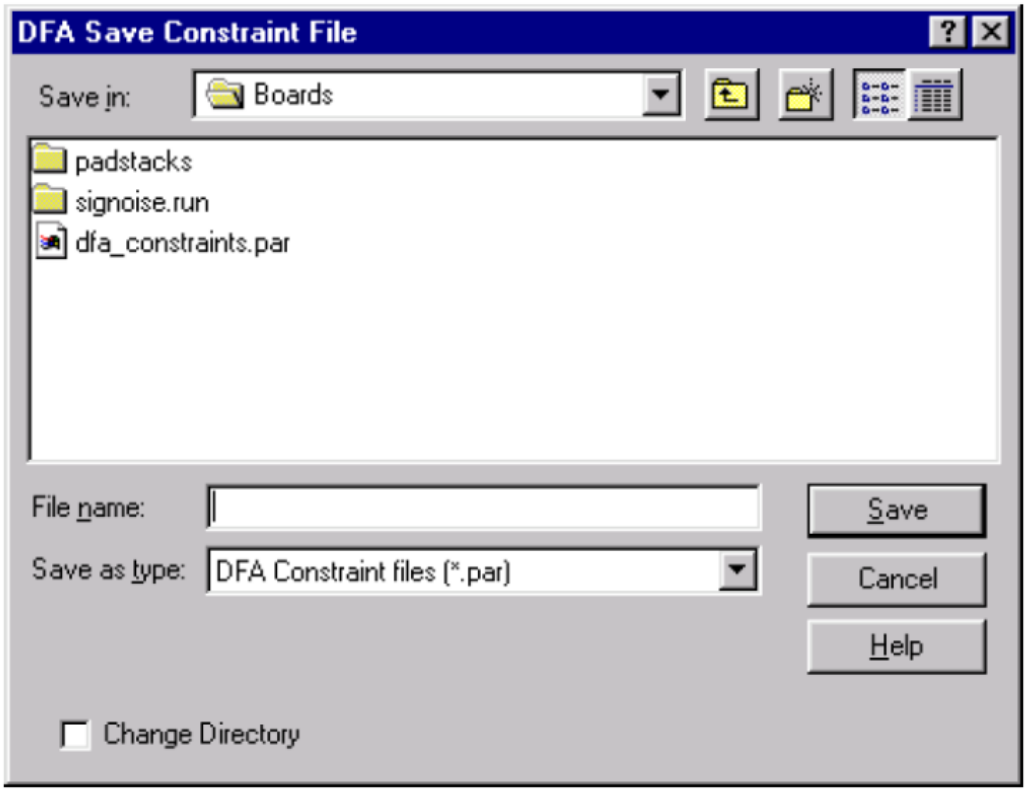
During an audit, the layout editor evaluates specialized constraint sets before less stringent constraints in a bottom-to-top order. You can maintain this progression of less stringent to more stringent constraints by inserting new or modified constraint sets between existing ones.

Each child item in an audit can be considered a place holder for the constraints associated with it. The most general constraint—default—must be at the top of the list of child items in audits that require prioritization of constraints because the layout editor evaluates constraints from bottom to top. When the layout editor finds a match for a specialized constraint, higher level—more generalized—constraints are not evaluated. The audits that require prioritization are:

- Component Assembly Clearance
- Component Orientation and Mounting Layer
- Lead Span
- Hole Audit
- Test Point Audit

Saving Constraints

After you have defined a new set of constraints in the DFA Audit Setup dialog box, save the constraint set by clicking the *Save* button in the main dialog box (Design For Assembly). Clicking *Save* opens a constraint file browser.



The information entered in the Audit Setup dialog box can be written to a new or existing `.par` file and is read when future audits of that constraint set are performed.

⚠ Clicking the *OK* button in the Audit Setup dialog box saves the modified constraints in the directory pointed to by the browser, then closes the dialog box. Clicking *Cancel* closes the dialog box, discarding the changes made in that session.

All the design assembly checks constraint files are ASCII files, thus they can be edited; however, we recommend that changes made to constraints be performed in the Audit Setup dialog box. Constraint files are structured in the following format:

```
STARTENV DFA
STARTRULE    component_orientation_layer_audit
    PARAM    SELECTOR_TYPE_     "Component"
    PARAM    SELECTOR_NUMBER_     1
    PARAM    PARAMETER_TYPES_     "S" , "S" , "S" , "NL"
    PARAM    PARAMETER_WIDGETS_     "P" , "E"
    PARAM    PARAMETER_POPUPS_     "POPUP_LAYER"
    PARAM    CONSTRAINTS     "Constraint_1" , "smd2smd" , "Default"
    PARAM    SELECT_BY1     "Symbol" , "Property" , "Any"
    PARAM    VALUE1     "ocs" , "DFA_DEV_CLASS=ALL" , "*"
    PARAM    LAYER     "BOTTOM" , "TOP" , "Either"
```

```
    PARAM    ORIENTATION     "90" , "0" , "0 90 180 270"
ENDRULE

STARTRULE test_point_audit
 PARAM MIN_SIZE 50
 PARAM MAX_SIZE 100
ENDRULE
ENDENV
```

The sample above defines the:

- Environment in which the file is to be used
- Constraints and their values
- Prioritization of constraints (where supported)

The default `dfa_constraints.par` file exists in `$CDS_INST_DIR/share/pcb/assembly`.

Selecting the Audits To Run

Choose the audit and the child item (if prioritization is supported) in the Setup dialog box. The lower portion of the dialog box displays the constraint values associated with your selection. For details, see the dfa command in the *Allegro PCB and Package Physical Layout Command Reference*.

Performing the Audit

Run the audit(s) from the layout editor with the `dfa` command. Make sure you have set all parameters to your satisfaction, and you have chose the appropriate audit(s) to run.

When you run an audit on a constraint set, the following files are created in the design directory:

- `dfa.msg`
- `dfa.log`
- `dfa.mkr`

The message, log, and marker files are updated automatically and contain the violations that were generated during the most recent audit. Move/copy or rename files generated earlier for purposes of comparisons with more recent runs.

Viewing Violations and Reports

Design assembly checks assume a default value of 200 violations. You can reset the maximum by assigning a specific number as a property value, or by entering a value in the DFA dialog box.

You can view violations in two ways: on the board itself as DRCs or from the dfa.mkr file through the markers utility (when you click the *Explore Violations* button in the DFA dialog box). Clicking on a violation in the markers window highlights the violation in the layout editor's working area. All design assembly checks DRCs are cleared from the board at the beginning of each audit session.

Clicking the *Report* button in the DFA dialog box lets you view the `dfa.msg` file containing details of every violation. The `dfa.log` file provides details of the audit process.

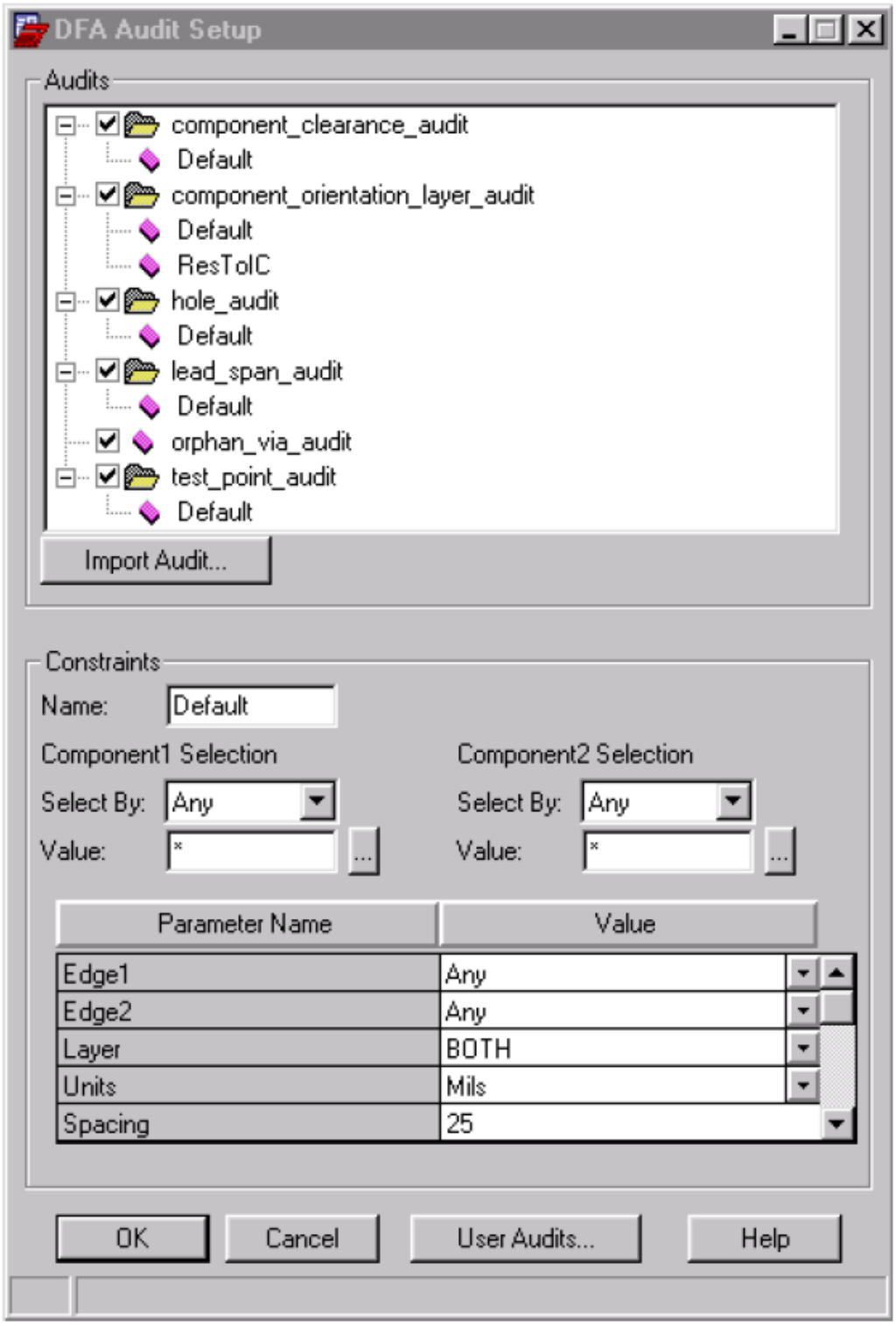
Completing the Process

If violations are present after running the audit, make the necessary corrections, then run the audit again. Perform this step until your board is free of violations.

Design for Assembly (DFA) Audit Descriptions

This section contains information on each of the design assembly checks audits. The accompanying illustrations depict the Audit Setup dialog box as configured for the separate audits.

Component Assembly Clearance



This audit checks the spacing between the components to accommodate assembly, inspection, and repair. A group of components can be chosen by defining a selection criteria using the "*Select By*" and "*Value*" fields. These values are described in the online Help found in the layout editor.

The Component Assembly Clearance audit supports prioritization of constraints. Distance is calculated in user-defined design units.

Example

Two child items, Default and ResToIC, exist in the Component Assembly Clearance Audit in the order shown. The value of each field in the constraints section of the Audit Setup dialog box is:

Child Item	Comp1	Value1	Edge1	Comp2	Value2	Edge2	Layer	Spacing
Default	All	*	Any	All	*	Any	Any	1200
ResToIC	RefDesR	*	Any	RefDesU	*	Any	Top	1000

In the example, the minimum distance between any two components on any layer is 1200 design units. The minimum distance between any resistor component with a reference designator (refdes) R and any IC with a reference designator U on the top layer is 1000 design units.

When you run the audit, design assembly checks for a minimum spacing of 1000 design units between resistors and ICs on the top layer. All other elements are checked for a minimum spacing of 1200 design units. If the priority is reversed— if ResToIC is moved to the top—the audit first checks for a minimum spacing of 1200 design units between any two components, and the ResToIC constraint is never checked.

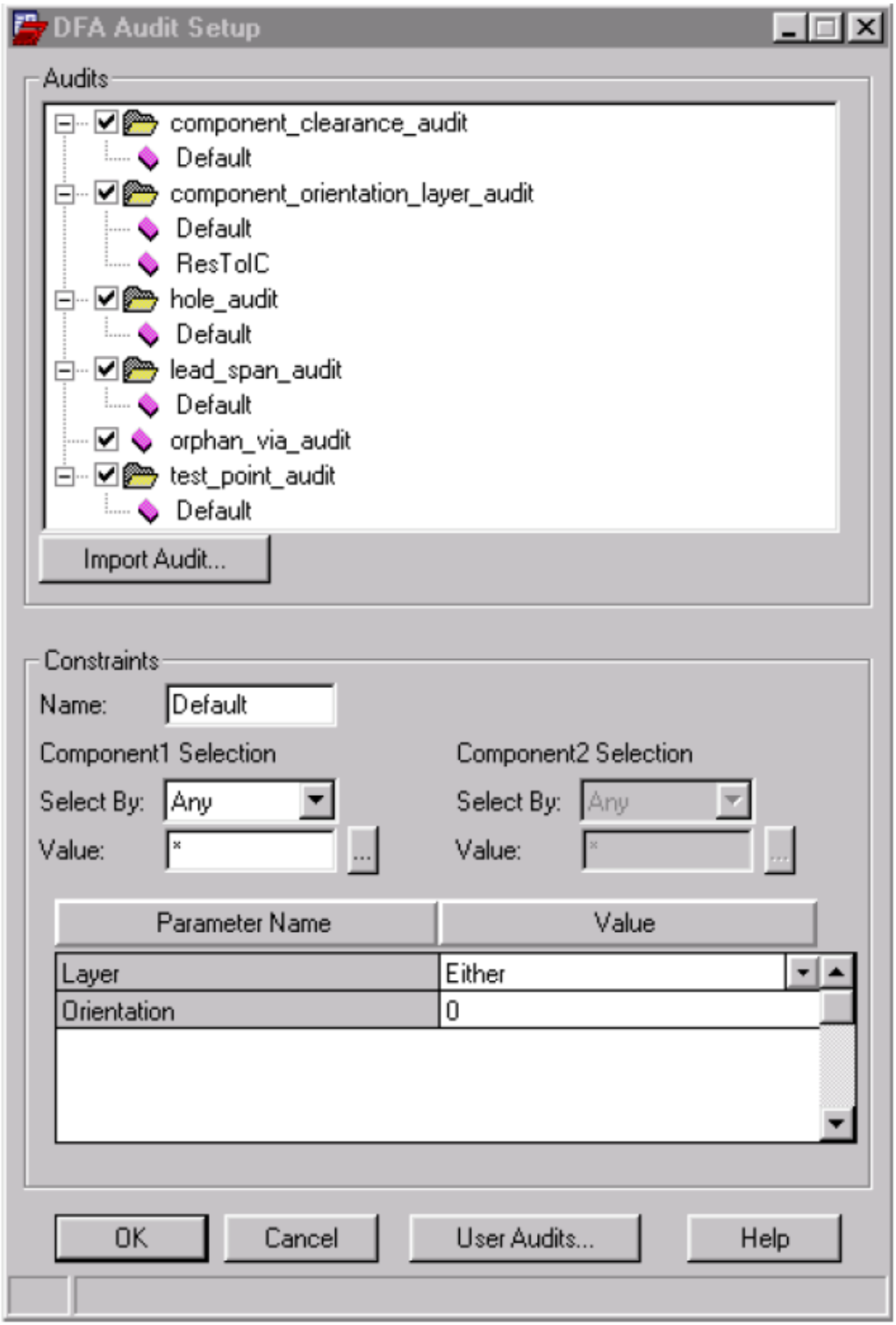
Error Messages

The error message generated by the audit is:

```
ERROR (component_clearance_audit)
Clearance between components U23 and C32 : 50
Minimum clearance required : 100
Categories used are category1: RefDes  * category 2: RefDes   *

ERROR (component_clearance_audit)
Clearance between components C32 and U23 : 50
Minimum clearance required : 100
Categories used are category1: RefDes  * category 2: RefDes   *

ERROR (component_clearance_audit)
```

```
Clearance between components U12 and U21 : 25
Minimum clearance required : 100
Categories used are category1: RefDes  * category 2: RefDes   *

ERROR (component_clearance_audit)
Clearance between components U21 and U12 : 25
Minimum clearance required : 100
Categories used are category1: RefDes  * category 2: RefDes   *
```

Component Orientation and Mounting Layer



This audit checks that component orientation is correct for the soldering process used, and that components are mounted only on allowed layers.

A group of components can be chosen by defining a selection criteria. Refer to the *Allegro PCB and Package Physical Layout Command Reference* for descriptions of field selection values and criteria. Prioritization of constraints is supported.

Example

The audit has two child items: Default and Resistor. The value of each field in the constraints section of the Audit Setup dialog box is:

Child Item	Select By	Value	Layer	Orientation
Default	All	*	Top	0
Resistor	RefDesR	*	Bottom	90

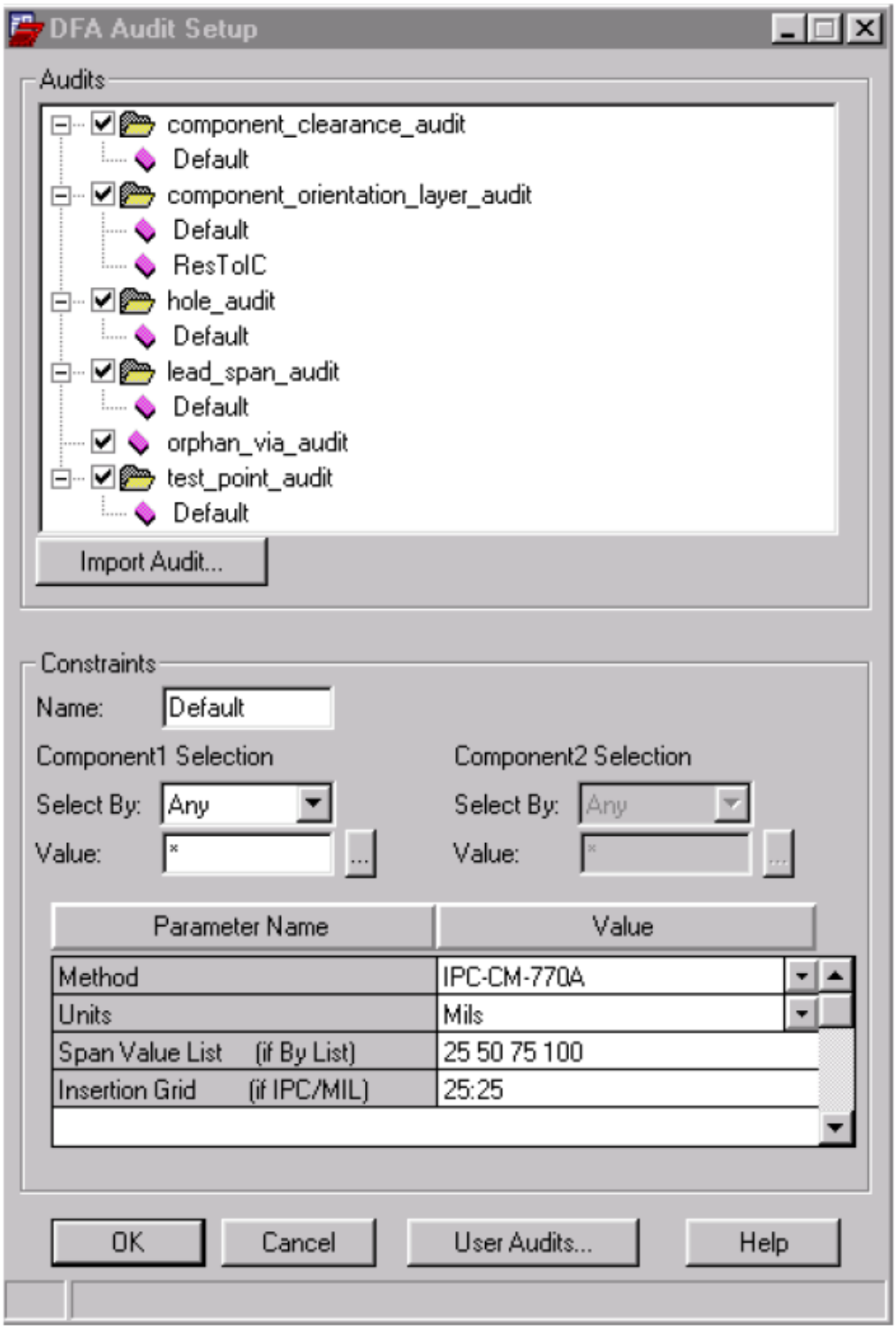
All resistors on the bottom layer should have an orientation of 90 or 270 degrees. Any component on the top layer should have an orientation of 0 or 180 degrees. The parameters of this audit specify that only resistors are allowed on the bottom layer.

If the two items were reversed, all components should be present on the top layer with allowed orientations of 0 or 180 degrees. Resistors on the bottom layer generate error messages.

The error messages generated by the audit are:

```
"ERROR : Component mounted on the wrong layer "

Component   : 'R1'

Layer   : TOP

Allowed Layer   : BOTTOM

Constraint name   : Constraint_1

ERROR : Component has wrong orientation"

Component   :'U2'

Orientation   : 0

Allowed Orientation : 90 180

Constraint name   : Default
```

Lead Span


DFA Audit Setup
Audits
component_clearance_audit
Default
component_orientation_layer_audit
Default
ResTolC
hole_audit
Default
lead_span_audit
Default
orphan_via_audit
test_point_audit
Default
Import Audit...
Constraints
Name:
Default
Component1 Selection
Select By:
Any
Value:
*
Component2 Selection
Select By:
Any
Value:
*
Parameter Name
Value
Method
IPC-CM-770A
Units
Mils
Span Value List (if By List)
25 50 75 100
Insertion Grid (if IPC/MIL)
25:25
OK
Cancel
User Audits...
Help

Axial components must be able to fit into the holes at the span defined in the symbol. This audit verifies the span values for components. Allowed span values can be specified by list or by equation.

The value list is entered on a separate form for both the options; that is, selecting span values by list, or by equation (shown above). Refer to the online Help for descriptions of field selection values and criteria. Prioritization of constraints is supported.

If span value checking is by list, the component span is the distance between the two pads of the component. With this method, you enter a space-separated list in the span value list field. The actual component span should match one in the list.

If the span value is calculated by equation, the equation used for determining the span values is

```
:X*Body Length + Y * LEAD_DIAMETER + Z
```

where X, Y, Z are constants which can be customized. The "*Base-Increment value list*" field has a space separated list of base increment pairs in the format base:increment. The actual span value is checked against the base:increment pair.

Example

Assume that X=Y =1 and Z=100 mils. Base: Increment = 300:100. Let the component span evaluate to 750 mils. The allowable span values are 300, 400, 500, 700, 700, 800 etc. Since 750 >700 and 750 < 800, the allowable span value is 800. So the actual pad-to- pad distance is checked against this value.

The audit is applied only to axial components where axial components are defined as package symbols with attached property DFA_DEV_CLASS and two pins with attached property LEAD_DIAMETER (when span values are chosen by equation).

The error message generated by the rule is:

```
"ERROR : Component does not have standard span value"

"ERROR (lead_span_audit)

Allowed span value(s) : 200

Component    : 'C91'

Span Value    : 140

Body Length    : 170

Span value specified  : By Equation

Constraint name    : Default

ERROR (lead_span_audit)
```

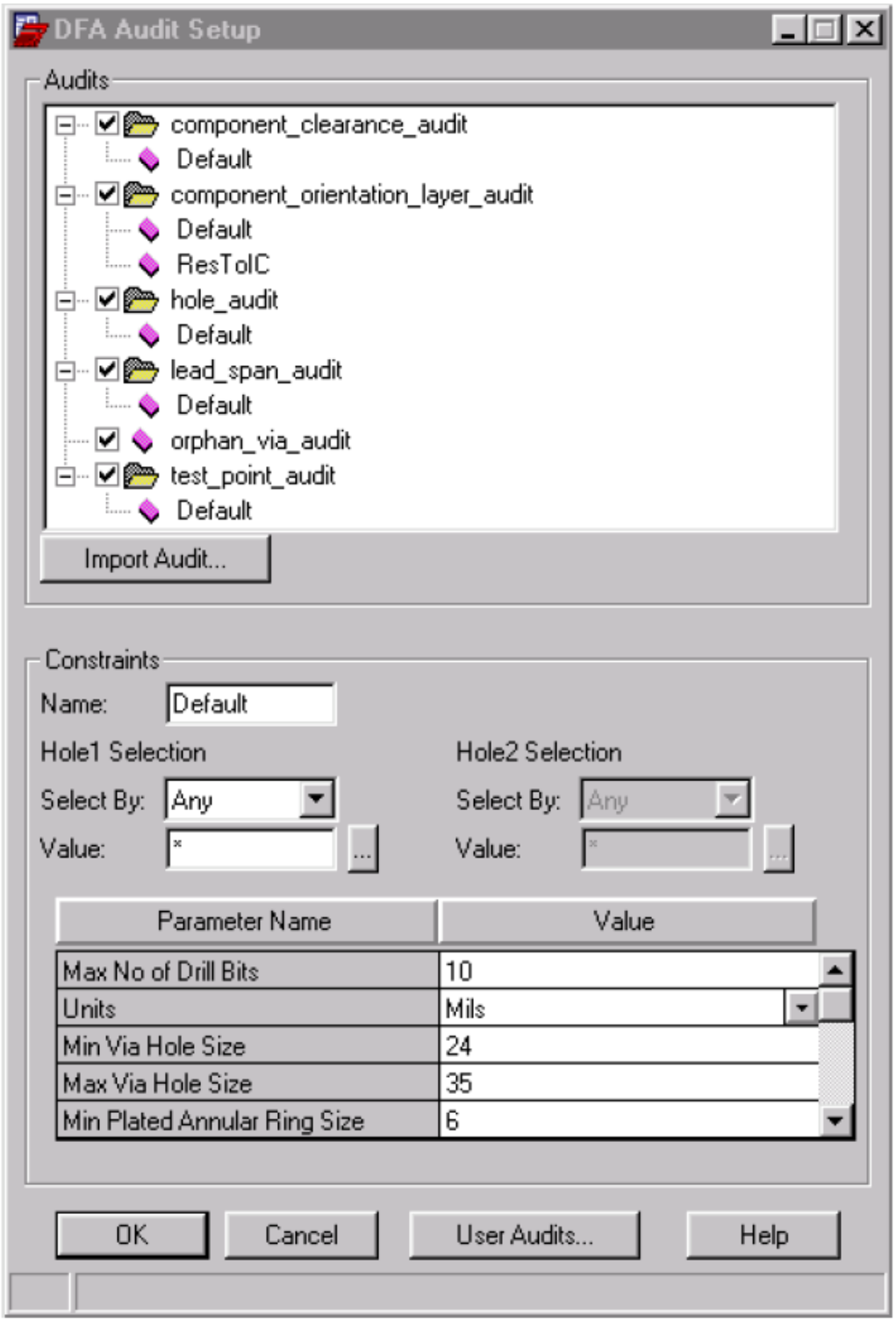
```
Allowed span value(s) : 205 220 220

Component    : 'C94'

Span Value    : 140

Body Length    : 170

Span value specified  : By List

Constraint name    : Default
```

Orphan Via

Choose this audit to report on vias that meet both of the following criteria:

- The via has no clines connected to it.
- The via is not on a net.

Hole



This audit:

- Verifies all pads have the minimum annular ring width
- Allows only permissible drill sizes in the design
- Checks minimum/maximum via size and whether duplicate holes are present in the same location

The audit reports the number of unique drill sizes in the design.

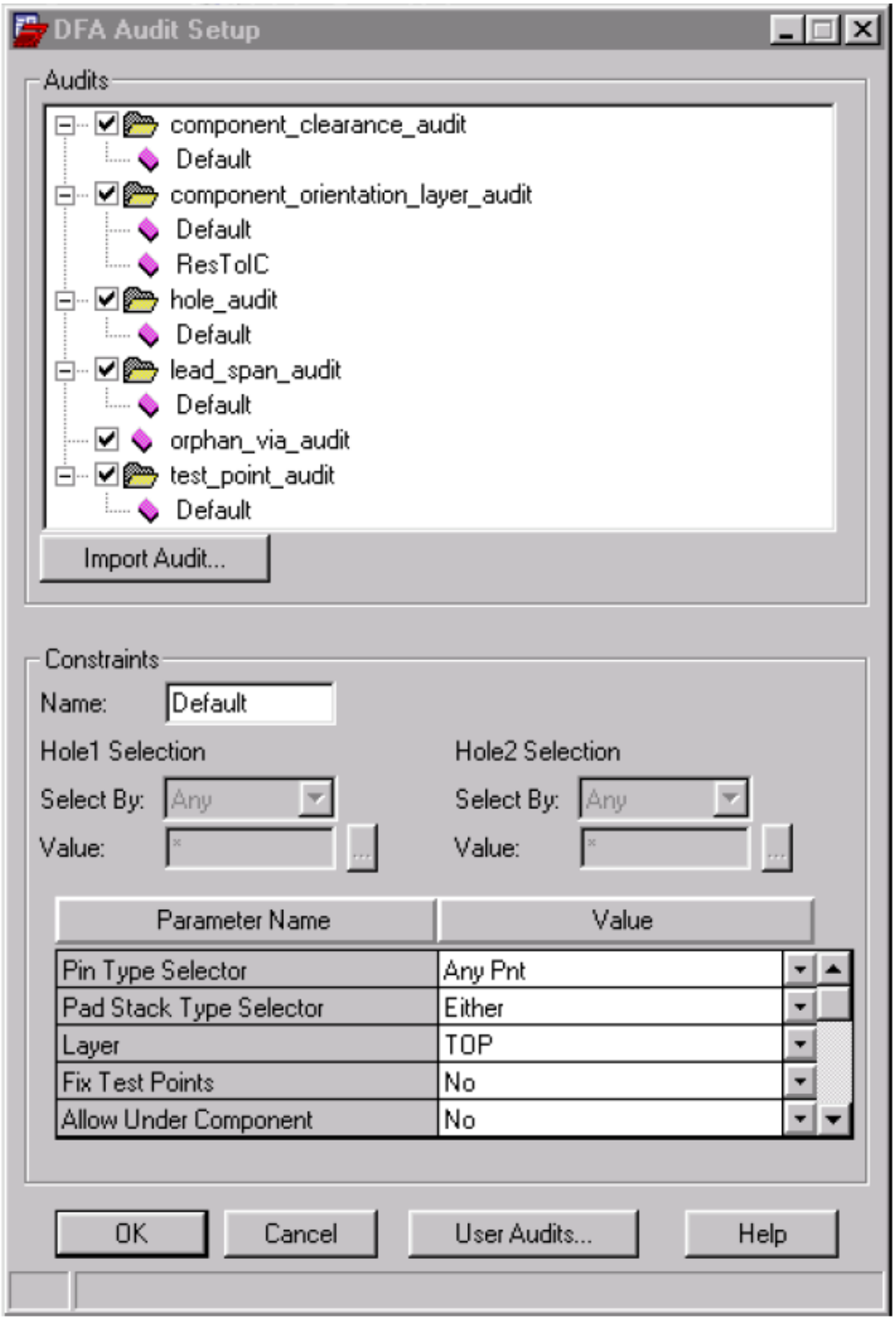
Refer to the *Allegro PCB and Package Physical Layout Command Reference* for details of field parameter values.

The error messages generated by this rule are:

```
INFO (hole_audit)
Maximum via size   : 2
Following vias violate this criteria:

        VIA1@(1000.0 2260.0)          14
        VIA1@(800.0 2540.0)           14
        VIA1@(1920.0 2560.0)          14
        VIA1@(2120.0 2260.0)          14

INFO (hole_audit)
Minimum via size   : 30
Following vias violate this criteria:

        VIA1@(1000.0 2260.0)          14
        VIA1@(800.0 2540.0)           14
        VIA1@(1920.0 2560.0)          14
        VIA1@(2120.0 2260.0)          14

ERROR (hole_audit)
Preferred plated hole size(s) : 5  6  7.

The following vias violate this criteria:

        VIA1@(1000.0 2260.0)          14
        VIA1@(800.0 2540.0)           14
        VIA1@(1920.0 2560.0)          14
        VIA1@(2120.0 2260.0)          14

ERROR (hole_audit)
Preferred plated hole size(s) : 5  6  7.
```

```
The following pins violate this criteria:

        R1.2                        14
        R2.2                        14
        R2.1                        14
        R1.1                        14

ERROR (hole_audit)
Minimum annular ring size for pads is 8.

The following vias violates this criteria:

        VIA1@(1000.0 2260.0)        3
        VIA1@(800.0 2540.0)         3
        VIA1@(1920.0 2560.0)        3
        VIA1@(2120.0 2260.0)        3

ERROR (hole_audit)
Minimum annular ring size for pads is 8 .

The following pins violates this criteria:

        R1.2                        3
        R2.2                        3
        R2.1                        3
        R1.1                        3
```

Testability



This audit checks for test points under components, the minimum/maximum size of test points, and tented test points (test points covered by solder mask).

A test point is defined as a pin/via with the attached property TESTPOINT and a value ETCH/CONDUCTOR/<SUBCLASS> placed on it, where <SUBCLASS> can be either top or bottom.

The error messages generated by this rule are:

```
ERROR(test_point_audit): Test point found under component

The following components have test points under them:

     U1 via1(70,85)

     U2 via2(30,908)

ERROR(test_point_audit):Test point size less than minimum

Minimum test point size   : 30
```

The following vias violate this criteria:

```
VIA1@(1000.0 2260.0)   3

VIA1@(800.0 2540.0)   3

VIA1@(1920.0 2560.0)   3

ERROR(test_point_audit):Test point covered by Solder Mask
```

The following test points are covered by solder mask:

```
VIA1@(1000.0 2260.0)

VIA1@(800.0 2540.0)
```

Hanging Trace

Choose this audit to report on cline segments that do not have a net assigned to them.

⚠ Cline segments assigned to dummy nets are not considered hanging traces, and this audit does not report them.