# Allegro/APD Design Guide: Getting Started

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>11</td>
</tr>
<tr>
<td>Design Tools</td>
<td>14</td>
</tr>
<tr>
<td>Important Information About Online Documentation</td>
<td>15</td>
</tr>
<tr>
<td>Online Books</td>
<td>16</td>
</tr>
<tr>
<td>Online Help</td>
<td>16</td>
</tr>
<tr>
<td>Printing Online Help</td>
<td>17</td>
</tr>
<tr>
<td>Using the Allegro/APD Design Guide</td>
<td>17</td>
</tr>
<tr>
<td>Conventions</td>
<td>17</td>
</tr>
<tr>
<td>Allegro/APD Terminology</td>
<td>21</td>
</tr>
<tr>
<td>Locating Information</td>
<td>21</td>
</tr>
<tr>
<td>Other Sources of Information</td>
<td>22</td>
</tr>
<tr>
<td>Product Installation</td>
<td>22</td>
</tr>
<tr>
<td>Related Documents</td>
<td>23</td>
</tr>
<tr>
<td>Late-Breaking Information</td>
<td>23</td>
</tr>
<tr>
<td>Customer Support</td>
<td>23</td>
</tr>
<tr>
<td>Contacting Cadence</td>
<td>24</td>
</tr>
<tr>
<td>Using SourceLink</td>
<td>24</td>
</tr>
<tr>
<td>Training</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>Getting Started with Allegro/APD</td>
<td>27</td>
</tr>
<tr>
<td>The Allegro/APD Program Suite</td>
<td>27</td>
</tr>
<tr>
<td>The Allegro/APD Design Editor</td>
<td>27</td>
</tr>
<tr>
<td>Padstack Designer</td>
<td>30</td>
</tr>
<tr>
<td>Maintaining databases</td>
<td>31</td>
</tr>
<tr>
<td>Setting up a UNIX Environment</td>
<td>39</td>
</tr>
<tr>
<td>Starting Allegro/APD Tools from an Operating-System Prompt</td>
<td>41</td>
</tr>
<tr>
<td>Starting Allegro/APD Tools from Windows NT</td>
<td>42</td>
</tr>
<tr>
<td>Opening and Saving Files</td>
<td>43</td>
</tr>
<tr>
<td>Opening New Designs</td>
<td>43</td>
</tr>
<tr>
<td>Creating a Symbol with the Package Symbol Wizard</td>
<td>44</td>
</tr>
</tbody>
</table>
Creating a Design with the Board Wizard ........................................ 51
Opening Existing Drawings .......................................................... 60
Setting Up Your Design .................................................................. 63
Saving Automatically ................................................................. 63
Saving to an Earlier Version ........................................................... 64
Protecting Files with Edit Locks ....................................................... 68
   Setting the Locking Mechanism through the User Interface .............. 68
   Setting the Locking Mechanism through the System Prompt ............ 72
Managing Files ............................................................................. 73
   Allegro/APD File Types ................................................................ 73
   Setting Up a Working Directory Structure ...................................... 74
Working With the Environment File on UNIX Systems ....................... 75
About the User Interface .................................................................. 76
   GUI Conventions ........................................................................ 77
   Using the Mouse ......................................................................... 77
   Viewing Your Design ................................................................... 77
The Allegro/APD Design Workspace .................................................. 81
The Design Window ........................................................................ 81
The Menu Bar ................................................................................. 82
The Icon Ribbon ............................................................................ 82
The Control Panel .......................................................................... 82
The Worldview Window ................................................................... 85
The Status Window ........................................................................ 89
The Console Window ...................................................................... 90
Pop-up Menus ................................................................................ 90
Changing Fonts ............................................................................. 91
Customizing the User Interface ....................................................... 92
Console/Keyboard Commands ............................................................ 92
Running Commands in the Background ............................................. 96

2
Generic Procedures ......................................................................... 97
Design Limitations ......................................................................... 97
Setting Up Your Drawing ................................................................ 98
   Specifying Drawing Parameters .................................................. 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Environment Commands with Scripts</td>
<td>148</td>
</tr>
<tr>
<td>Displaying Connectivity</td>
<td>150</td>
</tr>
<tr>
<td>Using Data Browsers</td>
<td>151</td>
</tr>
<tr>
<td>Displaying Quickview Information</td>
<td>151</td>
</tr>
<tr>
<td>Using Qvupdate to Display Quickview Information</td>
<td>153</td>
</tr>
<tr>
<td>Database and Library Selections</td>
<td>155</td>
</tr>
<tr>
<td>Running Commands with Strokes</td>
<td>156</td>
</tr>
<tr>
<td>Creating Strokes (UNIX only)</td>
<td>157</td>
</tr>
<tr>
<td>Defining Aliases</td>
<td>158</td>
</tr>
<tr>
<td>Creating Aliases</td>
<td>159</td>
</tr>
<tr>
<td>Deleting Aliases</td>
<td>160</td>
</tr>
<tr>
<td>Assigning Function and Control Keys</td>
<td>160</td>
</tr>
<tr>
<td>Function Keys</td>
<td>160</td>
</tr>
</tbody>
</table>

3

Managing Environment Variables                                       161

The Global Environment File                                           161
  Path Variables                                                       162
  System Variables                                                     164
  The Installed env File                                               164

Setting User-Defined Variables                                        168
  The set command                                                      168
  The settoggle command                                                169
  Creating a Local env File                                            171
  Defining Library Path Variables in a Local env File                  172

Setting Commands in the Console Window                                 173

The User Preferences Editor                                           173
  Setting User Preferences                                             173
  Customizing the Environment Variable Editor                          175

Setting Project Level and Site Customization Variables                181
  Project File Variables                                               181
  Site Customization                                                   183

Environment File Variables                                             186
  Autosave                                                              186
  Autovoid                                                             187
# Allegro/APD Design Guide: Getting Started

## 4 Command Mapping

<table>
<thead>
<tr>
<th>Module</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allegro</td>
<td>206</td>
</tr>
<tr>
<td>File</td>
<td>206</td>
</tr>
<tr>
<td>Edit</td>
<td>208</td>
</tr>
<tr>
<td>View</td>
<td>209</td>
</tr>
<tr>
<td>Add</td>
<td>209</td>
</tr>
<tr>
<td>Display</td>
<td>210</td>
</tr>
<tr>
<td>Setup</td>
<td>210</td>
</tr>
<tr>
<td>Layout</td>
<td>211</td>
</tr>
<tr>
<td>Void</td>
<td>211</td>
</tr>
<tr>
<td>Shape</td>
<td>212</td>
</tr>
<tr>
<td>Logic</td>
<td>212</td>
</tr>
<tr>
<td>Place</td>
<td>212</td>
</tr>
<tr>
<td>Route</td>
<td>213</td>
</tr>
<tr>
<td>Analyze</td>
<td>214</td>
</tr>
<tr>
<td>Manufacture</td>
<td>215</td>
</tr>
</tbody>
</table>
# Allegro/APD Design Guide: Getting Started

<table>
<thead>
<tr>
<th>Tools</th>
<th>216</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>216</td>
</tr>
<tr>
<td>APD</td>
<td>217</td>
</tr>
<tr>
<td>File</td>
<td>217</td>
</tr>
<tr>
<td>Edit</td>
<td>219</td>
</tr>
<tr>
<td>View</td>
<td>220</td>
</tr>
<tr>
<td>Add</td>
<td>220</td>
</tr>
<tr>
<td>Display</td>
<td>220</td>
</tr>
<tr>
<td>Setup</td>
<td>221</td>
</tr>
<tr>
<td>Layout</td>
<td>222</td>
</tr>
<tr>
<td>Void</td>
<td>222</td>
</tr>
<tr>
<td>Shape</td>
<td>222</td>
</tr>
<tr>
<td>Logic</td>
<td>222</td>
</tr>
<tr>
<td>Place</td>
<td>223</td>
</tr>
<tr>
<td>Route</td>
<td>224</td>
</tr>
<tr>
<td>Analyze</td>
<td>225</td>
</tr>
<tr>
<td>Manufacture</td>
<td>226</td>
</tr>
<tr>
<td>Manufacture–Plating Bar check</td>
<td>226</td>
</tr>
<tr>
<td>Tools</td>
<td>227</td>
</tr>
<tr>
<td>Help</td>
<td>228</td>
</tr>
</tbody>
</table>

## A

### Allegro/APD Configuration Guide

| System Requirements and Performance Enhancements | 231 |
| Operating Systems Requirements | 231 |
| Hardware and Operating System Requirements | 231 |
| Calculating Memory Requirements | 234 |
| Improving Performance (UNIX) | 236 |
| UNIX-Based Installation Directory Information and Troubleshooting | 237 |
| Files That Reference the Installation Directory | 237 |
| Checking File References to the Installation Directory | 237 |
| Automatically Correcting Installation Directory References | 238 |
| Displaying UI Dialog Boxes Correctly | 240 |
| NT-Based Installation Directory Information | 241 |
| Licensing Issues | 241 |
Allegro/APD Design Guide: Getting Started

Compatibility for Allegro/APD Libraries, Designs, and Scripts ...................... 241
Symbol Library and Padstacks ................................................................. 241
Database Compatibility across Platforms ......................................................... 242
Database Compatibility with Previous Software Releases ................................. 242
Database UPREV (DBDoctor) ........................................................................ 245
Script Compatibility ...................................................................................... 245
SKILL Compatibility ................................................................................. 245
IBM DFS ....................................................................................................... 245

Glossary ........................................................................................................ 247
Preface

This document set describes design methodologies and concepts for:

- Physical layout systems of printed circuit boards (PCBs) created with Allegro
- Microelectronic packages such as multichip modules (MCMs) or single chip modules (SCMs) created with Advanced Package Designer (APD).

With Allegro products, you can place and route a board design, and generate the output and documentation necessary for its manufacture. APD functionality lets you accomplish the major physical layout tasks of microelectronic package design.

This preface contains the following sections:

- Design Tools
- Important Information About Online Documentation
- Using the Allegro/APD Design Guide
- Other Sources of Information
- Customer Support

For information on platform or operating system information, see the .

For information on new features in Cadence PSD offerings, see the Product Notes section of your user documentation.

The Allegro and APD families of products are integrated suites of software tools that Cadence offers for systems design. These integrated tools help you perform the major tasks of PCB and SCM/MCM design, including:

- Logic design entry
  Create a printed circuit board design based on data from a Concept or Capture schematic, or based on a netlist from another CAE system. Then backannotate from your design to the schematic. Update your Allegro and APD designs by performing engineering change orders (ECOs).

- Physical layout
Place design elements and route them, either manually or automatically with SPECCTRA™.

- Design analysis
  Perform design analysis with SigNoise and EMControl.

- Manufacturing output
  Generate silkscreens and penplots, and create artwork.

Figure 3-1 shows the functional relationship between Allegro/APD and other Cadence/EDA tools for logic design, physical layout activities, and design analysis.
Figure 3-1: Functional Relationship Between System Design Tools

Figure 3-2 on page 14 defines the typical PCB design flow process.
Figure 3-2: Design Flow Process

LIBRARY DEVELOPMENT
- Create custom pad shapes
- Define library padstacks
- Define unique packages
- Define mechanical elements

LOGIC DATA TRANSFER
- Create design database
- Associate schematic

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 COMPLETION
- Rename reference designators
- Backannotate
- Add power and ground planes
- Run Design Rule Checking (DRC)

MANUFACTURING OUTPUT
- Generate pen plots
- Create artwork
- Generate numerical control output

DESIGN ANALYSIS
- Signal integrity analysis
- EMI Compliance

Design Tools

This table gives an overview of Allegro/APD and the tools designed to work with them.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Used For...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padstack Editor</td>
<td>Defining library padstacks</td>
</tr>
</tbody>
</table>
## Important Information About Online Documentation

Many Cadence products are sold and licensed in different configurations based on features and price. Online books and online Help describe the *full* set of features in a product (that is, those features available in Allegro PCB Design Expert or AP Engineer); therefore, the information in this user guide may contain information on features not supported in the configuration that you are using.

The following sections provide more information:

- **Online Books**
- **Online Help**
Online Books

Online books can help you learn about the Cadence products and how they work together to achieve a design objective. You can view and print the books independent of Cadence applications.

Online books complement the online help to provide you with the information you need to use the product. Generally, the books document design flow methodology, aspects of circuit board design, and details on how Allegro/APD performs specific design tasks. For details on how to utilize the Allegro/APD design guide, see Using the Allegro/APD Design Guide.

Online Help

Online help is based on Microsoft Windows NT Help. It is designed to be used while you work with the Cadence application. It provides step-by-step instructions for performing specific tasks as well as information on user interface windows and dialog boxes. You can use any of the following methods to invoke online help directly from the application:

- Highlight a command from a pull-down menu and press F1
- Click the Help button on the dialog box
- Select the Help > Contents menu item
- Click the “question mark” icon on the main user interface

Once inside Allegro/APD Help, you can navigate to various topics by clicking Contents or Index, or by clicking the green underlined text. You also can do a full text search by clicking the Find tab on the table of contents.

The Allegro/APD online help documents all the commands and associated tasks that relate to Allegro/APD products.

Note: For information regarding the functionality covered by your Allegro/APD product license agreement, select Help > Product Notes.

Allegro/APD Help contains the following types of information:

- Overviews - Information that applies to the overall use of Allegro/APD.
- Executables - Information about the Allegro/APD Padstack Editor, and batch program capabilities
- Using Allegro/APD - Information about the tasks in Allegro/APD.
- Command Reference - Explanations of each menu command in Allegro/APD
Technical Definitions - General terminology associated with printed circuit board development and design, and terminology specific to Allegro/APD.

Printing Online Help

You can print specific online Help topics or complete books from the Help’s table of contents. If you are working from a UNIX workstation, you may have to set the PRINTER variable in your environment file to the desired printer; for example:

`setenv PRINTER <printer_name>`

Your system will use that printer for all print jobs.

Using the Allegro/APD Design Guide

This design guide employs certain conventions to describe the Allegro/APD user interface.

Conventions

Running Commands

The Allegro/APD user interface uses toolbars (menu items and icon buttons) to run commands. It also provides a console prompt (>) from where you can enter commands. Because Allegro/APD lets you customize the toolbars for your specific needs, this user guide documents the console command rather than the default toolbar selections that activate the command. For example:

Instead of...
Choose Route–Gloss–Parameters.

the convention used in this document is...

Run the gloss param command. This directive means that you type the command, as it is written in the design guide, at the Allegro or APD prompt.

For complete console command-to-menu selection mapping, see Command Mapping.

- Default values in dialog boxes

Dialog boxes are usually shown with the system default selections:

- Filled buttons are the default selections
- Filled-in fields are the default values

Figure 3-3 on page 19 shows the Test Prep dialog box.
Command Syntax Conventions

Although most Allegro/APD commands are run by menu selection from the user interface, some commands must be entered in a Run form or on a command line (for example, `a_extract` commands).
This list describes the command line syntax conventions used in this documentation. For information on command syntax for the Allegro Extension Language (AXL) see the *Allegro SKILL Reference Manual*.

**Courier**

This guide shows all Run/command line names and examples in Courier font.

**nonitalic**

Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.

**<variable>**

Words in italics indicate variables for which you must substitute a name or value. Names are case sensitive. Angle brackets (**<variable>**) may also enclose variables.

For example

Use the switch option `-s <symfile>` to refresh specified symbols.

```
| command argument | argument
```

**[]**

Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.

**{}**

Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

```
... Three dots (...) indicate that you can repeat the previous argument. If you use the dots with brackets, you can specify zero or more arguments. If they are used without brackets, you must specify at least one argument, but you can specify more.

```

**argument...** : specify at least one, but more are possible

```

[argument]... : you can specify zero or more

```

```
,... A comma and three dots together indicate that if you specify more than one argument, you must separate those arguments by commas.

```

The following is an example of the *bbvia* batch command:
bbvia [-p <prefix>] [-t] [-c <cns>] <padname> <startlayer> <endlayer> <input_layout> [<output_layout>]

### Allegro/APD Terminology

Most terms used in this book are common to both Allegro and to APD. Instances where terminology differs are addressed throughout the book (for example, ETCH/CONDUCTOR). The table below lists terms that are uncommon to the tools.

<table>
<thead>
<tr>
<th>Allegro</th>
<th>APD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.brd</td>
<td>.mcm</td>
<td>File name extension</td>
</tr>
<tr>
<td>ETCH/TOP</td>
<td>CONDUCTOR/SURFACE</td>
<td>Layer subclass name</td>
</tr>
<tr>
<td>ETCH/BOTTOM</td>
<td>CONDUCTOR/BASE</td>
<td>Layer subclass name</td>
</tr>
<tr>
<td>BOARD</td>
<td>SUBSTRATE</td>
<td>Foundation</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>PART</td>
<td>General elements</td>
</tr>
</tbody>
</table>

Class names that derive from any of the above follow the naming conventions of the tool; for example, Package Keepin/Keepout in Allegro is Part Keepin/Keepout in APD.

### Locating Information

This table lists the documentation contained in the Allegro/APD Design Guide. Note that information is sequenced to reflect the standard PCB design flow, as illustrated in Figure 3-2 on page 14.

<table>
<thead>
<tr>
<th>For Information About...</th>
<th>See...</th>
</tr>
</thead>
<tbody>
<tr>
<td>System requirements and performance. The user interface; using Allegro/APD, starting and exiting, controlling the graphic display, graphic and text elements; design and system information; environment settings; command-to-menu mapping; glossary definitions.</td>
<td>Getting Started</td>
</tr>
<tr>
<td>Building and managing libraries, including defining padstacks, custom pads, packages, electrical attributes, and formats.</td>
<td>Developing and Defining Libraries</td>
</tr>
</tbody>
</table>
**For Information About...** | **See...**
---|---
Loading logic design data and converting third-party mechanical data, including loading data from Concept, Capture, netlists, and board mechanical data. Includes information on creating netlists. | Logic Transfer Data
Design rules, DRC, dictionaries of properties and constraints | Design Rules
Laying out cross-sections, defining etch shapes, adding graphic elements. | Layout Preparation
Placing components, including automatic and interactive placement; working with groups and modules. | Placement
Routing, including interactive and SPECCTRA routing; test preparation; glossing. | Routing
Design output, including renaming reference designators, creating drill and silkscreen data, running constraint audits. | Design Completion
Extracting views, generating penplots, and drafting and dimensioning. | Manufacturing
APD-specific chapters and appendices: AXL-SKILL | APD-Specific Information
AXL-SKILL | SKILL Reference Manual

**Other Sources of Information**

This section provides pointers to other sources of information for this product.

**Product Installation**

Cadence ships installation and licensing instructions with the product. The instructions include information about using Cadence’s document viewer, *cdsdoc*, to view online books.
Related Documents

The documentation CD-ROM contains documentation for all NT-related Cadence products. For detailed information on various aspects of electronic design automation as related to Allegro/APD but not addressed in this design guide, refer to the information in the table below. Note that some documents may be provided in an online Help format.

<table>
<thead>
<tr>
<th>For Information About...</th>
<th>See...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front end (schematic) tools and logic conversion; HDL library architecture</td>
<td>Getting started with PCB Design Studio Concept User Guides; Capture User Guide; Packager-XL User Guide; HDL Direct User Guide</td>
</tr>
<tr>
<td>Signal Analysis</td>
<td>SigNoise User Guide; SigXplorer User Guide</td>
</tr>
<tr>
<td>EMControl</td>
<td>Allegro/APD Online Help</td>
</tr>
<tr>
<td>Constraint Manager</td>
<td>Constraint Manager Online Help</td>
</tr>
<tr>
<td>Core SKILL use and functions</td>
<td>SKILL Language User Guide; SKILL Language Reference; SKILL Development Functions: SKILL IPC Functions Reference; SKILL++ Object Systems Functions</td>
</tr>
</tbody>
</table>

Late-Breaking Information

Information about Allegro/APD and other Cadence products that becomes available after the product has been shipped may be published in a Release Alert on SourceLink or the Cadence ftp site. See Using SourceLink for more information.

Customer Support

Technical support is available for customers who have a maintenance agreement with Cadence. The Cadence Customer Response Center operates from 8 a.m. to 8 p.m. U.S. Eastern Time.

This section describes the following:

- Contacting Cadence
- Using SourceLink
Contacting Cadence

You can use the following methods to contact Cadence:

- By phone:
  1-800-CADENC2 (1-800-223-3622)
- By electronic mail:
  support@cadence.com (North America)
- By fax:
  East Coast (Chelmsford, MA) 978-446-6007
  West Coast (San Jose, CA) 408-944-7155

Using SourceLink

SourceLink can be accessed by customers who have a SourceLink account with Cadence. The details of how to contact Cadence by phone, fax or email from locations throughout the world are available at the Web page

http://sourcelink.cadence.com/supportcontacts.html

You can also use SourceLink to:

- Search the Customer Response Center Solutions Database
- Create Service Requests directly with the Customer Response Center
- Check the status of your Service Requests and Product Change Requests (PCRs) (customers in North America only)
- Get information on current and upcoming releases
- Read technical application notes
- Get SKILL code written by application engineers and other customers

Details of SourceLink can be accessed at the Web page

http://www.cadence.com/support/
Training

Cadence offers many customer education services. Ask your sales representative for more information.
Getting Started with Allegro/APD

This chapter introduces you to Allegro/APD. If you are new to Allegro/APD, read this chapter to familiarize yourself with setting up and starting Allegro/APD and with the features and functions of the user interface; if you are an experienced Allegro/APD user, this chapter will acquaint you with changes from earlier versions.

**Note:** This design guide describes the features and functionality of the Allegro product suite and of Advanced Package Designer (APD). For information specific to APD, see the APD chapters of the user guide.

### The Allegro/APD Program Suite

When you install Allegro/APD on your computer, the program automatically creates these tools:

- Allegro/APD Workspace Editor
- Padstack Designer
- SPECCTRA Automatic Router
- DBdoctor
- SPECCTRAQuest (with Allegro Expert)
- Online documentation

Also installed are a number of programs that you run from an operating system prompt. These programs may display graphical user interfaces when run, or they may require that you enter arguments and options from the keyboard. These programs are documented in the appropriate sections of this design guide.

### The Allegro/APD Design Editor

The Allegro/APD Design Editor lets you create or modify designs. It can be started and run as a stand-alone tool or as the layout portion of a complete design solution managed with
Project Manager. For further information on Project Manager, see “Getting Started with PCB Design Studio” and the online Help in Project Manager.

**Figure 1-1 The Allegro/APD User Interface.**

**Note:** The illustrations in this document depict Allegro Design Expert interfaces, in most cases. Where there are significant differences between different offerings of Allegro and between APD, such differences are noted.

**Design Editing Modes**

The Allegro/APD design workspace takes many forms—or design editing *mode*—depending on the type of design activity. This affords you the convenience of using a single, variable-mode editor to complete your design.
Note: You must use a separate editor to create or modify a library padstack. See “The Allegro/APD Program Suite” on page 27 for information on invoking the padstack editor.

The commands (menu picks and icons) available from the Allegro/APD workspace change to reflect one of the following major design tasks:

- **Layout creation and modification**
  
  Create board file (.brd) or design file (.mcm) databases in this editing mode. Use this mode to perform such tasks as component placement, board/design routing, and other functions.

- **Symbol creation and modification**
  
  Create symbols for your design in the symbol editing mode. Allegro/APD appends the appropriate filename extension when you save your symbol.

There is also a mode for Shape editing, which is accessible from the Layout or Symbol editing modes.

You enter a design editing mode by specifying a file type when you use the new or open commands from the editor. If you are running Allegro/APD on Windows, you can invoke the file from the Windows NT Explorer (assuming you have set up a file association).

You can also invoke the shape editing mode from the layout editing menus with the shape edit or appropriate add shape commands.

**Note:** If you are working with a shape symbol, you must first fill the shape (shape fill) before you can exit the shape editing mode and subsequently exit the Allegro/APD workspace editor.
Padstack Designer

The Padstack Designer lets you create or edit library padstacks, including:

- Defining the parameters of your padstacks
- Creating blind and buried via padstacks
- Adding padstack layers
- Copying padstack layers
- Deleting layers in a padstack

Figure 1-2 The Padstack Designer User Interface

A library padstack defines pad data for all layers. You must define padstacks before you create any package/part symbols, because each pin in a package/part symbol must have an associated padstack.
When you double-click the Pad Designer icon (in Windows) or type `pad_designer` at your UNIX system prompt, the Padstack Designer appears.

For information on the Padstack Editor, see “Using the Padstack Designer” in “Library Padstacks”.

**Maintaining databases**

The **DBdoctor** program checks your database for errors and other problems and reports them as they occur. DBdoctor supports `.brd`, `.mcm`, `.mdd`, `.psm`, `.dra`, `.pad`, `.sav`, and `.scf` databases. DBdoctor can:

- analyze and fix database problems
- eliminate duplicate vias
- perform batch design rule checking (DRC)
- uprev databases more than one revision old

**Running DBdoctor**

To verify the integrity of an Allegro/APD drawing database at any time during the design cycle, run DBdoctor at regular intervals but always after completing a design and prior to creating an artwork file.

You can run DBdoctor from within Allegro/APD to verify work in progress, or from a terminal window outside Allegro/APD, perhaps to check multiple input boards in batch mode by using wildcards and various switches. You need not have Allegro/APD running to use DBdoctor.

During processing, DBdoctor generates `dbdoctor.log`, which records check summaries and detailed information on records that contain errors, as well as names of symbols and nets and x.y coordinate information. If DBdoctor finds an error, then it adds the `dbdoctor.log` to the board as an attachment. Allegro/APD only saves the log file from the last run of DBdoctor that found an error.

DBdoctor uses the input file name by default and copies it as `<board_name>.orig` in the same directory, thereby permitting you use wildcards. If you use wildcards with the input file, then each board you enter is copied under `<boardname>.orig`, unless you select the *No Backup* field on the dialog box that appears when you launch DBdoctor externally or use the `-no_backup` switch, in which case, Allegro/APD overwrites the original board. The following are available options:
To run DBdoctor:

1. Use one of the following methods to launch DBdoctor.

   a. Open the drawing in Allegro/APD whose database you wish to check and select *Tools*– *Database Check*.

   b. Use *Start* – *Run* on Windows NT and type `dbdoctor_ui`.

   c. Type `dbdoctor_ui` in a terminal window.

   d. Open a terminal window and navigate to the Allegro/APD working or user-defined directory. At the Unix command line, type:

```
  dbdoctor [-drc_only | -drc ] [-no_backup] [-outfile <output_boardname>] input_boardname ...
```

   To run `dbdoctor` on all the boards in a directory, type:

```
  dbdoctor [-drc updates all DRCs, including BATCH only.]
  -drc_only deletes all XTALK_REL records from all nets and updates all DRCs (including BATCH only) as the batch_drc program does.
  -shapes analyzes all shape elements of the database for problems and deletes rectangles that are straight lines. Errors found in shapes indicate the segment on which the error was found.
  -no_backup prevents copying the original board to `<boardname.brd>.orig`. This overwrites the original board without a backup.
  -outfile `<newboardname.brd>` saves the processed file to another file name, which is available as an option only if you enter a single board name in the *Input design* field. If you don’t specify a file name, Allegro/APD saves the processed file to the input design file name, overwriting the design. If you use wildcards with the input design name, then each board entered is copied to `<boardname>.orig`, unless you use the `-no_backup` switch: In that case the original board is overwritten. If you enter a single board and an output file, `-no_backup` is automatically selected.
2. The **DBDoctor dialog box** in Figure 1-3 on page 33 displays if you run DBDoctor externally using `dbdoctor_ui`; the dialog box in Figure 1-4 on page 34 displays if you run DBDoctor within Allegro/APD.

**Figure 1-3  DBDoctor dialog box launched externally**
3. If you launched DBdoctor externally using `dboctor_ui`, the DBdoctor dialog box as shown in Figure 1-3 on page 33 appears. Enter the following:

   a. Enter a file name in the *Input design* field. This can be any design, drawing, pad, or symbol file. You may also enter wildcard names when updating library data or multiple boards in a directory. To search for existing filenames, click ... to display the file browser from which you can choose an existing filename. To update multiple files, start the `Dbdoctor_ui` program from the desired data location and type wildcard names in this field.

   b. Enter a file name in the *Output design* field to save the processed file to another file name, which is available as an option only if you enter a single board name in the *Input design* field. If you don’t specify a file name, the processed file is saved to the *Input design* file name, overwriting the design.

   You may also enter wildcard names when updating library data or multiple boards in a directory. To search for existing filenames, click ... to display the file browser from which you can choose an existing filename.

   If you use wildcards with the *Input design* name, then each board you enter is copied to `<boardname>.orig`, unless you select *No Backup*: In that case the original board is overwritten. If you enter a single board and an output file, *No Backup* is automatically selected.

   c. Select *No Backup* to prevent copying the original board to `<boardname.brd>.orig`. This overwrites the original board without a backup.
4. Select *Update ALL DRC (including BATCH)* to re-compute DRC in the entire design for all constraints that have a DRC Mode of either Always or Batch.

5. Select *Shapes* to analyze all shape elements of the database for problems. Rectangles that are straight lines are deleted.

6. Click *Check* to initiate the database check. If you entered `-drc` or `-drc-only` or selected *Update ALL DRC (including BATCH)*, the console window displays the message “Performing DRC. Please wait,” and the dialog box displays “DBdoctor in progress.” When the database check is finished, the console window displays the message, “DRC done, xxx errors detected. Done dbdoctor,” and the dialog box displays, “DBdoctor completed.”

7. Click *Viewlog* to review the `dbdoctor.log` containing the results of the database check.

   If DBdoctor detects errors in an Allegro/APD database, the log lists erroneous drawing elements along with their X, Y location coordinates or a net/symbol name.

   a. Examine those regions in the drawing; then delete and reinstate the erroneous elements.

   b. Run DBdoctor on the drawing again to ensure the errors have been resolved.

**Example of a dbdoctor.log file**

The following is a sample `dbdoctor.log` file.

```
****************************************
DBDOCTOR of drawing D:\drive\cards\slide\samples\45angle.brd
****************************************
ERROR IN PAD STACK name = 0X0_SP
   ILLEGAL NULL PAD
   Error cannot be fixed.
ERROR IN T location = (15079.527, 559.526)
   ILLEGAL CONNECTION
   Error was fixed.
ERROR IN T location = (4555.999, 501.271)
   ILLEGAL CONNECTION
   Error was fixed.
ERROR IN T location = (4713.000, 557.581)
   ILLEGAL CONNECTION
   Error was fixed.
ERROR IN T location = (5907.000, 7630.000)
```
Partial versus full database consistency checks on saving

When you save a design, Allegro/APD executes a partial database consistency check by default, in essence, a quickcheck.

The `dbsave_full_check` environment variable indicates to the database save utility when to do a full check rather than a quickcheck. A number of 1, or 0 specifies that each time a board is saved, execute a full check. If you set the variable to 100, then every 100 checks a full check occurs.

For example, to set the `dbsave_full_check` environment variable to do a full check every five saves, at the console window command line type:

```
set dbsave_full_check = 5
```

If Allegro/APD detects errors, it saves the file as `<boardname>.SAV`.

**Note:** A full database check may considerably lengthen the time required to save large databases.

Working with `.SAV` databases

Allegro/APD creates a `.SAV` database due to an abnormal exit or an error during execution of a quickcheck. DBdoctor saves the drawing as `<boardname>.SAV`. (On a PC, a Dr. Watson error is generated.)

A database is considered a `.SAV` based on its internal contents, meaning that even if a `.SAV` database is renamed as a `.brd`, it is still considered a `.SAV` database. Databases that become `.SAV` databases have a write lock attached to them. DBdoctor can save `.SAV` databases with write locks, and if that occurs, DBdoctor deletes the write lock. Dbdoctor changes the state of a `.SAV` board only if it does not find any FATAL errors. All applications can open a `.SAV` files but cannot save them unless you remove the write lock using **File – Properties**.

When DBdoctor discovers an error in the database, Cadence recommends that you restore an earlier, error-free version of the file and work around the procedure that caused the error,
or fix problems before continuing to modify the database. Since it is not possible to repair every database, you may still have to send the database to customer support to determine if it can be repaired.

A work session resulting in a crash and a .SAV can be recovered by converting the .jrl file into a script and running it on the original board. Use the .jrl file to re-run the work session and reproduce the corruption using one of the methods outlined as follows:

**Method 1: Converting a .jrl File Into an Allegro Script - Unix only**

When an Allegro crash or system failure occurs, you can replay a session as a script if a journal (.jrl) file (for example, allegro_layout.jrl or allegro_interactive.jrl) exists by extracting and modifying the commands in the .jrl file.

1. Edit the .jrl file and search for the last save of the board.
2. Remove the data from the beginning of the script up to and including the save, leaving the next command as the first line of the .jrl file.
3. Enter the following one line command, where the \i entries in the .jrl file are the commands used for replay. Each \i entry contains \i (time) command, grep for the \i entry, and print the record without the first two words.
   
   ```bash
   grep "\i " allegro_layout.jrl | awk '{print $3 " $4" "$5" "$6" "$7" "$8" "$9 "$10" "$11" "$12') > new_filename.scr
   ```
4. Delete the last two lines in the new script before replaying it to avoid crashing again.
5. Save the board after running the script.
6. Manually execute the last commands in the original journal file.
7. If the next step causes repeated crashes, notify Cadence customer support.

**Method 2: Converting a .jrl file into a script file on Unix or NT**

To convert a .jrl file into a script file, use a PERL script, which necessitates that the PERL executable be located in your path. The following PERL script requires that you change the top line to contain the path to PERL on your UNIX system. Removing the top line may be needed on NT.

The script name in this example is `replay_jrl.pl`.

**Sample PERL script**

--------------Script ------- Cut Here--------------
#!/usr/local/bin/perl

$prog_name = $0;
$prog_name =~ s|^.*/(.*)|\1|;       # Keep only base name

$| = 1;                             # don’t buffer output
{
    # make sure we have an argument - <ALLEGRO_APPLICATION>.jrl file name
    die "Usage: $prog_name jrl_file\n" if ($#ARGV != 0);

    $jrl_file = $ARGV[0];
    open (FILE, "$jrl_file") || die "Cannot find or open $jrl_file\n";
    while (<FILE>) {
        # look for records beginning with \i
        if (/\i <\S+> (.+)/) {
            print "$1 \n";
        }
    }
}

---------------------------Script end-------------Cut Here----------------

To use the script on Unix, type:

    replay_jrl.pl journal file name > new script name

For example, for allegro_layout, the journal file name is allegro_layout.jrl. Type:

    replay_jrl.pl allegro_layout.jrl > allegro_layout.scr

If you used other Allegro editors, you can replace allegro_layout.jrl with the .jrl file created from that particular editor.

To use the script on NT

Redirect input to PERL by formatting the command as follows:

    perl.exe -F replay_jrl.pl allegro.jrl > allegro.scr

Upreving

Opening a previous version of Allegro/APD in the current version update the database automatically. For example, you can open an Allegro 13.0 database in Allegro 14.0 and use File – Save to save it as an Allegro 14.0 database.
On either Windows NT or Unix, you can also use Dbdoctor to uprev the database to the current revision of software and move boards forward multiple revs. Use wildcard options to uprev an entire directory of boards.

For example, use Dbdoctor for a board that originated in version 11.0, thereby preserving the original version of the board and upreving it to a new name in the current version.

The oldest database revision support for uprev on a platform is based on when Allegro/APD initially supported that platform. The following table lists the older database that can be upreved.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Allegro version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun 4V</td>
<td>1.0</td>
</tr>
<tr>
<td>HP</td>
<td>5.0</td>
</tr>
<tr>
<td>AIX</td>
<td>4.0</td>
</tr>
<tr>
<td>Windows</td>
<td>11.0</td>
</tr>
</tbody>
</table>

### Setting up a UNIX Environment

The Allegro/APD tool set operates in a windows environment on UNIX workstations. If your workstation does not start its window system automatically, you must start it.

To start the window system

- Type one of the following commands at an operating-system prompt:
  - `openwin` to start the X Window System™ on a Sun workstation
  - `xinit` to start the X window system on an IBM workstation
  - To avoid an IBM X Server bug, Cadence recommends you type `xinit` as follows:
    ```
    xinit -- -bs
    ```

### Editing Your `.cshrc` File

If you are working in a C shell, you must source Allegro/APD’s `.cshrc` file to initialize your environment before starting Allegro/APD. You can do this in two ways:

- Source Allegro/APD’s `.cshrc` file each time you start Allegro/APD.
Copy the contents of Allegro/APD’s cshrc file in your own .cshrc file.

To source Allegro/APD’s cshrc file

➤ At an operating-system prompt, type

```
source install_dir/tools/pcb/bin/cshrc
```

The `install_dir` is the directory in which Allegro/APD was installed.

To copy the contents of Allegro/APD’s cshrc file into your own .cshrc file

1. If you do not yet have a .cshrc file in your home directory, use a text editor to create the file.

2. Copy the contents of the following file into your .cshrc file:

```
install_dir/tools/pcb/bin/cshrc
```

The `install_dir` is the directory in which Allegro/APD was installed.

Incorporating the Allegro/APD profile file into your korn shell environment

If you are working in a korn shell, you must incorporate the Allegro/APD profile file into your environment before starting Allegro/APD. You can do this in two ways:

■ Source Allegro/APD’s profile file each time you start Allegro/APD.

■ Copy the contents of Allegro/APD’s profile file in your own .profile file.

To incorporate Allegro/APD’s profile file into your korn shell environment

➤ At an operating-system prompt, type

```
.
install_dir/tools/pcb/bin/profile
```

The `install_dir` is the directory in which Allegro/APD was installed.

To copy the contents of Allegro/APD’s profile file into your own .profile file

1. If you do not yet have a .profile file in your home directory, use a text editor to create the file.

2. Copy the contents of the following file into your .profile file:

```
install_dir/tools/pcb/bin/profile
```

The `install_dir` is the directory in which Allegro/APD was installed.
Starting Allegro/APD Tools from an Operating-System Prompt

When you start an Allegro/APD tool from an operating-system prompt, you have the following options:

- Type an Allegro/APD editor command and do not include a drawing name, or include the name of a new drawing.
- Type an Allegro/APD editor command and include the name of an existing padstack, symbol, or layout (without an extension) to be opened in Allegro/APD.

To start an Allegro/APD tool from an operating-system prompt

➤ Type one of the following Allegro/APD editor commands:

  To open padstacks:
  
  pad_designer [-s script] [-o/-j journal]
  [-p startdir] [filename]

  To open symbols or layouts:
  
  allegro (or apd) [-product product_name] [-s script_name]
  [-o/-j journal] [-p start directory] [filename] [license_filename]

The arguments for the Allegro/APD commands *allegro* and *apd* are

- `-product product_Name` Determines the product tier (PCB Design Expert, PCB Studio, etc.) that is run.
- `-s script_name` Runs a specified script file.
- `-o journal` [Default] Starts a journal file that records your Allegro/APD work session. The name of the file is `<program>.jrl`.
- `-j journal` Lets you specify a startup directory. If you start Allegro/APD with a drawing name that includes a path to the drawing (for example, `/home/joe/pcb/boards/layout_name(.brd)`, other files created during processing (log and jrl files) are created in the directory you specified and not the directory in which the drawing is located.
- `filename` Specifies a design file. You do not have to include the file type (extension).
- `-product license_filename` Starts the product based upon the name of the product license file.
-proj cpm_file
  Reads the HDL-indicated cpm file at start-up.

-mpsXXX
  Standard Cadence mps argument support (This is not typically required.)

database_name
  Starts the product with the indicated database name.

-version
  Prints the version of the product, then exits. UNIX operating systems only.

-noGraphic
  Runs Allegro/APD in a non-graphic mode but still requires an X display. UNIX operating systems only.

For example

allegro -product <product_name> -s <scriptfile> <filename>

If you do not include a design name, Allegro/APD displays the editor you have selected and opens a default file called unnamed.pad, unnamed.dra, or unnamed.brd. You can then use the Allegro/APD open or new command to open an existing or new drawing from the user interface.

If you have previously opened sessions of Allegro/APD, the last saved design in the previous session will open, based on information written to the master.tag file.

The master.tag file is a text file automatically generated when you launch a session of Allegro/APD. The file contains the name of the last database that you saved before ending a session. Allegro/APD reads this file when you next launch a session and opens the database of that name.

If, for any reason, you do not want Allegro/APD to open to the last saved database, you can move or delete master.tag. Allegro/APD will then open to a new, unnamed board file.

To locate master.tag, open the Allegro or APD initialization (.INI) file, located in your pcbenv directory. Search on directory= to locate the file.

Starting Allegro/APD Tools from Windows NT

- Double-click the appropriate icon in the Allegro/APD Program Group.
  The graphical user interface (GUI) of that tool appears.
Opening and Saving Files

Once Allegro/APD is up and running, you can open new and existing drawings using the appropriate items in the File menu. (If you have created designs in previous sessions, Allegro/APD will open to the last saved design, based on information written to the master.tag file, described above.)

Opening New Designs

When you create a new design file, you must specify the type of design you want to create, using the New Drawing dialog box to select whether you want to create a board file or a symbol file.

Figure 1-5: The New Drawing Dialog Box

The choices are:

Layout

Creates a board file (.brd) or design file (.mcm). You create your design database in this editing mode. Use this file to perform such tasks as component placement, board/design routing, and other functions.

Board (wizard)

The board wizard provides an easy way for you to create a design layout. The wizard is designed either to help beginning Allegro users create a design, or for experienced users who want a quick way to perform routine setup procedures as a foundation for a more complex design database. The New Board Wizard is not available on APD. For details, see Creating a Design with the Board Wizard on page 51.
Symbol

You create symbols for your design in the symbol editing mode. Allegro/APD appends the appropriate filename extension when you save your symbol.

There are two files associated with a symbol. The raw, unprocessed, drawing file has a .dra filename extension. When you run `create symbol` from the symbol editing mode, the .dra file is compiled into the appropriate binary file — Package (.psm), Format (.osm), Mechanical (.bsm), Shape (.ssm), or Flash (.fsm). See Working with Symbols for information about symbol files. The symbol editor lets you create the following types of symbols:

**Package/ Part Symbol**

Creates a new component symbol such as an IC or a discrete. Allegro/APD saves package/part symbols to the symbol library, by means of File > Create > Symbol, and appends the file name that you specify with a .psm extension.

**Mechanical Symbol**

Creates a drawing symbol such as a card edge connector or a board/design outline. Allegro/APD saves mechanical symbols to the symbol library and appends the file name that you specify with a .bsm extension.

**Format Symbol**

Creates a drawing symbol such as a legend or a company logo. Allegro/APD saves format symbols to the symbol library and appends the file name that you specify with an .osm extension.

**Shape Symbol**

Creates a drawing symbol such as a special shape for a padstack. Allegro/APD saves mechanical symbols to the symbol library and appends the file name that you specify with an .ssm extension.

**Flash Symbol**

Creates a flash symbol such as a thermal pad for Rastar formats. Allegro/APD saves flash symbols to the symbol library and appends the file name that you specify with an .fsm extension.

**Creating a Symbol with the Package Symbol Wizard**

The Package Symbol wizard provides an easy way for you to create a package symbol. The wizard is designed either to help beginning Allegro users create a simple package symbol, or
for experienced designers who want a quick way to create a base Package Symbol that they can modify into a more complex symbol.

During the Package Symbol wizard process you are expected to specify padstacks to use with your symbol. When you select a padstack the data browser lists all the padstacks available in your current PADSTACK path (PADPATH). If you want to create your own padstacks for use with the package symbol you can use the Padstack Designer to do this before using the Package Symbol Wizard. You can also use the Padstack Designer to find out the details of the padstacks already in the Allegro/APD Padstack Library.

The Package Symbol Wizard lets you create simple package symbols. Following the instructions on the screens, you enter data and see visual representations of the package symbol. When you have finished creating your package symbol, the corresponding symbol drawing is automatically opened in the Symbol Editor. If you cancel out of the wizard during the process, a new drawing, with the name you specified in the New Drawing dialog box, opens in the symbol editor.

At all times you can move backward and forward through the wizard screens to modify your package.

**Package Selection**

When you select the package symbol, the thumbnail graphic shows you what your symbol looks like. You select the package symbol from the list of basic symbol types:

- **DIP (Dual Sided Through Hole Package)**
  The DIP Setup page for dual sided through-hole package type allows you to customize the following package attributes:

  Number of pins in the package (N). The default is 14 pins.

  Lead pitch (e). This is an editable field with a pop-up containing some industry-standard values. The default is 100 mils.

  Spacing between the two rows (e1). This is an editable field with a pop-up containing some industry-standard values. The default is 300 mils.

  Package width (E). The default is 250 mils.

  Package length (D). the default is 800 mils.

- **SOIC (Small Outline Package)**
  The SOIC Setup page for small outline package type allows you to customize the following package attributes:
Number of pins in the package (N). The default is 14 pins.

Lead pitch (e). This is an editable field with a pop-up containing some industry-standard values. The default is 50 mils.

Spacing between the two rows (e1). This is an editable field with a pop-up containing some industry-standard values. The default is 225 mils.

Package width (E). The default is 175 mils.

Package length (D). the default is 395 mils.

- **PLCC/QFP (Quad Flat Package)**

  The first PLCC/QFP Setup page for PLCC and QFP package type allows you to customize the following package attributes:

  Number of pins in the package along the vertical boundary (Nv).
  
  Number of pins in the package along the horizontal boundary (Nh).
  
  The location of pin number 1 may be selected from:
  
  The top left corner of the package
  
  The middle of the top row of pins (the default selection)
  
  Lead pitch (e). This is an editable field with a pop-up containing some industry-standard values. The default is 31.50 mils (0.8 mm).
  
  Spacing between the two rows (e1). This is an editable field with a pop-up containing some industry-standard values. The default is 225 mils.

  The second PLCC/QFP Setup page for PLCC and QFP package type allows you to customize the following package attributes:

  Spacing between the two terminal columns (e1). This is an editable field. The default is 425 mils (10.8 mm).
  
  Spacing between the two terminal rows (e2). This is an editable field. The default is 425 mils (10.8 mm).
  
  Package width (E). The default is 394 mils (10 mm).
  
  Package length (D). the default is 394 mils (10 mm).

- **PGA/BGA (Pin Grid Array Package)**

  The first PGA/BGA Setup page for PGA package type allows you to customize the following package attributes:
Maximum matrix size:
Pin count along vertical boundary (MD)
Pin count along horizontal boundary (ME)
Pin arrangement
Full matrix perimeter
Matrix arrangement:
Outer rows
Core rows
Staggered pin (on or off)

The second PGA/BGA Setup page for PGA package type allows you to customize the following package attributes:

Pin numbering scheme:
Number right - letter down
Number right - letter up
Number left - letter down
Number left - letter up
Letter right - number down
Letter right - number up
Letter left - number down
Letter left - number up
Number horizontally
Number vertically
Letter horizontally
Letter vertically
JEDEC standard
Pad numbered with zeros
The third PGA/BGA Setup page for PGA package type allows you to customize the following package attributes:

- Lead pitch along vertical columns (ev). This is an editable field with a pop-up containing some industry-standard values. The default is 100 mils.
- Lead pitch along horizontal rows (eh). This is an editable field. This is an editable field with a pop-up containing some industry-standard values. The default is 100 mils.
- Package width (E). The default is 1575 mils.
- Package length (D). The default is 1575 mils.

■ TH Discrete (Through Hole Discrete) Through Hole Discrete Page

The TH Discrete page for Through Hole Discrete Package type allows you to customize following package attributes:

- Spacing between the two terminal pins (e1). This is an editable field with a popup containing some industry-standard values. The default is 1000 mils.
- Package Width (E). This is an editable field. The default is 750 mils.
- Package Length (D). This is an editable field. The default is 300 mils.

The SMD Discrete page for SMD Discrete Package type allows you to customize following package attributes:

- Spacing between the two terminal pins (e1). This is an editable field with a popup containing industry-standard values. The default is 195 mils.
- Package Width (E). This is an editable field. The default is 145 mils.
- Package Length (D). This is an editable field. The default is 145 mils.

■ SIP (Single Inline Package)

The SIP Setup page for Single Inline Package type allows you to customize following package attributes:

- Number of pins in the package (N). The default is 8 pins.
- Lead Pitch (e). This is an editable field with a popup containing some industry-standard values. The default is 100 mils.
- Package Width (E). This is an editable field. The default is 110 mils.
- Package Length (D). This is an editable field. The default is 750 mils.

■ ZIP (Zig-zag Inline Package)
The ZIP Setup page for Zig-zag Inline Package type allows you to customize following package attributes:

- Number of pins in the package (N). The default is 14 pins.
- Lead Pitch (e). This is an editable field with a popup containing some industry standard values. The default is 50 mils.
- Spacing between the two terminal rows (e1). This is an editable field with a popup containing some industry standard values. The default is 100 mils.
- Package Width (E). This is an editable field. The default is 110 mils.
- Package Length (D). This is an editable field. The default is 750 mils.

When you have chosen the package type, you must complete the Template and General Setup screens. After you complete this information, the wizard presents specific screens that request information about your package choice. Default values are inserted in fields where relevant, and industry standard choices are available where applicable.

You choose the package symbol you want to create and then customize it. When you are satisfied with your choices, the package symbol is created with the following Allegro/APD objects:

**Pins**

Pins are added to the symbol at locations depending on the basic symbol you choose and the parameters you set for the symbol. You define a unique padstack for pin 1 and another padstack for all the other pins.

**Component Outline**

The Component Outline is a rectangle drawn to your specifications. This outline is added to class PACKAGE_GEOMETRY, subclasses ASSEMBLY_TOP and SILKSCREEN_TOP.

**Labels**

The Reference Designator you specify is added at default locations on class REF DES, subclass SILKSCREEN_TOP. It is also added to class REF DES, subclass ASSEMBLY_TOP.

**Constraint Areas**

The package keepout area is automatically generated from the component outline and pin information you provide.
Template Selection

A template is a .dra file that contains basic information for the package symbol. Cadence supplies a default template, or you can create your own template that contains basic information on colors, text sizes, or documentation for your symbol. Cadence recommends that if you are a new user of Allegro, you should use the default template.

**Note:** The default template is located at

```
<cdsroot>/share/pcb/pcb_lib/symbols/template/sym_template.dra
```

The default file is replaced each time you apply a hotfix or QSR. Therefore, if you edit sym_template.dra, you should store a copy outside the Cadence hierarchy.

You should use a customized template only after you are familiar with the Package Symbol wizard. When you use a customized template you can add pre-existing data or common data to your drawing. This data can include your units and accuracy, company logo, specific colors, text block definitions, and any other layer-specific data that you want to use.

Make sure your customized template is available before starting the Package Symbol wizard. The browser lets you navigate to your custom template.

**Note:** There is no path variable available to find customized template .dras. You must enter the path to customized files in the query field.

When you prepare your custom template note the following guidelines:

- The custom template should not contain any data that might interfere with the symbol generation process. In particular, it should have no data on the following classes:
  - PIN
  - ETCH
  - PACKAGE GEOMETRY
  - PACKAGE KEEPIN
  - PACKAGE KEEPOUT
  - REF DES
  - ROUTE KEEPIN
  - ROUTE KEEPOUT
  - VIA CLASS
  - VIA KEEPOUT
The symbol wizard uses Text Block 3 for the Reference Designator Prefix text size.

If you put more than one reference designator prefix in a template, the package symbol wizard will only recognize the first reference designator. This does not preclude you from changing the reference designator prefix in the wizard.

The symbol wizard uses Text Block 1 for Pin Number text size.

**General Setup**

The General Setup screen lets you set the units for the package symbol, the accuracy for the drawing, and the Reference Designator Prefix.

At all times you can select the choices from drop-down list boxes.

The default reference designator is U*. If you forget to append an asterisk ( * ) to your reference designator, the wizard appends one for you.

Generally you specify one padstack for Pin 1 and another padstack for the rest of the pins in your package.

When you select the padstack for Pin 1, it automatically appears in the list box on the screen for the other padstacks. Use the Browse button to select a different padstack for the other pins.

The Compiled Symbol Generation screen lets you specify whether or not you want to create the compiled symbol. In order to use a symbol within a board it must be compiled. The compile step checks the symbol for errors and produces a compact file.

If you choose to compile the symbol, the wizard creates both a .dra file and a .psm file in your current working directory.

If you do not compile the symbol, only the symbol drawing (.dra) is created. You can open the symbol drawing and continue editing it in Allegro/APD. When you finish editing the symbol use the File > Create Symbol command to create the compiled symbol (.psm) file.

**Creating a Design with the Board Wizard**

The Board Wizard provides an easy way for you to create a design layout. The wizard is designed either to help beginning Allegro users create a design, or for experienced users who want a quick way to perform routine setup procedures as a foundation for a more complex design database. Board wizard lets you create a basic framework for your design by defining these parameters with data you provide:
You can also use the board wizard to import custom design data by way of user-defined templates and technology files. Templates and technology files that you can import into your design database should contain the following types of data:

- Default parameter settings
- Company-default subclasses
- Color-to-layer assignments
- Constraint (DRC) rules
- Layer stack-up information
- Mechanical (.bsm) symbols

If you choose not to load data from template or technology files, Board Wizard lets you input the data manually, from the wizard's user interface screens.

**Note:** The Board Wizard is not available on APD.

**To create a new layout:**

1. Run `new` to open the New Drawing dialog box.
2. Enter a drawing name, select `Board (wizard)` as the drawing type, and click OK.

   The initial board wizard screen appears.

3. Follow the instructions for entering the required data on each of the wizard's screens, then click Next to move forward to the next screen. At any time before finishing the process, you can click:

   *Back* to review or modify data
Cancel to end the wizard process. A new drawing containing no design data opens in the editor with the name you specified in the New Drawing dialog box.

4. When you have completed the last step in the wizard process, click Finish.

The drawing is automatically opened in the Layout Editor.

The following sections describe the individual Board Wizard screens, and the actions required to move through the wizard process.

Introduction

This informational screen provides a brief summary of the capabilities and operating behavior of the wizard.

Template

Lets you choose whether to import a template file to your new layout. A template file is an existing user-created .brd file containing customized data. Information that you should include in a .brd template file includes:

- Default parameter settings
- Company-default subclasses
- Color-to-layer assignments

Note: Your template file should not contain any data on the following classes:

- ETCH
- PIN
- VIA

The board wizard will accept the following data from a template file. These parameters cannot be over-written by the wizard, but can be modified after your new layout has been created.

- Drawing size
- Board outline
- Spacing constraints
  - Minimum line width
  - Minimum line to line spacing
Minimum line to pad spacing
Minimum pad to pad spacing
Package and route keepins
Grid definitions
Cross-section definitions
If your template file contains only two ETCH layers, the wizard will let you
add more layers
define them as routing layers or power planes
If additional layers are defined in your template, this functionality is disabled in the wizard.

Data contained in your template file that can be over-written are:

- Board units
- Board origin

To import a board template file:

1. Select Yes in the Template screen, and click the browse button.
   The Board Wizard Template Browser appears, listing all the template files in the path
   WIZARD_TEMPLATE_PATH. Reset this path to the location you want to keep your
   templates in, if different than the default location. (You must restart Allegro to activate the
   change.)

2. Select a template from the list, and click OK.
   The file name appears in the text field of the Template screen.

3. Click Next.

Tech File

In addition to a template file, you can also import data to your layout by way of a technology
(tech) file. (See “Creating and Using Technology Files” in the Allegro/APD Design Guide: Defining and Developing Libraries for details on technology files.) If you import data using
a template file and a tech file, note that the data in the tech file will take precedence over data
brought in from your template. Information that you should include in a tech file template
includes:

- Constraint (DRC) rules
Layer stack-up information

To import a technology file:

1. Select Yes in the Tech File screen, and click the browse button.

   The Board Wizard Tech File Browser appears, listing all the technology files in the path TECHPATH. Reset this path to the location you want to keep your tech files in, if different than the default location. (You must restart Allegro to activate the change.)

2. Select a tech file from the list, and click OK.

   The file name appears in the text field of the Tech File screen.

3. Click Next.

Board Symbol

The Board Symbol screen lets you import a mechanical (.bsm) symbol as your board outline.

**Note**: If you imported template data that included a board outline, you should not import a symbol from this screen. Doing so will result in your new layout containing two board outlines.

Information that you should include in a .bsm template file includes:

- Board outline
- Company-standard design sheet

To import a board symbol:

1. Select Yes in the Board Symbol screen, and click the browse button.

   The Board Wizard Mechanical Symbol Browser appears, listing all the mechanical symbols in the path PSMPATH. Reset this path to the location you want to keep your board symbols in, if different than the default location. (You must restart Allegro to activate the change.)

2. Select a symbol from the list, and click OK.

   The file name appears in the text field of the Board Symbol screen.

3. Click Next.

   One of two screens appears:
   - Import Default Data
   - General Parameters
Proceed to the appropriate section.

Import Default Data

This screen is displayed only if you selected a data file (template, technology, or board symbol) to import into your new layout.

1. Choose *Import default parameter data now* to load the selected data files at this stage in the process. This makes the data available for viewing and modifying in subsequent screens.
   —or—

   Choose *Import the parameter data at the end of the wizard*. With this action, some of the parameters shown on subsequent screens and contained in the loaded data files, will be disabled by the wizard. They will not be available to view or to modify. Instead, those parameters will be loaded into your new layout at the end of the wizard process when the board is created.

2. Click *Next*.

General Parameters

This screen lets you set up design units, drawing size and origin. Functionality is based on whether you selected and/or loaded data files. If enabled, data that you enter into General Parameters takes precedence over parameters in the data files.

General Parameter settings with data files selected/loaded:

- Units: enabled (defaults to Mils)
- Accuracy: determined by selected design units
- Size: disabled
- Origin: all options enabled

General Parameter settings with no data files selected/loaded:

- Units: enabled (defaults to Mils)
- Accuracy: determined by selected design units
- Size: enabled
- Origin: corner and center options enabled

➤ When you have accepted or modified the parameters, click *Next*. 
One of three screens appears:

- General Parameters (Continued)
- Board Outline
- Keepins

Proceed to the appropriate section.

**General Parameters (Continued)**

This set up screen lets you specify additional parameters for your new layout. Functionality is based on data defined in loaded data files.

General Parameter settings with data files selected/loaded:

- Grid spacing: disabled
- Etch layer count: disabled, if more than two defined in a data file
- Artwork film generation: enabled

General Parameter settings with no data files selected/loaded:

- Grid spacing: enabled
- Etch layer count: enabled
- Artwork film generation: enabled

You can add etch layers (up to a total of 128 layers) through the wizard when Etch layer count is enabled. This function is disabled if a loaded data file has defined more than two etch layers.

Artwork files that you generate are defined for each etch layer and take the name of the defined layer, as specified in a data file or in the Etch Cross-section Details screen.

One of three screens appears:

- Keepins
- Board outline
- Etch Cross-section details

Proceed to the appropriate section.
Keepins

This screen appears if a loaded data file contains geometry on BOARD GEOMETRY/OUTLINE, but no data on ROUTE KEEPIN and PACKAGE KEEPIN.

1. Enter values for route keepin and package keepin distances from the board edge. Values are in design units.

2. Click Next.

Board Outline

This screen appears only if no data files are loaded or data does not exist on BOARD GEOMETRY/OUTLINE.

1. Select a board outline.

2. Click Next to proceed to the Board Parameters screen for the type of board outline you selected.

Circular Board Parameters

1. Enter values for the diameter of the board outline, route keepin distance, and package keepin distance from the board edge. Values are in design units.

2. Click Next.

Rectangular Board Parameters

1. Enter values for the height and width of the board outline, route keepin distance, and package keepin distance from the board edge.

2. If your board will contain corner cutoffs, check the option and enter a value for cut lengths.

3. Click Next.

➤ When you have accepted or modified the parameters, click Next.

One of three screens appears:

- Etch Cross-section Details
- Spacing Constraints
- Summary
Etch Cross-section Details

This screen lets you define etch layer names and types from the layers that you added in the previous General Parameters screen. The wizard creates top and bottom layers by default and does not allow you to change their names. Any other layer that you created from the wizard can be renamed and defined as a routing layer or power plane, with the option of defining power planes as negative layers.

Note: This screen does not appear if data you have imported from a template or tech file contains more than two etch layer definitions.

1. Define the name and type for each etch layer.
2. To define power planes as negative layers, check the negative layers option.
3. Click Next.

One of two screens appears:

- Spacing Constraints
- Summary

Proceed to the appropriate section.

Spacing Constraints

This screen lets you define basic spacing constraints and a default via padstack for your layout.

Note: This screen does not appear if data you have imported from a template or tech file contains constraint definitions.

1. Enter a value for minimum line width.
   - The same value is automatically assigned to the other spacing constraints.
2. If necessary, modify the other spacing constraints.
3. Select a default via padstack from the Board Wizard Padstack Browser, or by entering the padstack name in the via padstack field.
4. Click Next.
Summary

This is the final screen of the board wizard, containing the name of your new layout file.

➤ Click Finish to create the new layout.

The named layout file is created (or over-writes an identically named file) in your current working directory.

This completes the New Board Wizard process.

Opening Existing Drawings

You can open existing drawings in two ways:

■ From an operating-system prompt, as described in “Starting Allegro/APD Tools from an Operating-System Prompt” on page 41.

■ From within Allegro/APD using the File– Open menu selection or by typing open at the command prompt.

Note: You are prompted to save any changes made to an open design before opening a new file, but may be prohibited from doing so if the database has been locked. For details on database locking, see Protecting Files with Edit Locks on page 68.

You can display information for an existing drawing before opening it by using the Quickview window in the Open dialog box. Quickviews provide a high-level graphic overview or a summary of properties of the database you select from the list. The information that displays is based on the icon you press in the dialog box. Figure 1-6 on page 61 is an example.
Note to PC users: Quickviews is available only if you use the default file browser. If your browsers do not display Quickviews, your system may be using the Windows 3.1 browser, which does not support Quickview.

For additional information on Quickview, see “Using Data Browsers” in Generic Procedures.
Using Dbstat for Existing Drawings

Because functionality of software may change from one version to the next, you may find it helpful to determine what revision of software an existing drawing was last saved on. The stand-alone program, `dbstat`, lets you quickly view from what version and type of operating system a design database was previously updated. (This may not always be the version the design was created on.)

Dbstat supports the following file types (extensions):

`brd`  `mdd`  `scf`  `pad`  `dra`
`mcm`  `psm`  `ssm`  `fsm`

Dbstat has two default options:

- `-v` The version on which the design was last saved
- `-p` The platform on which the design was last saved (either UNIX or NT)

Because these are default options, you do not need to type them in; dbstat automatically returns the information to you. Therefore, the following command

```
dbstat test.brd
```

returns information similar to the following:

```
test.brd: 14 UNIX
```

**Note:** For padstack (.pad) designs saved prior to version 10, dbstat returns the message, “Pre-rev 10 pad file.”

Dbstat also accepts the wildcard character `*.<ext>` for instances when you want to display information on all the designs of a particular type in your directory.

**Note:** If you run dbstat from a remote location, you must provide the full path to the designs whose information you want to access.

To run dbstat from your operating system prompt, enter:

```
dbstat [-vp] [-v] [-p] <filename.ext>
```
Setting Up Your Design

When you create a new design file you must specify the size, units, and offset of the new design file. You do this through dialog boxes accessed in the Setup menu or as commands typed at the Command prompt of the user interface.

For details on setting up new designs, see “Setting Up Your Drawing” in Generic Procedures.

After defining the drawing parameters, you are ready to add an outline from the symbols library, or create a new outline. The drawing will be displayed in the design window, according to the values in the Drawing Parameters/Drawing Options forms. (For a description of the Allegro/APD design window and entire UI, see “About the User Interface” on page 76.)

Saving Automatically

Allegro/APD lets you automatically save an active design or symbol at regular intervals when you set the autosave environment variable. When Allegro/APD saves a design, it automatically generates a file named AUTOSAVE.brd (a symbol is saved to a file named AUTOSAVE.dra) and places it in the directory that was active when you opened Allegro/APD. If you change directories, Allegro/APD saves the file to the original working directory. The saved file is kept after you have closed and saved the design or symbol and exited from Allegro/APD.

Note: The autosave option is automatically disabled if you invoke the database locking command, file_property. For details on this feature, see Protecting Files with Edit Locks on page 68.

If the autosave time is reached when a command or non-filled shape is active, Allegro/APD displays a message that reads “Save Pending.” The save executes when the command is completed or when the shape is filled. If no command has been executed since the last autosave, Allegro/APD will not resave the design.

To activate the autosave utility

➤ Set the autosave variable in the Environment Editor by running the enved command
(See Managing Environment Variables for details on environment variables.)

—or—

➤ Before opening a design, execute the following command from the Allegro/APD command line:

    set autosave
You can specify the interval at which checkpoint saves are made by using the `set` command and the `autosave_time` variable as follows:

```
set autosave_time = <time>
```

The `<time>` can be set from 10 to 300 minutes. The default is 30 minutes.

You can change the default name (AUTOSAVE) of the file generated by Allegro/APD by

- Running the `enved` command to display the User Preferences Editor dialog box and entering a new value for `autosave_name` in the Autosave Category
- Using the following command and the `autosave_name` variable

```
set autosave_name = <filename>
```

Allegro/APD lets you specify whether a database check is performed when a design or symbol is saved with the autosave facility. To enable a database check

- Set `autosave_dbcheck` in the User Preferences Editor dialog box
- Execute the following command from the Allegro/APD command line:

```
set autosave_dbcheck
```

Note that enabling the database check during autosave requires additional processing time. The default is disabled.

To disable the autosave facility

- Uncheck `autosave` from the User Preferences Editor dialog box
- After opening a design, execute the following command from the Allegro/APD command line:

```
unset autosave
```

### Saving to an Earlier Version

Allegro/APD databases are backward compatible with their major version number (the number to the left of the dot). This means that databases created in or upreved to any revision within a major version (for example, to 14.1) can migrate between revisions of that version. You cannot save any major version to an earlier one, such as 14.x to 13.x, 13.x to 12.x, etc.
In version 14.2, use of certain features will prevent you from opening your design in 14.0. Specifically, POWER_AND_GROUND scheduling, auto thieving, and insertion of dialectric layers between the air layer above TOP and the air layer below BOTTOM are not available in version 14.0. Attempting to open a design in 14.0 that uses this functionality will generate this error message in the tool's command console:

W - ILLEGAL SYMBOL REVISION

and this message at your operating system prompt:

THIS DATABASE WAS SAVED WITH A NEWER VERSION OF 14.0. THIS VERSION OF 14.0 CANNOT OPEN THE DATABASE.

Therefore, you must run the downrev command from the newer revision while it is open. Doing so lets you then migrate the newer version backward by stripping out the feature(s) that the earlier revision does not support. Because downrev deletes the data that prevents you from opening the database in earlier releases, it is important that you do not over-write your 14.2 database.

**Note:** Output data derived from technology files in 14.2 designs are not downward compatible. Earlier versions cannot process the information from a 14.2 techfile.

Newer revisions of a design that do not employ functionality unsupported in earlier revisions can be migrated freely within the major version without using the downrev feature.

To use downrev:

1. From an open design, run downrev.

   A dialog box appears in one of two configurations.

   a. If your design database does not require downrev to open in an earlier revision, the dialog box appears as shown in Figure 1-7. If this configuration appears, close the dialog box, and use the normal procedure for opening the database in the earlier revision.
b. If your design database requires `downrev` to open in an earlier revision, the dialog box appears as shown in Figure 1-8. Click the **Save design to 14.0** button in the dialog box to display the file browser, and go to step 2.

2. Select or type in the file name you want to save the file to, as shown in Figure 1-9.

   **Note:** Selecting the same name as the current open design will over-write your 14.2-compatible design.
3. Click **Save**.

The Save design to 14.0 level dialog box presents a completion message, as shown in Figure 1-10.

**Figure 1-10 Downrev Completion Dialog Box**

The message reminds you that the current open design retains the functionality that will be unsupported when you attempt to open the database in the earlier revision.
Protecting Files with Edit Locks

You can secure any design database file by running `file_property` to set an optional password-protected database lock. Doing so marks your file as read-only in the database (as opposed to on your platform’s operating system). This ensures that your design is not accidentally over-written by you or an unauthorized user when attempting to save without saving as a different file name. In addition, you can set database locking to disable the export of design data such as writing techfiles, exporting libraries, and creating modules. Database locking also turns off the autosave environment variable. The locking mechanism does *not* prohibit you from performing an uprev of the database in batch mode; however, batch programs that open databases for writing, such as netrev and netin, are unable to perform their operations when the database is locked. Down-revving of locked databases is also prohibited.

When a database lock has been set, any attempts to edit the file results in an error message, warning the user that the database has been locked for saving. (Edit locking will *not* inform you if another user has the file open.) The lock can be disabled only by entering the password established when the file was locked or, if a password was not set, by unlocking it in the File Properties dialog box or through `dbdoctor`.

⚠️ **Important**

*It is extremely important that you keep a record of any passwords used to lock databases. Cadence does not support the recovery of databases in a locked state due to forgotten passwords.*

Because a design might be legitimately opened for updating by any number of users in a large, networked system environment, the File Property dialog box displays the name of the user who locked the file, when it was locked, and on which system it was locked. A comment field allows you to provide additional information. These comments, as well as the option for prohibiting design data export, cannot be altered when the file is locked.

You can set file locking in two ways: through the user interface, or at your system prompt.

**Setting the Locking Mechanism through the User Interface**

To lock an active file:

1. Run `file_property` at the user interface command prompt.

   The *File Properties* dialog box opens. If the file has not been previously locked, the dialog box opens in the configuration displayed in Figure 1-11.
2. Check *Lock database*.

   The fields and checkbox option become active.

3. Perform any of these optional actions:

   □ Enter a password. It may contain a maximum of 20 alphanumeric characters. Invalid characters are: spaces, backslashes (\), and dashes (-). Passwords are case-sensitive.

   **Note:** *It is extremely important that you keep a record of any passwords used to lock databases. Cadence does not support the recovery of databases in a locked state due to forgotten passwords.*

   □ Check *Disable export of design data*. This option prohibits the generation of file data by most library export programs.

   □ Enter additional comments.

4. When you have selected the options for locking the database, click *OK*.

   A File Property browser opens and prompts you to save the file to disk.
**Note:** If you cancel out of the File Property browser, the database remains locked in memory, allowing you one additional save of your file. Cancelling the File Property browser does *not* undo the OK action in the File Properties dialog box.

To unlock an active file:

1. Run `file_property` at the user interface command prompt.

   The File Properties dialog box opens. In Figure 1-12, the dialog box on the left shows the configuration displayed if the file was saved without a password. In the figure on the right, all options have been used.

**Figure 1-12 File Properties Dialog Box**

![File Properties Dialog Box](image)

2. Perform the appropriate action, as described below:

   ➤ To unlock a file without password protection, click **Unlock**, then **OK**.

   The dialog box closes and the file is now unlocked.

   —or—
a. If the file is password protected, click *Unlock*. The password window opens, as shown in Figure 1-13.

**Figure 1-13  Lock Password Window**

![Lock Password Window](image)

b. Enter the password, and click *OK*.

If the password is correct, the password window closes.
—or—
If you enter an incorrect password, an error message is displayed. Enter the correct password and click *OK*.

*Note:* It is extremely important that you keep a record of any passwords used to lock databases. Cadence does not support the recovery of databases in a locked state due to forgotten passwords.

c. Click *OK* to close the File Properties dialog box.

The file is now unlocked for editing.
Setting the Locking Mechanism through the System Prompt

You can lock files in batch mode with `dbdoctor`. Enter the following command and arguments at your operating system prompt:

```
dbdoctor [-lock] [-unlock] [-password <text>] [-exports <ENABLED or DISABLED>]
         [-lockComment <text>] <filename>
```

- **-lock**
  Locks the database. If already locked, this argument requires the password option to update additional command line lock attributes.

- **-unlock**
  Unlocks the database. If locked with a password, requires the password option to unlock.

- **-password**
  Sets a password. Allows a maximum of 20 legal alphanumeric characters. Illegal characters are: spaces, backslashes (\), and dashes (-). Passwords are case-sensitive and cannot be changed without first unlocking the database file.

- **-lockComment**
  Optional. Provides new—or updates existing—user comments. Using this option without providing text removes an existing comment string. Password option is required if the database is password-protected. Comment can be provided only when locking the database and cannot be changed without first unlocking the database file. Comments containing spaces must be enclosed in quotes; for example, “This file locked against unauthorized editing.”

- **-exports**
  Optional. Provides new—or updates existing—export status. Valid arguments are ENABLED or DISABLED. Password option is required if the database is password-protected. Exports can be provided only when locking the database and cannot be changed without first unlocking the database file.

To determine the status of a database file, use `dbdoctor` with the `-islocked` argument, or `dbstat` and the file name as its sole argument, as shown in the following examples.

```
dbdoctor -islocked <filename>
```

- **-islocked**
  When locked, returns the user name (owner of the lock), the system on which the file was locked, and any additional comments.

```
dbstat <filename>
<filename>: 14 UNIX LOCKED
```
Managing Files

Allegro/APD File Types

Allegro/APD automatically attaches the appropriate extension to the base filename that you specify. These extensions indicate the following file types:

<table>
<thead>
<tr>
<th>Extension</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>.brd</td>
<td>Board/Substrate file that represents the drawing database</td>
</tr>
<tr>
<td>.dra</td>
<td>Drawing file. You must create one of these before you create a symbol file. Later, this file is compiled into a binary symbol file.</td>
</tr>
<tr>
<td>.pad</td>
<td>Padstack file</td>
</tr>
<tr>
<td>.mcm</td>
<td>Multi-chip module file (APD)</td>
</tr>
<tr>
<td>.osm</td>
<td>Library file that stores format symbols</td>
</tr>
<tr>
<td>.psm</td>
<td>Library file that stores package/part symbols</td>
</tr>
<tr>
<td>.bsm</td>
<td>Library file that stores drawing or board/substrate symbols</td>
</tr>
<tr>
<td>.fsm</td>
<td>Library file that stores flash symbols</td>
</tr>
<tr>
<td>.ssm</td>
<td>Library file that stores shape symbols</td>
</tr>
<tr>
<td>.mdd</td>
<td>Library file that stores module definitions</td>
</tr>
<tr>
<td>.ncr</td>
<td>Output file in Excellon Format 2 for numerically controlled routers</td>
</tr>
<tr>
<td>.tap</td>
<td>Output text files that contain NC drill data</td>
</tr>
<tr>
<td>.txt</td>
<td>Text file, such as that used for parameters</td>
</tr>
<tr>
<td>.scr</td>
<td>Script and macro files</td>
</tr>
<tr>
<td>.log</td>
<td>Log file that contains data on processes</td>
</tr>
<tr>
<td>.art</td>
<td>Artwork files</td>
</tr>
<tr>
<td>.dat</td>
<td>Data files</td>
</tr>
<tr>
<td>.jrl</td>
<td>A journal file which contains a record of events — menu picks, keyboard activity, and so on — which are recorded for each session in Allegro/APD. You can share this data with Cadence Usability staff to help us learn how you use the product, which will assist us in our efforts to improve the user interface.</td>
</tr>
</tbody>
</table>
Opening a .pad file invokes the Padstack Tool. Opening a .brd file starts the Workspace Editor with the layout menu set. Opening a .bsm, .osm, .psm, .fsm for .ssm file starts the Workspace Editor with the symbol menu set.

When you finish with a .dra file in the symbol editor, run the create symbol command. Allegro/APD converts the file to a binary, symbol type file.

**Setting Up a Working Directory Structure**

Figure 1-14 on page 74 shows a suggested directory structure for Allegro/APD projects. This structure lets you have several project directories (for example, proj1 and proj2) and have subdirectories under each project.

**Figure 1-14  Suggested Directory Structure for Allegro/APD Projects**

```
Your home directory /
  /pcbenv / proj1 / proj2 / proj3 /
  | env
  | allegro_ui.col
  | version.txt
  / devices / symbols / routetest2 / routetest1 /
  | Project specific devices
  | *.txt
  | Project specific devices
  | *.psm
  | *.osm
  | *.dra
  | *.pad
  / pcb1.brd
  / pcb2.brd
  / router.log
  / drc.rpt
  / pcb1.brd
  / router.log
```

Note: / indicates a directory
The symbols and devices directories beneath a project directory contain symbols and devices that are unique to that project. These subdirectories parallel the structure of the library directories supplied by Allegro/APD in `<install_dir>/share/lib/pcb_lib`.

A project can also contain other subdirectories, such as temporary directories for routing tests that let you run batch routes without overwriting log or design files.

### Working With the Environment File on UNIX Systems

If you want to run Allegro/APD batch commands at the UNIX operating-system level, you must set the environment variables before you start the tools. For detailed information on the Allegro/APD environment file, see Managing Environment Variables.

To set the Allegro/APD environment variable:

- Do one of the following:
  - To enable the Allegro/APD environment variable by default for all windows in your system, create and edit a local environment file so that it includes the Allegro/APD env.

  The local environment file overrides the global environment file.
For details on using a local environment file, see “Creating a Local env File” in Managing Environment Variables.

- To enable the Allegro/APD environment variable by default for all windows in your system, edit your .cshrc file (using a text editor on your system) and add the following line (appropriate to the tool you are working in):
  ```
  setenv Allegro
  setenv APD
  ```

- To enable the Allegro/APD environment variable for only the current work session in a window, enter the following command at an operating-system prompt (appropriate to the tool you are working in):
  ```
  setenv Allegro
  setenv APD
  ```

**Note:** If you use this method of specifying the Allegro/APD environment variable, remember that Allegro/APD runs only in the window in which you specify the Allegro/APD environment variable. If you use multiple windows and want to run Allegro/APD in different windows, you should set the Allegro/APD environment variable in your .cshrc file or in a local environment file (see “Creating a Local env File” in Managing Environment Variables).

### About the User Interface

Allegro/APD features a task-oriented user interface, which consists of three menu sets offering a wide range of design options and graphic functions.

You perform different operations in Allegro/APD by opening a new or existing file in one of three editing modes:

- Layout Editor
- Symbol Editor
- Shape Editor

The Shape Editor is accessible from the Layout or Symbol editing modes.

<table>
<thead>
<tr>
<th>For Information On...</th>
<th>See...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Allegro/APD program suite</td>
<td>“The Allegro/APD Program Suite” on page 27</td>
</tr>
<tr>
<td>Working with the user interface</td>
<td>“About the User Interface” on page 76</td>
</tr>
</tbody>
</table>
GUI Conventions

The Allegro/APD graphical user interface (GUI) adheres to most Microsoft Windows™ standards for pull-down menus, accelerator keys, mouse use, icons, and so on. Allegro/APD differs from most Windows applications, however, in that it follows the command–then–object method of command execution:

1. First you choose a command
2. Then you choose the object(s) to be acted upon

Using the Mouse

Allegro/APD supports a three-button mouse. The following table shows the functions of each mouse button:

Mouse Button Functions

<table>
<thead>
<tr>
<th>Mouse Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>To select a design element as the target for the active command and to manipulate the GUI (resize design window, choose commands from menus, tabs, or icons, and so on)</td>
</tr>
<tr>
<td>Middle</td>
<td>To pan and use the view (zoom) features (see Viewing Your Design)</td>
</tr>
<tr>
<td>Right</td>
<td>To extend the functionality of the active command. For example, depending on the current command, clicking the right mouse button displays a pop-up menu that contains extended commands, such as: Oops (cancel), Done, Reject, Group, Align, Next, and so on.</td>
</tr>
</tbody>
</table>

Viewing Your Design

The easiest way to zoom in, zoom out, and move across your design display in the UI workspace is by way of the middle mouse button. The button gives you access to all the zoom features available from the menu bar or keyboard commands (except zoom in, which is
Allegro/APD Design Guide: Getting Started
Getting Started with Allegro/APD

integrated into zoom points) without the need to make a menu selection or enter a command at the console. Use of the middle mouse button also enables you to roam—or pan—across your design.

Roaming

Roaming—or scanning—are the terms used to describe the action of moving across your design in the workspace. To scan your design:

➤ With your cursor located inside the design workspace, click and hold the middle mouse button as you drag the cursor across the design. As long as the mouse button remains pressed, you can move all areas of your design into full view. You cannot drag the cursor outside the boundaries of the design.

Zooming

Zoom functionality is dependent on the position of the cursor relative to its location when you first click the middle button (your “starting pick”). Movement of the cursor up or down, left or right of this coordinate determines what zoom function is active (as shown in Figure 1-15). Zoom center is the active zoom mode when the cursor is at its starting pick (dynamically displayed in the design as concentric circles). The mode you are in is displayed in the status window and by way of dynamic shapes that bound the affected areas. The shape geometry associated with each command is:

zoom center

![zoom center](image)

zoom points

![zoom points](image)
To enhance performance, zoom out repaints your design in a minimalized draw mode. This "skeletal" view is maintained until your second middle-button click completes the zoom operation. While you are in this mode, the following conditions apply:

- Pins, vias, and ratsnests are not drawn
- Line segments are drawn without endcaps
- Lines are drawn in single pixel width
- Shapes are unfilled
- Only reference designator text is drawn
- Layer visibility settings are ignored (all layers are visible)
The zoom function remains active until you click the middle mouse button a second time. (Clicking the left mouse button also takes you out of zoom mode.)

If you prefer not to use the dynamic zoom features, you can disable the functionality by setting the environment variable `no_dynamic_zoom` in the User Preferences Editor. By setting this variable, middle-button functionality is limited to zooming in or zooming out.
The Allegro/APD Design Workspace

Figure 1-16 The Allegro/APD Design Workspace

Menu Bar
Icon Ribbons
Control Panel

Design Window
World view Window
Console Window
Status Window

The Design Window

The Design Window is the part of the workspace editor in which you create your design. You can change the default background color, using the color command. See “Assigning.”

January 2002

81

Product Version 14.2
Colors to Grids, Ratsnest Lines, Highlighting, and Background” in Generic Procedures for details.

The Menu Bar

The pull-down menus in the menu bar provide all of the commands that you need to create or modify a design. The menu command sets (Layout, Symbol, Shape) that are available to you depend on the task that you are performing and the Allegro/APD product you are running.

You can also use the accelerator key combinations to execute the command. The key combinations appear in the pull-down menu, to the right of the command.

The Icon Ribbon

The Allegro/APD icon ribbon contains icons that give you a quick way to access common Allegro/APD commands. To learn what a tool icon does, position the cursor over the icon without depressing the mouse button. A terse description of the icon appears. Icons can be customized to suit your specific needs.

The Control Panel

The Control Panel lets you toggle between the Options, Find, and Visibility tabs, and provides a Worldview Window that lets you zoom in or out on a design to let you see where you are in the design. For more information on the Worldview Window, see “The Worldview Window” on page 85.

To provide you with more space for your design, Allegro/APD lets you undock the Control Panel using the display param command.

Once you have undocked the Control Panel, you can shrink or minimize it as you would any Windows dialog box.

The Options Tab

The Options tab displays current parameters and values for the active command, and also contains fields that let you control the actions performed by the current command.
The fields that appear in the Options tab differ according to the active command. When you choose an Allegro/APD command, the Options tab changes to reflect the appropriate class and the default subclass (the first subclass on the list for that class). For more information, see “About Classes and Subclasses” in Generic Procedures.

The parameters and values you set in the Options tab take effect immediately and override definitions for the same parameters and values that may exist elsewhere in Allegro/APD. For example, Allegro/APD looks to the Drawing Options dialog box (status) for the rotation and text values. If you place a different value in the Options tab, however, Allegro/APD ignores the information in the Drawing Options dialog box.

**Note:** When you update values in the Drawing Options dialog box, the values in the Options tab change as well.

**The Find Filter**

The Find Filter lets you specify which design objects will be affected by the active command. When you run an interactive command, such as move, delete, or vertex, the Find Filter displays the elements that are required by the command.
For detailed information on using the Find filter and on finding elements in your design, see “Finding Design Elements” in Generic Procedures.

The Visibility Form

The Visibility form lets you selectively display or hide conductive elements in your design. Once you have assigned colors to each class of design element (from color), you can use the Visibility Form to selectively display etch/conductor, pins, and vias on each layer in the design. See “Assigning Colors to Design Elements” in Generic Procedures for more information on assigning colors.

The Visibility form displays the color assigned to a design element when that element is visible, and displays the background color of the design window when the design element is invisible.

When the button displays the assigned color, visibility is enabled and the design element is visible. When the button displays the background color, visibility is disabled and the design element is hidden. You can quickly control the visibility of all layers by clicking the All button associated with the desired design element.

You can delete plane layers in the Visibility form by clicking the Planes checkbox, a convenience if your design has a large number of layers that you might have to scroll through.
Figure 1-19  The Visibility Form.

![Visibility Form]

The Worldview Window

The worldview window gives you a quick way to zoom in or out on a particular section of a design. There are three ways you can control your view of the design in the design window:

- Use the worldview window to display specific areas of your design, or to zoom in and out
- Scroll the design window over the design
- Use Allegro/APD zoom functions
Using Worldview

The worldview window gives you a quick way to display a particular section of a design. It lets you zoom in to display a smaller area of your design outline in the design window, or zoom out to display a larger area of your design outline in the design window.

Displaying Specific Areas of Your Design

To use the worldview window to display specific areas of your design

➢ In the worldview window, click left and drag the display window over the area of the design that you want displayed in the design window.

   **Note**: If you are using a three-button UNIX mouse, the middle button gives you a greater degree of control when performing this operation.

If you size the display window over a small area of the board/substrate outline (using the left button), the design window zooms in on that area.

If you size the display window over a larger area of the board/substrate outline (using the left button), the design window zooms out to display that area.

You can use the worldview window alone or in conjunction with the View menu commands and accelerator keys.
Modifying the display area bounding lines

The display area in the worldview window is visible as a square or rectangle within the larger area of the design outline. This square or rectangle is typically displayed with white lines, but can be changed in the Display Group of the Color and Visibility dialog box. To change the line color of the display area:

1. Run color.
2. Select Display from the Group drop-down.
3. Select the last (furthest right) color box in the second row of the Palette.
4. Click Modify.
5. In the Color Editor, adjust the color setting accordingly, then click OK.
6. In the Color and Visibility dialog box, click Apply or OK.

Display areas in the worldview window will now be drawn in the color you designated.

The Worldview Window Pop-Up Menu

To display the worldview window pop-up menu

- Click right in the worldview window.

  The pop-up menu appears.

Following are descriptions of the options in the Worldview pop-up menu.

*Move Display* lets you move the display window to select an area of your drawing for display in the design window.
**Resize Display** zooms the design window on an area you define by selecting points in the worldview window.

You can also type `window center` at the Allegro/APD console window to perform the same function, but you then specify the new window area by selecting its center in the design window.

**Find Next** advances through the list of any highlighted items, centering the display on each of them, in the order in which they were highlighted.

**Find Previous** reverses through the list of highlighted items.

You can continue selecting Find Next or Find Previous by clicking left in the worldview window. The click repeats the last command. Find Next is the default command in effect with a left click after new elements have been highlighted.

The find direction and identification of the element being located is displayed in the Allegro/ APD status window. The `>` in the message line indicates the result of Find Next and `<` indicates Find Previous. For example, after centering around a line with Find Next, the message is `> Line`.

**Panning the Design Window**

You can remain at a zoomed in view, and move the design window across your design in any direction.

➤ Hold down the **Shift** key, click right, and move the mouse pointer toward the section of the design that you want to move into view.
For example, if you move the mouse pointer left, the design will appear to move to the right in the design window.

View Functions

You can control the view of your design by choosing commands from the View menu, or by using designated icons, function keys, keyboard accelerators, or mouse strokes. (See “Running Commands with Strokes” in Generic Procedures for details on command strokes.)

The table below lists the ways you can zoom in or out on a design, or move the design in the design window.

<table>
<thead>
<tr>
<th>Zoom Functions...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zooming in on your design</td>
</tr>
<tr>
<td>Zooming in on a specific section of your design</td>
</tr>
<tr>
<td>Zooming out on your design</td>
</tr>
<tr>
<td>Zooming out to a full view of your design</td>
</tr>
<tr>
<td>Zooming out to a full view of your design, excluding legends and borders</td>
</tr>
<tr>
<td>Centering an element in the design area</td>
</tr>
<tr>
<td>Zooming back to the last previous window extents</td>
</tr>
</tbody>
</table>

The Status Window

The Status Window shows the active command and the current x, y coordinates of the cross hairs. These coordinates change as you move the mouse.
The Status Window also has a *Stop* button, which halts execution of the currently active process.

**The Console Window**

The Console Window displays messages and command output, and is the command line in which you enter Allegro/APD commands.

**Figure 1-21 The Console Window**

```
Opening existing drawing...
Revising data for compatibility with current software.
Starting new drawing...
Grids are drawn 400.00, 400.00 apart for enhanced viewability.
Command > drawing param
```

**Pop-up Menus**

Several Allegro/APD commands provide pop-up menus that let you perform additional functions while the command is active.

To display the pop-up menu, click the right mouse button.

Depending on the active command, the pop-up menu that appears will look similar to the following:
The first three commands appear on every pop-up menu. The options below the first three vary, depending on the active command.

**Changing Fonts**

Allegro/APD lets you customize the look of the graphical user interface by changing the size and type of the fonts in the console, status, and Options windows, and in the Find Filter. This can be convenient if you find it difficult to read information presented in the default size and type.

To change fonts in the Allegro/APD user interface

1. Exit Allegro/APD, if you have it running.
2. Set the font variables in your environment file. ([Managing Environment Variables](#)) provides details on environment variables.

These variables can also be set in the System dialog box in the Control Panel.

```
fontSize = -12
```

where -12 represents the default font size. A larger negative number (for example -20) makes the font larger. Do not use positive numbers in this value.
fontFace = helvetica where helvetica represents the default font type. Fonts available to you depend on your platform and any user-installed fonts. The value is always a font name.

fontWeight = 500 where 500 represents bolded type. Change the value to 300 to produce unbolded type.

3. Restart Allegro/APD.

4. Resize the window if necessary to display all information in the larger font size.

You can also change font variables in the User Preferences Editor dialog box by running the `enved` command. Note that you must restart Allegro/APD to see the change.

**Customizing the User Interface**

Allegro/APD lets you customize its user interface to suit your site or individual needs. Menu customization is done through the menu selections under View > Customization. You can also customize the UI with two AXL-SKILL functions: `axlUIMenuDump` and `axlUIMenuLoad`. See the AXL-SKILL Reference Manual for details.

**Console/Keyboard Commands**

You can enter most Allegro/APD commands in the console window, though commands are often accessed through the menus. For example, you can type `new` in the console window, or choose File–New (the default menu selection) from the menu bar.

**Environment and System-Level Commands**

Environment and system-level commands can be accessed only through the console window. They include (but are not limited to):

### cd

The `cd` (change directory) command opens the active design in another location. The correct syntax is `cd <dir>`.

### pwd

The working directory name command (pwd) shows you the location of the active design.

### set/unset

The set command displays a window–Defined Variables–that lists the path names of all the environment variables in Allegro/APD’s directory hierarchy.
The `unset` command returns a variable setting to its previous value.

Note: You can also access a read-only Defined Variables window through the List All button in the User Preferences Editor dialog box (`enved`). For detailed information on setting environment variables, see Managing Environment Variables.

**pause**

The `pause` command lets you suspend processing of scripts. When the command is accompanied by an integer, as shown below

```plaintext
pause 30
```

the script suspends processing for the specified number of seconds before continuing.

**alias/unalias**

The `alias` command lets you:

- List aliases defined in your Allegro/APD environment
- Create a unique command vocabulary for frequently used commands or series of commands
- Assign commands to function keys

The `unalias` command deletes aliases that have been set for commands and function keys.

For details on these commands, see “Defining Aliases” in Generic Procedures.

**scriptmode**

This command enables and disables script replay and record options. For details, see “Using Environment Commands with Scripts” in Generic Procedures.

**stop**

This command stops both script and macro recording processes and closes the `.scr` file into which the script or macro was being recorded.

**source**

This command reads a file, typically your global environment file, when formatted as shown here.
source <filename>

The command can be nested up to four levels.

**skill**

This command lets you access the AXL-SKILL language processor. For details on AXL-SKILL, refer to the AXL-SKILL Reference.

**ifvar/ifnvar**

The ifvar command lets you include variables in scripts and environment files to change from new to old names. The ifnvar performs the opposite function.

For details on this command, see “Using Environment Commands with Scripts” in Generic Procedures.

---

**Command Browser**

You can access the complete selection of keyboard commands through the **Command Browser**. This dialog box lets you either run the command or view any online Help associated with that feature. To display Command Browser:

1. **Type helpcmd at the Allegro/APD command prompt.**

   The Command Browser appears.
2. Click either *Execute* (to run the command) or *Help* (to display documentation).

   **Note:** Not every command has a Help topic associated with it.

3. Select a command by clicking on the name.

4. To limit the display of command selections, use the *Filter* field and wildcards (\? for any single character and \* for multiple characters).

   a. Type a command name or enter a partial string with wildcards to display one or more commands.

   b. Press the Tab key.

   Example: Typing \?ol\* results in the display of the following commands:
   - color
   - color priority
   - colorview create
   - colorview load
colorview restore
polar

**Keyboard Shortcuts**

Keyboard shortcuts and accelerators let you perform a number of actions without using the mouse, including changing the view of your design and displaying dialog boxes from the user interface.

**Running Commands in the Background**

This section is specific to Allegro/APD on UNIX workstations.

Normally, when you run an Allegro/APD command from the Allegro/APD command line, you cannot use the Allegro/APD design window until the command is complete. When you type an Allegro/APD command at an operating-system prompt in a UNIX shell window, you cannot use the shell window until the command completes. By running Allegro/APD commands in the *background* you are able to continue using the design window or shell window.

While the background job is running, you can look at the contents of the output file with the UNIX commands *more* or *type*.

When a job completes, you are notified with a message in your console window.
Generic Procedures

This chapter describes generic operations that apply to a variety of Allegro/APD processes.

The chapter seeks to familiarize you with the user interface and its relationship to functions in Allegro/APD. Detailed descriptions of selection options in the various dialog boxes are available in online Help. (See Important Information About Online Documentation in the Preface for details.) Some of the functionality described in this chapter may not be available in all versions of Allegro/APD.

Design Limitations

Keep in mind the following limitations before setting up the parameters of your designs, as described in this section:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of connections per design</td>
<td>No limit</td>
</tr>
<tr>
<td>Maximum number of connections per net</td>
<td>No limit</td>
</tr>
<tr>
<td>Database resolution in inches</td>
<td>4 places</td>
</tr>
<tr>
<td>Database resolution in mils</td>
<td>2 places</td>
</tr>
<tr>
<td></td>
<td>The environment variable drawing_no4mils allows a setting of 4 places but may result in rounding problems in artwork output.</td>
</tr>
<tr>
<td>Database resolution in microns</td>
<td>2 places</td>
</tr>
<tr>
<td>Database resolution in millimeters</td>
<td>4 places</td>
</tr>
<tr>
<td>Database resolution in centimeters</td>
<td>4 places</td>
</tr>
<tr>
<td>Maximum design area size</td>
<td>Dependent upon units chosen and number of decimal points. Example: with Mil at 2 decimal places, you can design a board in excess of one million square inches.</td>
</tr>
</tbody>
</table>
Setting Up Your Drawing

This section describes the tasks that are necessary when setting up your drawing, including

- Specifying drawing parameters
- Defining text size
- Specifying grids
- Setting drawing options concerning display, design rule checks, text, lines, and symbols

Other setup parameters, such as design constraints and user-defined properties, are covered in other sections of your Allegro/APD documentation.

Specifying Drawing Parameters

The Allegro/APD Drawing Parameters dialog box lets you specify the following:

- Drawing type
- User units
The flow for specifying drawing parameters is:

1. Run the `drawing param` command.
   
   The Drawing Parameters dialog box appears.

2. Select the drawing type from the drop-down list box in the *Type* field.

3. Select the user units from the drop-down list box in the *User Units* field.

4. Select the size of the active design from the drop-down list box in the *Size* field.

5. Click on the up or down arrow in the *Accuracy* field to determine the database accuracy of the drawing.

6. Set the drawing’s physical boundaries using the *Drawing Extents* fields.

7. Click *OK*. 
To change the location of the drawing’s origin:
➤ Enter the new location in the \( X \) and \( Y \) fields of the Move Origin section.

**Specifying the Text Size**

The Text Setup dialog box lets you specify how text is to appear in your design.

To specify text parameters:
➤ Run the `define text` command.
   The Text Setup dialog box appears.

**Figure 2-2 Text Setup Dialog Box**

This dialog box lets you set the width of each character in a particular block, the height of each character, the distance between the top and bottom of characters in two lines, the amount of space between characters (kerning), and the width of the line that is used to photoplot the characters.

For more information on formatting text, see “Working with Text” on page 132.
Specifying Grids

The Grid Display dialog box lets you set the x and y values for both etch/conductor and non-etch/conductor grids in a design. It also lets you customize the grid for each etch/conductor layer in a design.

To specify grid parameters

➤ Run the `define grid` command.

The Grids Display dialog box appears.

All drawings, except Autoplacement, interactive routing, and Autorouting use non-etch/conductor grid. All non-etch/conductor layers use the same, single-increment grid with the grid points spaced evenly across the design.

Etch/Conductor grids are dedicated routing grids for both interactive and autorouting. You can use a separate x, y grid for each etch/conductor layer in a design. In addition, you can set a
single increment value for each etch/conductor grid, or you can set different values for nonetch/conductor grids and etch/conductor grids.

You can enter values into the Grids Display dialog box to reset the point of origin for x and y, as well as the spacing between the grid points for x and y. The default point of origin for all layers is x=0, y=0. The default increment setting for nonetch/conductor layers is x=100, y=100. For etch/conductor layers, the default setting is x=25, y=25.

Using the Drawing Options Dialog Box

The Drawing Options dialog box lets you control the way Allegro/APD displays many elements of your design.

To display the Drawing Options dialog box:

➢ Run the status command.

The Drawing Options dialog box appears. Select the appropriate tab to re-focus the section of the dialog box you need to work in. Each of the tabs are described in the next sections.

From the Drawing Options dialog box you can:

■ Control the Allegro/APD display
■ Set the Line Lock Parameters
■ Run online design rule checks.
■ Specify the angle that Allegro/APD uses when you place symbols. It also lets you mirror any symbols that you add to a drawing.
DRC (Design Rule Checking) Tab

Figure 2-4  DRC Tab of the Drawing Options Dialog Box

These are the settings, fields, and buttons on this tab:

*Status* indicates whether DRC markers are up-to-date. Options are *UP TO DATE* and *OUT OF DATE BATCH*.

*On-Line* specifies whether you will run DRC online (*On*) or in batch mode (*Off*). The default is *On*.

*Marker size* determines the size, in user-defined units, of the DRC markers that are displayed in a design. The default is *25*.

*Symbol height* indicates the default height of a symbol on the design if you did not specify the height when you created the symbol. Use this value to check package-to-package/part-to-part and package-to-keepout/part-to-keepout DRC. Note that the default symbol height of *150* is in database units, which should be used in all data that pertains to symbol height.

*Update DRC* displays the total number of errors when you click this button. It is only enabled when you are running DRC online.
Display Tab

Figure 2-5 Display Tab of the Drawing Options Dialog Box

These are the settings and fields on this tab:

*Connect point size* specifies the size of a connect point in user units. The default is 10.

*Max rband count* lets you specify the maximum number of rubberband lines that can be displayed in a window. The default is 500.

*Ratsnest geometry* displays a pop-up menu that lets you choose the shape of the ratsnest lines. Options are *Jogged* and *Straight*. The default is *Jogged*.

*Ratsnest points* displays a pop-up menu that lets you choose the closest distance on a line (*Closest endpoint*) or between two pins (*Pin to pin*). The default is *Closest endpoint*.

*Bus rats* displays the middle portion of ratsnest lines with the same BUS_NAME property so that they appear to be merged together into a thick line. By default, this is not checked.

*Grid* lets you display the grid. By default, this is checked.

*Thermal pads* lets you display thermal pads when you have a negative plane. By default, this is not checked.
Filled pads (Windows version) or Filled pads and cline endcaps (UNIX version) lets you control how the design is displayed on the screen. Pins are filled and the line vertices are rounded to more closely approximate artwork. By default, this is not checked.

Display drill holes displays all drill holes in your design, whatever the source. By default, this is not checked.

Text Tab

Figure 2-6 Text Tab of the Drawing Options Dialog Box

These are the fields on this tab:

Justification indicates the anchor point within text that you add to the layout. The anchor location determines how the text appears in the text block. The options are Left, Right, and Center. The default is Left.

Parameter block lets you choose a text parameter block number. The size and spacing of any text that you add to a drawing are controlled by the current block number in this field. You can change the values in each parameter block with the Add – Text menu option. The default is 1.
Marker size determines the size, in user units, of the displayed text markers. These markers indicate locations for text that will be entered later. The default is 50.

Line Lock Tab

Figure 2-7  Line Lock Tab of the Drawing Options Dialog Box

These are the settings and fields on this tab:

Lock direction displays a pop-up menu that lets you specify the direction of lines that you add to a drawing. The options are Off, 45, and 90. The default is 45.

Lock mode specifies the type of segments to use when adding lines or connect lines. The options are Line and Arc. The default is Line.

Minimum radius determines the minimum radius allowed for an arc. The default is 0.

Fixed 45 length specifies the length, in user units, of 45-degree segments. By default, this is unchecked. If checked, the default is 25.

Fixed radius specifies the radius, in user-defined units, of arcs. By default, this is unchecked. If checked, the default is 25.
Tangent specifies whether tangent lines are locked. If checked, arcs construct tangent to lines. By default, this is checked.

**Symbol Tab**

**Figure 2-8 Symbol Tab of the Drawing Options Dialog Box**

These are the setting and field on this tab:

*Angle* specifies the default angle that is used to place symbols. You can enter an angle with up to three decimal places, or you can display a pop-up menu and choose one of the available angles. The default is 0.

*Mirror* lets you mirror symbols that you add to a drawing. By default, this is not checked.

**Drawing size Button**

This button displays the Drawing Parameters dialog box, described in “Specifying Drawing Parameters” on page 98.
About Classes and Subclasses

In Allegro/APD, categories of drawing elements are called classes. Classes represent all types of visible items in the design. A few examples of classes are

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETCH/CONDUCTOR</td>
<td>Represents pieces of copper forming electrical connections.</td>
</tr>
<tr>
<td>PINS</td>
<td>Represents defined pads and holes.</td>
</tr>
<tr>
<td>BOARD/SUBSTRATE GEOMETRY</td>
<td>Represents the physical outline of the design and other geometry related to the PCB.</td>
</tr>
<tr>
<td>PACKAGE/PART GEOMETRY</td>
<td>Represents the physical components of the design.</td>
</tr>
</tbody>
</table>

The parts of the drawing in each class are called subclasses. Each class can contain many subclasses, including some that you define.

Classes and subclasses are used to identify how every element is to be used in a design. For example, the command Add–Line used when Board/Substrate Geometry is the active class adds a simple geometric graphic element to a design. The same command used when Etch/Conductor is the active class adds a connecting line of etch/conductor to the design because the command correlates the function with the class of element.

Subclasses allow a further degree of classification that allows Allegro/APD to treat data more specifically. For example, Etch/Conductor has two pre-defined subclasses associated with it: Top/Surface and Bottom/Base (thus eliminating the necessity of referring to element types by layer number). You also have the option of defining subclasses. (See “Creating User-Defined Subclasses” on page 117.)

Table 2-1 on page 110 lists groups of Allegro/APD classes and their pre-defined subclasses. Note that the Allegro/APD product you are running may not include all the classes/subclasses listed here.

To display groups of classes and subclasses

1. Run the color command.

   The Color and Visibility dialog box appears.
2. Open the drop-down menu by clicking the arrow button to the right of the *Group* field.

The menu displays the five groups that, together, contain all the classes and subclasses in Allegro/APD. They are

- Geometry
- Manufacturing
- Stack-Up
- Components
- Areas

**Note:** The menu also contains the Display item, which allows you to set non-class design objects and to use Shadow Mode for distinguishing key objects in your design.

- See “Assigning Colors to Grids, Ratsnest Lines, Highlighting, and Background” on page 128 for details on setting non-class design objects.
- See Controlling the Visibility of Individual Objects on page 129 for details on using Shadow Mode.

Figure 2-10 shows the classes and subclasses in the Manufacturing group.
As you select different groups, the associated classes and subclasses appear in the dialog box as shown in Table 2-1 on page 110. See “Controlling Visibility” on page 113 for details.

**Note:** Subclasses will vary depending on layers added to—or deleted from—specific designs.

### Table 2-1 Allegro/APD Classes and Subclasses

<table>
<thead>
<tr>
<th>Group</th>
<th>Class</th>
<th>Subclasses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry</td>
<td>Board/ Substrate Geometry</td>
<td>OUTLINE PLATING_BAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASSEMBLY_NOTES TOOLING_CORNERS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIMENSION PLACE_GRID_TOP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLACE_GRID_BOTTOM TOP_ROOM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BOTTOM_ROOM BOTH_ROOMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SWITCH_AREA_TOP SWITCH_AREA_BOTTOM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SILKSCREEN_TOP SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td>Group</td>
<td>Class</td>
<td>Subclasses</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Package/Part</td>
<td>ASSEMBLY_DETAIL</td>
<td>SOLDERMASK_TOP</td>
</tr>
<tr>
<td>Geometry</td>
<td></td>
<td>OFF_GRID-AREA</td>
</tr>
<tr>
<td></td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>PLACE_BOUND_TOP</td>
<td>PLACE_BOUND_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>PIN_NUMBER</td>
<td>PAD_STACK_NAME</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>BODY_CENTER</td>
<td>SOLDERMASK_TOP</td>
</tr>
<tr>
<td></td>
<td>SOLDERMASK_BOTTOM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DISPLAY_BOTTOM</td>
<td>DISPLAY_TOP</td>
</tr>
<tr>
<td></td>
<td>DISPLA Y_TOP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DISPLA Y_BOTTOM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MFG_GLUE</td>
<td></td>
</tr>
<tr>
<td>Manufacturing</td>
<td>SHAPE PROBLEMS</td>
<td>NO_GLOSS_ALL</td>
</tr>
<tr>
<td></td>
<td>PHOTOPOINT_OUTLINE</td>
<td>NO_GLOSS_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>NO_GLOSS_TOP</td>
<td>NCDRILL_LEGEND</td>
</tr>
<tr>
<td></td>
<td>NO_GLOSS_INTERNAL</td>
<td>PROBE_TOP</td>
</tr>
<tr>
<td></td>
<td>NCDRILL_FIGURE</td>
<td>AUTOSILK_TOP</td>
</tr>
<tr>
<td></td>
<td>PROBE_BOTTOM</td>
<td>NO_PROBE_TOP</td>
</tr>
<tr>
<td></td>
<td>AUTOSILK_BOTTOM</td>
<td>SHAPE PROBLEMS</td>
</tr>
<tr>
<td></td>
<td>NO_PROBE_BOTTOM</td>
<td></td>
</tr>
<tr>
<td>Drawing</td>
<td>OUTLINE</td>
<td>TITLE_BLOCK</td>
</tr>
<tr>
<td>Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLE_DATA</td>
<td>REVISION_BLOCK</td>
</tr>
<tr>
<td></td>
<td>REVISION_DATA</td>
<td></td>
</tr>
<tr>
<td>Stack-Up</td>
<td>DRC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOP/SURFACE</td>
<td>BOTTOM/BASE</td>
</tr>
<tr>
<td></td>
<td>THROUGH ALL</td>
<td>PACKAGE_TOP</td>
</tr>
<tr>
<td></td>
<td>PACKAGE_BOTTOM</td>
<td>I</td>
</tr>
<tr>
<td>Etch/Conductor</td>
<td>TOP</td>
<td>BOTTOM</td>
</tr>
</tbody>
</table>
### Table 2-1 Allegro/APD Classes and Subclasses, continued

<table>
<thead>
<tr>
<th>Group</th>
<th>Class</th>
<th>Subclasses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-Etch/Conductor</td>
<td>TOP</td>
<td>BOTTOM</td>
</tr>
<tr>
<td></td>
<td>INTERNAL LAYERS</td>
<td>THROUGH ALL</td>
</tr>
<tr>
<td>Pin</td>
<td>TOP</td>
<td>BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SOLDERMASK_TOP</td>
<td>SOLDERMASK_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>PASTEMASK_TOP</td>
<td>PASTEMASK_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>FILMMASKTOP</td>
<td>FILMMASKBOTTOM</td>
</tr>
<tr>
<td>Via</td>
<td>TOP</td>
<td>BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SOLDERMASK_TOP</td>
<td>SOLDERMASK_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>PASTEMASK_TOP</td>
<td>PASTEMASK_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>FILMMASKTOP</td>
<td>FILMMASKBOTTOM</td>
</tr>
<tr>
<td>Components Refdes</td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>DISPLAY_TOP</td>
<td>DISPLAY_BOTTOM</td>
</tr>
<tr>
<td>Comp Value</td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>DISPLAY_TOP</td>
<td>DISPLAY_BOTTOM</td>
</tr>
<tr>
<td>Dev Type</td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>DISPLAY_TOP</td>
<td>DISPLAY_BOTTOM</td>
</tr>
<tr>
<td>Toler</td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>DISPLAY_TOP</td>
<td>DISPLAY_BOTTOM</td>
</tr>
<tr>
<td>User Part</td>
<td>ASSEMBLY_TOP</td>
<td>ASSEMBLY_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>SILKSCREEN_TOP</td>
<td>SILKSCREEN_BOTTOM</td>
</tr>
<tr>
<td></td>
<td>DISPLAY_TOP</td>
<td>DISPLAY_BOTTOM</td>
</tr>
<tr>
<td>Areas</td>
<td>Route Keepin</td>
<td>THROUGH ALL</td>
</tr>
</tbody>
</table>
Controlling Visibility

To control class and sub-class visibility:

1. Enter the `color` command on the command line to display the Color and Visibility dialog box.

2. Open the drop-down menu in the `Group` field.

3. Select a group.

   The classes and subclasses associated with that group appear in the dialog box, as shown in Figure 2-11.
Figure 2-11 Classes and Subclasses Associated with a Group

4. To control the visibility of individual subclasses, click the associated checkbox. (In the illustration, subclass NCDRILL_LEGEND is the only visible element.)

   —or—

   To control class visibility, click the checkbox under the appropriate class name. (In the illustration, MANUFACTURING and DRAWING FORMAT are unselected.)

   All the subclasses in that class become visible when you enable the checkbox.

   —or—

   To turn on/off all classes and subclasses, open the drop-down menu in the global visibility selector field, and choose All Visible or All Invisible.

   All classes in all groups become visible or invisible, based on your selection.

   You can also use the following arguments with the color command to control layer visibility.
Table 2-2  Color command arguments (for customer extension scripts)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
<th>Syntax example</th>
</tr>
</thead>
<tbody>
<tr>
<td>globvis [off</td>
<td>Disables/enables visibility for all layers.</td>
<td>color -globvis “off”</td>
</tr>
<tr>
<td>or on]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vis [&lt;class</td>
<td>Enables visibility for an entire class/subclass.</td>
<td>color -vis “BOARD GEOMETRY/</td>
</tr>
<tr>
<td>name&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASSEMBLY_NOTES”</td>
</tr>
<tr>
<td>invis [&lt;class</td>
<td>Disables layer visibility for the specified class/</td>
<td>color -invis “ETCH/TOP”</td>
</tr>
<tr>
<td>name&gt;</td>
<td>subclass name&gt;]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>toggle &lt;class/</td>
<td>Reverses the current layer visibility for the</td>
<td>color -toggle “BOARD/</td>
</tr>
<tr>
<td>subclass name&gt;</td>
<td>specified class/subclass.</td>
<td>GEOMETRY/OUTLINE”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using Color Visibility Views

A color visibility view saves the visibility of the Allegro/APD classes and subclasses as a collection of layer visibility settings that you can apply to subsequent Allegro/APD designs using the Views field on the Visibility window of the control panel. You can save your settings in a file that is stored in the current directory with a .color extension.

A color view also displays film record visibility settings stored in the current design.

Color views (.color files) display in the Views field as File: <name>. Film record names display there as Film: <name>, unless you suppress the film record names from the list of color views in the Visibility window of the control panel by selecting the color_nofilmrecord environment variable in the Display section of the User Preferences Editor dialog box. (See “The User Preferences Editor” on page 173.) Restart Allegro/APD for changes to the variable’s value to take effect.

Creating or changing a color visibility view

1. Choose View – Color View Save or colorview create to display the Color Views dialog box.
2. Enter the name of the color view in the Save View field. Allegro/APD automatically appends the .color file extension in the current working directory.

3. Select the View Replacement Method:

- **Complete** saves the current layer visibility settings to a color view file. When you load the file later, then it completely replaces the design's visibility settings, analogous to how the film option to color views works.

- **Partial** allows a color view file to be created such that when loaded, it partially replaces the design's visibility settings. Allegro/APD prompts you to use the Color View dialog box to create the partial changes you desire.

- **Partial with toggle** functions the same as the Partial view replacement method; however, settings that you change toggle when you load the color view file. Toggle means that if the visibility for a layer is on, it is turned off; if off, it is turned on.

    Allegro/APD displays the following message in the Color Views dialog box:

    Change layer visibility

4. Change the desired visibility settings by using the Color/Visibility dialog box or the visibility settings in the control panel.

5. Click Close.

    Allegro/APD saves the settings for the color view in a file with the name you specified, in the Save View field; otherwise closes the dialog box without creating a color view file in the current working directory.
Loading a color visibility view

Use the `colorview load` command to load a specified color view. Colorview load treats a `.color` file as a script to be replayed, allowing you to assign function keys to your favorite views.

The syntax is `colorview load <filename> .color`.

1. Type `colorview load` in the console window. The Colorview file browser appears.
2. Enter a filename to load a specified color view or select one from the file browser.
3. Click `Save` to load the color view.

Deleting a color visibility view

To delete a color view:

1. Locate the directory in which the color views reside.
2. Delete the color view files you no longer need.

Applying the previous color visibility view

You can apply the previous color view or toggle back and forth between two color views.

1. Choose `View – Color View Restore Last` to apply the color view that preceded the current color view.
2. Re-select `View – Color View Restore Last` to toggle back and forth between two color views.

Creating User-Defined Subclasses

Run the `subclass` command to create your own subclasses. You use the same procedure for creating the following nonetch/conductor subclasses:

- Board/Substrate Geometry
- Component Value
- Device Type
- Drawing Format
You use a separate procedure for creating an Etch/Conductor subclass (see “Creating Etch/Conductor Subclasses” on page 119). Etch/Conductor subclasses identify the layers or cross section of the printed circuit board/substrate.

**Creating Non-Etch/Conductor Subclasses**

To create a user-defined, non-etch/conductor subclass

1. Run the *define subclass* command to display the Define Subclass dialog box.

**Figure 2-13 Define Subclass Dialog Box**

2. Click the desired class button to display the Define Non-Etch/Conductor Subclass dialog box, listing the current subclass elements in the class.
For example, the MANUFACTURING class displays the list of its subclass element.

In the Define Non-Etch/Conductor Subclass dialog box, any existing user-defined subclasses have a button to the left of their name.

3. For non-etch/conductor subclasses, such as BOARD/SUBSTRATE GEOMETRY, type a name in the New Subclass field, and press Enter.

4. Click OK in the Class dialog box to dismiss both dialog boxes.

Creating Etch/Conductor Subclasses

This section explains how to create a user-defined, etch/conductor subclass.
To define etch/conductor subclasses:

1. Run the `define subclass` command to display the Define Subclass dialog box.

2. Click `ETCH/CONDUCTOR` to display the Layout Cross Section dialog box.

**Figure 2-14 Layout Cross Section Dialog Box**

3. Click the *Edit* button opposite the layer in which you want to add a subclass. Select *Insert* from the pop-up menu.
Allegro/APD Design Guide: Getting Started
Generic Procedures

Allegro/APD adds an unnamed layer above the subclass associated with the Insert button that you clicked.

4. Enter a name for the layer.
5. Click to select the appropriate film type: Positive or Negative.
6. Click to select the appropriate layer type: Conductor or Plane.
7. Continue to add layers, or click OK to save your edits and dismiss the define etch/conductor subclass dialog box.

Note: Adding an ETCH/CONDUCTOR subclass (a layer) also adds the same subclass to the Pins, Vias, and DRC classes.

Removing a Subclass

This section explains how to remove a user-defined subclass.

To remove an Etch/Conductor or non-Etch/Conductor subclass
1. Run the `define subclass` command.

2. Click the desired Class button.

3. Click the `Edit` button to the left of the user-defined subclass that you want to remove. Choose `Delete` from the pop-up menu.

4. Click `OK`.

**Note:** Removing an Etch/Conductor subclass (a layer) also removes the same subclass from the Pins, Vias, and DRC classes.

---

## Customizing Colors Used in Your Design

### The Color Palette

You can define colors to be used in your design through the Palette section of the Color and Visibility dialog box. Allegro/APD lets you assign the colors defined in the color palette to specified object types in your design (see “Assigning Colors to Design Elements” on page 125). You can also modify the elements of the colors themselves, using the Color Editor dialog box.

### Color Hierarchy

Allegro/APD provides a *global* palette that is used initially for all drawings. The global palette can be applied to designs or saved with different values as a local palette, but it cannot be modified or deleted. If you have not previously created a local palette for your current drawing, the colors that you see when you open Color and Visibility represent the global palette. If you have previously created a local palette, you can switch to the global palette by choosing the `Read Global` option in the dialog box’s drop-down menu.

You can modify the global palette to create a custom *local* palette. Once created, this palette can be read into any new or existing designs, allowing you to consistently assign colors of your choice to objects.

Changes that you make to a palette can be used for a specific design only, through the creation of a *database* palette.

For details on saving color changes to local and database palettes, see “Saving the Modified Color Palette” on page 124.
Modifying Color Elements

1. Run the color command to display the Color and Visibility dialog box.

2. In the Palette section of the dialog box, click the color box for the color you want to change.

3. Click Modify.

   The Color Editor dialog box appears.

Figure 2-15  Color Editor Dialog Box

4. Adjust the color settings by doing one of the following:

   - Enter numerical values in the fields.
   - or -
     a. Hold down the left mouse button while you move the cursor around the color selection chart.
     b. Release the mouse button when the color in the viewer box displays the color you want.
     c. Adjust the luminosity setting in the field or on the slide bar.

   The customized color appears in the color viewer box.

5. Click OK.
The Color Editor closes and the color box in the Color and Visibility dialog box changes to the customized color.

–or–

Click *Cancel*.

The dialog box closes without saving your changes.

6. Do one of the following:

- Click *Apply* in the Color and Visibility dialog box.
  
  The design is updated with your color changes.

- or –

- Click *Reset* to return the palette to the previously committed values.

### Saving the Modified Color Palette

After you have completed making your color changes as described in “Modifying Color Elements” on page 123 you can save the modifications as a *local* palette or as a *database* palette.

**Note:** A *local* palette is not specific to only the current design; it can be read into any other board/substrate design. A design-specific color scheme is a *database* palette.

To save a local palette

1. Open the palette selector drop-down menu after making your color changes.

2. Highlight *Write Local*.

   The drop-down menu closes and your color selections are saved as your local palette.

To save a database palette:

1. Click *Apply* after making your color changes.

   The design is updated with your color changes. The *Apply* button is grayed-out until you make another change.

2. Run the *save* or *save_as* commands.

   The design is saved with the current (database) palette.
Note: The database palette remains current until you apply a different color palette to your design. For example, if you apply the global palette to your design, the database palette then reflects that color scheme.

Assigning Colors to Design Elements

The Color and Visibility dialog box is how you identify and control the display of Allegro/APD design elements. This section explains how to:

- Assign colors to subclasses
- Control the colors used for grids, ratsnest lines, and highlighting
- Prioritize subclasses to be displayed above other subclasses in the current drawing for easy reference
- Control whether individual classes and subclasses appear in the current drawing

Assigning Colors to Subclasses

To assign colors to subclasses in the current drawing

1. Run the `color` command to display the Color and Visibility dialog box.
2. In the Palette section of the dialog box, click one of the basic color choices.

3. Click the color box associated with the subclass whose color you want changed.

   The color box for the subclass changes to the color you selected from the basic color palette.

4. Click OK.

   Allegro/APD updates the active drawing.

   The Options tab displays the color assigned to a subclass in a box next to the name of the subclass.

### Assigning a Display Priority to Colors

After you assign colors to subclasses (described in the previous section), you can control which colors appear on top of others in the active drawing for easier recognition. For example, during interactive routing you can display etch/conductor on the active layer on top of all other colors for easier editing. To do this, set each color's priority. The color with the highest priority is displayed on top of other colors and displays a complete element. You can also prevent colors from being displayed.

![Highest priority color displays over color with lesser priority](image)

To define color priority

1. Run the `color priority` command to display the Priority color panel.
Figure 2-16 Color Priority Dialog Box

The color with the highest priority is at the top of the color panel. All elements of that color appear on top of other elements in the drawing.

2. Click the color box of the color you want to move in the priority list.

3. Determine where in the hierarchy you want to move the selected color.
4. Click the color box of the color that occupies the position to which you want to move the selected color.

![Diagram showing color selection and visibility]

The first color you selected (green in these examples) moves to the new location; the previous color for that position moves down one level.

5. Click OK.

Allegro/APD displays the changes.

You can prevent a class from being displayed by clicking on the Vis button that is associated with the class.

![Diagram showing visibility control]

**Assigning Colors to Grids, Ratsnest Lines, Highlighting, and Background**

To assign colors to grids, ratsnest lines, and highlighting:

1. Run the *color* command to display the Color and Visibility dialog box.
2. In the Palette section of the dialog box, click one of the basic color choices.

3. Click the color box associated with the element whose color you want changed.

   The color box for the element changes to the color you selected from the basic color palette.

4. Click OK.

Controlling the Visibility of Individual Objects

You may occasionally want to highlight specific objects in your design without affecting the visibility settings of that object's entire subclass (as described in Assigning Colors to Subclasses on page 125). To control the visibility of individual design objects, use the Shadow Mode feature in the Display area of the Color and Visibility dialog box (shown in Figure 2-17). Shadow Mode is used in conjunction with the hilight and dehilight commands, as well as various interactive commands.

When you turn on Shadow Mode, the following conditions occur:

- The Brightness setting slide bar moves to its last applied percentage of brightness. The initial default percentage setting is 50%.
The colors in the Palette section dim to the selected percentage of brightness in the slide bar. This allows you to “preview” how the colors in your design will be displayed if you click Apply or OK.

A “Dim active layer” check box lets you dim the active layer of your design. Dimming the active layer if it contains a large number of objects displayed normally (non-highlighted) can increase the effectiveness of Shadow Mode. You can dim the active layer by way of the check box in the Color and Visibility dialog box or in the Options tab when shadow mode is turned on.

The design elements of the current active drawing dim to the percentage of brightness set in the slide bar.

Display Modes

With Shadow Mode active, objects in your design can be displayed in the following ways:

- **Normal.** Objects on the active layer of your design remain unaffected by Shadow Mode unless you select the “Dim active layer” control in the Options tab.

- **Highlighted,** either permanently by way of the **hilight** command, or temporarily when you run an interactive command. In this state, objects are not affected by Shadow Mode. Objects affected or added by a current interactive command are temporarily highlighted while the command is active. For example, if you run **add connect** with Shadow Mode on, the objects highlighted would include:
  - Interconnecting pins
  - Existing etch being tied into
  - Connect lines, vias, and DRCs

  When you complete the command, the added/affected objects are dimmed.

- **Dim.** The objects are unaffected by the conditions described above. The degree of dimming depends on the percentage of brightness set in the Color and Visibility dialog box.

Shadow Mode Keyboard Commands

The syntax for setting Shadow Mode at the command prompt is

\[
\text{shadow [on] [off] <+/-n>}
\]

Examples:
### Generic Procedures

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>shadow on</code></td>
<td>Turns Shadow Mode on</td>
</tr>
<tr>
<td><code>shadow 50</code></td>
<td>Turns Shadow Mode on and sets the brightness factor to 50%</td>
</tr>
<tr>
<td><code>shadow +10</code></td>
<td>Increases the current brightness factor by 10%</td>
</tr>
<tr>
<td><code>shadow off</code></td>
<td>Turns Shadow Mode off</td>
</tr>
<tr>
<td><code>shadow toggle</code></td>
<td>Toggles shadow on and off</td>
</tr>
</tbody>
</table>

You can set global Shadow Mode parameters through the use of keyboard commands entered at the Allegro/APD command prompt, allowing you to assign function keys or toolbars to the dimming controls. For example, to toggle shadow mode on and off using the F3 key, you would enter the following at the Allegro/APD command prompt:

```plaintext
alias SF3 shadow toggle
```

### Plotting a Design

The method by which Allegro/APD plots your design to a plotter or printer differs according to which platform you are on (UNIX or Windows) and which tools you run.

- **Allegro/APD on Windows** uses Windows Print Manager for controlling printing operations.
  
  For information on installing a driver that supports your printer or plotter, consult the Microsoft Windows NT documentation.

- **Allegro/APD on UNIX** uses the `allegro_plot` program which is based upon the Cadence corporate plotting package, `plotServ`.
  
  Windows **does not support** `allegro_plot`. If you create an Intermediate Plot (IPF) file, which is a representation of an Allegro/APD database, you can copy it to a UNIX workstation that runs `allegro_plot` or to third-party plotting software.

- **On either platform**, Allegro/APD lets you import IPF files or create them for export using the `load plot` and `create plot` commands.

Full details of these methods appear in Chapter 4, “Plotting,” of the *Allegro/APD Design Guide: Manufacturing Processes.*
Working with Text

You can add, edit, and delete text in a drawing. Text can provide additional information about the design or it can be included as labels that are attached to graphic elements. This section describes:

- Defining text characteristics
- Adding text to drawings
- Editing text in drawings

Defining Text Characteristics

Allegro/APD lets you define the size and spacing characteristics of text that appears in the drawing. You can assign text parameters to up to 16 text blocks, which makes it easy to specify the appearance of text that you subsequently add to your design. You specify the text parameters as you add the text or label.

To define text parameters:

1. Run the define text command.
   
   The Text Setup dialog box appears.
2. Verify the default entries for each text block.

3. Update the entries to meet your site requirements.

4. Apply the text parameters.

Adding Text to Drawings

You can use text in Allegro/APD drawings as simple notes and as logical labels of elements. Labels include reference designators, device type, value, tolerance, and user part number.

Note: The add text command does not let you enter an exclamation point (!) in an Allegro/APD database, since a_extract uses that character as a field delimiter. Be aware of the possible consequences of this condition if you read into your database a file that contains an exclamation point.

Some label commands require not only the data for text location and content, but also the identity of the element to be labeled, such as labeling placement room areas in your layout.

Use the add text command to annotate design elements. Use the Layout–Labels menu selection (in Symbol mode) to add text labels (Ref Des, Device, Value, Tolerance, User Part Number) to symbols.
To add text to a design

1. Run the `add text` command.

2. Complete the Options tab.

3. Position the cursor and click at the location for the text.

4. Enter the text in the design window.
   - Limit text lines to 80 characters, including spaces.
     To correct errors, press `Delete` or `Backspace`.
   - Press `Enter` to start a new line of text with line spacing set by the parameter block.

5. When you have entered all text required for the current point, click right to display the pop-up menu, and choose `Done`.

**Note:** To import a text file into the design, run `add text`, click right, and choose `Read from file`.

### Editing Existing Text or Labels

You can edit text in a drawing. If the text is a reference designator label, editing the text changes the reference designator in the database. This can have other side effects, as explained in this section. You cannot edit a device type label in a drawing, because it would redefine the logical structure of the component.

In general, when you edit text, Allegro/APD

- Highlights the text and displays the text cursor on the first character location of the text string
- Overwrites the existing text
- Lets you select another text string for editing

**Note:** You cannot enter an exclamation point (!) in an Allegro/APD database, since `a_extract` uses that character as a field delimiter. If an exclamation point is part of existing text that you are editing from an older version of Allegro/APD, be aware that `edit text` will not be able to replace that character if removed.

To edit text

1. Run the `tex edit` command.

2. Position the cursor over the text to be edited and click.
3. Enter the new text for that text string and press Enter.

4. Pick another string to edit.

–or–

Click right and choose Done from the pop-up menu.

**Finding Design Elements**

The *Find Filter* is located in the upper portion of the Control panel, and allows you to find design elements by directly selecting objects in the design or by using the Find By Name/Property dialog box.

**Figure 2-19 Find Filter**

The elements in the Find filter that are available for the active command are in bold text and have their check boxes selected. The elements available for selecting depends on the command that is active.

You can select or deselect any elements by clicking the check box on or off, or you can select/deselect all the elements with the All On/All Off buttons.

If you try to find an invalid element type, Allegro/APD displays the following message:
<element types> are not selectable at this time.
Name Function Failed.

Determining the Element Selection Hierarchy

The Allegro/APD database maintains a hierarchy of elements to simplify the selection process. When you select an element, Allegro/APD selects the highest level element that is associated with that selection. If you disable the higher level elements, such as connect lines or nets, Allegro/APD selects lower level elements, such as line segments.

For example, a pin can be part of a function, net, symbol, component, or group. When determining the proper element to highlight, Allegro/APD uses the following hierarchy:

- Groups
- Components
- Symbols
- Functions
- Nets
- Pins

There are two primary methods you can use to locate design elements in Allegro/APD: the show element command and the show property command. Both allow you to find elements by name or property, but do so in different ways.

Using Show Element

You can use the show element command with the Find filter to locate and identify design elements by property or by name. You can further refine a find operation by entering a value for the element you want to find. You perform these operation using the Find By Name/Property dialog box.

Finding Elements by Name or Property

With the show element command active, click More in the Find filter to display the Find by Name/Property dialog box which lists all available object types for selected elements.
Figure 2-20  Find By Name/Property Dialog Box

Depending on the object type you select, the Find By Name/Property dialog box allows you to identify an element that you want to find by listing those elements by object type. You can then select individual items, and by clicking *Apply*:

- Display the Show Element dialog box on the element(s)
- Display the location of the element(s) in the World View area of the UI
- Highlight the selected element(s) in the design area of the UI.
If you know the name of the element that you want to locate (such as U13), you can find it by entering its designation in the Name Filter field and selecting the appropriate object type from the menu.
Using Show Property

Unlike show element, the show property command is not used with the Find filter, though it can help you locate elements in a design. When you run the command, the Show Property dialog box appears.

**Figure 2-23 : Show Property dialog box with Value and Definition views**

By selecting a property (sorted by property or element) and pressing the appropriate Show button, you can display a definition of the property or its value relative to the object to which it is attached. The Name and Value fields let you qualify an element further. When you enter a name or value, Allegro/APD searches only for those elements that match both the Name and Value that you entered, and that are valid for the active command.
Using Find by Property from the Console Window Prompt

You can also use the console window prompt to find elements by property. The Find Filter must be activated with elements that allow property assignments.

To use *Find by Property* from the console window prompt

➤ At the console window prompt, type

```
find property name <property value>
```

All elements are selected for the active command that have the defined property name and value.

You can use wildcard characters for both the property name and value. The property name field is not case sensitive.

Finding by Name from the Console Window Prompt

You can also use the prompt in the console window to find elements by name. The Find Filter must be activated with elements that appear in the design.

When you use the command line, you can enter character strings, including the element type plus a name or list file, and wildcard characters. Character strings are not case sensitive.

The following table lists keywords, the way in which Allegro/APD matches that keyword, and an example of each keyword type.

**Find by Name Commands**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Allegro/APD Match</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net</td>
<td>Net that matches name</td>
<td>data1</td>
</tr>
<tr>
<td></td>
<td>Net with BUS_NAME property matching netname</td>
<td>data&lt;1&gt;</td>
</tr>
<tr>
<td>Comp</td>
<td>Component instance that matches refdes</td>
<td>U34</td>
</tr>
<tr>
<td></td>
<td>Symbol pin that matches refdes</td>
<td>U34.1</td>
</tr>
<tr>
<td>Symbol</td>
<td>Symbol instance that matches refdes</td>
<td>U34</td>
</tr>
<tr>
<td></td>
<td>Symbol pin that matches refdes.pin</td>
<td>U34.1</td>
</tr>
<tr>
<td>Func</td>
<td>Function instance that matches funcdes</td>
<td>TF–7</td>
</tr>
</tbody>
</table>
Find by Name Commands, continued

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Allegro/APD Match</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devtype</td>
<td>Component or symbol instances that match device type—components are selected if</td>
<td>74LS74</td>
</tr>
<tr>
<td></td>
<td>the command allows; otherwise, symbols are selected</td>
<td></td>
</tr>
<tr>
<td>Symtype</td>
<td>Symbol instances that match symbol name</td>
<td>dip14</td>
</tr>
</tbody>
</table>

You must enter the keyword exactly as it appears in the drop-down list in the Find Filter. In other words, type `comp` or `symtype` instead of component or symbol. If you enter multiple names, put a space between the element names. If the element name contains a space, put quotation marks around it.

For example, the following command selects the nets MEM17, DATA4, and CLOCK for processing.

```
net mem17 data4 clock
```

Likewise, when you enter multiple lists, you must put a space between each list file. For example, the following command selects all components in the files `U.lst` and `R.lst` for processing.

```
list comp U(.lst) R(.lst)
```

Using Wild Cards

Allegro/APD lets you use wild card characters when you try to find elements by name or by list. The following table lists the valid wild card characters.

### Valid Wild Card Characters

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Match Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Any number of characters</td>
<td>name* = name1, name12, name ANY</td>
</tr>
<tr>
<td>?</td>
<td>Any single character</td>
<td>name? = name1, nameA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>name? ≠ name12, name ANY</td>
</tr>
</tbody>
</table>
Highlighting Selected Elements

When you select elements by group or window, Allegro/APD lets you specify the temporary highlight color. The following table summarizes the way in which Allegro/APD highlights element types:

**How Allegro/APD Highlights Element Types**

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net</td>
<td>All vias, connect lines, shapes, frects, ratsnests, and pins on a selected net</td>
</tr>
<tr>
<td>Component Instance</td>
<td>Symbol instance linked to selected component instance. Only placed components can be highlighted. Symbols highlighted by symbol and component instance appear the same on the display.</td>
</tr>
<tr>
<td>Ratsnest</td>
<td>Selected ratsnest line</td>
</tr>
<tr>
<td>Function Instance</td>
<td>All pins for the selected function instance</td>
</tr>
</tbody>
</table>

Finding Elements by Using the pick Commands

In addition to using the mouse to highlight elements in a drawing, you can use *pick* or *ipick* commands to enter x, y coordinates for the elements.

1. In the console window, type one of the following commands at the command prompt:
   - `pick, pickx, picky`
   - `ipick, ipickx, ipicky`

   A dialog box similar to the following appears:

   ![Allegro Expert](image)

   **Note:** You must be in a command mode—for instance, *add fshape*—for the dialog box to appear.

2. Enter the x coordinates or incremental x coordinates (*ipick* commands).
3. Click *OK*. 
4. Enter the same information for y coordinates.

   If Allegro/APD finds the appropriate element at those coordinates or at the incremental
distance from those coordinates, it highlights the element.

**Using Temporary Group Mode**

When you run an interactive command in temporary group mode, you can identify elements
by name, list, pick, window, or any combination of these until you click right and choose
*Complete* from the pop-up menu.

To deselect elements that you select in temporary group mode:

➤ Press *Ctrl* and click the mouse button.

   If you are working with a congested board/substrate and multiple elements are selected
by a pick, the elements that you do not deselect go into the reject buffer.

**Finding Buses in Composer/Concept and Allegro/APD**

When you draw a schematic in Composer or Concept, you can identify groups of nets as
buses. The Find Filter lets you use this bus identification to process nets that are members
of the buses. In Composer/Concept, each net in a bus has a bus name, followed by a number
that is enclosed in angle brackets. This number specifies the bit position in the bus. For
example, a four-bit data bus can consist of the nets DATA<0>, DATA<1>, DATA<2>, and
DATA<3>.

**Identifying Buses in Allegro/APD**

When you choose the *File–Import Netlist* command and select *Concept* from the Netlist In
dialog box, each bus is assigned a BUS_NAME property and value that matches its net
name. For example, in the bus described in the preceding section, each net receives a
BUS_NAME property with DATA as the assigned value.

The net name that is assigned is the original bus name plus the associated number without
the angle brackets. For example, the corresponding Allegro/APD net names for the four-bit
data bus are DATA0, DATA1, DATA2, and DATA3.

This association between the net name and the bus name lets you use the Find by Name
function to identify the net. Also, by using the *Edit–Properties* command to add the
BUS_NAME property interactively.
Bus Selection Syntax

You can specify designated bus nets on the command line or, if you select Nets in the Find Filter, in the Name field.

To specify a group of nets in a bus

➤ Enter the bus name and a bit subscript field using the following formats:

- `<bit>` Specifies a single bit of the bus. For example, DATA<3> defines net DATA3.
- `<bit1:bit2>` Specifies a subrange of bits. For example, DATA<3:1> defines nets DATA3, DATA2, and DATA1. (The order of this subrange does not matter; DATA<3:1> is the same as DATA<1:3>.
- `<bit_list>` Specifies a list of bit subscript fields that can have either of the preceding formats. Separate each list with a comma. For example, DATA<1:3,7,10:12> defines bits 1, 2, 3, 7, 10, 11, and 12.

In each of these formats, angle brackets delimit the bit subscript field; the `bit` variable specifies a bit number and must be an integer greater than or equal to zero. If you leave the angle brackets empty, Allegro/APD selects all nets of the bus. To select bus members, the bus name must match the net name and bit number exactly.

The following command selects the DATA1, DATA3, DATA4, DATA5, DATA6, and DATA7 nets for processing.

```plaintext
net data<1,3:7>
```

In addition, you can assign a BUS_NAME to nets that do not have a bit number in the name or that match the bus name, but that can be found by using the `busname<>` syntax. For example, if you assign the BUS_NAME property DATA to the DATA0, DATA1, DATA2, and DATA3 nets and enter the following command in the Name field, you would select all of the nets.

```plaintext
net data<>  
```

Using Buses in Allegro/APD

The following Allegro/APD commands accept bus names:

- `hilight`
- `dehilight`
- `show element`
You can also use the select by bus name option to expedite the following operations.

- Highlighting the bus nets
- Assigning placement weights to a bus by defining the WEIGHT property on bus nets
- Routing buses before the other nets by setting the ROUTE_PRIORITY property on bus nets

**Highlighting and Dehighlighting Design Objects**

Allegro/APD lets you highlight and dehighlight objects as a means of locating them and determining their display priority. These commands use both the Find Filter and the Options tab.

**Highlighting Design Objects**

1. Run the `hilight` command.

   Allegro/APD displays the Find Filter or the Visibility dialog box, depending on which one you select before you run `hilight`. 
The Find Filter lists the objects that you can highlight for this command. The Options tab shows the colors that you can use to highlight a design object. The current highlight color appears in the PERM HIGHLIGHT field.

By default, the Options tab contains 24 colors. To change any of these colors, run the color command.

2. Make sure that the available selections in the Find Filter include the design object that you want to highlight.

   To deselect any checked objects, click the check box.

3. Click the highlight color in the Options tab that you want to assign.

   The color in the PERM HIGHLIGHT fields changes to the color you selected.

4. Click on the design objects in the design window.

   —or—

   Hold down the left mouse button to define an area that contains the design objects, then release the button.

   The object(s) is highlighted with the color that you selected in the Options tab.

   This procedure changes the color only of the design object that you selected. It does not change the color of all instances of that type of object.

**Dehighlighting Design Objects**

To dehighlight design objects:

1. Run the dehiligh command.

   The Find Filter appears, listing the valid objects for the active command.

2. Click on a highlighted object.
Hold down the left mouse button to define an area that contains the design objects, then release the button.

The highlight color disappears from the selected object(s).

Automating Design Tasks with Scripts and Macros

If you find yourself repeating certain design tasks on a regular basis, you can create Allegro/APD scripts and macros to automatically perform those tasks.

While you can use both scripts and macros across multiple drawings, scripts always start and end at the same coordinate, whereas a macro lets you start at a different coordinate each time you use the macro. Every action included in the macro takes place relative to the starting point.

To create a script or macro:

1. Run the script command.

   The Scripting dialog box appears.

   Figure 2-24  Scripting dialog Box

Creating and Replaying Scripts

Scripts are useful when performing repetitive tasks such as setting up fields in dialog boxes, adding objects to multiple databases at the same location, and duplicating drawings.

To create a script

1. Run the script command.

2. In the File field of the dialog box, enter the name you want to give to the script.
Click the Browse button to display the script file browser.

a. Select a script from the list. The list is composed of scripts previously recorded
b. Click OK.

   The script file you selected in the browser appears in the File field of the Scripting dialog box.

3. Click Record.
4. Perform the tasks that you want the script to run.
5. Run script again, and click Stop in the dialog box.

To replay a script:

1. Run script.
2. In the File field of the dialog box, enter the name of the script you want to replay.
3. Click Replay.

   Allegro/APD replays the script.

Using Environment Commands with Scripts

You can modify the behavior of script recording and replaying through the use of environment commands entered at the Allegro/APD command line.

ifvar/ifnvar

The ifvar command lets you include variables in scripts and environment files to change from old to new names. The syntax for ifvar is as follows (use of quotes is recommended):

ifvar <variable> "<then command>" "<else command>"
ifvar <variable> <then command>
ifvar <variable> ";" "<else command>"

Note: The command ifnvar performs the opposite function.

The following example illustrates how ifvar can be used to switch between the old and new menu sets.

ifvar OLDCUI "set MENU = $GLOBAL/menus" "set MENU = $GLOBAL/cuimenus"
set MENUPATH = .\$MENU

**scriptmode**

The `scriptmode` command enables and disables script replay and record options. The syntax for `scriptmode` is:

`scriptmode [+ -] [<arguments>]`

When specifying options, + enables options and - disables them. If you execute the `scriptmode` command without specifying one or more options, Allegro/APD displays the options that are currently set. Options that are modified in a script are restored to their original values when the script is terminated. You can embed this command in the script file, and more than one option can be specified. Only the first letter of the option is required.

**Arguments (switches) for scriptmode are:**

- **f flush**
  During record, each command is written to the disk script file. If disabled (default for better performance), Allegro/APD uses a memory buffer that is written to disk when full, or when the script is terminated.

- **pause**
  Will pause until you click the mouse or press a key. If you provide a time specification (in seconds), scripting continues with no further interaction when the time period has elapsed.

- **confirm**
  Stops the script and displays a Yes/No confirmer with an optional message. When you dismiss the confirmer, an environmental variable is set or unset as follows:

    Yes sets confirm = yes
    No unsets confirm

  The default message is "Do it?"

- **c continue**
  During replay, the script continues if an error is encountered. If disabled (default), the script terminates when an error is encountered.

- **e echo**
  During replay, the script echoes the command to the appropriate window before executing the command. If disabled (default), no echo is performed.
If you execute the `scriptmode` command without specifying one or more options, Allegro/APD displays the options that are currently set. Options that are modified in a script are restored to their original values when the script is complete.

### Displaying Connectivity

Allegro/APD uses ratsnest lines to display the connectivity in a design. These lines show the logical connections between pins, lines, or vias that are on the same net.

To display ratsnest lines:

- Run the appropriate `rats` command:
  - `rats all` (displays all ratsnest lines in the design)
You can refine your selection by filling out the Find by Name section of the Find filter.

To remove ratsnest lines from a design:

➤ Run the appropriate `unrats` command.

- `unrats all` (removes all ratsnest lines in the design)
- `unrats component` (removes all ratsnest lines to pins on the component(s) that you select)
- `unrats net` (removes all ratsnest lines to pins on the net(s) that you select)

You can refine your selection by filling out the Find by Name section of the Find filter.

Using Data Browsers

Data browsers are dialog boxes that present objects of the type required by the current command. You can select objects listed in a data browser, but you cannot delete, rename, or otherwise control the type of data displayed. Data browsers list all named objects in an Allegro/APD design or within libraries outside the design, based on parameters that you set in the dialog box.

Displaying Quickview Information

Data browsers support quick views of the database objects that you select from the list in the dialog box. Quickviews let you see a graphic preview of a database or a selection of the properties that make up the database. Supported databases include the following file types:

- `.brd` / `.mcm`
- `.bsm`
- `.dra`
- `.osm`
- `.mdd`
- `.ssm`
- `.psm`
- `.fsm`

File browsers that open scripts, logs, and other text files do not support quickviews.
Note: Older databases must be upgraded to version 14.0 (or subsequent versions) with Qvupdate before you can display them in quickview. See Using Qvupdate to Display Quickview Information for details.

By selecting one of the two quickview buttons, you can view different data associated with your selection:

- Text
  
  The Text button displays text information, such as the information for a package symbol.
  
  Name: SSOP28
  Type: Allegro Symbol
  Units: MILS
  Accuracy: 2
  Pins: 28

- Preview
  
  The preview button displays a simple graphic of the database, the image of which depends on the type of database you are viewing.

  ❑ Quickviews of .brd, .mcm, and .mdd databases display a board outline, package keepin, or a rectangle of the drawing extents and a selected set of the largest pin-count components in the database.

  ❑ Quickviews of symbols display a symbol outline and the number of pins on the symbol. If the symbol contains a large number of pins, the quickview does not display all of them. (But that information can be derived from the text view.)

Figure 2-25 on page 153 shows the data browser that opens from the place manual command and a quickview of the properties of the selected object. The title bar reflects the object type you are browsing.
Figure 2-25  Data Browser for Manual Placement

If Quickview cannot display the preview or the properties of the element, a “Not Available” message appears in the quickview window.

Using Qvupdate to Display Quickview Information

This stand-alone program lets you update footprint information in design (.brd), drawing (.dra), padstack (.pad), or module (.mdd) databases that were created prior to release 14.0 so that text and graphics associated with them can be displayed in the Quickview window of file/library browsers. Without running Qvupdate, such information can be displayed in Quickview only by opening the pre-14.0 database in the Allegro graphic environment and overwriting the database with the save command. Qvupdate lets you update the footprint information for all your pre-14.0 libraries in one operation though the use of the * wildcard character.

Note: Qvupdate does not update symbols; you must update corresponding .dra files. Qvupdate automatically generates symbols from the .dra file.

Note the following conditions:

- Saving pre-14.0 databases in Allegro batch mode does not update the footprint information.
- Running uprev will not add the Quickview data to an Allegro database.
Databases that were created prior to release 13.0 may have to be upreved before running Qvupdate.

Running Qvupdate on Windows

1. Open the Qvupdate dialog box by double-clicking the qvupdate.exe icon located in your install directory.

![Qvupdate Dialog Box](image)

2. Enter, or browse for, the name of the single database you want to update. Include the full path if the database name you enter is in a different location than Qvupdate.
   —or—
   Enter * and the appropriate extension to update to update all files of a particular type; for example, C:\Boards\*.brd.

3. Click Execute.

The program updates the footprint information for the database(s). When the operation is complete, a log file is displayed containing the status of the update.

Running Qvupdate on UNIX

1. Enter qvupdate and a single database name at your operating system prompt.
   —or—
   Enter qvupdate and * with the appropriate extension to update to update all files of a particular type.
   
   qvupdate /boards/pinesc.brd
   qvupdate /boards/*.brd

The program updates the footprint information for the database(s) and returns information containing the status of the update.
Database and Library Selections

In default mode (Database), data browsers list all the objects in your design's database. You can also view all named objects in the Allegro/APD libraries when you check Library. The objects listed in Library mode may sometimes include items already in the design. This is because database items remain displayed in the list box when the library option is checked.

If an object in the database has the same name as an object in the library but contains different content, the database object takes precedence in the data browser; that is, the database object is selected.

When you check the Library option, it reopens in Library mode for the duration of the design session, or until you de-select the library option.

To select a database object:

1. Choose an application that prompts you for data by opening a data browser. (Specific instances are covered in the appropriate sections of this user guide.)

2. If the object you are looking for is not listed in the design’s database, click Library to get a listing of all objects in the library.

   You can filter the objects displayed in the list box by typing a string (partial object name) and a “wildcard” character in the field. For example:

   - Type FLAT* to display all object names that begin with FLAT.
   - Type FLAT*x to display all object names that begin with FLAT and end with x.
   - Type FLAT ?, where ? represents any single character.

   Data browsers remember filters that you enter in the field. They can be reviewed by clicking the arrow button to the right of the field.

   - Highlight a filter by clicking on it or by using the up-arrow/down-arrow keys on your keyboard.
   - Close the filter history menu by clicking the arrow button.

3. Select the object name you want to place in the design using one of these methods:

   - Click the object name.
     The object name is highlighted and appears in the field.
   - Type the object name in the field.
The data browser searches the design database, then the library files for the object. If the name you are looking for is in the library, the Library checkbox turns on to indicate the object’s location.

- Double-click on the object name.
  
  The object is selected and the data browser closes.

4. Do one of the following:

- Click **OK**.
  The data browser closes and the selected valid object is ready to be placed in the design. (*OK* does not close the browser until a valid object name is selected.)

  --or--

- Click **Cancel** to close the data browser without placing an object.

---

### Running Commands with Strokes

You can run certain commands using predefined patterns of mouse strokes that Allegro/APD lets you draw in the user interface. Allegro/APD interprets the patterns as commands and executes the command when you complete the stroke.

You can run these commands using stroke patterns:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>World View</td>
</tr>
<tr>
<td>Z</td>
<td>Zoom In</td>
</tr>
<tr>
<td>M</td>
<td>Move</td>
</tr>
<tr>
<td>C</td>
<td>Copy</td>
</tr>
<tr>
<td>A</td>
<td>Delete</td>
</tr>
<tr>
<td>U</td>
<td>Oops</td>
</tr>
</tbody>
</table>

When you use strokes to run these commands, the following conditions apply:

- The world view (W) stroke can be drawn anywhere on the design.
The zoom in (Z) stroke zooms in to the area in which you draw the Z.

The move (M), copy (C), and delete (D) strokes select the object under the first point of the stroke, shown here as circles in the patterns.

For more information on the commands listed here, see the appropriate sections of your Allegro/APD documentation.

To run commands using strokes

1. In the design work area, locate the cursor over the object you want to move, copy or delete, or over the area you want to zoom in to. (The world command can be drawn anywhere.)

2. Hold down the Control key and right mouse button at the same time to make a stroke.
   As you move the mouse, you see the pattern being drawn.

3. When the stroke is complete, release the right mouse button.
   If Allegro/APD recognizes the stroke, the command runs and the object is highlighted. If the stroke is unrecognized, Allegro/APD displays the following message:
   Stroke not recognized.

Note: If you draw a stroke in the design area but not over an object, the command runs without a selected object.

Allegro/APD does not let you create your own stroke patterns or edit the predefined patterns.

Creating Strokes (UNIX only)

If you are running Allegro/APD on a UNIX workstation, you can create your own strokes by customizing the default Allegro/APD.strokes file provided by Cadence.

To specify a file containing your own strokes

➤ At the Allegro/APD command line, type

   strokefile <filename>
Allegro/APD will automatically use your customized file, stored in the user’s `pcbenv` directory.

To create your own stroke file, or edit an existing stroke file:

1. Type `sted` at a command line prompt.
   
The STED Editor user interface appears.

2. Follow the instructions displayed in the STED Help files for information on creating and editing strokes.
   
   **Note:** You must enter strokes in the same direction in which they were created. This lets you have two strokes that look the same but issue different commands.

   For example, if two strokes appear as diagonal lines, one can represent the *Edit–Copy* command, and the other the *Window* command. The difference is that one stroke is drawn from upper left to lower right and the other from lower left to upper right. In the stroke editor file, a cross signifies the starting point of the stroke.

### Defining Aliases

The alias feature lets you define a command vocabulary and create shorthand for commands you use most often. You can also program function keys (on most keyboards) to execute commands to increase speed and ease of work.

The alias is an alternative way of entering the command, but it does not disable the full commands. You can still use the standard form of the command.

This section describes how to establish an alias for typed entries and for function keys. Note that aliases work only in Allegro/APD, not at the operating system level.

A command alias entered at the Allegro/APD command prompt is active only for the current work session. When you exit from Allegro/APD and return to the operating system, aliases are lost.

To use command aliases repeatedly

- Define and save them in a local Allegro/APD environment file as described in Managing Environment Variables

Some default command aliases are provided with Allegro/APD. The sample Allegro/APD global environment file lists the default aliases for the function keys and for the typed commands.

**Note:** `a` is used as an alias for `alias`. 
You have several options at the keyboard. You can

- Use standard commands
- Use the default aliases
- Define aliases for personal use
- Define temporary aliases for an individual work session by entering the alias command on the Allegro/APD command line
- Establish aliases in a local environment file that remain in effect at every Allegro/APD login until you change the environment file

Creating Aliases

To create a command alias for the current work session:

➤ Enter the **alias** command at the Allegro/APD command line.

Type the keyword **alias**, your abbreviated form of the command, and the command string to be abbreviated.

```
alias name definition
```

—or–

```
alias <your abbreviation> <Allegro/APD command>
```

For example, to abbreviate the standard **gloss** command as **gl**, type the keyword **alias**, your abbreviation **gl**, followed by the **gloss** command:

```
alias gl gloss
```

—or–

```
a gl gloss
```

After creating this alias, when you type **gl** and press **Return**, Allegro/APD runs the **gloss** command as though you had typed **gloss**.

Chained commands, representing more than one consecutive action or macro command file, can be entered on the command line or defined as an alias. Use a semicolon (;) to separate the commands and enclose the commands in quotes. For example:

```
alias ee "class etch/conductor; menuload etch/conductor"
```

**Note:** When you define an alias, the original form of the command can still be used. Additionally, the name portion of the alias cannot contain any blank spaces.
Deleting Aliases

Use the `unalias` command to delete an alias. At the Allegro/APD command line

To delete an alias

➤ Enter `unalias` and the alias you want to delete.

```
unalias <alias name>
```

For example, to delete `gl` as an alias for `gloss`, enter

```
unalias gl
```

Assigning Function and Control Keys

Allegro/APD function and control keys take advantage of the capability provided by the native windowing systems in which you can execute Allegro/APD. This section describes the function and control keys.

**Note:** Some keyboards may not support all function key assignments.

**Function Keys**

You can assign Allegro/APD commands to any function key that Allegro/APD can access through a native windowing system. Allegro/APD defines function keys for F1 through F10 and SF1 through SF10. Check the sample environment file for a list of the predefined function and control key aliases.

**Example**

To alias the `netin param` command to the F2 key

1. In the Allegro/APD Console Window, type `alias` followed by a space.
2. Press the F2 key.
3. Type `netin param`.

   The command echoed on the command line appears in the following form:

   `alias F2 netin param`

   Remember to type a space after `alias` and F2.
Managing Environment Variables

This chapter describes the structure and purpose of the Allegro/APD environment files (env, env_local.txt, and user-defined .prf) and how you can use them to set operating conditions at the site level and at local levels. Allegro/APD lets you define environment variables in a variety of ways:

- Using the set command
- Editing a local env file
- Editing a user preference file

Important

Allegro/APD is designed to use environment variables specified in the local env file or in the interactive User Preferences Editor dialog box in the Allegro/APD user interface to set padstack, footprint, etc. search paths. If you define search paths such as PADPATH in Project Manager rather than Allegro/APD, you must use Project Manager to launch Allegro/APD to locate the desired files.

The Global Environment File

The environment file is an ASCII text file that contains system and configuration information critical to the operation of your Allegro/APD software. This information is contained in two forms:

- Variables
- Aliases

Variables control system behavior as well as define search paths so that Allegro/APD can locate data files needed during processing. For example:

\[ \text{autosave\_time} = 30 \]

is a system variable that tells Allegro/APD to automatically save an active file every 30 minutes. Another type of variable is a path variable such as
set libpath = .D:\PCBENV\share\PCB\pcb_lib

which identifies the pcb_lib directory in the current working directory as the location of device and symbol files provided by Allegro/APD.

**Aliases** are function key strokes or typed commands that you can assign to affect a command or task. An example is

```bash
alias F4 cancel
```

which allows you to cancel your last action by pressing the F4 key.

The global environment file (env) is located in the text directory of the Allegro/APD install directory. Allegro/APD looks for the env file in this location whenever it is started and, if not found there, generates an error message.

**Note:** Do not move the env file or copy/modify its contents. Changes made to the file will be lost if you reinstall Allegro/APD or if you upgrade the software. See “Setting User-Defined Variables” on page 168 for details on how to customize environment variables.

### Path Variables

The variable set automatically by Allegro/APD upon startup is:

```bash
set GLOBAL = $ALLEGRO_INSTALL_DIR/text
```

Allegro/APD uses other configuration variables to locate system files for menus, forms, and messages.

In addition to these configuration variables, the env file also contains library search path variables that determine how Allegro/APD searches for symbol, device, and help files.

In each case, moving system directories without making the appropriate changes in the path variables cause Allegro/APD to generate errors and fail to locate information. It is NOT recommended that you move these directories.

The following environment variables should NOT be changed:

- **System directory path variables that control APD executable programs and processes:**

  ```bash
  # Configuration variables (Don't change)
  set FORMPATH = . $GLOBAL/forms
  set MENU = $GLOBAL/cuimenus
  set MENUPATH = . $MENU
  set FILEEXTS = $GLOBAL/extensions
  set UNITS = $GLOBAL/units.dat
  set ICONS = $GLOBAL/icons
  ```
set BMPPATH = $GLOBAL/icons
set RESOURCE = resource

- System variables that control system configuration behavior:
  - ANSIFONT = ansifont
    Text font vector file, used with VECTORPATH variable.
  - ICONS = $GLOBAL/icons
    Pointer to icon directory for Allegro.
  - FILEEXTS = $GLOBAL/extensions
    Extensions to file-types rule file for the Allegro browser.
  - RESOURCE = resource
    Resource (print) configuration file for the editors.
  - UNITS = $GLOBAL/units.dat
    Units rule conversion file.

- Internal variables:
  - BASE
    Pointer to the directory where the editor was started
  - ENV_CLASS
    Last class name set by the class command
  - ENV_SUBCLASS
    Last subclass name set by the subclass command
  - LOCALENV
    Pointer to the local environment directory
  - GLOBAL, GLOBALPATH
    Pointers to the global environment directory
  - MENULOAD
    Currently loaded menu file for the editor
  - MODULE
Current design name

❑ CWD

Current working directory

Library Path Variables

The global environment file contains the library search paths to all the libraries that are provided with Allegro/APD. In a local environment file, you can add or modify environment variables that define custom library search paths; for example, to locate component libraries for specific design projects. This procedure is explained in more detail in “Defining Library Path Variables in a Local env File” on page 172.

System Variables

The env file controls the appearance and behavior of Allegro/APD through variables that modify graphics displays, control automatic save functions and plotting, allow file versioning, influence glossing, change the contents of backannotation files, and perform other functions. However, not every variable is included in the installed env file. See “Setting User-Defined Variables” on page 168 for a list of variables that can be added at the local level.

The Installed env File

The following is a representation of the global environment file.

# # ALLEGRO GLOBAL Environment file # # $Header: env,v 1.166 01/07/16 14:14:21 fxf $

# The following variables are set by the software if not preset by user:
# CDS_SITE, ALLEGRO_SITE, TELENV, ALLEGRO_INSTALL_DIR,
# ALLEGRO_INSTALL_TOOLS, ALLEGRO_INSTALL_ROOT, ALLEGRO_TYPE,
# __UNIX (if UNIX), _PROGRAM, HOME, LOCALPATH, LOCALENV
# Variable names are case insensitive

#-----------------------------------------------
# System Variables
set GLOBAL = $ALLEGRO_INSTALL_DIR/text
# change from LIBPATH because it conflicts with IBM’s shared library
# environment variable
set ALIBPATH = $ALLEGRO_INSTALL_DIR/pcb_lib

set COMPLIBPATH = $ALLEGRO_INSTALL_DIR/allegrolib

set GLOBALPATH = . $GLOBAL

# Allegro Data Services Variable
set ADSPATH = $GLOBALPATH

# Present for compatibility with older versions of Allegro.
ifnvar ALLEGRO_SITE "set ALLEGRO_SITE ."

#--------------------------------------------------------------------
# Configuration variables (Don’t change)
set BMPPATH = . $ALLEGRO_SITE/icons $GLOBAL/icons
set FORMPATH = . $ALLEGRO_SITE/forms $GLOBAL/forms
set MENUPATH = . $ALLEGRO_SITE/menus $GLOBAL/cuimenus

set UNITS = $GLOBAL/units.dat

# Configuration paths
set PRFEDITPATH = . configure/prfedit $LOCALENV/configure/prfedit
$ALLEGRO_INSTALL_DIR/configure/prfedit
set BATCHHELPATH = . $ALLEGRO_INSTALL_DIR/batchhelp
set SROUTEPATH = $ALLEGRO_INSTALL_DIR/configure/sroute

set HELPPATH = . $ALLEGRO_INSTALL_DIR/help $GLOBAL/help
set PDFPATH = . $ALLEGRO_INSTALL_DIR/help/pdf

# Graphics demo env
set IMAGEPATH = . $ALLEGRO_INSTALL_DIR/examples/image

#--------------------------------------------------------------------
# Drawing font
set VECTORFONTPATH = . $GLOBAL
set ANSIFONT = ansifont
set KANJIFONTPATH = . $GLOBAL/fonts/kanji
set KANJIFONT1 = kanjifont1
set KANJIFONT2 = kanjifont2
# Display Variable

# disable repair under ratsnests
set display_norepair rats
set display_shapefill_analysis 1

# HDL Supported Design Library Search Path Variables
set MODULEPATH = . $ALLELEGRO_SITE/modules
set PADPATH = . symbols .. $ALLEGRO_SITE/padstacks $compalib
set PSMPATH = . symbols .. $ALLEGRO_SITE/symbols $compalib
set TECHPATH = . $ALLELEGRO_SITE/tech $GLOBAL/tech
set TOPOLOGY_TEMPLATE_PATH = . templates .. $ALLEGRO_SITE/topology $topfilelib

# SigNoise data installation directory
set SIGNOISEPATH = . $LOCALENV $ALLELEGRO_SITE/signal $signal_install_dir $signal_optlib_dir $GLOBAL

# Non-HDL Supported Design Search Path Variables
set ARTPATH = ..
set APTPATH = ..
set CLIPPATH = .
set DCLPATH = .. $ALIBPATH $COMPLIBPATH
set DEVPATH = .. $ALLELEGRO_SITE/devices $ALLEGRO_SITE/devices $COMPLIBPATH/devices
set DFAAUDITPATH = . $ALLELEGRO_SITE/assembly $ALLELEGRO_INSTALL_DIR/assembly
set NCDPATH = .. $ALLELEGRO_SITE/nclegend $GLOBAL/nclegend
set SCRIPTPATH = . $ALLELEGRO_SITE/scripts $GLOBAL/script
set TEXTPATH = .. $ALLELEGRO_SITE/extracta $GLOBAL/views
set VIEWPATH = .. $ALLELEGRO_SITE/views
set XTALK_TABLE_PATH = .. xtalk_tables $ALLELEGRO_SITE/xtalk $ALIBPATH/xtalk_tables
# System Configuration .scf file search path
set SCFPATH = . scfs .. ../scfs

#--------------------------------------------
#--------------------------------------------
# Aliases
#--------------------------------------------
# Bind roam operations to function keys
set roamInc = 96
alias Up roam y -$roamInc
alias Down roam y $roamInc
alias Left roam x -$roamInc
alias Right roam x $roamInc

#--------------------------------------------
# NEW STYLE FUNCTION KEY ALIASES
# F1 is normally reserved by the system for Help so we don’t use it
alias F2 done
alias F3 oops
alias F4 cancel
alias F5 show element
alias F6 property nets
alias F7 vertex
alias F8 zoom points
alias F9 zoom fit
alias F10 zoom in
alias F11 zoom out
alias F12 property refdes
alias SF1 add connect
alias SF2 grid
alias SF3 hilight pick
alias SF4 dehilight all
alias SF5 redisplay
alias SF6 slide
alias SF7 move
alias SF9 save_as temp
alias ~N new
alias ~O open
Setting User-Defined Variables

User-defined variables allow you to add or modify certain behaviors to Allegro/APD. Variables can be set at the local level to provide you with pathways to individual project directories and associated libraries, accommodate individual display preferences, or set certain behaviors. For example, if you want to ensure that files are saved to disk at least once every 15 minutes, you can set an autosave variable at the local level to accomplish this.

The set command

The set command is one way that you can define or replace an environment variable. (Another method is described in “The User Preferences Editor” on page 173.)
The syntax for the `set` command is:

```
set variable_name = value(s)
```

**Note:** Methods for setting environment variables vary according to the shell you are using. If you are using `csh`, for example, you can set variables using the `setenv` command. If you do not know what shell you are using, refer to your operating-system documentation or see your system administrator.

A simple example is setting the database to save your work automatically every 30 minutes. In your local `.env` file, you enter

```
set autosave_time = 30
```

To disable settings in your local file, you can delete the entry or use the `unset` command.

**The settoggle command**

Use the `settoggle` command to change the value of an environment variable based on its current value and a list of possible values. The syntax for the `settoggle` command is:

```
settoggle <variable name> [value1] [value2] ... [value n]
```
Managing Environment Variables

**variable name**  required environment variable name

**values [1 - n]**  an optional list of possible values for the environment variable

If you specify no optional values...

...and the variable is unset, Allegro/APD sets the variable with a value of " ", which is equivalent to:

```
set <variable name>
```

...and the variable is currently set, Allegro/APD unsets the variable, which is equivalent to:

```
unset <variable name>
```

If you specify one value...

...and the variable is unset, Allegro/APD sets the variable to that of the specified value, which is equivalent to:

```
set <variable name> value 1
```

...and the variable is currently set, Allegro/APD unsets the variable, which is equivalent to:

```
unset <variable name>
```

If you specify more than one value...

...Allegro/APD substitutes the value listed immediately after the current environment variable value for the current variable. The comparison is case insensitive. Allegro/APD sets the environment variable to the first value in the value list when the variable:

- is currently unset
- has a value not in the list
- has the same value as the last item in the value list

This is equivalent to:

```
set <variable name> value 1
```
Settoggle command examples

Example 1

1. The following unsets the `pcb_cursor` environment variable:
   ```
   unset pcb_cursor
   ```

2. The following sets the `pcb_cursor` environment variable to `infinite`:
   ```
   settoggle pcb_cursor infinite cross
   ```

3. The following sets the `pcb_cursor` environment variable to `cross`:
   ```
   settoggle pcb_cursor infinite cross
   ```

Example 2

1. The following unsets the `display_drcfill` environment variable:
   ```
   unset display_drcfill
   ```

2. The following sets the `display_drcfill` environment variable:
   ```
   settoggle display_drcfill
   ```

3. The following unsets the `display_drcfill` environment variable:
   ```
   settoggle display_drcfill
   ```

Creating a Local env File

This section describes how to set up environment files. You do this by adding variables and/or aliases to a `local` environment file.

**Do not move, copy, or modify the global environment file in the software install directory.**

1. Copy the file named `env_local.txt` in the Allegro/APD install directory to a file named `env` in your home directory, using the copy method appropriate to your platform.

   **Hint:** If you are unsure where the `env` files are located, type the `set` command in the console window of Allegro/APD. In the Defined Variables window, find “set envpath” in the list of variables. The `env` files are located in the text directory at the end of the path.
2. Open the file.

```
# ALLEGRO local user's environment file
#                                           # - this indicates a comment
#
# read global environment file
source $ALLEGRO_INSTALL_DIR/text/env
```

The first 5 lines in the example are comments. (The pound sign as the first non-
whitespace character tells the system to ignore the information on that line. Comment
lines can be inserted anywhere in the file.)

Line six contains the source command which tells the system to read and execute all the
information in the global environment file. Any data that you now enter into the local file
becomes part of the “instruction set” of the Allegro/APD software.

**Note:** The location of information in the local file is pertinent. Path and system variables
and aliases that are entered *under* the source command line in the local file will
supercede similar variables in the global file.

3. When you have finished adding or modifying variables in the local env file, save your
changes and close the file.

### Defining Library Path Variables in a Local env File

When you create a new library, you can enter a library path variable in your local environment
file that will access that library instead of the default libraries provided with Allegro/APD.

The pathname is a directory search list. Allegro/APD looks for data in the order listed in the
path. For example:

```
set PSMPATH = . symbols ....../symbols $LIBPATH/symbols
```

defines a search path that looks for the required directory (symbols) in the current working
directory. If there is a symbols library, Allegro/APD accesses it for symbols as needed. If there
is no symbols library in the current working directory, Allegro/APD continues to look in the
next directories higher up. If no user-created symbols libraries are found, Allegro/APD uses
the installed symbols library.

1. Place project directories containing your custom libraries in a location other than the
   Allegro/APD-installed libraries.

2. In your local environment file, enter the new library search path variable.
Example: You have created a custom symbols library for a project, and have placed that library in a directory called `sym_pro1` in the current working directory. In your local `env` file, under the line that sources the global `env`, add this line:

```
set PSMPATH = ./sym_pro1 ..../symbols $LIBPATH/symbols
```

Allegro/APD will search this path for the symbol library instead of the symbols path name in the global `env` file.

### Setting Commands in the Console Window

You can override variables and aliases in your local environment file by entering `set` commands in the console window of the Allegro/APD user interface. Commands that you set here remain in effect for the duration of the current work session.

**Note:** Some variables, such as `autosave`, must be set *before* launching Allegro/APD. Such variables must be set in the local `env` file, or through other methods provided by the operating system environment. In Windows NT, for example, variables can be set in the System dialog box in the Control Panel.

### The User Preferences Editor

If you prefer not to work directly in your local `env` file or by using the `set` command at the Allegro/APD command prompt, you can set or unset environment variables from the User Preferences Editor, a graphical user interface that you open by running `enved` from the Allegro/APD command prompt. A summary description of each variable displays at the bottom of the dialog box when you change a value.

### Setting User Preferences

To set preferences in the User Preferences Editor:

1. Run the `enved` command to display the User Preferences Editor.
Figure 3-1 User Preferences Editor

2. Select a category from the *Categories* list.

   —or—

Enter a preference name in the *Search for preference* field and press the *Tab* key.

The list of preferences associated with the category appears.

3. Change values for preferences in any of these ways:

   - Checking or unchecking a box.
   
   - Entering or deleting data in a field by typing or selecting a value from a menu, where available. To delete a value using a drop-down menu, select the blank value.
Resetting paths in physical path windows. This relates to the Design_paths and Ui_paths categories.

As you click on a checkbox, field, or button, information related to the preference appears in the Summary description section.

Notice when changes to preferences become effective:

- **Command** – When you run the next command related to the preference
- **Immediate** – As soon as you click OK in this dialog box
- **Repaint** – When you reset your view of the Allegro/APD work area
- **Restart** – After restarting Allegro/APD

4. If you want to view a text file listing all current settings, click List All.

5. Click OK to save your changes.

**Customizing the Environment Variable Editor**

The User Preferences Editor dialog box displays user-defined variables through a mechanism called a user preference file. Preference (.prf) files are used to categorize user preferences (user-defined environment variables). You can use preference files to customize the tree view control in the Category section of the User Interface dialog box. In Figure 3-2, the dialog box is displaying the preferences found under the category UI.
This category and its preferences (environment variables) are determined by the contents of its associated preference file, *ui.prf*, shown below.

```
# $Header: ui.prf,v 1.2 02/10/08 15:47:16 rez $
#
# User Interface Variables
#
# See User.prf for complete descriptions of fields in this file.

noconfirm:TYPE=CHECK:EFF_ONRESTART:
```
HELP=If set Allegro does not prompt you to confirm certain actions. This variable should be used only with scripts.

readme_never:TYPE=CHECK:EFF_ONRESTART:
HELP=If set, the allegro README is suppressed each time allegro is activated.

noformsbutton:TYPE=CHECK:EFF_ONRESTART:
HELP=When set, print to script for a form containing add/reset buttons can include add and reset as commands. Replay of the script will update values and then perform RESET. With the variable set, script record will not record the form buttons.

nodragpopup:TYPE=CHECK:EFF_ONRESTART:
HELP=Normally, to use strokes you must hold down the Ctrl key when pressing down the right mouse button. Setting this option allows strokes to be input by just dragging with the right mouse button depressed. With this option, you lose the ability to select popup menu items by doing a right mouse button press, drag and release. Instead, you will have to click twice with the right mouse button: once to see the popup and a second time to select a popup item (works like the ValidUI based allegro_layout).

show_max_manhattan_pins:TYPE=LONG:MIN=0:EFF_IMMEDIATE:
HELP=Show element will not display information when nets contain more than 50 pins for performance reasons. Value can be any number greater than 0. Not applicable for nets with NO_RAT property.

noshow_current_selections:TYPE=CHECK:EFF_ONRESTART:
HELP=If set, disables show_current_selections.

find_nongui_reject:TYPE=CHECK:EFF_ONRESTART:
HELP=When the user rejects an object and there are multiple objects in the reject buffer, a form is now presented. This variable restores old functionality, which allows the user to step through the list, continually selecting reject from the popup.

form_olddo:TYPE=CHECK:EFF_ONRESTART:
HELP=When a carriage return is used after adding a value in an Allegro form, the new UI will close the form. Use this variable to keep the form active when a carriage return is used while adding values.

control_auto_raise:TYPE=ENUM:POPUP=options,find,visibility:DEFAULT=options:
EFF_IMMEDIATE:\
HELP=Sets the default tab in control panel. If a command had changed the \
tab, upon termination of a command, system reverts control panel to the \
default tab. If not set, default tab is options.

fontfixedface:TYPE=STRING:EFF_ONRESTART:\
HELP=Fixed font name used in forms using fixed fonts. Default is “courier”.

fontface:TYPE=STRING:EFF_ONRESTART:\
HELP=Font name used in forms. Default is “MS Sans Serif”.

fontsize:TYPE=LONG:MIN=0:EFF_ONRESTART:\
HELP=Font size used in forms. Default is -12 for Unix and -9 for Windows NT.

fontweight:TYPE=LONG:MIN=0:EFF_ONRESTART:\
HELP=Font weight used in forms. Default is 400.

Note the following:

- The name of the preference file determines the name of the catagory that appears in the dialog box.
- Each entry in the file describes an environment variable that corresponds to that displayed in the User Preferences Editor dialog box.
- The entries in the .prf do not contain or store value settings, they contain only the descriptions of the variables contained in the catagory. Settings are saved to a user preferences section of your local env file.
- Only 16 entries per file is allowed. If additional entries are required, you must create a new preference file.

⚠️ Important

When you create .prf files, make sure there are no blank lines at the end of the file. Blank lines in a user-created .prf may cause Allegro/APD to crash.

Default reference files supplied by Cadence are located in your Allegro/APD install directory

$ALLEGRO_INSTALL_DIRECTORY/share/pcb/configure/prfedit

Each preference file corresponds to a catagory in the User Preferences Editor dialog box. The file user.prf, shown below, is a comment file that explains the particulars of each entry type in a preference file.
# Allegro User Defined Env Variables

# An env editor file.

# Below items could be set in this file.

# SUMMARY:HELP    - (string) Describe a brief overview of the usage of this
# group of env variables.
# <ENV VAR>     - (string) The name of env var. Every entry must start
# with one.
# LABEL         - (string) Optional label to be displayed instead of
# ENV VAR itself to make it less cryptic for the user.
# TYPE          - (string) Data type for the env var. Choices are:
#                   STRING,
#                   LONG,
#                   REAL,
#                   CHECK,
#                   BUTTON,
#                   ENUM (popup) used for multiple selection choices
# MIN           - (string) optional, minimum value for LONG and REAL fields.
# MAX           - (string) optional, maximum value for LONG and REAL fields.
# POPUP         - (string) optional, list of choices for ENUM fields.
#                   Popup items are seperated by ",".
# DEFAULT       - (string) is used to restore env var to default value when
# user hits defaults button in the pref editor. This
# is specially useful for POPUPs.
# EFF_IMMEDIATE - (bool) if the change will take effect immediately or
# EFF_ONREPAINT - (bool) if the change will take effect after repaint
# operation is performed.
# EFF_ONRESTART - (bool) if the application has to be restarted for the
# change to take effect.
# HELP          - (string) Brief description of the usage of the env var.

# Providing values to above options; all options separated by ";"
# string        - <name>=<string value>
# int           - <name>#<value>
# bool          - <name>

# Note1, MIN and MAX fields expect numerical values in long or double.
# Note 2, '\' escapes ':' when ':' is not used as a separator of fields.
#
# Maximum number of entries per file is 16. A new file should be created
# if the number exceeds 16.
#

SUMMARY:HELP=User Specific Settings. \\nCopy this file to local configure/prfedit directory to add your \\own preferences to it. See prfeditpath setting for more details.

You can modify *user.prf* to display custom variables that will display when you select the User catagory in the dialog box. You can also create new preference files by copying existing .pfrs to new file names. The new file name will be displayed as a catagory in the dialog box.

The path variable PRFEDITPATH searches for preference files at these locations

```
.
configure/prfedit
<home directory>/pcbenv/configure/prfedit
<customer_site> /configure/prfedit
$ALLEGRO_INSTALL_DIRECTORY/share/pcb/configure/prfedit
```

Searching begins at the local level, so that preference files stored locally or in your home directory take precedence over preference files of the same name located elsewhere; for example, at a customer site location available to a group of users.

**Path variables**

Unlike system variables, selecting a value for a path variable (for example, PRFEDITPATH) displays a Physical Paths window in which you can change the order of a path search, delete part or all of a path, or enter a new path string. Each of these actions is controlled by buttons on the user interface. Figure 3-3 illustrates a Physical Paths window containing the path variable for PRFEDITPATH.
Setting Project Level and Site Customization Variables

If you are working within an HDL-based project, you may want to specify design library search paths at the project level to enhance integration of these tools into your design flow. If you are a CAD site administrator, you can customize your Allegro/APD environment for your workplace.

Project File Variables

To better integrate your project into HDL-based design flows, you can base your design path variables on the contents of the standard Concept-HDL project (.cpm) file. This file controls variable settings when you work in the Concept-HDL environment; that is, when you open Allegro/APD through Program Manager. You can also set environment variables when you run `enved` with a special option (see Setting .cpm variables on page 182).

If design path variables are not set in the .cpm file, your design tool uses the variables defined in the PATH settings of your local environment file. The .cpm file supports any of the following design variables:

- PSMPATH
- PADPATH
The `.cpm` design path settings defines the `.cpm` project file at your user-defined location.

**Setting .cpm variables**
You can set `.cpm`-based design path variables by:

- Editing the `.cpm` file
- Accessing the env file
- Running Tool Setup in Project Manager
- Using the `enved` command with the `-proj <.cpm file location>` option. This is the recommended method and the one described in this section.

1. Run the `enved` command from your operating system prompt with the `-proj` option, as shown in the example
   
   `enved -proj <.cpm file location>
   
   The User Preferences Editor opens.

   **Note:** You cannot run `enved` with the `-proj` option from the command prompt in Allegro/APD.

2. Select **Design paths** from the Categories list.

   The design path preferences are listed in the dialog box, as shown in Figure 3-4. The CPM column appears only if you run `enved` with the `-proj` option.
3. Check the CPM boxes for the design paths you want defined by the .cpm file.

4. Click OK to save the changes and close the dialog box.

5. Restart Allegro/APD to put the changes into effect. (This step is necessary only if you are running enved in stand-alone mode, or if you are running setup from Project Manager)

Site Customization

Site customization through the operating system variable CDS_SITE lets you customize the Cadence-supplied environment by overriding the Allegro/APD default location, <cdsroot>/share/local. It allows you to create a directory hierarchy in CDS_SITE where you can place personalized files that extend or enhance your site's entire suite of Cadence tools. In addition to the CDS_SITE variable at the operating system level, you can set a variable,
ALLEGRO_SITE, within Allegro/APD for individual users. ALLEGRO_SITE lets you locate specific configuration files outside the standard default location, $CDS_SITE/pcb. Site customization does not require any changes to the installation hierarchy or modification to the local environment.

**Note:** This feature is designed for use by CAD site administrators.

Allegro/APD searches for site-specific locations in the following order:

- $ALLEGRO_SITE (default location: CDS_SITE/pcb)
- $CDS_SITE/pcb
- <cds_root> /local/pcb

### Using the CDS_SITE Functionality

The CDS_SITE variable allows you to create directories and files that support Allegro/APD functionality. Directories you might choose to create at the CDS_SITE location could include:

- Standard script files in a “scripts” directory
- Locally developed Skill programs—and a *allegro.ilinit* file to load them—in a “skill” directory

Allegro/APD searches for Skill files in the following order:

- <cdsroot>/share/pcb/etc/skill (or a user-defined location specified by CDS_SITE)
- $ALLEGRO_SITE/skill
- $HOME/pcbenv
- . (the program’s start directory)

**Note:** You can reverse the search order by setting the environment variable *skill_old_ilinit*.

You can also create a *site.env* file containing variable settings that would propagate across an entire design site. For example:

- Infinite cross-hair cursor
- Replacement of default Allegro/APD symbol paths with your own

To effect these settings, your *site.env* file would need to contain the following data:

```plaintext
set pcb_cursor = infinite
set psmpath = . $allegro_site/symbol1 $allegro_site/symbol12
```
set padpath = . $allegro_site/symbol1 $allegro_site/symbol12

To load Skill files, your allegro.ilinit file must contain specific data. The following is an example of a .ilinit file. (This example file can be found at `<cdsroot>/share/local/pcb/skill`.)

```
; This example file shows how to load Skill files (those with the
; extension ".il" in the current directory.
; To use, copy to allegro.ilinit if to be used by all Allegro-based programs
; or <programName>.ilinit if intended for only one program
;
; Setting Allegro environment variable, LoadSkillFilesDebug will turn
; on printing the name of each file as it is loaded.

unless(boundp(`LoadSkillFilesDebug)
    LoadSkillFilesDebug = axlGetVariable("LoadSkillFilesDebug")
)

when(LoadSkillFilesDebug printf("\n"))
    (foreach file (rexMatchList "\.il$" (getDirFiles ".")
        when(LoadSkillFilesDebug printf("Loading Skill file: %s\n" file))
            (load strcat("./" file))
    )
when(LoadSkillFilesDebug printf("\n"))
```

**Environment Compatibility**

HDL design path information is ignored when you open designs in pre 14.2 releases. In these instances, traditional environment path variables are used.

Site-based changes that you make through the CDS_SITE variable are ignored in older shell environments unless you use the -q option when you source your environment file. Doing so appends a line to your master env file that reads the `site.env` file, when present. The format for the command is

```
source [-q] <environment_filename>
```
Environment File Variables

This section describes the effect of environment variables when set. The table below lists the types of variables and their location.

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>See ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autosave</td>
<td>“Autosave” on page 186</td>
</tr>
<tr>
<td>Autovoid</td>
<td>“Autovoid” on page 187</td>
</tr>
<tr>
<td>Browser</td>
<td>“Browser” on page 188</td>
</tr>
<tr>
<td>Config_Paths</td>
<td>“Config_paths” on page 189</td>
</tr>
<tr>
<td>Control_Panel</td>
<td>“Control_Panel” on page 190</td>
</tr>
<tr>
<td>Design_Paths</td>
<td>“Design_Paths” on page 191</td>
</tr>
<tr>
<td>Display</td>
<td>“Display” on page 191</td>
</tr>
<tr>
<td>Drawing</td>
<td>“Drawing” on page 193</td>
</tr>
<tr>
<td>DRC</td>
<td>“DRC” on page 194</td>
</tr>
<tr>
<td>Etch</td>
<td>“Etch” on page 194</td>
</tr>
<tr>
<td>File_management</td>
<td>“File Management” on page 196</td>
</tr>
<tr>
<td>Gloss</td>
<td>“Gloss” on page 196</td>
</tr>
<tr>
<td>Misc</td>
<td>“Misc” on page 196</td>
</tr>
<tr>
<td>Plot</td>
<td>“Plot” on page 199</td>
</tr>
<tr>
<td>Roam</td>
<td>“Etch” on page 194</td>
</tr>
<tr>
<td>Skill</td>
<td>“Skill” on page 200</td>
</tr>
<tr>
<td>Simulation</td>
<td>“Simulation” on page 201</td>
</tr>
<tr>
<td>Ui</td>
<td>“Ui” on page 201</td>
</tr>
<tr>
<td>Ui_paths</td>
<td>“Ui_Paths” on page 203</td>
</tr>
</tbody>
</table>

**Autosave**

These variables control the automatic Allegro/APD database save facility.

- autosave

---

January 2002 186 Product Version 14.2
Enables automatic save. This variable must be set or unset before starting Allegro/APD. The command must be used in conjunction with the following autosave variables.

- **autosave_time = n**
  Controls the autosave interval. The default is 30 minutes. The minimum is 10 minutes and the maximum is 300 minutes.

- **autosave_name = filename**
  Saves the autosaved file to a desired file name. The default is AUTOSAVE.brd.

- **autosave_dbcheck**
  Enables a quick database check before an autosave. In default mode, this is turned off because it increases the time needed to save.

- **dbsave_full_check**
  Indicates to the database save utility when to do a full check rather than a quick check. A number of 1, or 0 specifies that each time a board is saved, execute a full check. If you set the variable to 100, then every 100 checks a full check occurs.

### Autovoid

- **av_clineplus <n>**
  Alleviates DRCs generated by autovoid for cline-to-shape spacing problems by forcing autovoid to increase the void size around clines. <n> represents an addition to the DRC value when clearing clines.

- **ed_shape_keep_drcs**
  Use to prevent the Edit – Shape command from removing the current DRCs for the shape from the display.

- **av_endcapstyle = <round, square, octagon>**
  Displays the void around a cline in the named style. Affects the geometry of voids created around cline endpoints. Changes void results when via/pin pad sizes are less than or equal to the line thickness.

- **av_inline = n**
  Controls the distance between pins during autovoid processing to determine whether pins are voided together where n is a number greater than 0. The default is 100.
Restores the 13.5 autovoid program.

- **av_padplus <n>**
  
  Same as `pad_drcplus`. Alleviates DRCs generated by autovoid for pin-to-shape spacing problems by forcing autovoid to increase the void size around pads. `<n>` represents an addition to the pad circumference; for example, setting `n` to 2 increases the radius of the void by 1 mil.

- **av_shapeplus <n>**
  
  Alleviates DRCs generated by autovoid for shape-to-shape spacing problems by forcing autovoid to increase the void size around shape. `<n>` represents an addition to the DRC value when clearing shapes.

- **av_thermal_extend = n**
  
  Controls the thermal-relief clines generated during autovoid processing. Use this variable to specify how far into the shape the clines will extend. The default location is 5 mils inside a shape. The value must be increased by a factor of 10 for each additional unit of accuracy, or thermal connection failures may occur. Example: accuracy = mils 1, value for 3 mil extension = 30

### Browser

- **browser_nodircheck**
  
  Causes all browsers to have the change directory checkbox initially unset. By default, the main file browser (for instance, File – Open) have the checkbox initially set while secondary file browsers (for instance, scripts) have it initially unset.

- **browser_nosticky**
  
  Causes file browsers to always open to the current working directory. If unset, file browsers open to the directory selected the last time it was invoked and a file opened.

- **browser_win31**
  
  Uses the Windows 3.1 file browser: two sections exist; one containing directories to filter and the other containing file data. The new browser lists all directories and files in one section and supports preview of certain file types.

- **clip_filebrowser**
  
  Causes the `clpcopy` command to use the file browser instead of the library path browser. This restores 13.6 behavior.

- **nolast_directory**
Prevents Allegro-based products from using the last directory stored in the `~/pcbenv/allegro.ini` file. Consequently, Allegro launched without command line arguments opens in the current working directory.

- **nolast_file**
  Prevents Allegro-based products launched without command line arguments from using the master.tag file in the current working directory to determine the initial board to open. It starts with unnamed.brd.

- **old_scriptbrowser**
  Uses a file browser for replaying scripts. If unset, uses an Allegro data browser, which supports SCRIPTPATH. This restores 13.6 behavior.

### Config_paths

These variables determine how Allegro/APD searches for library or system files.

- **APTPATH= $APTPATH**
  Search paths for aperture flash files. Obsolete with .fsm support (.bsm)

- **ARTPATH= . ..**
  Search list for artwork control, aperture, and parameter files

- **CLIPPATH = $CLIPPATH**
  Search paths for sub-drawing files (.clp)

- **DEVPATH=. devices .. ../devices $LIBPATH/devices**
  Search path for library devices (.txt)

- **DFAAUDITPATH**
  Search paths for DFA Audit (.arl, .rle)

- **LIBPATH=$ALLEGRO_INSTALL_ROOT/share/pcb/pcb_lib**
  Pointer to master Allegro/APD directory

- **NCDPATH=. ..**
  Search list for NCDrill control files

- **SCRIPTPATH=.. $GLOBAL/script**
  Search list for script files
Managing Environment Variables

- TECHPATH = $TECHPATH
  Search paths for technology files (.tech)

- TEXTPATH=.$GLOBAL/views
  Search list for extract control files

- THERMPATH= ..$GLOBAL
  Search list for thermal control files

- VIEWPATH = $VIEWPATH
  Search paths for visibility schema files (.color)

- WIZARD_TEMPLATE_PATH = $WIZARD_TEMPLATE_PATH
  Search paths for Allegro templates (.brd, .dra)

- XTALK_TABLE_PATH = $XTALK_TABLE_PATH
  Search paths for cross talk tables (.xtb)

Control_Panel

These settings affect control panel operations, including the Find Filter, Visibility, and Options tabs as well as the Worldview window.

- control_auto_raise [options | find |visibility]
  Raises the indicated tab window to the top so long as a specific command doesn't change the window stacking (example, the groupedit command always makes the options tab display on top).

- find_nongui_reject
  Restores pre-13.0 functionality that allows rejection of objects by continually selecting reject from the pop-up, as opposed to rejecting one or more objects from a reject form.

- find_reject_graphics
  When the reject list is active, the selected element in the list can have its graphics altered to assist in identifying it. Allows you to change the behavior so the element either blinks on and off, highlighted, or unchanged. The default is blink. Performance issues could occur with highlighting some elements.

- no_zoom_to_object
Disables the zoom operation in Find By Name.

- wv_voltage_nets
  Displays all highlighted VOLTAGE net elements in the worldview window (pre-14.0 mode). By default, only displays pins of nets with the NO_RAT or VOLTAGE property. Default mode may improve interactive performance on boards with a large number of clines on highlighted voltage nets.

**Design Paths**

These describe design search path settings.

- MODULEPATH = $MODULEPATH
  Search paths for design reuse modules (.mdd).

- PADMIN = . symbols .. ./symbols $LIBPATH/symbols
  Search list for library padstacks (.pad)

- PSMPATH = . $PSMPATH
  Search list for library symbol files (.psm, .osm, .bsm, .ssm, and .fsm)

- SIGNOISEPATH = . ..$GLOBAL
  Search list for signal noise control files

- TECHPATH = $TECHPATH
  Search paths for technology files (.tech)

- TOPOLOGY_TEMPLATE_PATH = $TOPOLOGY_TEMPLATE_PATH
  Search paths for topology template files (.top)

**Display**

These variables modify the Allegro/APD graphics display:

- bug_solaris_hit_lines
  On some Solaris 7 systems, long, odd, angled highlighted liens (that is, ratsnets) can cause graphics issue. Setting this variable causes the line draw to use a line width of 1 rather than 0 to work around the problem. This may slow the line draw slightly but is needed until the platform vendors fixes the problem.
Allegro/APD Design Guide: Getting Started
Managing Environment Variables

- **bug_noxorfatlines**
  Affects IBM platforms when set and displays fat line segments in the dynamic cursor buffer as zero-width lines (during the move command for example).

- **display_backingstore**
  An X window only feature that stores an image of the screen in memory so repainting is unnecessary when a dialog box is closed or a window is moved. Saves repainting but requires additional memory. Applies to drawing data not dialog boxes.

- **display_drcfill**
  Displays DRC markers as “filled” butterflies.

- **display_manual_colorpriority**
  Disables Allegro/APD from being updated to put the new active subclass on top when you drill or swap when adding etch interactively. This increases performance, but Allegro/APD does not redisplay the screen to bring the new layer to the top of the priority. The default is auto-priority.

- **display_nocolor_dynamics**
  Restores the dynamic color behavior present in versions prior to 13.5. When set, dynamic cursor buffer draws elements as white. Unset, elements are drawn in their true color.

- **display_nodynamicarcwidth**
  Displays arcs and circles drawn with a width as zero width for performance reasons while using interactive commands such as move or copy.

- **display_nohilitefont**
  Displays all highlighted elements with a solid highlight color. The default is to display highlighted elements using a combination of the highlight color and the element's original color.

- **display_nolinewidth**
  All lines are displayed at a width of one pixel. This can increase performance at the expense of an accurate display. The default is lines at width.

- **display_norepair = rats, all**
  Display repair is disabled for the described elements. Options are rats or all. When not set, display repair mode is enabled for all elements.

  rats disables display repair for ratsnest lines. Results in better performance.
all provides the best performance while leaving the variable unset. Results in the best display appearance.

The master configuration file sets the rats option. You might want to set it to all for slow machines.

- **display_nosaved_geometry**
  Disables the feature that remembers user positioning of windows.

- **display_noskeletal_draw**
  Disables skeletal draw, which is used when you require graphics to be refreshed very quickly, such as with dynamic zoom out mode.

- **display_raster_ops**
  Controls the use of raster operations for improving the appearance of display features. Additional memory is used and performance can be slow depending on your graphics hardware. If set to “on” any feature is allowed to use it. For example, this greatly improves the look of dynamic graphics. If set to “slow,” features that would required frequent use of the capability are not allowed to use it. The default is “on.”

- **display_shapefill = n**
  Fills in solid shapes with specified line spacing for all classes except analysis. The default spacing is 4. To disable, set the variable to 0.

- **display_shapefill_analysis = <n>**
  Changes the fill lines in shapes on the analysis class.

- **display_thintext**
  Draws text at one pixel width, ignoring photoplot width. This restores 13.5 Allegro behavior.

- **no_dynamic_zoom**
  Performs dynamic zoom with middle mouse click, providing zoom in, zoom out, window fit, and zoom center operations. If disabled, middle mouse key behaves as previously, providing zoom in and zoom out only.

### Drawing

These variables control overrides in the drawing parameters dialog box.

- **drawing_4mils**
Allegro/APD Design Guide: Getting Started
Managing Environment Variables

Allows a database in MILS to be set to an accuracy of three or four decimal places. When unset, which is the default, the maximum accuracy in MILS is two. Using more than two decimal places of accuracy causes rounding issues when fabricating drawings.

- **drawing_no_4mils_msg**
  Hides the warning message in the drawing parameter dialog box if you set the accuracy to more than two when using MILS.

**DRC**

These variables control DRC settings.

- **drc_diff_pair_overide**
  If set, this environment variable causes any Line/Line DRCs to be filtered out if they are on differential pairs (where the primary spacing is less than the Line/Line spacing and produces no violation).

- **drc_old_pad_pad**
  If set, this environment variable uses (for pad/pad) the antipad for spacing checks, if a via or pin is outside a negative shape or inside a negative shape and on a different net. Also uses the thermal pad for spacing checks if inside a negative shape and on the same net as the shape.

**Etch**

These variables control etch settings.

- **acon_route_on_active_subclass**
  Limits the add connect command so that it always begins a route on the active subclass. Otherwise, it may change the subclass to match an element that is findable at the starting pick.

- **allegro_dynam_timing**
  If unset or set to "on," enables dynamic timing feedback. You can alias a function key to toggle the timing display enable using the settoggle command as follows: alias F2 "settoggle allegro_dynam_timing off on"

- **allegro_etch_length_on**
  Makes the add connect command display pin-to-pin etch-length information and net length information during interactive routing.
Allegro/APD Design Guide: Getting Started
Managing Environment Variables

- **bubble_no_display_invisible**
  Prevents elements on nonvisible subclasses from appearing in the etch editing dynamics when they are bubbled. The default behavior is for any bubbled etch to be displayed in the dynamics even if the element would not normally display.

- **bubble_shove_bondpads**
  Allows bondpad vias to be bubbled when via shoving is enabled. The default behavior is for bondpads to not be shoved.

- **noswapripup**
  Does not rip up etch during component swap. The default is rip up etch.

- **rats_factor = n**
  Controls the preference to select straight pin to pin horizontal and vertical ratsnest connections. The value must be a decimal number between 0.0 and 1.0. If not specified, a value of 0.5 is assumed.

  The range of values is:
  - **0.0** — No preference is given to the straightness of a connection. The shortest manhattan length is the sole determining value.
  - **1.0** — Any straight horizontal or vertical connection is preferred over any non-straight connection, regardless of total distance between pins.
  - **0.5** — Both total distance and straightness is considered when selecting pin to pin connections.

  The closer the value is to 0.0, the more distance and the less straightness is considered. The closer the value is to 1.0, the more straightness and the less distance is considered.

- **ratt_off_if_connected**
  Controls the visibility of rat Ts that are fully connected (all of its ratsnests are connected). By default, a rat T remains visible when its ratsnests are all connected. A value of “on” causes a fully connected rat T to be invisible. A value of “unhighlighted” causes a fully connected rat T to be invisible unless it is highlighted.

- **slide_arcs**
  Select arcs when using the `slide` command. If disabled, line segments containing arcs and arc segments will not slide.
File Management

The following variables control the automatic file versioning facility. If you enable file versioning, Allegro/APD appends a file version number to the end of a filename. The default mode for these variables is disabled. The numeric value represents the number of versions you want to maintain.

- **ads_boardrevs = n**
  Enables file versioning for Allegro/APD layout (.brd/.mcm) and symbol (.sm) files. The maximum number of revisions \(n\) allowed is 10.

- **ads_logrevs = n**
  Enables file versioning for Allegro/APD log files. The maximum number of revisions \(n\) allowed is 10.

- **ads_textrevs = n**
  Enables file versioning for Allegro/APD files that are not layout, symbol, or log files. The maximum number of revisions \(n\) allowed is 10.

- **TEMP or TMP**
  Causes Allegro processes to use the specified directory for temporary storage of data files to /temp or /tmp, <directory path with a large quantity of free disk space.> Often artwork requires more temporary space than the /tmp directory contains.

Gloss

These variables influence glossing.

- **cbd_check**
  Allegro/APD performs a DRC check on the connections that have been changed during glossing processing.

- **gloss_pad_shape**
  Allegro/APD considers the outline of a pad as a square or rectangle and enhances line entry into pads

Misc

These variables cover a variety of uses and situations.
Managing Environment Variables

- **art_circvects <n segments in a circle>**
  Gerber 6.x artwork will vectorize arcs differently from the following:
  - Small arcs (500 mils or less) vectorize to 72 segments per circle.
  - Large arcs (3000 or greater) vectorize to 360 segments per circle

- **autosilk_disregard_solder_mask**
  Causes autosilk to clear vias that have no soldermask pads defined.

- **dcnets_delete_norat**
  Causes Identify DC net command to delete the NO_RAT property from those nets getting the power and ground schedule. Use as a migration aid for legacy boards converting from NO_RAT to this type of scheduling.

- **draft_show_trailing_zeroes**
  Disables the removal of trailing zeroes for metric dimensional values.

- **fst_ref_des <value of starting RefDes number>**
  Use when requiring auto rename to start at a specific RefDes number

- **idf_nodelete**
  Causes idf_in interface to not delete existing data in a board while importing new data; by default, idf_in overrides data when importing.

- **iges_arcs**
  Reverses the direction of arcs in iges data transformed by a matrix that contains information for the Z-axis. (in other words, to correct arcs that may appear to be inverted in the output.)

- **logic_edit_enabled**
  Enables the net logic command. By default this feature is disabled to prevent inadvertent changes to the logic.

- **include_terminators**
  Controls the delay rule checker. If checking across the entire net, also includes the terminator length in the calculation.

- **nclegend_file**
Overrides the default filename convention for NC Drill Legend. By default, NC Drill Legend uses a name of default-<units> where units is the current board units. The override name should contain only the filename not a PATH component.

- **netrev_fullmessages**
  Causes netrev to issue additional warning messages when you attempt to add properties to Allegro objects where those properties are illegal.

- **obsolete_color**
  To use the color dialog box in versions 6 through 12.

- **old_style_flash_symbols**
  Uses old style flashes for thermals when you create new boards in Allegro, rather than the new WYSIWYG thermal flash symbols.

- **pcb_baf_pin_number**
  When set, backannotation uses the pin number—not the pin name—for the “was” part of the PIN statement for preassigned and not yet assigned functions.

- **preserve_symbol_textblocks**
  Use to preserve the symbol block number when placing components. Allegro 12.0 and beyond tries to match the symbol block text size to a board text block size when placing components. If a match is not found, a new textblock is created until all text blocks are used.

- **schematic_editor**
  Presets the brand of a new drawing. Possible values are “capture” for capture designs and “hdl concept” for HDL-based designs.

- **scriptmode +invisible**
  Improves script performance of script replay. Insert at the top of a script to make forms invisible during replay.

- **scriptmode +echo**
  Insert at the top of a script to echo script commands to the status line in the command console.

- **scriptmode +step**
  Allows you to step through a script one line at a time. Each script command requires a carriage return or any key to advance.
show_current_selections

Show element command (Display – Element) will not complete automatically when something is selected. Instead, a message window regarding the object is displayed.

TUNEPCB_NOMSG

Eliminates messages generated by the tune_pcb program.

Plot

These variables control plotting and print settings. For plotting, you must set them before creating intermediate plot (IPF) files.

plot_shape_spacing = n

Controls the spacing of the lines in shapes during plotting, where \( n \) is a database unit greater than 1.

print_unix_command

Allows you to use a command other than the Unix default of \( \text{lp} \) for printing to your default printer, which is defined by either the Unix PRINTER environment variable or your system administrator. Because PRINTER is a standard Unix variable, it does not appear in the environment editor and thereby precludes modification.

The syntax is:

\[
\text{set print_unix_command} = \text{"<cmd> [<arguments>] %s"}
\]

The substitution string (%s) indicates where the filename to be printed appears in the command string. The following example uses \( \text{lp} \), but disables the banner and sends the file to a printer named “myprinter.”

\[
\text{set print_unix_command} = \text{"lp -o nobanner -d myprinter %s"}
\]

print_nt_extension

Changes the default filename extension from .txt. Printing on NT sends files to be printed to Notepad, the program that handles printing in a base Windows environment. Notepad uses your default printer, which cannot be changed from within Allegro. If the file to be printed does not end in .txt, Allegro creates a copy of the file with the required extension. Depending upon the driver your printer supplier provides, additional print options may be available, and your printing may not be scriptable as a result.

When you change the extension, you must ensure that you have a print method associated with that extension type in the Windows file association.
The syntax is:

\[ \text{set print_nt_extension} = \text{<extension>} \]

### Roam

These variables detail roam settings.

- **pcb_autoroam**
  Defines time interval in milliseconds between roam increments. If set to 1000, roam increments are 1 per second. If not set, the default is 250 milliseconds.

- **roaminC**
  Defines roam increments in pixels, when using arrow keys to roam. For best performance, set to multiples of 16. Minimum is 16 and maximum is 256 pixels. The default value is 96 pixels.

### Skill

These variables detail Skill options.

- **skill_height**
  Controls the height of the telskill input window for MS Windows only. Value can be any number between 10 and 40. Default is 20.

- **skill_linebuffer**
  Controls the scroll buffer of the telskill input window for MS Windows only. Value can be any number between 40 and 500.

- **skill_old_ilinit**
  RESTORES the old mode of loading the first .ilinit file found. New mode is to load any .ilinit file found in `<cdsroot>/share/pcb/etc/skill, $ALLEGRO_SITE/skill, $HOME/pcbenv or starting directory.`

- **skill_width**
  Controls the width of the telskill input window. Value can be any number between 40 and 140. Default is 80.

- **TELSKILL**
  Allows SKILL information to be entered in the shell/xterm used to open Allegro
Simulation

- **analyze_unspec_pins_as=[name]**
  Provides a global default for pin-use codes. Any pin that is otherwise unspecified will take the value of this variable. Pins with unspecified pinuse codes are treated as though they have the default pin-use code that you have specified.

- **CROSSTALK_CALC=[NET_SUM, PEAK_NET]**
  Determines crosstalk value reported on Signal Noise Analysis form. The default is NET_SUM.

- **CROSSTALK_RANGE=<value>**
  Specifies maximum etch separation distance examined during signal noise analysis. The default value is 0.050 inch. Setting a large value (such as 10) will increase processing times.

- **DRIVER_EDGE=[RISE, FALL]**
  Determines default value for Rise/Fall switch on the Signal Noise Analysis form.

- **NET_SIMRESULT_FILE=<file>**
  If set, NET_SIMRESOLUTION is used and waveform results are written to the specified file.

- **NET_SIMRESOLUTION=<# of points>**
  Controls resolution of the optional sim waveform files. The default is 100 points.

- **NET_SIMTIME=<# of points>**
  Determines simulation time (in nanoseconds) during signal noise analysis. If this variable is not set, the time is a function of the length of the net and the driver’s rise or fall time.

Ui

These variables modify the behavior of the user interface.

- **cancel_key**
  Reassigns the cancel operation. The default assignment for the cancel operation is to use Ctrl-C. Any alphanumeric key can be used with optional control/alpha key or the escape key.

- **fontface**
Sets the font name used in dialog boxes. Default is MS San Serif.

- `fontfixedface`
  Sets the fixed font name used in dialog boxes using fixed fonts. Default is Courier.

- `fontsize`
  Sets the font size used in dialog boxes. Default is 12 points for Unix and 9 points for Windows NT.

- `fontweight`
  Defines the font weight used in dialog boxes. Default is 400.

- `form_oldreturn`
  Keeps forms/dialog boxes open when a carriage return is used while adding values.

- `no_dragpopup`
  Allows strokes to be input by drawing with only the right mouse button depressed, rather than the right button-\text{Ctrl} key combination.

  \textbf{NOTE:} This variable disables the ability to select popup menu items with a right-button click. Instead, you must click twice; once to display the menu, the second time to select an item.

- `noconfirm`
  Allegro/APD does not prompt you to confirm certain actions. This variable should only be used in scripts, and then with caution.

- `noformscriptbutton`
  Prints to script for a form containing add/reset buttons can include add and reset as commands. Replay of the script will update values and then perform \text{RESET}. With the variable set, script record will not record the form buttons.

- `noshow_current_selections`
  Disables the default \text{show_current_selection}.

- ` pcb_cursor = infinite, cross`
  Displays the cursor in one of the two configurations. The default is cross.

- ` pcb_cursor`
  Displays the desired cursor type. The default value is “cross.”
Allegro/APD Design Guide: Getting Started
Managing Environment Variables

- **pcb_cursor_angle**
  
  Applies only to infinite cursor (see the pcb_cursor variable) and specifies the angle for the infinite cursor. By default, the infinite cross-hair is shown orthogonally. Values are between 0 and 90.

- **pcb_stoptracking = n**
  
  Whenever the number of elements in the cursor buffer exceeds \( n \), the dynamic cursor buffer is not redrawn as you move the cursor. The redrawing takes place when the cursor stops.

- **pcb_visibility_down**
  
  You are required to manually invoke the visibility dialog box when adding etch interactively during connect processing. The default automatically invokes the visibility dialog box.

- **readme_never**
  
  Suppresses the Allegro README is suppressed each time allegro is activated.

- **report_height**
  
  Specifies the height of the text window used in the report command. Values are 20 to 70 lines.

- **show_max_manhattan_pins**
  
  Show element will not display information when nets contain more than 50 pins for performance reasons. Value can be any number greater than 0. Not applicable for nets with NO_RAT property.

- **showelement_autovoid**
  
  Automatically positions the show element window to avoid the main window. When unset, which is the default, you can position the window as remembered when next opened.

- **showelement_basearea**
  
  Reports an area in square inches or centimeters instead of the default board units.

**Ui_Paths**

- **formpath**
  
  Search paths for forms.
- **menupath**
  Search paths for menus.

- **prfeditpath**
  Search path for user preferences.
Command Mapping

This chapter provides a mapping of Allegro/APD default menu selections to console commands.

- **Allegro**
  - File
  - Edit
  - View
  - Add
  - Display
  - Setup
  - Layout
  - Void
  - Shape
  - Logic
  - Place
  - Route
  - Analyze
  - Manufacture
  - Tools
  - Help

- **APD**
  - File
### Allegro

The command mapping in the following table lists the menu selections that can be found in the different modes (Layout, Shape, and Symbol) of Allegro Expert. If you are running a different flavor of Allegro (for example, Allegro PCB), some commands listed here might not be available.

<table>
<thead>
<tr>
<th>Menu Selection</th>
<th>Console Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td></td>
</tr>
<tr>
<td>File–New</td>
<td>new</td>
</tr>
<tr>
<td>File–Open</td>
<td>open</td>
</tr>
<tr>
<td>File–Save</td>
<td>save</td>
</tr>
<tr>
<td>File–Save As</td>
<td>save_as</td>
</tr>
<tr>
<td>Command Mapping</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>File–Create Symbol</td>
<td>create symbol (in Symbol Editor only)</td>
</tr>
<tr>
<td>File–Import–Logic</td>
<td>netin param</td>
</tr>
<tr>
<td>File–Import–Artwork</td>
<td>load photoplot</td>
</tr>
<tr>
<td>File–Import–Stream</td>
<td>load stream</td>
</tr>
<tr>
<td>File–Import–IPF</td>
<td>load plot</td>
</tr>
<tr>
<td>File–Import–DXF</td>
<td>dxf in</td>
</tr>
<tr>
<td>File–Import–IDF</td>
<td>idf in</td>
</tr>
<tr>
<td>File–Import–IFF</td>
<td>iff in</td>
</tr>
<tr>
<td>File–Import–SPECCTRA</td>
<td>spectra in</td>
</tr>
<tr>
<td>File–Import–Redac</td>
<td>redac in</td>
</tr>
<tr>
<td>File–Import–Visula</td>
<td>visula in</td>
</tr>
<tr>
<td>File–Import–PADS</td>
<td>pads in</td>
</tr>
<tr>
<td>File–Import–PCAD</td>
<td>pcad in</td>
</tr>
<tr>
<td>File–Import–Sub-Drawing</td>
<td>clppaste</td>
</tr>
<tr>
<td>File–Import–Techfile</td>
<td>techfile in</td>
</tr>
<tr>
<td>File–Import–Active Times</td>
<td>signal atimes</td>
</tr>
<tr>
<td>File–Import–Placement</td>
<td>plctxt in</td>
</tr>
<tr>
<td>File–Annotations</td>
<td>annotation in</td>
</tr>
<tr>
<td>File–Export–Logic</td>
<td>feedback</td>
</tr>
<tr>
<td>File–Export–Netlist w/Properties</td>
<td>netout</td>
</tr>
<tr>
<td>File–Export–IPF</td>
<td>create plot</td>
</tr>
<tr>
<td>File–Export–DXF</td>
<td>dxf out</td>
</tr>
<tr>
<td>File–Export–IDF</td>
<td>idf out</td>
</tr>
<tr>
<td>File–Export–SPECCTRA</td>
<td>spectctra_out</td>
</tr>
<tr>
<td>File–Export–Sub-Drawing</td>
<td>clpcopy</td>
</tr>
<tr>
<td>File–Export–Libraries</td>
<td>dlib</td>
</tr>
<tr>
<td>File–Export–Techfile</td>
<td>techfile out</td>
</tr>
<tr>
<td>File–Export–Placement</td>
<td>plctxt out</td>
</tr>
</tbody>
</table>
Allegro/APD Design Guide: Getting Started
Command Mapping

File–Export–Annotations  annotation out
File–Export–IPC 356  ipc356 out
File–Export–Valor ODB ++ inside  odb_out
File–Export–Save design to 14.0  downrev
File–Viewlog  viewlog
File–File Viewer  No corresponding command
File–Plot Setup  plot setup
File–Plot Preview (Windows NT only)  plot preview
File–Plot  plot
File–Properties  file_property
File–Change Editor  toolswap
File–Script  script
File–Exit  exit

Edit
Edit–Move  move
Edit–Copy  copy
Edit–Mirror  mirror
Edit–Spin  spin
Edit–Change  change
Edit–Delete  delete
Edit–Shape  shape edit
Edit–Z-Copy  zcopy shape
Edit–Delete Unconnected Shapes  delete unconnected
Edit–Split Plane–Parameters  split plane params
Edit–Split Plane–Create  split plane create
Edit–Split Plane–Locate Islands  locate islands
Edit–Compose Shape  compose shape
Edit–Decompose Shape  decompose shape
Edit–Vertex  vertex
<table>
<thead>
<tr>
<th>Command Mapping</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit–Delete Vertex</td>
<td>delete vertex</td>
</tr>
<tr>
<td>Edit–Boundary (Shape editor only)</td>
<td>boundary</td>
</tr>
<tr>
<td>Edit–Change Net (Pick) (Shape editor only)</td>
<td>changenet pick</td>
</tr>
<tr>
<td>Edit–Change Net (Name) (Shape editor only)</td>
<td>changenet name</td>
</tr>
<tr>
<td>Edit–Text</td>
<td>text edit</td>
</tr>
<tr>
<td>Edit–Chamfer (in Designer and Studio series)</td>
<td>draft chamfer</td>
</tr>
<tr>
<td>Edit–Fillet (in Designer and Studio series)</td>
<td>draft fillet</td>
</tr>
<tr>
<td>Edit–Groups</td>
<td>groupedit</td>
</tr>
<tr>
<td>Edit–Properties</td>
<td>property edit</td>
</tr>
<tr>
<td><strong>View</strong></td>
<td></td>
</tr>
<tr>
<td>View–Zoom By Points</td>
<td>zoom points</td>
</tr>
<tr>
<td>View–Zoom Fit</td>
<td>zoom fit</td>
</tr>
<tr>
<td>View–Zoom In</td>
<td>zoom in</td>
</tr>
<tr>
<td>View–Zoom Out</td>
<td>zoom out</td>
</tr>
<tr>
<td>View–Zoom World</td>
<td>zoom center</td>
</tr>
<tr>
<td>View–Zoom Center</td>
<td>zoom center</td>
</tr>
<tr>
<td>View–Zoom Previous</td>
<td>zoom previous</td>
</tr>
<tr>
<td>View–Color View Save</td>
<td>colorview create</td>
</tr>
<tr>
<td>View–Color View Restore Last</td>
<td>colorview restore</td>
</tr>
<tr>
<td>View–Refresh</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>View–Customization–Display</td>
<td>display param</td>
</tr>
<tr>
<td>View–Customization–Toolbar</td>
<td>No corresponding command</td>
</tr>
<tr>
<td><strong>Add</strong></td>
<td></td>
</tr>
<tr>
<td>Add–Line</td>
<td>add line</td>
</tr>
<tr>
<td>Add–Arc w/Radius</td>
<td>add rarc</td>
</tr>
<tr>
<td>Add–3pt Arc</td>
<td>add arc</td>
</tr>
<tr>
<td>Add–Circle</td>
<td>add circle</td>
</tr>
<tr>
<td>Add–Rectangle</td>
<td>add rect</td>
</tr>
<tr>
<td>Add–Frectangle</td>
<td>add frect</td>
</tr>
</tbody>
</table>
Add–Text  
Add–Shapes–Solid Fill  
Add–Shapes–Unfilled  
Add–Shapes–Cross Hatch Fill  
**Display**  
Display–Color/Visibility  
Display–Color Priority  
Display–Element  
Display–Measure  
Display–Parasitic  
Display–Property  
Display–Highlight  
Display–Dehighlight  
Display–Show Rats–All  
Display–Show Rats–Components  
Display–Show Rats–Net  
Display–Blank Rats–All  
Display–Blank Rats–Components  
Display–Blank Rats–Nets  
**Setup**  
Setup–Drawing Size  
Setup–Drawing Options  
Setup–Text Sizes  
Setup–Grids  
Setup–Subclasses  
Setup–Cross-section  
Setup–Vias–Define B/B Via  
Setup–Vias–Auto Define B/B Via  
Setup–Constraints
| Setup–Electrical Constraint Spreadsheet | cmgr |
| Setup–Property Definitions         | define property |
| Setup–Define Lists                 | define list |
| Setup–Areas–Package Keepin        | keepin package |
| Setup–Areas–Package Keepout       | keepout package |
| Setup–Areas–Package Height        | package_height |
| Setup–Areas–Route Keepin          | keepin router |
| Setup–Areas–Route Keepout         | keepout router |
| Setup–Areas–Via Keepout           | keepout via |
| Setup–Areas–Probe Keepout         | keepout probe |
| Setup–Areas–Gloss Keepout         | keepout gloss |
| Setup–Areas–Photoplot Outline     | keepin photo |
| Setup–User Preferences            | env editor |

**Layout**

Layout menu selections are available only in the Symbol Editor

| Layout–Pins          | add pin |
| Layout–Connections   | add connect |
| Layout–Slide         | slide |
| Layout–Labels–RefDes | label refdes |
| Layout–Labels–Device | label device |
| Layout–Labels–Value  | label value |
| Layout–Labels–Tolerance | label tolerance |
| Layout–Labels–Part Number | label part |

**Void**

Void menu selections are available only in the Shape Editor

| Void–Shape            | void shape |
| Void–Circle           | void circle |
| Void–Element          | void element |
| Void–Auto             | void all |
**Shape**
Shape menu selections are available only in the Shape Editor
- Shape–Parameters
- Shape–Check
- Shape–Fill

**Logic**
- Logic–Net Logic
- Logic–Net Schedule
- Logic–Assign Differential Pair
- Logic–Identify DC Nets
- Logic–Assign RefDes
- Logic–Auto Rename RefDes–Rename
- Logic–Auto Rename RefDes–Design
- Logic–Auto Rename RefDes–Room
- Logic–Auto Rename RefDes–Window
- Logic–Auto Rename RefDes–List
- Logic–Change Parts
- Logic–Terminator Assignment

**Place**
- Place–Manually
- Place–Quickplace
- Place–SPECCTRA
- Place–Autoplace–Insight
- Place–Autoplace–Parameters
- Place–Autoplace–Top Grids
- Place–Autoplace–Bottom Grids
- Place–Autoplace–Design
- Place–Autoplace–Room
- Place–Autoplace–Window
<table>
<thead>
<tr>
<th>Place–Autoplace–List</th>
<th>place area list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place–Interactive</td>
<td>place interactive</td>
</tr>
<tr>
<td>Place–Swap–Pins</td>
<td>swap pins</td>
</tr>
<tr>
<td>Place–Swap–Functions</td>
<td>swap functions</td>
</tr>
<tr>
<td>Place–Swap–Components</td>
<td>swap components</td>
</tr>
<tr>
<td>Place–Autoswap–Parameters</td>
<td>swap param</td>
</tr>
<tr>
<td>Place–Autoswap–Design</td>
<td>swap area design</td>
</tr>
<tr>
<td>Place–Autoswap–Room</td>
<td>swap area room</td>
</tr>
<tr>
<td>Place–Autoswap–Window</td>
<td>swap area window</td>
</tr>
<tr>
<td>Place–Autoswap–List</td>
<td>swap area list</td>
</tr>
<tr>
<td>Place–Evaluate–Parameters</td>
<td>eval param</td>
</tr>
<tr>
<td>Place–Evaluate–Design</td>
<td>eval area design</td>
</tr>
<tr>
<td>Place–Evaluate–Room</td>
<td>eval area room</td>
</tr>
<tr>
<td>Place–Evaluate–Window</td>
<td>eval area window</td>
</tr>
<tr>
<td>Place–Evaluate–List</td>
<td>eval area list</td>
</tr>
<tr>
<td>Place–Update Symbols</td>
<td>refresh symbol</td>
</tr>
<tr>
<td>Place–Replace SQ Temporary–Devices</td>
<td>replace temp_device</td>
</tr>
<tr>
<td>Place–Replace SQ Temporary–Symbols</td>
<td>replace temp_symbols</td>
</tr>
</tbody>
</table>

**Route**

<table>
<thead>
<tr>
<th>Route–Connect</th>
<th>add connect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route–Slide</td>
<td>slide</td>
</tr>
<tr>
<td>Route–Custom Smooth</td>
<td>custom smooth</td>
</tr>
<tr>
<td>Route–SPECCTRA–Run Router Checks</td>
<td>specctra checks</td>
</tr>
<tr>
<td>Route–SPECCTRA–Route by Pick</td>
<td>route_by_pick</td>
</tr>
<tr>
<td>Route–SPECCTRA–Route Automatic</td>
<td>auto_route</td>
</tr>
<tr>
<td>Route–SPECCTRA–Interactive Editor</td>
<td>specctra</td>
</tr>
<tr>
<td>Route–Gloss–Parameters</td>
<td>gloss param</td>
</tr>
<tr>
<td>Route–Gloss–Design</td>
<td>gloss area design</td>
</tr>
<tr>
<td>Route–Gloss–Room</td>
<td>gloss area room</td>
</tr>
</tbody>
</table>
Route–Gloss–Window  
gloss area window
Route–Gloss–Highlight  
No corresponding command
Route–Gloss–List  
gloss area list
Route–Testprep–Auto  
testpreop param
Route–Testprep–Create Probe  
probe create
Route–Testprep–Delete Probe  
probe delete
Route–Testprep–Swap Probe  
probe swap
Route–Testprep–NC Tape Probes  
nctape

**Analyze**

Analyze–SI/EMI Sim–Initialize  
signal init
Analyze–SI/EMI Sim–Library  
signal library
Analyze–SI/EMI Sim–Model  
signal model
Analyze–SI/EMI Sim–Model Dump/Refresh  
signal model refresh
Analyze–SI/EMI Sim–Preferences  
signal prefs
Analyze–SI/EMI Sim–Audit–  
signal audit
Design Audit
Analyze–SI/EMI Sim–Audit–Net Audit  
signal audit net
Analyze–SI/EMI Sim–Audit–Audit One Library  
signal lib audit
Analyze–SI/EMI Sim–Audit–Audit List of Libraries  
signal libs audit
Analyze–SI/EMI Sim–Probe  
signal probe
Analyze–SI/EMI Sim–Xtalk Table  
signal xtalktable
Analyze–EMI Rules–Initialize  
signal eminit
Analyze–EMI Rules–Auto Setup  
signal emiautopropmain
Analyze–EMI Rules–Manual Setup  
signal emimanualpropmain
Analyze–EMI Rules–Rule Select  
signal emiruleselect
Analyze–EMI Rules–Audit  
signal emiverify
Analyze–EMI Rules–Execute  
signal emiexecute
### Allegro/APD Design Guide: Getting Started

### Command Mapping

<table>
<thead>
<tr>
<th>Command</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze–EMI Rules–Results</td>
<td>emiresults</td>
</tr>
<tr>
<td>Analyze–EMI Rules–Audit Report</td>
<td>emiverifyslreport</td>
</tr>
<tr>
<td>Analyze–EMI Rules–Execute Report</td>
<td>emiexecutereport</td>
</tr>
<tr>
<td><strong>Manufacture</strong></td>
<td></td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft commands in the Layout Editor are accessed under the Dimension menu item in the Symbol Editor</td>
<td></td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Parameters</td>
<td>draft param</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–LineFont</td>
<td>linefont</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Linear Dim</td>
<td>dimension linear</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Datum Dim</td>
<td>dimension datum</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Angular Dim</td>
<td>dimension angular</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Leader Lines</td>
<td>leader only</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Diametral Leader</td>
<td>leader diametral</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Radial Leader</td>
<td>leader radial</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Balloon Leader</td>
<td>leader balloon</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Chamfer Leader</td>
<td>leader chamfer</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Chamfer</td>
<td>draft chamfer</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Fillet</td>
<td>draft fillet</td>
</tr>
<tr>
<td>Manufacture–Dimension/Draft–Create Detail</td>
<td>create detail</td>
</tr>
<tr>
<td>Manufacture–Artwork</td>
<td>film param</td>
</tr>
<tr>
<td>Manufacture–Stream Out</td>
<td>stream out</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Parameters</td>
<td>ncdrill param</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Legend</td>
<td>ncdrill legend</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Tape</td>
<td>nctape_full</td>
</tr>
<tr>
<td>Manufacture–NC–Route</td>
<td>ncroute</td>
</tr>
<tr>
<td>Manufacture–Cut Marks</td>
<td>cut marks</td>
</tr>
<tr>
<td>Manufacture–DFA Check</td>
<td>dfa</td>
</tr>
</tbody>
</table>
### Allegro/APD Design Guide: Getting Started

#### Command Mapping

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacture–Create Coupons</td>
<td>create coupons</td>
</tr>
<tr>
<td>Manufacture–Silkscreen</td>
<td>silkscreen param</td>
</tr>
<tr>
<td>Manufacture–Variants–Create Assembly Drawing</td>
<td>variant assembly</td>
</tr>
<tr>
<td>Manufacture–Variants–Create Bill of Materials</td>
<td>variant bom</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td></td>
</tr>
<tr>
<td>Tools–Create Module</td>
<td>create module</td>
</tr>
<tr>
<td>Tools–Padstack–Modify Design Padstack</td>
<td>padeditdb</td>
</tr>
<tr>
<td>Tools–Padstack–Modify Library Padstack</td>
<td>padeditlib</td>
</tr>
<tr>
<td>Tools–Padstack–Replace</td>
<td>replace padstack</td>
</tr>
<tr>
<td>Tools–Padstack–Group Edit</td>
<td>multipadedit</td>
</tr>
<tr>
<td>Tools–Padstack–Refresh</td>
<td>refresh padstack</td>
</tr>
<tr>
<td>Tools–Pad–Boundary</td>
<td>editpad boundary</td>
</tr>
<tr>
<td>Tools–Pad–Restore</td>
<td>editpad restore</td>
</tr>
<tr>
<td>Tools–Pad–Restore ALL</td>
<td>editpad restore all</td>
</tr>
<tr>
<td>Tools–Silkscreen</td>
<td>silkscreen param</td>
</tr>
<tr>
<td>Tools–Derive Connectivity</td>
<td>derive connectivity</td>
</tr>
<tr>
<td>Tools–Reports</td>
<td>reports</td>
</tr>
<tr>
<td>Tools–Technology File Compare</td>
<td>techfile compare</td>
</tr>
<tr>
<td>Tools–Setup Advisor</td>
<td>setup advisor</td>
</tr>
<tr>
<td>Tools–Database Check</td>
<td>dbcheck</td>
</tr>
<tr>
<td>Tools–Update DRC</td>
<td>drc update</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td></td>
</tr>
<tr>
<td>Help–Allegro Help</td>
<td>help</td>
</tr>
<tr>
<td>Help–Product Notes</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Known Problems and Solutions</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–Sourcelink</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–Education</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–pcb.cadence.com</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Manuals</td>
<td>cdsdoc</td>
</tr>
</tbody>
</table>
The command mapping in the following table lists the menu selections that can be found in the different modes (Layout, Shape, and Symbol) of Advanced Package Designer.

<table>
<thead>
<tr>
<th>Menu Selection</th>
<th>Console Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help–Design Flow</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–About Allegro</td>
<td>No corresponding command</td>
</tr>
</tbody>
</table>

**APD**

**Menu Selection** | **Console Command**
--- | ---
File–New | new
File–Open | open
File–Save | save
File–Save As | save_as
File–Create Symbol | create symbol (in Symbol Editor only)
File–Import–Logic | netin param
File–Import–Artwork | load photoplot
File–Import–Stream | load stream
File–Import–IPF | load plot
dx in
File–Import–DXF | idf in
File–Import–IFF | iff in
File–Import–SPECCTRA | specctra_in
File–Import–Redac | redac in
File–Import–Visula | visula in
File–Import–PADS | pads in
File–Import–PCAD | pcad in
File–Import–Sub-Drawing | clppaste
## Allegro/APD Design Guide: Getting Started

### Command Mapping

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File–Import–Techfile</td>
<td>techfile in</td>
</tr>
<tr>
<td>File–Import–Active Times</td>
<td>signal atimes</td>
</tr>
<tr>
<td>File–Import–Placement</td>
<td>plctxt in</td>
</tr>
<tr>
<td>File–Import–Paste Resistor</td>
<td>film res</td>
</tr>
<tr>
<td>File–Import–Annotations</td>
<td>annotation in</td>
</tr>
<tr>
<td>File–Export–BGA/Die (ASCII) Wizard</td>
<td>package data out</td>
</tr>
<tr>
<td>File–Export–Logic</td>
<td>feedback</td>
</tr>
<tr>
<td>File–Export–Netlist w/Properties</td>
<td>netout</td>
</tr>
<tr>
<td>File–Export–IPF</td>
<td>create plot</td>
</tr>
<tr>
<td>File–Export–DXF</td>
<td>dxf out</td>
</tr>
<tr>
<td>File–Export–IDF</td>
<td>idf out</td>
</tr>
<tr>
<td>File–Export–SPECCTRA</td>
<td>specttra_out</td>
</tr>
<tr>
<td>File–Export–Sub-Drawing</td>
<td>clpcopy</td>
</tr>
<tr>
<td>File–Export–Libraries</td>
<td>dlib</td>
</tr>
<tr>
<td>File–Export–Techfile</td>
<td>techfile out</td>
</tr>
<tr>
<td>File–Export–Placement</td>
<td>plctxt out</td>
</tr>
<tr>
<td>File–Export–Board Level Component</td>
<td>allegro_component</td>
</tr>
<tr>
<td>File–Export–Annotations</td>
<td>annotation out</td>
</tr>
<tr>
<td>File–Export–Valor ODB ++ inside</td>
<td>odb_out</td>
</tr>
<tr>
<td>File–Export–Save design to 14.0</td>
<td>downrev</td>
</tr>
<tr>
<td>File–Viewlog</td>
<td>viewlog</td>
</tr>
<tr>
<td>File–File Viewer</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>File–Plot Setup</td>
<td>plot setup</td>
</tr>
<tr>
<td>File–Plot Preview (Windows NT only)</td>
<td>plot preview</td>
</tr>
<tr>
<td>File–Plot</td>
<td>plot</td>
</tr>
<tr>
<td>File–Properties</td>
<td>file_property</td>
</tr>
<tr>
<td>File–Change Editor</td>
<td>toolswap</td>
</tr>
<tr>
<td>File–Script</td>
<td>script</td>
</tr>
<tr>
<td>File–Exit</td>
<td>exit</td>
</tr>
</tbody>
</table>
### Edit

<table>
<thead>
<tr>
<th>Command Mapping</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit–Move</td>
<td>move</td>
</tr>
<tr>
<td>Edit–Copy</td>
<td>copy</td>
</tr>
<tr>
<td>Edit–Mirror</td>
<td>mirror</td>
</tr>
<tr>
<td>Edit–Spin</td>
<td>spin</td>
</tr>
<tr>
<td>Edit–Change</td>
<td>change</td>
</tr>
<tr>
<td>Edit–Delete</td>
<td>delete</td>
</tr>
<tr>
<td>Edit–Shape</td>
<td>shape edit</td>
</tr>
<tr>
<td>Edit–Z-Copy</td>
<td>zcopy shape</td>
</tr>
<tr>
<td>Edit–Delete Unconnected Shapes</td>
<td>delete unconnected</td>
</tr>
<tr>
<td>Edit–Split Plane–Parameters</td>
<td>split plane params</td>
</tr>
<tr>
<td>Edit–Split Plane–Create</td>
<td>split plane create</td>
</tr>
<tr>
<td>Edit–Split Plane–Locate Islands</td>
<td>locate islands</td>
</tr>
<tr>
<td>Edit–Compose Shape</td>
<td>compose shape</td>
</tr>
<tr>
<td>Edit–Decompose Shape</td>
<td>decompose shape</td>
</tr>
<tr>
<td>Edit–Vertex</td>
<td>vertex</td>
</tr>
<tr>
<td>Edit–Delete Vertex</td>
<td>delete vertex</td>
</tr>
<tr>
<td>Edit–Boundary (Shape editor only)</td>
<td>boundary</td>
</tr>
<tr>
<td>Edit–Change Net (Pick) (Shape editor only)</td>
<td>changenet pick</td>
</tr>
<tr>
<td>Edit–Change Net (Name) (Shape editor only)</td>
<td>changenet name</td>
</tr>
<tr>
<td>Edit–Text</td>
<td>text edit</td>
</tr>
<tr>
<td>Edit–Groups</td>
<td>groupedit</td>
</tr>
<tr>
<td>Edit–Properties</td>
<td>property edit</td>
</tr>
<tr>
<td>Edit–Wirebond–Unlock Wirebond</td>
<td>unlock wirebond</td>
</tr>
<tr>
<td>Edit–Wirebond–Swap Wirebond</td>
<td>swap wirebond</td>
</tr>
<tr>
<td>Edit–Wirebond–Shift Wirebond</td>
<td>shift wirebond</td>
</tr>
<tr>
<td>Edit–Wirebond–Assign Wirebond</td>
<td>auto assign wirebond</td>
</tr>
<tr>
<td>Edit–Wirebond–Re-Align Bond Pads</td>
<td>realign bond pads</td>
</tr>
<tr>
<td>Edit–BGA</td>
<td>bga editor</td>
</tr>
</tbody>
</table>
# Allegro/APD Design Guide: Getting Started

## Command Mapping

### View

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View–Zoom By Points</td>
<td>zoom points</td>
</tr>
<tr>
<td>View–Zoom Fit</td>
<td>zoom fit</td>
</tr>
<tr>
<td>View–Zoom In</td>
<td>zoom in</td>
</tr>
<tr>
<td>View–Zoom Out</td>
<td>zoom out</td>
</tr>
<tr>
<td>View–Zoom World</td>
<td>zoom center</td>
</tr>
<tr>
<td>View–Zoom Center</td>
<td>zoom center</td>
</tr>
<tr>
<td>View–Zoom Previous</td>
<td>zoom previous</td>
</tr>
<tr>
<td>View–Color View Save</td>
<td>colorview create</td>
</tr>
<tr>
<td>View–Color View Restore Last</td>
<td>colorview restore</td>
</tr>
<tr>
<td>View–Refresh</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>View–Customization–Display</td>
<td>display param</td>
</tr>
<tr>
<td>View–Customization–Toolbar</td>
<td>No corresponding command</td>
</tr>
</tbody>
</table>

### Add

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add–Line</td>
<td>add line</td>
</tr>
<tr>
<td>Add–Arc w/Radius</td>
<td>add rarc</td>
</tr>
<tr>
<td>Add–3pt Arc</td>
<td>add arc</td>
</tr>
<tr>
<td>Add–Circle</td>
<td>add circle</td>
</tr>
<tr>
<td>Add–Rectangle</td>
<td>add rect</td>
</tr>
<tr>
<td>Add–Frectangle</td>
<td>add frect</td>
</tr>
<tr>
<td>Add–Text</td>
<td>add text</td>
</tr>
<tr>
<td>Add–Shapes–Solid Fill</td>
<td>add fshape</td>
</tr>
<tr>
<td>Add–Shapes–Unfilled</td>
<td>add ufshape</td>
</tr>
<tr>
<td>Add–Shapes–Cross Hatch Fill</td>
<td>add xshape</td>
</tr>
</tbody>
</table>

### Display

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display–Color/Visibility</td>
<td>color</td>
</tr>
<tr>
<td>Display–Color Priority</td>
<td>color priority</td>
</tr>
<tr>
<td>Display–Element</td>
<td>show element</td>
</tr>
<tr>
<td>Display–Measure</td>
<td>show measure</td>
</tr>
</tbody>
</table>
## Command Mapping

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display–Parasitic</td>
<td>show parasitic</td>
</tr>
<tr>
<td>Display–Property</td>
<td>show property</td>
</tr>
<tr>
<td>Display–Highlight</td>
<td>hilight</td>
</tr>
<tr>
<td>Display–Dehighlight</td>
<td>dehilight</td>
</tr>
<tr>
<td>Display–Show Rats–All</td>
<td>rats all</td>
</tr>
<tr>
<td>Display–Show Rats–Components</td>
<td>rats component</td>
</tr>
<tr>
<td>Display–Show Rats–Net</td>
<td>rats net</td>
</tr>
<tr>
<td>Display–Blank Rats–All</td>
<td>unrats all</td>
</tr>
<tr>
<td>Display–Blank Rats–Components</td>
<td>unrats component</td>
</tr>
<tr>
<td>Display–Blank Rats–Nets</td>
<td>unrats net</td>
</tr>
</tbody>
</table>

### Setup

<table>
<thead>
<tr>
<th>Setup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup–Drawing Size</td>
<td>drawing param</td>
</tr>
<tr>
<td>Setup–Drawing Options</td>
<td>status</td>
</tr>
<tr>
<td>Setup–Text Sizes</td>
<td>define text</td>
</tr>
<tr>
<td>Setup–Grids</td>
<td>define grid</td>
</tr>
<tr>
<td>Setup–Subclasses</td>
<td>define subclass</td>
</tr>
<tr>
<td>Setup–Cross-section</td>
<td>define xsection</td>
</tr>
<tr>
<td>Setup–Vias–Define B/B Via</td>
<td>define bbvia</td>
</tr>
<tr>
<td>Setup–Vias–Auto Define B/B Via</td>
<td>auto define bbvia</td>
</tr>
<tr>
<td>Setup–Constraints</td>
<td>cns</td>
</tr>
<tr>
<td>Setup–Electrical Constraint Spreadsheet</td>
<td>cmgr</td>
</tr>
<tr>
<td>Setup–Property Definitions</td>
<td>define property</td>
</tr>
<tr>
<td>Setup–Define Lists</td>
<td>define list</td>
</tr>
<tr>
<td>Setup–Areas–Part Keepin</td>
<td>keepin package</td>
</tr>
<tr>
<td>Setup–Areas–Part Keepout</td>
<td>keepout package</td>
</tr>
<tr>
<td>Setup–Areas–Part Height</td>
<td>package_height</td>
</tr>
<tr>
<td>Setup–Areas–Route Keepin</td>
<td>keepin router</td>
</tr>
<tr>
<td>Setup–Areas–Route Keepout</td>
<td>keepout router</td>
</tr>
<tr>
<td>Setup–Areas–Via Keepout</td>
<td>keepout via</td>
</tr>
</tbody>
</table>
Allegro/APD Design Guide: Getting Started
Command Mapping

Setup–Areas–Probe Keepout  keepout probe
Setup–Areas–Gloss Keepout  keepout gloss
Setup–Areas–Photoplot Outline  keepin photo

**Layout**

Layout menu selections are available only in the Symbol Editor

Layout–Pins  add pin
Layout–Connections  add connect
Layout–Slide  slide
Layout–Labels–RefDes  label refdes
Layout–Labels–Device  label device
Layout–Labels–Value  label value
Layout–Labels–Tolerance  label tolerance
Layout–Labels–Part Number  label part

**Void**

Void menu selections are available only in the Shape Editor

Void–Shape  void shape
Void–Circle  void circle
Void–Element  void element
Void–Auto  void all

**Shape**

Shape menu selections are available only in the Shape Editor

Shape–Parameters  shape param
Shape–Check  shape check
Shape–Fill  shape fill

**Logic**

Logic–Auto Create Net  auto create net
Logic–Create Net  create net
Logic–Auto Assign Net  auto assign net
Logic–Assign Net  assign net
<table>
<thead>
<tr>
<th>Command Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logic</strong>–Desassign Net</td>
</tr>
<tr>
<td>Logic–Derive Assignment</td>
</tr>
<tr>
<td>Logic–Identify DC Nets</td>
</tr>
<tr>
<td>Logic–Net Schedule</td>
</tr>
<tr>
<td>Logic–Assign Differential Pair</td>
</tr>
<tr>
<td>Logic–Assign RefDes</td>
</tr>
<tr>
<td>Logic–Auto Rename RefDes–Rename</td>
</tr>
<tr>
<td>Logic–Auto Rename RefDes–Design</td>
</tr>
<tr>
<td>Logic–Auto Rename RefDes–Room</td>
</tr>
<tr>
<td>Logic–Auto Rename RefDes–Window</td>
</tr>
<tr>
<td>Logic–Auto Rename RefDes–List</td>
</tr>
<tr>
<td>Logic–Edit Parts List</td>
</tr>
<tr>
<td>Logic–Terminator Assignment</td>
</tr>
</tbody>
</table>

**Place**

<table>
<thead>
<tr>
<th>Command Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place–Manually</td>
</tr>
<tr>
<td>Place–SPECCTRA</td>
</tr>
<tr>
<td>Place–Autoplace–Insight</td>
</tr>
<tr>
<td>Place–Autoplace–Parameters</td>
</tr>
<tr>
<td>Place–Autoplace–Top Grids</td>
</tr>
<tr>
<td>Place–Autoplace–Bottom Grids</td>
</tr>
<tr>
<td>Place–Autoplace–Design</td>
</tr>
<tr>
<td>Place–Autoplace–Room</td>
</tr>
<tr>
<td>Place–Autoplace–Window</td>
</tr>
<tr>
<td>Place–Autoplace–List</td>
</tr>
<tr>
<td>Place–Interactive</td>
</tr>
<tr>
<td>Place–Swap–Pins</td>
</tr>
<tr>
<td>Place–Swap–Functions</td>
</tr>
<tr>
<td>Place–Swap–Components</td>
</tr>
<tr>
<td>Place–Autoswap–Parameters</td>
</tr>
</tbody>
</table>
## Allegro/APD Design Guide: Getting Started

### Command Mapping

<table>
<thead>
<tr>
<th>Command Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place–Autoswap–Design</td>
<td>swap area design</td>
</tr>
<tr>
<td>Place–Autoswap–Room</td>
<td>swap area room</td>
</tr>
<tr>
<td>Place–Autoswap–Window</td>
<td>swap area window</td>
</tr>
<tr>
<td>Place–Autoswap–List</td>
<td>swap area list</td>
</tr>
<tr>
<td>Place–Evaluate–Parameters</td>
<td>eval param</td>
</tr>
<tr>
<td>Place–Evaluate–Design</td>
<td>eval area design</td>
</tr>
<tr>
<td>Place–Evaluate–Room</td>
<td>eval area room</td>
</tr>
<tr>
<td>Place–Evaluate–Window</td>
<td>eval area window</td>
</tr>
<tr>
<td>Place–Evaluate–List</td>
<td>eval area list</td>
</tr>
<tr>
<td>Place–Update Symbols</td>
<td>refresh symbol</td>
</tr>
<tr>
<td>Place–Replace SQ Temporary–Devices</td>
<td>replace temp_device</td>
</tr>
<tr>
<td>Place–Replace SQ Temporary–Symbols</td>
<td>replace temp_symbols</td>
</tr>
</tbody>
</table>

### Route

<table>
<thead>
<tr>
<th>Command Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route–Wirebond</td>
<td>add wirebond</td>
</tr>
<tr>
<td>Route–Connect</td>
<td>add connect</td>
</tr>
<tr>
<td>Route–Slide</td>
<td>slide</td>
</tr>
<tr>
<td>Route–Spider Router</td>
<td>spider router</td>
</tr>
<tr>
<td>Route–Custom Smooth</td>
<td>custom smooth</td>
</tr>
<tr>
<td>Route–Define Short</td>
<td>define shorting scheme</td>
</tr>
<tr>
<td>Route–Create Short</td>
<td>create short</td>
</tr>
<tr>
<td>Route–ZRouter</td>
<td>zrouter</td>
</tr>
<tr>
<td>Route–SPECCTRA–Route Radial</td>
<td>radial router</td>
</tr>
<tr>
<td>Route–SPECCTRA–Run Router Checks</td>
<td>spectra checks</td>
</tr>
<tr>
<td>Route–SPECCTRA–Route by Pick</td>
<td>route_by_pick</td>
</tr>
<tr>
<td>Route–SPECCTRA–Route Automatic</td>
<td>auto_route</td>
</tr>
<tr>
<td>Route–SPECCTRA–Interactive Editor</td>
<td>spectra</td>
</tr>
<tr>
<td>Route–Add Fillet</td>
<td>add fillet</td>
</tr>
<tr>
<td>Route–Delete Fillet</td>
<td>delete fillet</td>
</tr>
<tr>
<td>Route–Gloss–Parameters</td>
<td>gloss param</td>
</tr>
</tbody>
</table>
## Allegro/APD Design Guide: Getting Started
### Command Mapping

| Route–Gloss–Design          | gloss area design         |
| Route–Gloss–Room            | gloss area room           |
| Route–Gloss–Window          | gloss area window         |
| Route–Gloss–Highlight       | No corresponding command  |
| Route–Gloss–List            | gloss area list           |
| Route–Testprep–Auto         | testpreop param           |
| Route–Testprep–Create Probe | probe create              |
| Route–Testprep–Delete Probe | probe delete              |
| Route–Testprep–Swap Probe   | probe swap                |
| Route–Testprep–NC Tape Probes | nctape                 |

### Analyze

| Analyze–SI/EMI Sim–Initialize | signal init               |
| Analyze–SI/EMI Sim–Library    | signal library            |
| Analyze–SI/EMI Sim–Model      | signal model              |
| Analyze–SI/EMI Sim–Model Dump/Refresh | signal model refresh |
| Analyze–SI/EMI Sim–Preferences | signal prefs             |
| Analyze–SI/EMI Sim–Audit–Design Audit | signal audit     |
| Analyze–SI/EMI Sim–Audit–Net Audit | signal audit net         |
| Analyze–SI/EMI Sim–Audit–Audit One Library | signal lib audit |
| Analyze–SI/EMI Sim–Audit–Audit List of Libraries | signal libs audit   |
| Analyze–SI/EMI Sim–Probe      | signal probe              |
| Analyze–SI/EMI Sim–Xtalk Table | signal xtalktable         |
| Analyze–EMC–Initialize        | signal eminit             |
| Analyze–EMC–Auto Setup        | signal emiautopropmain    |
| Analyze–EMC–Manual Setup      | signal emimanualpropmain  |
| Analyze–EMC–Rule Select       | signal emiruleselect      |
Analyze–EMC–Audit          signal emiverify
Analyze–EMC–Execute         signal emiexecute
Analyze–EMC–Results         signal emiresults
Analyze–EMC–Audit Report    signal emiverifiyreport
Analyze–EMC–Execute Report  signal emiexecutereport
Analyze–PNC Interface       pnc
Analyze–Package Model       package model

**Manufacture**

Manufacture–Create Plating Bar pbar create
Manufacture–Delete Plating Bar pbar delete
Manufacture–Plating Bar check pbar check

Manufacture–Dimension/Draft commands in the Layout Editor
are accessed under the Dimension menu item in the Symbol Editor

Manufacture–Dimension/Draft–Parameters draft param
Manufacture–Dimension/Draft–LineFont linefont
Manufacture–Dimension/Draft–Linear Dim dimension linear
Manufacture–Dimension/Draft–Datum Dim dimension datum
Manufacture–Dimension/Draft–Angular Dim dimension angular
Manufacture–Dimension/Draft–Leader Lines leader only
Manufacture–Dimension/Draft–Diametral Leader leader diametral
Manufacture–Dimension/Draft–Radial Leader leader radial
Manufacture–Dimension/Draft–Balloon Leader leader balloon
Manufacture–Dimension/Draft–Chamfer Leader leader chamfer
Manufacture–Dimension/Draft–Chamfer draft chamfer
Manufacture–Dimension/Draft–Fillet draft fillet
Manufacture–Dimension/Draft–Create Detail create detail
<table>
<thead>
<tr>
<th>Command Mapping</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacture–Stream Out</td>
<td>stream out</td>
</tr>
<tr>
<td>Manufacture–Artwork</td>
<td>film param</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Parameters</td>
<td>ncdrill param</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Legend</td>
<td>ncdrill legend</td>
</tr>
<tr>
<td>Manufacture–NC–Drill Tape</td>
<td>nctape_full</td>
</tr>
<tr>
<td>Manufacture–NC–Route</td>
<td>ncroute</td>
</tr>
<tr>
<td>Manufacture–DFA Check</td>
<td>dfa</td>
</tr>
<tr>
<td>Manufacture–Create Coupons</td>
<td>create coupons</td>
</tr>
<tr>
<td>Manufacture–Silkscreen</td>
<td>silkscreen param</td>
</tr>
<tr>
<td>Manufacture–Documentation–BondPad Text</td>
<td>bpa</td>
</tr>
<tr>
<td>Manufacture–Documentation–BondPad Display–Highlight</td>
<td>bondpad hilite</td>
</tr>
<tr>
<td>Manufacture–Documentation—BondPad Display–Dehighlight</td>
<td>bondpad dehilite</td>
</tr>
<tr>
<td>Manufacture–Documentation–Display Pin Text</td>
<td>dpn</td>
</tr>
<tr>
<td>Manufacture–Documentation–Package Report–Sorted by Die Pin</td>
<td>dbp_report die</td>
</tr>
</tbody>
</table>

**Tools**

<table>
<thead>
<tr>
<th>Tools</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools–Create Module</td>
<td></td>
</tr>
<tr>
<td>Tools–Padstack–Modify Design Padstack</td>
<td>padeditdb</td>
</tr>
<tr>
<td>Tools–Padstack–Modify Library Padstack</td>
<td>padeditlib</td>
</tr>
<tr>
<td>Tools–Padstack–Replace</td>
<td>replace padstack</td>
</tr>
<tr>
<td>Tools–Padstack–Group Edit</td>
<td>multipadedit</td>
</tr>
<tr>
<td>Tools–Padstack–Refresh</td>
<td>refresh padstack</td>
</tr>
<tr>
<td>Tools–Pad–Boundary</td>
<td>editpad boundary</td>
</tr>
<tr>
<td>Command</td>
<td>Equivalent Command</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Tools–Pad–Restore</td>
<td>editpad restore</td>
</tr>
<tr>
<td>Tools–Pad–Restore ALL</td>
<td>editpad restore all</td>
</tr>
<tr>
<td>Tools–Derive Connectivity</td>
<td>derive connectivity</td>
</tr>
<tr>
<td>Tools–Reports</td>
<td>reports</td>
</tr>
<tr>
<td>Tools–Technology File Compare</td>
<td>techfile compare</td>
</tr>
<tr>
<td>Tools–Setup Advisor</td>
<td>setup advisor</td>
</tr>
<tr>
<td>Tools–Database Check</td>
<td>dbcheck</td>
</tr>
<tr>
<td>Tools–Update DRC</td>
<td>drc update</td>
</tr>
<tr>
<td>Help–APD Help</td>
<td>help</td>
</tr>
<tr>
<td>Help–Product Notes</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Known Problems and Solutions</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–Sourcelink</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–Education</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Web Resources–pcb.cadence.com</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Manuals</td>
<td>cdsdoc</td>
</tr>
<tr>
<td>Help–Design Flow</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–Sourcelink</td>
<td>No corresponding command</td>
</tr>
<tr>
<td>Help–About APD</td>
<td>No corresponding command</td>
</tr>
</tbody>
</table>
Allegro/APD Configuration Guide

This appendix explains how to configure Allegro/APD after you install the Cadence software. It also includes information on system requirements. For additional information, see the Product Notes section of your PCB Systems Division documentation.

This chapter covers the following:

- **System Requirements and Performance Enhancements**
  - Operating Systems Requirements
  - Hardware and Operating System Requirements

- **Calculating Memory Requirements**

- **Improving Performance (UNIX)**

- **UNIX-Based Installation Directory Information and Troubleshooting**
  - Files That Reference the Installation Directory
  - Checking File References to the Installation Directory
  - Automatically Correcting Installation Directory References

- **NT-Based Installation Directory Information**

- **Licensing Issues**

- **Compatibility for Allegro/APD Libraries, Designs, and Scripts**
  - Symbol Library and Padstacks
  - Database Compatibility across Platforms
  - Database Compatibility with Previous Software Releases
  - Database UPREV (DBDoctor)
  - Script Compatibility
  - SKILL Compatibility
IBM DFS
System Requirements and Performance Enhancements

This section describes the system requirements for Allegro/APD, including

- Operating systems for each platform
- Disk space
- Virtual and physical memory
- How to improve graphics performance

Operating Systems Requirements

<table>
<thead>
<tr>
<th>Platform</th>
<th>WINDOWS NT</th>
<th>Solaris</th>
<th>HP7000</th>
<th>IBM RS6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Pentium processor or higher</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform ID</td>
<td>wint</td>
<td>sun4v</td>
<td>hppa</td>
<td>ibmrs</td>
</tr>
<tr>
<td>Operating System</td>
<td>NT 4.0, Service Pack 6a, and Windows 2000</td>
<td>Solaris 7 and 8</td>
<td>HP-UX 11.00</td>
<td>AIX 4.3.3</td>
</tr>
</tbody>
</table>

Hardware and Operating System Requirements

HP-UX

- HP7000 workstation
  - PA7000 or PA1.1 or PA1.2
- HP-UX 11.0 or 11.11 OS
- CDE
  - When you run CDE, Cadence recommends that the Save Under option with the X server be enabled. To do this:
    1. Login as root.
    2. Edit /usr/lpp/X11/defaults/xserverrc
3. Change the following line:

```
EXTENSIVE =" "
to read
EXTENSIVE =" -bs"
```

4. Reboot your machine.

- 256 MB (or greater) system memory
- 300 MB swap space
- 2 GB (or greater) available disk space
- 24-color display recommended

**IBM/AIX**

- RS6000-based UNIX workstation
- AIX 4.3.3 operating system
- 256 MB (or greater) system memory
- 300 MB swap space
- 3 GB (or greater) available disk space
- 24-bit color display (recommended)
- Backingstore needs to be enabled in the following manner:
  1. Login as root.
  2. Edit `/usr/lpp/X11/defaults/xserverrc`
  3. Change the following line:

```
EXTENSIVE =" "
to read
EXTENSIVE =" -bs"
```

4. Reboot your machine.

**Solaris/SunOS**

- Sun ULTRA 1 or better
Solaris 7 or 8

CDE

1. Login as root.
2. Edit /usr/lpp/X11/defaults/xserverrc
3. Change the following line:
   ```
   EXTENSIVE =" "
   to read
   EXTENSIVE =" -bs"
   ```
4. Reboot your machine.

- 256 MB (or greater) system memory
- 300 MB swap space
- 2 GB (or greater) available disk space
- 24-bit color display (recommended)

For a list of required patches, see `<install_dir>/tools/pcb/ospatches/sun4v/README.cdn`

Windows NT

Allegro/APD NT products run on PCs with Intel processors and Microsoft Windows NT 4.0. Because Allegro/APD is integrated directly with Windows NT, hardware and peripherals supported by Windows NT are also supported by Allegro/APD. A list of hardware and peripherals officially supported by Windows NT can be obtained from the Microsoft web page.

Windows Operating System Requirements

Allegro/APD products run on Microsoft Windows NT 4.0 SP6a, and Windows 2000 (any service pack). Other Windows versions, such as Windows 3.1, Windows for Workgroups, Windows 95 and 98, and Windows XP are not supported.

Note: As a general rule, directories into which you install Cadence products should not have spaces in their names (for example, a directory name should be `cdsapd`, not `cds apd`). Spaces in directory names could cause problems in generating reports, particularly on the Windows 2000 platform.
Minimum Hardware Requirements

The following list contains the minimum hardware requirements for Allegro/APD on a personal computer. These requirements are for small to medium-size designs.

- Pentium ii 400MHz processor
- 256 MB physical memory
- 600 MB disk space
- 300 MB swap space
- Three-button Microsoft-compatible mouse
- Ethernet card (for network communications and security hostID)
- CD-ROM drive
- 800 x 600 SVGA color monitor (32768 minimum)

Recommended Hardware Configuration

The following list contains the recommended hardware requirements for Allegro/APD on a personal computer. These requirements are for large designs.

- Pentium ii 400MHz (or faster) processor
- 128 MB (or greater) physical memory
- 1 GB available disk space
- 300 MB swap space
- Three-button Microsoft-compatible mouse
- Ethernet card (for network communications and security hostID)
- CD-ROM drive
- 1024 x 768 SVGA color monitor or greater (32768 colors)

Calculating Memory Requirements

Your system must have adequate virtual memory (swap space) and physical memory (RAM) to run Allegro/APD. The complexity of designs requires that estimates for memory be based on more than the number of symbols being used.
Table A-1 is a worksheet to estimate Allegro/APD virtual memory requirements. Table A-2 is a worksheet to estimate physical memory requirements. Use this information as a guideline for judging how much total memory you need. All values are in megabytes (MB).

### Table A-1 Allegro/APD Virtual Memory Requirements

<table>
<thead>
<tr>
<th>Items That Require Virtual Memory</th>
<th>Item MB</th>
<th>Total MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Base Allegro/APD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Typical database size (disk size)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Total base memory (add lines 1 + 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Gloss or Testprep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4a. Grid density: &lt;min grid size&gt; x &lt;min grid size&gt; =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4b. &lt;4a&gt; x &lt;router keepin area&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4c. &lt;4b&gt; x &lt;number of layers&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5d. Memory: (&lt;4c&gt; x 16)/ 1,000,000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you are running one or the other, enter the larger value. If you are running both simultaneously, enter the sum of 4d and 5d.

| 7. SigNoise                       |         |          |
| 7a. Base                          | 8       |          |
| 7b. Library (80 MB for 11.5 Allegro/APD) |         |          |
| 7c. Total (add lines 7a + 7b)     |         |          |

8. If you plan never to run Gloss, Testprep, or SigNoise, add 4 MB

Grand total, Allegro/APD Virtual Memory (add lines 3 + 6 + 7c + 8)

The Allegro/APD virtual memory requirement does not determine the overall virtual memory needed by your system. The overall virtual memory requirement also depends on the resources required by your window environment plus any other tools you plan to run simultaneous with Allegro/APD. For the simplest case—no tools other than Allegro/APD and your window system—use Allegro/APD’s virtual memory requirements and add 32 MBs to determine your system’s virtual memory.
(line 8 in Table A-1) _____ + 32 = _____ (systems virtual memory)

If you require addition swap space in the future, please consult your hardware vendor systems administrator or documentation for swap space modification instructions.

**Table A-2 Allegro/APD Physical Memory Requirements**

<table>
<thead>
<tr>
<th>Items That Require Physical Memory</th>
<th>Item MB</th>
<th>Total MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Base OS</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>2. Base Allegro/APD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2a. Virtual Memory Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(line 10 in Table A-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2b. Divide the value for line 2c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>by 2; (line 3c / 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical memory requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(add lines 1 + 2b)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Improving Performance (UNIX)**

You may be able to greatly enhance your graphics performance on certain platforms if you run both X and Allegro/APD on the same platform.

**To run X and Allegro/APD on the same platform**

➤ Set the display variable to its local mode.

        This lets the X protocol to use shared memory instead of expensive TCP/IP transport.

**To run X and Allegro/APD on the same platform**

➤ At the system prompt, type

        setenv DISPLAY :0

        Not all platform vendors support this performance enhancement. On IBM systems, you will notice substantial performance increase.
UNIX-Based Installation Directory Information and Troubleshooting

After you install Cadence software, configuration files are modified to reference the installation directories you specified during installation. This section explains the files that reference installation directories include:

- How to verify the files are edited correctly
- A sample `cshrc` file
- A sample `profile` file
- How to automatically edit incorrect references

Files That Reference the Installation Directory

When you install Allegro/APD using `softload`, the following files are automatically edited to reference the installation directory you specified during `softload`:

- `<install_dir>/tools/pcb/bin/cshrc`
- `<install_dir>/tools/pcb/bin/profile`

where `<install_dir>` is the actual installation directory you specified during `softload`. For example, `<install_dir>` could be `/usr/cds`, as in the following example:

```
/usr/cds/tools/pcb/bin/cshrc
```

Checking File References to the Installation Directory

If you have problems running Allegro/APD after installation, verify that the following variables correctly reference the installation directory:

- In the `<install_dir>/tools/pcb/bin/cshrc` file
  ```
  set CDS_ROOT = `<install_dir>`
  ```
- In the `<install_dir>/tools/pcb/bin/profile` file
  ```
  CDS_ROOT=<`install_dir>`
  ```
In the `<install_dir>/tools/pcb/text/Allegro/APDHelp` file, many variables reference the installation directory.

If the `cshrc`, `profile`, and `Allegro/APDHelp` files contain incorrect references to the installation directory, you can edit them individually, or run the `setup_Allegro/APD` script to automatically correct all three files, as described in the following section.

Automatically Correcting Installation Directory References

During installation, the `setup_Allegro/APD` script saves the original versions of the `cshrc`, `profile`, and `Allegro/APDHelp` files in their respective directories. These original files have the extension `.fcs`

- `<install_dir>/tools/pcb/bin/cshrc.fcs`
- `<install_dir>/tools/pcb/text/profile.fcs`
- `<install_dir>/tools/pcb/text/Allegro/APDHelp.fcs`

The `setup_Allegro/APD` script edits these original versions to create the following files:

- `tools/pcb/bin/cshrc`
- `tools/pcb/text/profile`
- `tools/pcb/text/Allegro/APDHelp`

Do not edit the original `.fcs` files. The `setup_Allegro/APD` script will not run correctly if these files have been modified. You can, however, edit the resulting `cshrc`, `profile` or `Allegro/APDHelp` files.

To use `setup_Allegro/APD` to place the correct `install_dir` directory references in the `cshrc`, `profile` and `Allegro/APDHelp` files

1. Change directories to the installation directory:
   ```
   cd <install_dir>
   ```
   The installation directory is the directory immediately above the `tools` directory.

2. Type
   ```
   tools/pcb/bin/setup_Allegro/APD <platform> [<path>]
   ```
   platform

   Is one of the following platform specifications:
path

Is the path to your specified install directory, *install_dir*. This enables designers to run Allegro/APD from a directory other than the one where Allegro/APD is installed. If not specified, *install_dir* defaults to the current working directory.

For example, if your administrator installed Allegro/APD on

```
AServer:/cds/9504
```

and you have NFS mounted the installation directory on your workstation as

```
Yourworkstation:/usr/cds
```

the paths in the installation directory will reference the path `/cds/9402` rather than your local path of `/usr/cds`. Portions of the Allegro/APD software will not run correctly unless your local path to the software matches the path specified in the configuration files.

There are several possible resolutions to this:

- Mount the server directories on your workstation just as they are installed on the server.

  ```
  mount AServer:/cds/9504 /cds/9504
  ```

- Create a symbolic link on your workstation that points from the installation directory to your local NFS mount point.

  ```
  ln -s /usr/cds /cds/9504
  ```

- Have your administrator modify the configuration files on the server to match the directory paths on your workstation.

  On the host where the software is installed (AServer), change directories to `/cds/9504` and enter the following command:

  ```
  tools/pcb/bin/setup_Allegro/APD sun4w /usr/cds/<install_dir>
  ```

The default working directory for the *install_dir* becomes `/usr/cds`.
Example cshrc File

The following is a sample of the tools/pcb/bin/cshrc file configured for an install_dir specified as /usr/cds.

```
set CDS_ROOT = /usr/cds
set TOOLSDIR = $CDS_ROOT/tools
setenv ALGROPATH $TOOLSDIR/pcb/bin
set path = ( $TOOLSDIR/bin \
             $ALGROPATH \
             $path )
```

Example profile File

To use a sample local profile file (~/.profile)

```
% At the Korn Shell command line, type
CDS_ROOT=/usr/cds
TOOLSDIR=$CDS_ROOT/tools
ALGROPATH=$TOOLSDIR/pcb/bin
PATH=$TOOLSDIR/bin:$ALGROPATH:$PATH;
export PATH
```

**Note:** Do not set the TELENV variable if you use the standard Cadence installation hierarchy.

Displaying UI Dialog Boxes Correctly

If the secondary (child) dialog boxes disappear behind the main UI of Allegro/APD, you need to modify your window manager to keep child windows on top.

- For HP and AIX
  
The typical window manager default configuration is
  
  `secondariesOnTop:True`

- For Solaris
  
The typical window manager default configuration is
  
  `secondariesOnTop:False`

If you run CDE, add the following to your ~/ .Xdefaults file

```
DTwm*secondariesOnTop:True
```
If you want to restrict this behavior to certain programs, add the following to your ~/.Xdefaults file

\[ DTwm*<program>*secondariesOnTop:True \]

For example:

\[ DTwm*Allegro*secondariesOnTop:True \]

Add an entry to the file for each program. When finished, restart your window manager.

**NT-Based Installation Directory Information**

The Allegro/APD directory structure for Windows NT is identical to that on UNIX platforms. However, NT does not install `chsrc` or `profile` files.

**Licensing Issues**

For detailed information on licensing Cadence products for Unix and Windows environments, see the installation manual that accompanies your CD-ROM.

**Compatibility for Allegro/APD Libraries, Designs, and Scripts**

**Symbol Library and Padstacks**

All library symbols created with previous releases of Allegro/APD are compatible with this release. Before loading the generic library, be sure that customized component symbols are not in the same directory in which you will load Allegro/APD.

\[ \text{Caution} \]

Your library will be overwritten when you load this Allegro/APD library. Your CAE libraries will not be affected. Before using any of the Allegro/APD library symbols, carefully review them to ensure that they meet your physical design criteria.
Database Compatibility across Platforms

Databases are compatible across all Allegro/APD configurations on all platforms. Allegro/APD uses the same database format for all versions of Allegro/APD on all platforms, so no conversion is necessary to move between them.

Database Compatibility with Previous Software Releases

Allegro/APD databases are backward compatible with their major version number (the number to the left of the dot). This means that databases created in or upreved to any revision within a major version (for example, to 14.1) can migrate between revisions of that version. You cannot save any major version to an earlier one, such as 14.x to 13.x, 13.x to 12.x, etc.

In version 14.2, use of certain features will prevent you from opening your design in 14.0. Specifically, POWER_AND_GROUND scheduling and insertion of dielectric layers between the air layer above TOP and the air layer below BOTTOM are not available in version 14.0. Attempting to open a design in 14.0 that uses this functionality will generate this error message in the tool’s command console:

W - ILLEGAL SYMBOL REVISION

and this message at your operating system prompt:

THIS DATABASE WAS SAVED WITH A NEWER VERSION OF 14.0. THIS VERSION OF 14.0 CANNOT OPEN THE DATABASE.

Therefore, you must run the downrev command from the newer revision while it is open. Doing so lets you then migrate the newer version backward by stripping out the feature(s) that the earlier revision does not support. Because downrev deletes the data that prevents you from opening the database in earlier releases, it is important that you do not over-write your 14.2 database.

Note: Output data derived from technology files in 14.2 designs are not downward compatible. Earlier versions cannot process the information from a 14.2 techfile.

Newer revisions of a design that do not employ functionality unsupported in earlier revisions can be migrated freely within the major version without using the downrev feature.

To use downrev:

1. From an open design, run downrev.

A dialog box appears in one of two configurations.

   a. If your design database does not require downrev to open in an earlier revision, the dialog box appears as shown in Figure A-1. If this configuration appears, close
the dialog box, and use the normal procedure for opening the database in the earlier revision.

**Figure A-1  Save Design (Configuration A) Dialog Box**

![Save Design to 14.0 Level](image1)

**b.** If your design database requires **downrev** to open in an earlier revision, the dialog box appears as shown in Figure A-2. Click the **Save design to 14.0** button in the dialog box to display the file browser, and go to step 2.

**Figure A-2  Save Design (Configuration B) Dialog Box**

![Save Design to 14.0 Level](image2)

2. Select or type in the file name you want to save the file to, as shown in Figure A-3.

**Note:** Selecting the same name as the current open design will over-write your 14.2-compatible design.
3. Click **Save**.

   The Save design to 14.0 level dialog box presents a completion message, as shown in Figure A-4.

**Figure A-4  Completion Dialog Box**

The message reminds you that the current open design retains the functionality that will be unsupported when you attempt to open the database in the earlier revision.
Database UPREV (DBDoctor)

Databases are automatically uprev'd from earlier Allegro/APD versions. The NT version of Allegro/APD does not have the ability to uprev designs created prior to Allegro/APD 10.0. Databases from earlier versions of Allegro/APD must be uprev'd on UNIX (to at least version 10.0) before they can be used in Allegro/APD on NT.

In version 14.2, databases more than one release removed can be upprevved to the current release by running DBDoctor. You can run DBDoctor from the user interface by typing:

- dbdoctor at the Allegro/APD command prompt
- dbdoctor <file_name> at your operating system prompt
- uprev_overwrite <file_name> at your operating system prompt

Script Compatibility

⚠️ Caution

*Cadence does not guarantee that scripts are 100% upwardly compatible from release to release.*

SKILL Compatibility

SKILL programs are fully compatible with the Allegro/APD environment and should run without modification.

IBM DFS

DFS is an IBM networking protocol. Before Allegro/APD 10.0, databases could become corrupted if the volume that contained the database became full during the database save. This corruption was because of a deficiency in DFS, not Allegro/APD.

Starting in Allegro/APD 10.0, Allegro/APD detects that a volume is DFS and takes appropriate action to ensure the database is written correctly. Unfortunately, this means that writing databases across the network will take about twice as long as in pre-10.0 versions of Allegro/APD because Allegro/APD verifies the data as it is written to disk. Database reads are not affected.
You can disable this DFS safety feature by setting the Allegro/APD environment variable `afs_nosync`. With this variable set, databases save as fast as in previous releases, but the file write safety check is not performed.

By default, this additional checking takes place automatically as long as your system uses the standard “/dfs” naming convention.
Glossary

A

**active devices**
Transistors or diodes that can change the basic character of a circuit.

**additive process**
A process that creates or screens-on a circuit by adding a conductor in a precise pattern.

**alias**
A user-defined abbreviation for a command. See [script files](#).

**allegro**
The Unix command that provides complete design functionality, including automatic placement, routing, post-processing, and third-party database translators.

**anti-pad**
A negative pad (clear, surrounded by black), usually a circle, to prevent the connection of a pin to an embedded metal layer.

**area optimization**
A Thick/Thin-Film Resistor Synthesizer command file directive that generates resistors with the smallest possible area.

**area resolution**
The process Allegro/APD uses to decide which constraint applies when two or more constraint areas overlap or an element such as a line extends over more than one constraint area.

Allegro/APD finds all possible constraints that may apply to a spacing or physical situation, then selects the most conservative constraint value.

**aspect ratio**
The ratio of the length to the width of a resistor. For example, if the length is 2 and the width is 1, the aspect ratio is 2:1 or 2 squares of resistance.
Automatic Placement
An Allegro/APD function that places components in a design, based on controls provided by the user.

Automatic Routing
An Allegro/APD function that automatically connects pins with etch.

autosave
Allegro/APD provides a built-in facility that regularly saves an active design or symbol. You must activate the autosave utility in Allegro/APD or in your local environment file.

B

BASE
An ETCH/CONDUCTOR subclass; an outer layer of a design (APD terminology).

bi-directional
A pin on an ECL net that sometimes acts as a load and sometimes as a driver.

blind via
A hole used to connect ETCH/CONDUCTOR subclass that does not go all the way through a design. A blind via can connect either outer ETCH/CONDUCTOR subclass to an inner ETCH/CONDUCTOR subclass. See buried via. In the industry, blind and buried are often used interchangeably to describe vias that do not go all the way through the design.

board geometry
The physical definitions of the design’s base material.

bonding pads
Connecting pads on the substrate that are connected to bonding wires.

bonding wires
Wires that connect pads on a chip to either pads in the package or pads on the layout or substrate.

BOTTOM
An ETCH/CONDUCTOR subclass; an outer layer of a design (Allegro terminology).

boundary
A line that defines the outside edge of a window.
**buried resistor**
Terminates high-speed nets. You can construct buried resistors by replacing pin pads with a new pad in the shape of a resistor plus a rectangle that represents the resistor paste. The insulation of the resistor is represented in the database (the lack of copper between the resistor and the power plane) and not the internal geometry of the resistor. Other methods include adding a symbol representing a resistor and creating positive artwork film for that etch layer or creating a stand-alone resistor routed to the terminating pin. The latter consists of two pins: one via-like (a pin connected to the terminating pin). A second pin consists of a single layer pad that flashes on an imbedded plane.

**buried via**
A hole used to connect ETCH/CONDUCTOR subclasses that does not go all the way through a design. A buried via can connect any internal ETCH/CONDUCTOR subclass to another internal ETCH/CONDUCTOR subclass. See blind via. Blind and buried are often used interchangeably to describe vias that do not go all the way through a design.

**C**

**characteristic impedance**
At a given instance in time, a transmission line appears to an electrical signal as a resistance whose value is called the characteristic impedance. The resistance, capacitance, and inductance of a transmission line combine to impede the flow of charge.

**check box**
A check box is used on an Allegro/APD form to specify whether an item is to be used or selected. A check mark indicates a selected check box.

**chip-on-board**
A chip that is glued directly to the board. Usually, a chip or integrated circuit is enclosed in a package and mounted on the board. Bonding wires attach pinouts to pads.

**clip-on-chip**
A chip that is glued on a board that is very small and enclosed in a package. One example is a CPU and cache memory together in the same package to reduce delays.

**circuit**
A set of electronic functions, such as gates and buffers, that when connected together constitute the electronic description of a printed circuit design. When this description is provided in ASCII form, it is called a netlist. Allegro/APD requires a readable netlist as input for automatic design and checking.
class
A category used to identify and refer to elements in a design. It eliminates the requirement of referring to elements by layer number. You can have up to 64 subclasses that further define a class.

colored ceramics
Refers to the clay substrate.

color palette selector
The color palette selector is an application window that controls colors used in design display on the Allegro/APD desktop.

color editing
You can select colors from the color pad that is displayed when you select the color menu option. Two other forms appear as well as the color pad. Pick a color from the color pad and then select the subclass color box in the subclass form to apply that color.

color priority form
Controls the order in which colors appear in a design. Colors with the highest priority are painted on top with other colors displayed underneath.

command
A string of characters typed at the operating system prompt that perform a specific action. See option.

command line
The line, identified in the design window by the > prompt, at which the user can enter commands.

component
Circuit elements that are logically part of the circuit contained in the design drawing but are physically distinct from the design. For example, ICs and resistors.

component pin
Conductors that protrude from packages. Pins allow the component to be connected electrically to the circuits in the printed circuit design.

conductance
A measure of the heat transfer rate of an object for a given temperature difference across a measured area (in W/cm-deg C).
CONDUCTOR
A routing layer (Allegro/APD). For example, SURFACE or BASE. See routing subclass.

conductivity
A material property that describes a heat transfer rate through a volume of the material for a given temperature difference (in W/cm-deg C).

conductors
Materials with a low resistivity that conduct electricity easily.

connection
The smallest logical unit the automatic routing tool considers when routing a net. See connect line.

connect line
The line of etch that connects two pins on a net. See stub.

constraint
A restriction that the DRC process applies to a physical element in a design. Allegro/APD searches for constraint violations during automatic and interactive processing and flags violations with DRC markers. Allegro/APD has 130 types of constraints. Each constraint type:

- Has a name (for example, Stub Length)
- Has a DRC mode that determines when the DRC process applies the rule
- Can have one or more values associated with that constraint

constraint area
A user-defined outline (unfilled shape or rectangle element) in a design that determines the area in which a constraint set is active. Allegro/APD considers all shapes and unfilled rectangle elements in class SUBSTRATE GEOMETRY on subclass CONSTRAINT_AREA to be constraint areas. Each constraint area must have either a NET_SPACING_TYPE or NET_PHYSICAL_TYPE property (or both) attached.

constraint set
A predefined group of constraints organized by the behavior and type of element to which the constraints apply. Allegro/APD has three types of constraint sets:

- Spacing
- Physical
- Electrical
The number and type of constraints in each set are fixed. When you create a constraint set, you give it a unique name, then specify values for each constraint in the set.

**construct**

An ASCII character string in an Allegro/APD technology file that starts with an opening parenthesis followed by a keyword, followed by one or more values or a nested construct, and ending with a closing parenthesis. The keywords in a technology file identify Allegro/APD design parameters and constraints.

**converters**

See data translators.

**crosstalk**

Signal transmittal from one wire to another by electromagnetic field effects. On a printed circuit, parallel etch can exhibit significant crosstalk.

**cursor**

An element of the graphic display controlled by the mouse and the keyboard.

**D**

**database**

An Allegro/APD file that contains complete information about a design.

**data translators**

These are Allegro/APD options that provide data translation between Allegro/APD and other products, including interfaces to Calma, SciCards, Prime Computervision (CV), Cadnetix, Redac, Gardner Denver, Greenfield, Applicon, and Gerber.

**default**

A value selected for a parameter that is displayed by Allegro/APD when a form is displayed on the screen or when the user executes a command.

**design**

In Allegro/APD, a database file with a .brd file name extension. A design drawing usually contains two outer ETCH/CONDUCTOR subclasses (TOP and BOTTOM), internal ETCH/CONDUCTOR subclasses, padstacks, vias, edge connectors, and components. See printed circuit board.

**design work window**

The window displayed from an option where you can create, edit, and manipulate a design drawing.
device

The set of information used to describe a component.

device file

An ASCII file that contains electrical part information. In Allegro/APD, you supply this information for new parts.

die

An unpackaged chip.

DIE

Die Information Exchange format.

dielectric

Material that does not conduct electricity. It is used for insulating conductors and making capacitors.

dielectric constant

A value that represents a material's ability to store a charge when used as a capacitor.

discrete

An analog component. For example, resistor, capacitor, or inductor.

DIP

Dual-In-line Package.

display area

The space within the boundary of a window used by the application program to display a design.

display color

The Display Color command shows the temporary highlight, rats, and permanent highlight. You can use multiple highlight colors.

doping

The addition of an impurity that alters a material's conductivity.

drafting symbols

Leader-oriented, linear, datum, and angular dimensions (lines, text, arrows, and so on) are stored in the Allegro/APD database as drafting (.dra) symbols. Like other Allegro/APD symbol types, a drafting symbol consists of lines, arcs, and text that can be individually manipulated.
Unlike other Allegro/APD symbol types, only dimensioning commands create drafting symbols. No .dra files are created. Drafting symbols are created internally to enable you to more easily manipulate dimensions within a design (select, move, and delete them, for example).

**drawing**
A plot produced by a plotting device or a design drawing.

**drawing grid**
A dot matrix grid on which the user creates non-etch geometries.

**DRC**
Design Rule Checking. Allegro/APD continuously checks the design for spacing violations based on user-defined rules and standards.

**DRC model**
A user-set control switch associated with each constraint type. It has three possible settings that determine whether the constraint will be computed.

- Every time there is a change to the design (*Always*)
- Only on batch command (*Batch*)
- Never (*Never*)

DRC modes cannot be different for particular Spacing or Physical Constraint Sets or for different constraint areas. A single setting of the DRC mode applies to all instances of a constraint type, such as *Line to Line Spacing*. For Electrical Constraint Sets, however, DRC modes can be different for each constraint in each constraint set.

**DRC rule sets**
Minimum spacing definitions between standard design elements.

**DRC rule sets by class**
Determines how Allegro/APD handles distances between diverse nets.

**driver**
A source pin on an active component where a signal originates. Important in high-frequency circuitry (ECL).

**driver terminator**
The terminator on the driver end of an ECL net.
ECL
Emitter-Coupled Logic. In Allegro/APD, refers to high-speed designing.

ECL net
A net designed using the principles of Emitter-Coupled Logic. In Allegro/APD, a net that has the ECL property attached to it.

detail connector
A set of surface mounted pins on the edge of a layout. Edge connectors are used to connect designs to other designs, or to external devices such as front panels.

event
Parameters that control the Allegro/APD operating environment. Default settings can be user-defined to meet site requirements.

etch
Conductive material used in manufacturing a design.

ETCH
A routing class.

ETCH subclass
A routing layer. For example, TOP or BOTTOM (Allegro), or SURFACE or BASE (APD). See routing subclass.

etch T
A connection that is routed between a pin and another connection. See stub.

execution
An attempt by an automatic tool to complete a step. For example, autorouting, auto swap, and auto placement.

F
fails
A connection that was attempted by the automatic routing tool but was not completed.

failure rate
In Allegro/APD, a term used to quantify hardware reliability for components. It indicates the number of times a component fails in one million hours of operation. See MTBF.
field
In a form, displays the text or numeric value for a parameter. In an application menu, a field contains the name of an application option. In pop-up or pull-down menus, a field is a menu option.

fill-in fields
These fields have a single underline next to the name of the field and are displayed where you are required to supply information. An icon may be attached that displays a pop-up menu that contains one or more choices used in the field.

flash
In photoplotting, the process of creating pads using standard apertures.

flip chip
An unpackaged integrated circuit that connects to a hybrid circuit by means of solder bumps on its faces that correspond to its pin-outs.

floorplanning
Allows you to specify locations on a design for automatically placing components. See placement evaluator.

form
A box or window that is displayed by Allegro/APD when you select some menu options. A form sets attributes and operating characteristics for a design. Also known as a dialog box.

format symbol
A set of information contained in a file with a .osm file name extension used to create the drawing format and represent standard drawing forms such as a border, title block, notes, and all applicable drawing information.

from-to
A routing term for pairs of certain design elements that are scheduled to be interconnected. Elements can be component pins or rat Ts. In Allegro/APD, a from-to is represented by a ratsnest.

funcdes
The identification code of a function or gate.

function
A logical unit of an electronic part such as an integrated circuit, also referred to as a gate.
function designator
The identification code for a function or gate.

G

gate array
A geometric pattern of basic gates contained in one chip. These gates can be interconnected during manufacture to form a complex function that can be reproduced.

gate
The schematic description of the logical symbol, or symbols in a device.

glossing
Glossing applications perform post-processing functions, including increasing the width of connections to ensure greater manufacturing reliability, converting corners to arcs, and adding dielectric patches to hybrid designs to insulate intersecting connection.

green tape
DuPont's process for co-fired ceramics. The color of the unfired substrate is green.

H

Help
The Help button displays on-line help about Allegro/APD in a separate window. The helpcmd and helpmenu commands entered on the command line display the command table for that design work window and menu option-to-command correspondence.

hot spot
A spot of color at the center of each component. The color of the hot spot indicates the operating temperature range of the associated component.

hybrid circuit
A special form of microelectronic design that interconnects passive and active devices. A hybrid circuit responds to semiconductor chip integration and packaging needs. It combines the use of thick film used with printed circuits and thin film used with integrated circuits. Multilayered ceramics or co-fired ceramics is another common hybrid.
IC
An integrated or microcircuit (monolithic) that consists of interconnected elements inseparably associated and formed on or within a single substrate to perform an electronic circuit function.

ink
See paste.

Insight
An expert application that automatically sets operating parameters for use during automatic placement and routing.

interfaces
See data translators.

ISHM
The International Society for Hybrid Microelectronics.

isotherm
A line or curve that connects points of constant temperature. The color of the line indicates the range of temperature for the locations along the line.

jog
A piece of etch that runs perpendicular to most etch on that ETCH/CONDUCTOR subclass.

keepin package
A constraint that specifies to Allegro/APD the area in which to place all packages.

keepout package
A constraint that specifies to Allegro/APD the area in which packages are forbidden.
L

layer
   An insulated plane in the design that contains lines of etch.

laser trimming
   The removal of resistive material by laser that raises the resistance value of a film resistor.

line
   See connect line.

line fattening
   A gloss application that increases the width of connect lines wherever possible for greater manufacturing reliability. See glossing.

line ripup
   A feature of the automatic router that removes existing connect lines to make room for new connections.

list picker
   A scroll area that displays a list. You can select an item from the list or scroll through it to review other choices. The chosen item is displayed in an identification field. Alternatively, keyboard input is permitted.

load
   Any pin on an ECL net that is not a driver or a terminator.

load terminator
   The terminator on the load end of an ECL net.

log
   A file that Allegro/APD creates as a by-product of many processes. For example, when you execute an option in an application menu, Allegro/APD creates a log file to record events that occurred during processing. See reports.

lossy
   A lossy transmission line has resistance, causing it to dissipate some power as current passes through it. See ohmic loss.

lump load
   A model of transmission line using a combination of capacitor, inductor, and resistor.
lump loading ratio
The longest distance between any two consecutive loads in a net divided by the total length of the net.

M

manhattan distance
The orthogonal distance between two points. The distance calculated as the sum of the distance between the points along the X axis and the distance between the points along the Y axis. DX + DY.

map
Associates a component with a particular row and column.

mask
A pattern on glass or fine mesh screen that serves as the template for exposing thin film photoresist or for screening thick film material.

mechanical symbol
A set of information contained in a file having a .bsm filename extension used to define mechanical and graphic elements on a design drawing. Typically, design symbols represent non-electrical elements, for example, design outlines, plating bars, mounting holes, or card ejectors. Mechanical-only fixtures with drill holes are represented by pins with no pin numbers. design symbols do not have a reference designator label.

menu option
Any of the selections that appear in a menu.

menu pop-up
Any of the selections that appear in a pop-up when you pick a menu option.

menutool
A Cadence application that lets you create your own menu definition files. You can reposition frequently used menu options, delete menu options, and define pop-up options.

message area
An area in the command line used by Allegro/APD to display messages to the user. Up to three lines of text can be displayed and a scroll bar can be used to display messages outside the confines of the display area.
MTBF
Mean Time Between Failures. In Allegro/APD, a term used to quantify hardware reliability for designs. MTBF indicates the number of hours of design operation before a failure. See failure rate.

N
net
Any set of pins and vias that are logically connected.

net layer rules
These rules establish line width, neck width, and whether etch is allowed on each etch.

netlist
An ASCII text file that provides the electrical blueprint for the circuit design.

noise immunity
The worst case between output voltages produced by a driver pin and input voltages that a receiver pin interprets correctly when operated in an ideal environment. This presumes equal junction temperatures and no other sources of signal noise other than typical device manufacturing variations.

noise margin
A more thorough calculation of noise immunity, accounting for expected noise sources. This is the “margin of safety” that remains after estimated noise levels are subtracted from the ideal noise immunity.

net schedule
A preferred order for the interconnection of a net's component pins. The schedule may be user-defined or determined by Allegro/APD. When determined by the software, scheduling is based on component placement, types of component pins in the design, timing rules, etc.

O
ohmic loss
Voltage drop across a resistor as current passes through it. Design etch has measurable resistance and, due to the signal current, some voltage is lost on its way to the receiver pin.
option
A menu choice that you select from an application menu to display a form or execute a process.

Options
In Allegro/APD, a tab display in the right side of the Allegro/APD window. The fields in the Options tab change to match the command or option you have selected. Fields typically identify the class, subclass, and color assigned to the subclass.

P

package geometry
Graphic elements that make up a physical component, commonly referred to as shapes or symbols.

package symbol
A set of information contained in a .psm file used to represent an electrical component. The symbol is a physical representation of the logical parts in a schematic design, such as a dual in-line package (DIP), resistor, capacitor, or edge connector. Package symbols have a reference designator label and at least one pin number.

pad
See SMD pad.

padstack
A list of all data for each pad definition in the design drawing; each pin and via refers to a padstack for size, shape, and drill information.

parameters
Text and numeric values that control what you see on the screen and the functions performed by automatic programs.

passive devices
Devices such as resistors, capacitors, and inductors that either absorb or store energy.

paste
A screenable thick-film material. The three categories of thick film are conductors, resistors, and dielectrics. Also known as ink.

path
A line of travel between two pins in a net.
pick

1. The act of positioning the cursor on a graphic element such as an option and clicking (pressing and releasing) the left mouse button. Compare with select.

2. A phase of execution of the Swap application.

pin escape

A line of etch and a via used to connect surface-mounted pins to internal ETCH/CONDUCTOR subclasses.

pin pair

A set of two design elements, either component pins or rat Ts, on a net or extended net (xNet) that is established for the purpose of specifying a timing constraint. Pin pairs do not necessarily form a from-to, since the elements do not have to be scheduled for direct connection.

pin-to-pin connection

A signal path from a particular driver pin to a particular receiver pin. For example, a network with two drivers and three receivers has six possible pin-to-pin connections.

placement

An Allegro/APD function that executes the placement of components in a design drawing. Allegro/APD provides both interactive and automatic placement capabilities.

placement evaluator

Allows you to judge where routing channels are blocked. The placement evaluator calculates statistics for routing a design. The placement evaluator analyzes the potential routing success of a placed design. You can start testing a placement for routability as soon as you place components in the design.

placement grid

A matrix of lines that you create using the Grid option in the Autoplacement menu, and edit using Edit commands. The grid defines locations for automatic component placement. Interactive placement uses the non-etch grid that you create using the Define – Grid option.

pop-up field

Displays multiple choices if you either toggle the field (click on it several times) or hold down the left mouse button in the field to display the pop-up. To select an item, release the mouse button on the highlighted item.
printed circuit board
A single, thin piece of material comprised of many laminations of a substrate, usually epoxy, on which an electrical circuit is printed, usually in copper. Online, an Allegro/APD design.

property
In Allegro/APD, a specific instruction as to how a net, function, reference designator, or pin is treated by an automatic program.

radio button
A group of buttons that you can select by toggling. A small, filled-in circle indicates a selected radio button.

ratsnest line
In a design drawing, a line that shows a logical connection between two pins, connect lines, or vias. Elements connected by the same ratsnest line are part of the same net. The ratsnest shows the circuit logic and, for ECL circuits, the order in which pins are to be connected.

rat T
A database object used to insert a branch in a net’s schedule at some point other than at a component pin. A rat T has a physical location that is often an approximate location for a ‘T’ or a via in the net’s physical interconnect.

rat T cluster
A group of component pins on a single net that are logically connected (that is, specified by the net schedule) indirectly through one or more rat Ts. A pin can belong to more than one rat T cluster.

reference designator
The designator, or identification code, for a component.

reflection
When a signal traversing a wire meets a sudden change in characteristic impedance, some of the signal is reflected backwards. This is similar to the splash-back caused when a stream of water is passed through a mesh screen. In an etch, multiple reflections are possible, producing “ringing” and “overshoot.”
refresh symbol
This command replaces existing symbols in a design with newer versions of symbols from a library. Options indicate the symbol type to refresh. You can refresh package symbols, mechanical symbols, a list of symbols that you provide in a text file, or all symbols.

regular pad
A positive pad (black) with a regular shape (circle, square, rectangle, oblong, shape, or aperture flash).

resistance
Extent to which an instance resists the passage of heat.

resistor packs
Components that contain many resistors. On ECL designs, the resistors in resistor packs are used as terminators.

reports
User-defined files that provide specific information about a design. For example, you may execute the report command from the operating system to create an ECL Loading Report, a file that lists any nets that do not meet design specifications. See log.

ripup
See line ripup.

room
A user–defined area of the design that is treated separately by several automatic programs. For example, the automatic placement program uses rooms to group related components. See window.

route keepin
A route constraint. An area you must add to the design to tell AutoRoute where to contain the routed connections.

route keepout
A route constraint. An area you can add to the design that tells AutoRoute where not to route connections.

routing
The process of connecting pins in a net with lines of etch.
routing area
The area of the design drawing in which you wish to route. Also, an area of the design drawing you can route separately from the rest of the design drawing, such as a room or a window.

routing channels
Horizontal and vertical paths that connect routing grid points.

routing grid
A matrix of dots or grid points that AutoRoute uses to route connections.
The space between routing grid points.

routing layer
A layer on which connections are routed. See routing subclass and ETCH subclass.

routing subclass
In Allegro/APD, any of the ETCH/CONDUCTOR subclasses that you have designated for routing. Routing subclasses are a subset of the ETCH/CONDUCTOR class. (All routing subclasses are ETCH/CONDUCTOR subclasses, but not all ETCH/CONDUCTOR subclasses are necessarily routing subclasses.) ETCH/CONDUCTOR subclasses that are not routing subclasses are just unused routing layers. See ETCH subclass and routing layer.

rubberbanding
A feature of interactive commands where, as you move an element of the design drawing with the mouse, lines attached to it “stretch” as you move.

rules-driven design
User-defined design characteristics that can be specified by the schematic that are recognized by Allegro/APD and determine processing results.

S

script files
Scripts let you perform repetitive tasks in Allegro/APD in a timely fashion. You can build a script by recording a file and executing the commands that you want the script to execute. You can use scripts to set up forms for routing, placing, and artwork or executing a series of check plots. Scripts can call other scripts.

scroll area
Scroll areas are used to display data that cannot be displayed within a single window.
scroll bar
A band along the right side of a window that is used to display the contents of a drawing
or file that does not fit within the confines of the window.

scheduling
The process of creating and updating the interactive ratsnest to reflect the order in which
pins are to be routed in an ECL net. Schedules are established in a netlist.

search pin
In an ECL net, the pin from which the closest terminator is searched, even if that is not
the pin to which the terminator is added.

select
The act of positioning the cursor on a menu option, form field, or pop–up, pressing and
holding the middle button, and releasing when your choice is highlighted. Compare with
pick.

signal analysis
An Allegro/APD option that predicts where layout-dependent noise problems such as
crosstalk and reflection might occur.

signal noise
Unwanted voltages that cause a received voltage signal to differ from the signal originally
transmitted.

skip
A connection that is not attempted by AutoRoute.

slide bar
The slide bar icon is positioned to the right of a fill-in field and displays a minimum and
maximum number at either end of a horizontal bar. These numbers appear when the
icon is selected. You can select from a range of values by sliding the bar with the cursor.

SMD
1. Surface-Mounted Device

2. A technology using surface-mounted components which have pins that are glued to the
surface of a design. Designs that contain SMDs can have components on both sides.
See through-hole component.
SMD pad
A piece of etch on TOP or BOTTOM/SURFACE or BASE where an SMD component pin is connected to the design.

SMD pin
A component pin that has a component pad belonging to only one ETCH/CONDUCTOR subclass—either TOP or BOTTOM/SURFACE or BASE.

source
A driver.

sputter deposition
Exposing the chip or board to atomized metal being sputtered at the chip that sticks where there is no mask.

staggered via
A via that spans more than two layers and adds a cline (user-defined stagger size) and another via for each set of layers. One through-drill can produce seven vias and six clines.

status area
A 3-line area on the design window that displays information about the current activity. The first and second lines display the current directory and filter. The third line identifies the current command, or “idle” if no command is active.

status message
The message displayed in the status area of a design work window. When you are using an interactive tool (for example, the Add > Line option), the status message reports the current command (“Add Line”). When you are executing an automatic program (for example, AutoRoute), the status message reports statistics indicating the progress the program has made. See MTBF.

step
A phase of routing with a distinct function or a goal and a unique set of parameters and number of executions defined for accomplishing that goal.

stub
When a pin is routed to a connect line, the stub is the line of etch between the pin and connect line. Compare with etch T.

subclass
Further defines a class. You can define subclasses for a class. Each class can have up to 64 subclasses.
substrate
The material with which an IC, printed circuit, or hybrid circuit starts. For instance, silicon, GaAs, fiberglass, or ceramic.

subtractive process
This process creates a circuit by etching away unwanted conductors already on the substrate.

SURFACE
A CONDUCTOR subclass. One of the outer layers. (APD)

surface-mounted pin
See SMD pin.

swap
Swap Function: To exchange the locations of two functions that are logically identical, either within a component or between components, to minimize the average net length. You can perform function swap either automatically or interactively in Allegro/APD.
Swap Pin: To exchange the locations of two pins within a function that are electrically identical to minimize the average ratsnest crossings. You can perform pin swap either automatically or interactively in Allegro/APD.
Swap Component: You can perform interactive swapping of two components in Allegro/APD to improve design placement.

switch area
SWITCH_AREA_TOP and SWITCH_AREA_BOTTOM are areas in which all etch is routed in the direction perpendicular to the preferred direction of most of the etch on that ETCH/CONDUCTOR subclass.

symbol
A set of data that can be used to represent any design element. There are three kinds of symbols:
- Package symbol (.psm)
- Mechanical symbol (.bsm)
- Format symbol (.osm)
- (.ssm)
Glossary

T

TAB
Tape-Automated Bonding. A method for attaching a chip to a substrate for chip-on-board.

technology file
(or tech file)
An ASCII file that can be read into an APD design to specify user preferred units, constraint and parameter values, and user properties.

terminator
A resistor pin where the other pin is attached to a negative voltage. Terminators are used to eliminate signal reflection on high-frequency (ECL) nets. The device file for a terminator always contains, on one line, PACKAGEPROP TERMINATOR_PACK.

terminator assignment
The process of assigning terminators to the load end and the driver end of every ECL net that has a LOAD_TERM_VAL and/or DRIVER_TERM_VAL property attached to it.

thermal analysis
An Allegro/APD option that lets you analyze the thermal conditions resulting from current design placement and the boundary conditions you specify, and retrieve junction and case temperatures (J_TEMPERATURE and C_TEMPERATURE property values) for the components in the design.

thermal-relief pad
A negative pad (clear, surrounded by black), often created with a special aperture flash, to connect a pin to an embedded metal layer that distributes a voltage, such as a power or ground.

thermal shift
A temperature-induced change in operating voltage. A silicon junction at room temperature operates at 0.6 volts. This value increases about 2 millivolts for every 1C of junction temperature rise. The junction temperature difference between driver and receiver is responsible for thermal shift in logic devices.

thick film
A hybrid circuit technology that selectively deposits materials on an insulating substrate. Several masks or layers occur on one or more metal or resistor prints. The conductor, dielectric, and resistor inks are screen-printed in their final circuit pattern and fired at temperatures up to 1000 degrees Centigrade on the ceramic substrate. Thick film is often used to create printed circuits.
thin film
A hybrid circuit technology that deposits metals and resistor materials across a substrate and then removes material through photoetching. The conductor, dielectric, and resistor films are vacuum- or vapor-deposited on a substrate in sheets. The circuit pattern is photolithographically masked and chemically etched. Thin film is more sensitive to assembly processes and more costly than thick film. It is often used to create integrated circuits.

third party
A drawing or schematic generated by an automated or mechanical process other than a Cadence tool.

through-hole component
A component that has pins that go through all layers in a design. The pins are adhered to the design with solder.

title bar
A band along the top of a window that displays the name of the window and information about that application.

TOP
An ETCH/CONDUCTOR subclass. One of the outer layers.

transmission line
An electric conductor exhibiting series inductance and shunt capacitance distributed along its length. A signal must charge up each chunk or inductance and capacitance before it is passed long to the next chunk, thus reducing the propagation velocity.

U
user-defined net
A net for which you establish the pin order by using the $SCHEDULE keyword in the $NETS section of the netlist. The schedule program does not change the pin order.

user unit
The unit of measure you select when creating a new design. The fill-in field for user units is in the Drawing Parameters form. Mils is the default; other choices include inches, millimeters, centimeters, and microns.
V

via
An opening in a dielectric layer that connects adjacent conductor layers. In Allegro/APD, a via is a plated-through hole with etch on every ETCH/CONDUCTOR subclass. Vias make it possible to route a single connection through more than one ETCH/CONDUCTOR subclass. Also called a feedthrough.

via pitch
The closest allowable center-to-center distance between vias.

Viable
A Cadence analysis tool that predicts design reliability. Viable uses project, library, method, setup, and template files.

via grid
An optional user-defined matrix of dots representing locations where AutoRoute can place vias. Without a via grid, AutoRoute uses route grid points to place vias.

via keepout
A route constraint that specifies to Allegro/APD the area in which vias are forbidden but etch is allowed.

visibility
Controls items that are displayed on the screen.

W

window
Any section, usually rectangular, of the graphic display where, if the cursor is within its borders, the mouse and keyboard assume functions that are different from their functions in the surrounding area. Examples include an application menu, forms that are displayed when you select the Param option, and borders that surround the design drawing. Also a user-defined area in which an automatic process is executed.

window cursor
The crosshair that is displayed when the cursor is positioned inside the boundaries of a design window. You can change the cursor shape using the Status form.

work window
Used to perform design tasks not specifically related to a design drawing that control how a drawing is manipulated. See design work window.
x

xNet

Extended net; a net composed of a group of nets in a system. The nets may be on one or more boards or modules. Typically, an XNet is composed of nets that are separated by passive devices such as resistors, connectors, cabling, etc.