T Commands

Product Version 23.1
September 2023

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

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

Contents

1

T Commands

About Text Displays

Cadence tools use a text display dialog box to provide various types of information, which may be a simple property definition or a comprehensive list of element types, values, location, and function. In some cases, the dialog box displays error messages.

Menu Bar Options

File – Save As	Saves the information in a text file. When you issue this command, the editor prompts you for a file name and appends the `.txt` extension.
File – Print	Prints the contents of the window on either UNIX or NT systems. Use the User Preferences Editor dialog box to set the *print_unix_command* environment variable governing UNIX printing or the *print_nt_extension* environment variable governing NT printing.
File – Stick	Makes the window remain on screen until you close the window, or the program terminates. Use this option to compare information between two windows. For example, you may use show element to obtain information about two design objects and use *File – Stick* to compare the contents of each window.
Close	Dismisses the window.

Related Topics

- Managing Environment Variables

tbx zdrc

The `tbx zdrc` command performs metal-to-metal spacing checks in the z-direction for the net or net class objects. All the objects (traces, pin, vias, and shapes) of a net on any layer are converted and treated as shapes before the Z-DRC spacing check is run. The spacing value can be specified manually or derived from the Constraint Manager and a DRC is generated for violations. The command also provides a graphical display of overlapping where a violation occurs.

When enabling the DRC system the following rules are applied:

- Z-DRC only reads CSet assignments on net classes. Any override values on net class, bus, differential pair, XNet, and net objects are ignored.
- If a net does not belong to any net class, it is being assigned to default net class NO_TYPE and *Default* CSet is assigned to it.
- Z-DRC performs a metal-to-metal check and uses the maximum value that appears in the CSet. Z-DRC holds one spacing value for each CSet.
- When running Z-DRC, the actual spacing value is calculated by taking the maximum value of the net class CSets from both nets.
- Z-DRC takes Class-Class rules into account by taking the maximum value of the CSet which is assigned to the Class-Class rule.

Access using:

- Menu path: *Tools – Z-DRC*

Related Topics

- *Z-DRC Dialog Box*
- Running Z-DRC Spacing Checks for Net Objects

Z-DRC Dialog Box

Main Tab		
Spacing value		
	DRC system	Choose to calculate the actual value based on the spacing rules defined in Constraint Manager
	Fix	Choose to manually specifying the spacing value
Layer distance calculation	Specify how Z-DRC accounts for thicknesses of copper layers. The three options are: • *Include all Cu layers:* The thicknesses of all dielectric layers and all copper layers from the current layer span are summed up. • *Exclude external Cu:* The thicknesses of all dielectric layers and internal copper layers from the current layer span are summed up. • *Exclude all Cu:* Only the thicknesses of the dielectric layers from the current layer span are summed up.	
Group A [Class/Nets]	Select at least one object (net or net class) that need to be checked against each other	
Group B [Class/Nets]	Select at least one object (net or net class) that need to be checked against each other	
Limit Selection to	Limits the selection to certain layers. Only clines, shapes, pins, and vias included in the selected layer span are considered for Z-DRC check	
	Start	Specify the start layer
	End	Specify the end layer
Output Tab		
DRC list	Displays all violations	
DRC Filter		
	Layer	Filters the violations to specific layer pairs between which a violation occurred
	Type	Filters the violations to objects involved, such as L-L (Line-Line), V-L (Via-Line)
	Net	Filters the violations to net names involved. Wildcards are supported. For example,*CLK*
DRC Visit	Displays crossprobing options	
	Zoom	Zooms to the location of a violation when selected in the DRC List
	Limit Layers	Displays only the layers involved (between which a violation occurred)

	Display overlap	Enable to display layer overlapping graphically. The overlapping is represented by shapes that are drawn on special BOARD GEOMETRY subclasses starting with ZDRC_.
Reports		
	DRC Errors	Generates two Z-DRC Error reports (in text and csv format)
	Layer distances	Generates a Layer Distance report that contains the distances for all layer pair combinations taking the options from Layer distance calculation into account.
	Rules	Generates Rule Spacing Report
OK		
Z-DRC Update	Click to run Z-DRC using the following two options: • Retain existing DRC • Cleanup all DRC	
Clear All	Clears all DRCs from the list and associated DRC markers from the design canvas	

Related Topics

- *tbx zdrc*
- Running Z-DRC Spacing Checks for Net Objects

Running Z-DRC Spacing Checks for Net Objects

1. Choose *Tools – Z-DRC.*
2. In the *Main* tab of the Z-DRC dialog, specify the spacing value.
3. For layer distance calculation, choose to include or exclude copper layers.
4. Select at least one net from Group A and Group B.
5. Click *Z-DRC Update* and choose either *Retain existing DRC* or *Cleanup all DRC*.
6. Open the *Output* tab, double-click the violation from *DRC List*.
   The violation is zoomed and highlighted in the design canvas.
7. Use *DRC Filter*, to filter the violations based on layer name and type.
8. In the *DRC Visit* section, enable the options to display layer overlapping and zoom level.
9. Click *Reports* options to generate DRC error and Rule Spacing reports.

Related Topics

- *tbx zdrc*
- Z-DRC Dialog Box

tdf_editor

The `tdf_editor` command automatically replaces footprints in a design and is an alternative to the legacy use model of alternate symbol selection during interactive placement. This command address the requirement of changing a footprint of a component when moving from a rigid to flex zone or when you need to accommodate mask requirements for a new manufacturing assembler.

The main advantage of this solution is that it is back-end driven and has no dependencies on the front-end libraries. This flow does not require the traditional assignment of jedec types and netlist export from the schematic design application. This solution is also compatible with non-Cadence schematic-driven flows.

> ⚠ By default, the Allegro X PCB Editor continues to support the alternate symbol methodology. The ALT_SYMBOLS property can be used in conjunction with Technology Dependent Footprints (TDF) to enable specific parts to override the mapping for a particular PACKAGE (JEDEC Type).

Using the technology-dependent zone-aware footprint solution, you can define multiple technologies for different zones that use alternate symbol choices for component placement in zones. The command opens the Technology Dependant Footprint Editor consists of an XML-based mapping table to define alternate padstack mapping for top, bottom, and embedded layers. The technology-dependent footprint file (`.tdfx`) promotes consistent usage across design teams and controlled by the environment variable TECHPATH.

This command is available with the Miniaturization product option. It is also available with the Allegro Venture PCB Designer Suite, Allegro Enterprise PCB Designer Suite, and Allegro X Advanced Package Designer+ (using SiP Layout XL / using SiP Layout Bundle) licenses.

Related Topics

- Technology Dependent Footprint Editor Dialog Box
- Package Symbol Mapper Dialog Box
- Applying Alternate Symbols Using Technology Dependent Footprint Mapping File
- Creating New Zone Technology in Technology Dependent Footprint Editor
- Applying New Zone Technology to a Zone

Technology Dependent Footprint Editor Dialog Box

- Menu path: *Setup – Technology Dependent Footprints*

Version	Displays version number of the technology dependent footprint (TDF) file loaded in editor	
Last edited by	Displays the name who has last edited the TDF file	
Keywords	Use to add keyword in the TDF file	
Comments	Use to add comments to the TDF file	
Read Only	Enable this checkbox to disable all the entries for editing TDF file	
Mapping Table	Accommodates for the TOP, EMBEDDED, and BOTTOM layer mapping assignments	
	Package Symbols	Displays list of package symbol names
	Default Symbols	Display alternate symbol for each package symbol. To modify alternate symbol assignment, double-click to launch Package Symbol Mapper Dialog Box.
	+	Click to add new column or row for a technology type or package name respectively
Show database symbols only	Enable this checkbox to view only those package symbols names in the mapping table that are placed in a design	
Refresh symbol options		
	Ignore FIXED property	Replace a symbol to which the FIXED property has been assigned.
	Reset symbol text and location	The symbol text and size is reset as it is defined in the symbol definition as opposed to how it is defined in a design, if different.
	Ripup etch	Etch associated with symbol pins is removed during refresh symbol.
	Reset fanouts	Reset predefined fanouts from the symbol.
Table utilities		
	Populate symbols	Load all the package symbols and default symbols that are used in a design in the mapping table

	Purge unused symbols	Removes unused package symbols from the mapping table
	Purge unused technologies	Removes unused technology column from the mapping table
Refresh symbols	Click to start the alternate symbol assignment in a design	
Mapping file		
	Load from library	Opens library browser to select the technology dependent footprint file. The browser lists all the TDF mapping files that are available in the path set for the TECHPATH path variable.
	Load	Opens file browser to select the technology dependent footprint file.
	Save	Opens file browser to save the current technology dependent footprint file for use in other designs.
	Delete	Deletes the alternate symbol mapping information and the attachment in the design.

Related Topics

- tdf_editor
- Package Symbol Mapper Dialog Box
- Applying Alternate Symbols Using Technology Dependent Footprint Mapping File
- Creating New Zone Technology in Technology Dependent Footprint Editor
- Applying New Zone Technology to a Zone

Package Symbol Mapper Dialog Box

Available Package Symbols	Display list of all the package symbols available for the selected symbol. Double-click the symbol name to add it to the list of *Alternate Symbols.*	
Alternate Symbols	Lists alternate symbols mapped to package symbol. Double-click the symbol name to remove it from the list.	
Filter in/out items from the list		
	Packages to show	Set filter to show relevant package symbol names
	Packages to hide	Set filter to hide package symbol names
	Database	Populate symbols from the current design in the *Available Package Symbols.*
	Library	Populate symbols from the library in the *Available Package Symbols.*
OK	Click to apply the mapping information	

Mapping Table Options

Add Default Symbols	Adds a new column listing default symbol names
Add Zone Technology	Adds a new column for specifying alternate symbols for a zone
Remove Zone Technology	Deletes zone technology column
Assign Alternate Symbol	Opens Package Symbol Mapper for assigning alternate symbol
Assign NONE	Removes symbol mapping for a the selected package symbol
Copy Table	Copies complete table for the selected layer
Paste Table	Pastes complete table on the selected layer
Remove all Mapping Data	Deletes mapping information for all the layers
Undo	Reverts the last action
Redo	Repeats the last action
Sort Ascending	Sorts the selected column in ascending order
Sort Descending	Sorts the selected column in descending order

Related Topics

- tdf_editor
- Technology Dependent Footprint Editor Dialog Box
- Applying Alternate Symbols Using Technology Dependent Footprint Mapping File
- Creating New Zone Technology in Technology Dependent Footprint Editor
- Applying New Zone Technology to a Zone

Applying Alternate Symbols Using Technology Dependent Footprint Mapping File

The choices of footprints vary from one manufacturer to other. Using technology-dependent footprint flow, you can assign alternate symbols in your design to meet the requirement of a specific manufacturer.

Mapping files with alternate symbol information are created along with symbol libraries. The TDF Editor interface provides options to automatically updates alternate symbols in the entire design as specified in the TDF mapping file and get all symbols to refresh using the alternate symbols specified for that technology.

1. Choose *Setup – Technology Dependent Footprints*.
   Technology Dependent Footprint Editor opens.
2. Choose *Mapping File – Load*.
3. Select a TDF mapping file from the library and click *OK*.
   The mapping file is loaded in the editor.
4. If required, modify the alternate symbol assignment in the mapping table for the TOP, EMBEDDED, and BOTTOM layers.
5. Set up the *Refresh symbol options*.
6. Click *Refresh symbols* to apply the changes to the design.
   The process starts and a progress meter is displayed.
7. Click *OK* in the confirmation message that pops up.
   Alternate symbols are assigned to all the components in the entire design as defined in the TDF mapping file.
8. Choose *File – Viewlog* to open `tdf_refresh.log` file to verify the status of all the symbols that have been refreshed or changed.

Related Topics

- tdf_editor
- Technology Dependent Footprint Editor Dialog Box
- Package Symbol Mapper Dialog Box
- Creating New Zone Technology in Technology Dependent Footprint Editor
- Applying New Zone Technology to a Zone

Creating New Zone Technology in Technology Dependent Footprint Editor

Some areas of a layout may require different footprints than the rest of the design. You can specify such areas by creating zones and then assign alternate footprints to the components that are placed in those zones. To do this, you should create a new zone technology in the TDF Editor and map it to a zone.

1. Choose *Setup – Technology Dependent Footprints*.
   Technology Dependent Footprint Editor opens.
2. To load the information of default symbols that are displayed in the canvas, click *Populate symbols* in *Table utilities.*
   The symbols are listed in the mapping table for the TOP, EMBEDDED, and BOTTOM layers.
3. Click the + sign in the *Mapping Table* to add a new zone technology.
   A new column is added to the mapping table.
4. Double-click the column header and enter a new name for the new zone technology.
5. To assign the alternate symbol to the new zone technology:
   1. Double-click the blank cell in the new zone technology column to open Package Symbols Mapper.
   2. Assign an alternate symbol from the list of available package symbols.
   3. Click *OK* to close the Package Symbols Mapper.
   4. Repeat the above three steps to complete the alternate symbol mapping for the new zone technology.
6. Choose *Mapping File – Save* and specify a name and location for the TDF file to save the mapping data.

Related Topics

- tdf_editor
- Technology Dependent Footprint Editor Dialog Box
- Package Symbol Mapper Dialog Box
- Applying Alternate Symbols Using Technology Dependent Footprint Mapping File
- Creating New Zone Technology in Technology Dependent Footprint Editor
- Applying New Zone Technology to a Zone

Applying New Zone Technology to a Zone

To use different pad geometries in the zones of the flex area of the design, you can assign specific zone technology to that area using the technology-dependent footprint flow. You can assign alternate symbols mapping to a zone, so that when placing components into that zone or moving components into the zone they are automatically swapped the symbols based on the specified alternate mappings.

Do the following to assign a zone technology to a zone:

1. Choose *Setup – Zone – Create* to create a new zone*.*
2. In the Options tab, specify the zone data.
3. Draw a shape in the design canvas to define the zone boundary.
4. Right-click and choose *Done*.
   The zone is created in the design canvas.
5. To open Zone Manager, choose *Setup – Zone – Manage.*
6. In the *Zone Technology* column, select technology from the pull-down menu.
7. Click *Apply* and *OK* to save the settings and to start the refresh symbol process.
   The technology-dependent footprints are changed in the selected zone.
8. To verify, select a symbol and move it out of the zone.
   The symbol automatically changes to its default footprint.

Related Topics

- tdf_editor
- Technology Dependent Footprint Editor Dialog Box
- Package Symbol Mapper Dialog Box
- Applying Alternate Symbols Using Technology Dependent Footprint Mapping File
- Creating New Zone Technology in Technology Dependent Footprint Editor

techfile

The `techfile` batch command lets you read in or write out technology or a parameter file from back-end databases. Arguments associated with the command let you:

- Read a tech or a parameter file into a design.
- Write (create) a tech or a parameter file from a design.
- Compare a design to a tech file.
- Uprev tech files from pre-Release 16.0 EDIF-like tech files (`.tech`) to the Release 16.0 XML-based version of the Technology Constraint File (`.tcf`).

Tech files contain parameters, design-level constraint data and modes, including the cross-section, and user-defined properties. Typically stored on disk, you use tech files to preserve company standards while creating new databases. Upon completion of the command, the tool reports its results to the `techfile.log` that appears in the current directory.

Database parameter (`.prm`) files contain customized parameters exported from one design and imported into another when you initially begin a design. You create the `.prm` text file with *File – Export – Parameters* (`param out` command).

The `techfile` command uses the following parameters.

`techfile -r|-w|-c|-u [-q][-d][-n][-i][-p] [-o` *<paramtype>*`...][-t` *<drawing_type>*]

-r	Reads a tech or a parameter file into a layout.
-w	`w`rites a tech or a parameter file from a layout.
-c	Compares a tech file with a layout.
-u	`U`prevs an older tech file to the new `.tcf` format.
`-n`	Creates a new layout for the specified function (use with the `-r` argument only).
`-d`	Does not run Design Rule Checking (DRC) on layout after a tech file is read (use with the `-r` argument only).
`-i`	Ignores layers if they do not already exist in the design (use with `-r` and `-c` arguments only).
-o <paramtype>	Specifies to export a particular database parameter <paramtype> from a design. By default, all supported parameter records are exported. Use with `-w` and `-p` options.
-p	Reads or writes a database parameter file (`.prm`) into an existing design file; otherwise reads or writes the techfile. Works with `-r` or `-w` options only.
`-q`	Quiet mode, continues with warnings.

`-t`	Targets <*drawing type*> as either a `.brd`, or `.mcm` database: - Default drawing type is `.brd` - Works with `-n` or `-u` options only - Use "mcm" for `.mcm` (quotes not required) Example: `techfile -rn -t mcm my_pkg.tech my_pkg.mcm`
-version	Prints the version of the command.
techfile	Existing techfile (`.tech`).
new_techfile	Generates this techfile (`.tcf`).
layout	Existing database (`.brd`, or `.mcm`)
new_layout	Optional output database. If you do not use this variable, the tool overwrites the existing layout.
drawing_type	Uses file extension (`mcm` or `brd`).
paramfile	Database parameter file (`.prm`).

Examples

This table provides examples and describes the command syntax.

To…	Use this command syntax…
Uprev a tech file to the latest release	`techfile -u` [`-t` <*drawing_type*>] [<*techfile*> <*new_techfile*>] Description This tool uprevs an older tech file (`.tech`) to the Release 16.0 version of the tech file (`.tcf`).
Import (read) a tech file into a design	`techfile -r` [`-n`][`-d`] [`-t` <*drawing_type*>] [<*techfile*> <*layout*>] [*new_layout*] Description Reads <*techfile*> into your design <*layout*>. With the `-d` option, the tool does not run DRC on the design after reading the tech file. The combination of the `-r` argument and the `-n` option creates a new default (blank) design in memory, reads the < *techfile* > into it, then writes the resulting design to < *new_layout* > .
Export (write or create) a tech file from a design	`techfile -w` [<*layout*> <*new_techfile*>] Description With Release 16.0, the tool creates a Constraint Manager Technology Constraint File (`.tcf`) when exporting the tech file. The `-w` argument opens your design <*layout*> and writes a tech file to <*techfile*>. If you use the `-w` argument and the `-n` option, you can create a tech file containing the tool's defaults. The tool creates a new default (blank) design in memory and writes its tech file to <*techfile*>. The command then deletes the default design from memory.

Compare a tech file to a design	`techfile -c` [`-i`] [<*techfile*> <*layout*>] Description Compares the parameters and constraints in <*techfile*> to the values in your design <*layout*> and writes a comparison of the values in the tech file and design into the `techfile.log` file, and generates a report. **If you use the** `-c` **argument and the** `-n` **option,** the tool creates a new default (blank) design in memory, compares the parameters and constraints in <*techfile*> with the default values in the unnamed design, and writes the comparison. Then it deletes the default design from memory.
Import (read) a database parameter file into an existing design	`techfile -r -p` [<*in_paramfile*> <*in_design*>] [<*out_design*>] Description Reads the contents of a <*in_paramfile*> into the specified design <*in_design*>, then writes the resulting design to < *out_design* > . If you do not specify an <*out_design*>, the input design is overwritten.
Export (write) all supported database parameters from an existing design	`techfile -w -p` [<*in_design*> <*out_paramfile*>] Description Writes all supported database parameters from <*in_design*>, then writes the results to < *out_paramfile* >.
Export (write) specific database parameters from an existing design	`techfile -w -p` [`-o` <*paramtype*> `-o` <*paramtype*>...] [<*in_design*> <*out_paramfile*>] Description Writes a group of database parameters `o` [`-o` <*paramtype*> `-o` <*paramtype*>...] you specify from <*in_design*>, then writes the results to < *out_paramfile* >.

Related Topics

- param out
- *Defining and Developing Libraries*
- techfile compare

techfile compare

Use this command to compare a tech file to a design. You can determine if the values in a design conform to the intended values residing in the tech file, before you send the design to manufacturing.

The `techfile.log` records the values of the file and the design for side-by-side comparison. Only the constraints specifically contained in the tech file are checked against their counterparts in the design. The `techfile.log` also contains any warnings or errors encountered while reading the tech file.

You can run this operation in batch mode with the `techfile` command.

Related Topics

- techfile
- Tech File Compare Dialog Box
- Comparing the Design to the Technology File
- Defining and Developing Libraries

Tech File Compare Dialog Box

- Menu path: *Tools – Technology File Compare*

Tech file name	Specifies the name of the tech file that you want to compare.
Browse	Click **...** to display an Open browser window from which you can choose the tech file name.
Library	Click to display the Select Tech File to Compare dialog box, which contains all tech files in the directories defined in the TECHPATH environment variable in the *Design_paths* category of the User Preferences Editor, available by choosing *Setup – User Preferences* (enved command).
Compare	Click to start the comparison program.
Viewlog	Click to review the `techfile.log` file for any warnings or errors.
Close	Closes the dialog box.
Help	Displays help for this dialog box.

Related Topics

- techfile
- techfile compare
- Comparing the Design to the Technology File
- enved
- Defining and Developing Libraries

Comparing the Design to the Technology File

1. Run the `techfile compare` command.
   The Tech File Compare dialog box appears.
2. Type the technology file name in the text box
   -or -
   click *Browse* to choose another file name.
3. Click *Compare*.
4. Click *Viewlog* to review the `techfile.log` file for the differences.
5. When the program is done, click *Close*.

Related Topics

- techfile
- techfile compare
- Tech File Compare Dialog Box
- Defining and Developing Libraries

techfile import

Obsolete command. See techfile in.

techfile in

The `techfile in` command lets you read in technology files from Allegro back-end databases. Use this command to import either a legacy tech file (.`tech`) or a new Technology Constraint File .`tcf` file into your design.

Tech files contain parameters, design-level constraint data and modes, including the cross-section, and user-defined properties. Typically stored on disk, you use them to preserve company standards while creating new databases. Upon completion of the command, the tool reports its results to the `techfile.log` that appears in the current directory.

The `techfile in` command, imports the technology file in merge mode. In this mode, the layers in the existing technology file are not deleted. Instead, the layers are re-organized to include the new layers in the existing tech file. If a constraint in the tech file does not exist in the design, it is added.

⚠ To import the technology file in the overwrite mode, use the *File – Import – Techfile* command in Constraint Manager.

If an error occurs in the tech file, the tool continues reading the file, and writing warning and error messages, but does not update the design.

You can also run this operation in batch mode with the `techfile` command.

Related Topics

- techfile
- Defining and Developing Libraries
- Import a technology file(.tcfx) Dialog Box
- Tech File In Dialog Box
- Importing a Technology File
- techfile out

Tech File In Dialog Box

- Menu path: *File – Import – Techfile*

Input Tech File	Specifies the name of the technology file to be loaded into your drawing. Click **...** to display an Open browser window from which you can choose the tech file name.
Run DRC	Check this box to run DRC (Design Rule Checking) after reading a technology file into your design.
Viewlog	Click to review the `techfile.log` file for any warnings or errors.
Import	Click to start the import program.
Close	Click to exit the dialog box.

Related Topics

- techfile in
- Importing a Technology File
- techfile
- techfile out
- Defining and Developing Libraries

Importing a Technology File

1. Run `techfile out` on an existing design.
2. Run `new` to start a new design.
3. Run `techfile in` to display the *Tech file In* dialog box.
4. Enter the tech file name to import.
5. Click *Import*.
6. When the program finishes, click *Close*.
7. Choose *Viewlog* to review the `techfile.log` file for any warnings or errors.
   The new design now has same constraints, stackup, and size as your initial design.

Related Topics

- techfile in
- Tech File In Dialog Box
- techfile
- techfile out
- Defining and Developing Libraries

techfile out

Use this command to create a tech file (`.tcf`). All instances of all constraint sets are written to the tech file. You cannot write out only the constraints for a particular constraint set type or instance. Results of the export appear in the `techfile.log` file in the current directory.

You can also run this operation in batch mode with the `techfile` command.

Related Topics

- techfile
- *File – Export – Techfile* command in Constraint Manager
- Tech File Out Dialog Box
- Exporting a Technology File from Your Design
- techfile in
- Defining and Developing Libraries

Tech File Out Dialog Box

- Menu path: *File – Export – Techfile*

Output tech file	Specifies the name of the tech file you want created from the design. Click **...** to display an Open browser window from which you can choose the tech file name.
Export	Click to start the export process.
Close	Closes the dialog box.
Viewlog	Click to review the `techfile.log` file for any warnings or errors.
Help	Displays help for the dialog box.

Related Topics

- techfile out
- Exporting a Technology File from Your Design
- techfile
- techfile in
- Defining and Developing Libraries

Exporting a Technology File from Your Design

1. Run `techfile out` to display the Tech File Out dialog box.
2. Type the name in the Output Tech file field
   -or -
   click ... to choose another file name.
3. Click *Export*.
   When the program is done, click *Close*.
4. Choose *File – Viewlog* to review the `techfile.log` file for any warnings or errors.

Related Topics

- techfile out
- Tech File Out Dialog Box
- techfile
- techfile in
- Defining and Developing Libraries

Temp Group

Temp Group is available as an option on the right-button pop-up menu when you run certain editing and display commands; for instance, `property edit` and `show element`. This option allows you to choose multiple elements for simultaneous editing. It is unavailable when you are working in pre-selection use model.

Related Topics

- Choosing Multiple Elements for Editing

Choosing Multiple Elements for Editing

1. Run any command that supports the *Temp Group* option.
2. Before choosing an element to edit, click right to display the pop-up menu and choose *Temp Group*.
3. Choose the elements for editing. Each element you choose is highlighted.

   > ⚠ To deselect any element you highlighted, hold down the control (Ctrl) key and click on the highlighted element.

4. When you choose all the elements, click right again to display the pop-up menu and choose *Complete*.
5. You can now perform editing on the elements that you chose.

Related Topics

- Temp Group

termination edit

The termination edit command lets you view, modify, add, or delete terminators. Termination synthesis is usually an iterative process, sometimes requiring you to make changes or adjustments to terminators and synthesize again.

Related Topics

- Logic - Define Terminators Dialog Box
- Modifying, Adding, or Deleting Terminators

Logic - Define Terminators Dialog Box

- Menu path: *Logic – Define Terminators*

Use this dialog box to view, modify, add, or delete terminator types. You can "add" termination to a net, or adjust the termination value of existing termination for signal analysis. If you accept the solution of an added terminator, you can then run `termination package` to specify the physical part to use.

When the physical part is added, you can use `place manual` to place the part on the design.

Net Filter	Uses asterisk as wildcard to narrow the search of available nets. After you apply the filter, use the radio buttons to further limit the search:
Net list box	Shows all nets allowed by Net Filter and radio button selection.
Pin list box	Shows the pin details for a net highlighted in the Net - termination window. Highlight a line to choose a pin type for termination editing.
Termination Type	Choose a new termination type to apply to the specified pin.Depending on the type chosen, entry boxes for appropriate values display: Term Resistance, Term Capacitance, Term Voltage High - Low,Cutoff Volt High - Low,Delay Constraint
Modify	Apply the new termination type and values

Related Topics

- termination edit
- Modifying, Adding, or Deleting Terminators

Modifying, Adding, or Deleting Terminators

1. Run `termination edit`.
   The Define Terminators dialog box appears.
2. Use the Filter and the radio buttons to narrow the search. In the Pin field, choose an eligible pin for modification. You can only modify certain terminators, and even then the extent of modification may be limited. It depends on the use of the pin (pin use code) and whether it is a driver, receiver, or bi-directional termination.

Pin Use Code	Termination Type
UNSPEC	NONE
POWER	NONE
GROUND	NONE
NC	NONE
OUT	NONE, SERIES
TRI	NONE, SERIES
OCA	NONE, SERIES
OCL	NONE, SERIES
IN	NONE, SHUNT RC, GND DIODE, PWR DIODE, DUAL DIODE, SHUNT, THEVENIN
BI	NONE, SERIES, SHUNT RC, GND DIODE, PWR DIODE, DUAL DIODE, SHUNT, THEVENIN

⚠ Pins having a pin use code of OUT, TRI, OCA, and OCL are drivers; IN is a receiver and BI is bi-directional.

3. Use the Termination Type pull-down list to choose a different type for the specified pin.
   Depending on the type you choose, additional fields may appear below and to the right of the Termination Type field.
4. To change the information in any of the fields, enter the values of your choice.
5. Click *Modify*.
   The information in the list boxes updates to reflect your changes.

Related Topics

- termination edit

termination package

The `termination package` command lets you package newly created terminators and add them to the design. Terminators created for analysis purposes are unpackaged. They must be packaged before they can become part of the design.

You can specify package definitions by browsing Design Entry HDL or the editor libraries or by creating packages "on the fly" (create temporary packages). When you create terminators in the design window of your user interface, the program attempts to package those terminators according to the following criteria:

- The type of terminator specified
- Any constraint on the net
- The number of components in a terminator to be packaged.

Related Topics

- Package Terminators Dialog Box
- Adding Packages from Device Libraries
- Adding Packages from Design Entry HDL Component Libraries
- Creating Temporary Termination Packages
- Choosing Terminators for Packaging

Package Terminators Dialog Box

- Menu path: *Logic – Package Terminators*

Terminator Types window	Shows alt terminator types and values in the design.
Package	Shows the device name you choose from the editor or Design Entry HDL browser. Also shows auto-generated name if you choose *Create Part*.
Clear	Clears the Package: window.
Browse Concept	Opens the Design Entry HDL part library browser.
Browse Allegro	Opens the editor part library browser.
C*reate Part*	Creates a temporary package. Package High and Package Low fields may appear at the bottom of the dialog box when a Shunt RC or Thevenin termination is chosen. Voltage fields may also appear at the bottom of the dialog box to specify the voltage nets to which the termination packages will be wired.
Package	Adds the termination package to the netlist and adds the necessary wiring.

Related Topics

- termination package
- Adding Packages from Device Libraries
- Adding Packages from Design Entry HDL Component Libraries
- Creating Temporary Termination Packages
- Choosing Terminators for Packaging

Adding Packages from Device Libraries

1. Run `termination package`.
   The Package Terminators dialog box appears.
2. Click *Browse Allegro*.
3. Choose a device in the Device Browser to update the data in the Package area of the Package Terminators dialog box.
   Package High and `Package Low` fields may appear at the bottom of the dialog box when you choose a Shunt RC or Thevenin termination. Voltage fields may also appear at the bottom of the dialog box to specify the voltage nets to which the termination packages will be wired.
4. Click *Package* to add the termination package to the netlist and add the necessary connections.

Related Topics

- termination package
- Package Terminators Dialog Box
- Adding Packages from Design Entry HDL Component Libraries
- Creating Temporary Termination Packages
- Choosing Terminators for Packaging

Adding Packages from Design Entry HDL Component Libraries

1. Run `termination package`.
   The Package Terminators dialog box appears.
2. Click *Browse Concept*.
   If your library database has not been identified by SigNoise as HDL or SCALD format, a dialog appears prompting you to identify it.
3. Click the appropriate label in the prompt.
   A File Browser appears, chosen for the file type you clicked.
4. Choose a library file in the File browser.
   The Concept Browser appears.
5. Choose a device in the Concept Browser to update the data in the Package area of the Package Terminators dialog box.
   Package High and *Package Low* fields may appear at the bottom of the dialog box when you choose a Shunt RC or Thevenin termination. Voltage fields may also appear at the bottom of the dialog box to specify the voltage nets to which the termination packages will be wired.
6. Click *Package* to add the termination package to the netlist and add the necessary connections.

Related Topics

- termination package
- Package Terminators Dialog Box
- Adding Packages from Device Libraries
- Creating Temporary Termination Packages
- Choosing Terminators for Packaging

Creating Temporary Termination Packages

1. Run `termination package`.
   The Package Terminators dialog box appears.
2. Click *Create Part.*
   Package High and *Package Low* fields may appear at the bottom of the dialog box when you choose a Shunt RC or Thevenin termination. Voltage fields may also appear at the bottom of the dialog box to specify the voltage nets to which the termination packages will be wired.
3. Click *Package* to add the termination package to the netlist and add the necessary connections.

Related Topics

- termination package
- Package Terminators Dialog Box
- Adding Packages from Device Libraries
- Adding Packages from Design Entry HDL Component Libraries
- Choosing Terminators for Packaging

Choosing Terminators for Packaging

1. Run `termination package`.
   The Package Terminators dialog box appears.
2. Click a terminator type in the list box at the top of the dialog box.
   The type of terminator you determines whether one or more package fields appears near the bottom of the dialog box. If the list box is empty, you must synthesize a terminator or create one using the Define Terminators dialog box. Allegro PCB SI currently lets you specify seven different terminators:
   - Series
     This method of termination is power efficient. It effectively dampens ringing and overshoot.
   - Shunt (parallel)
     This method of termination gives the best speed performance for an interconnection and allows the use of distributed loads. However, the required power consumption makes it unsuitable for CMOS.
   - ShuntRC (AC termination)
     The RC combination in this method of termination dampens signal transients. It differs from the shunt because the capacitor blocks any DC current path and helps reduce power consumption.
   - GND Diode
     This method of termination, commonly used for DRAMs, dampens the signal overshoot at a level of -1V using a Schottky diode. It does not provide immunity to crosstalk and noise at high edge rates.
   - Dual Diode
     This method of termination is similar to the GND Diode, but uses two diodes to dampen the signal overshoot to both -1V and +1V.
   - PWR Diode
     This method of termination is similar to the GND Diode, but dampens the signal
   - Thevenin
     This method of termination uses two components (resistors) to provide effective termination for incident wave switching while not degrading VOL and VOH as much as the parallel termination. Virtually free of duty cycle effects and suitable in both high and low frequency buses.

Related Topics

- termination package
- Package Terminators Dialog Box
- Adding Packages from Device Libraries
- Adding Packages from Design Entry HDL Component Libraries
- Creating Temporary Termination Packages

testprep automatic

Lets you define parameters for the automatic testprep process and automatically generate testpoints.

For additional information related to testprep, see the *Preparing Manufacturing Data* user guide in your documentation set.

Related Topics:

- Testprep Automatic Dialog Box
- Generating Testpoints Automatically
- Preparing Manufacturing Data

Testprep Automatic Dialog Box

- Menu path: *Manufacture – Testprep – Automatic*
- Toolbar icon:

Allow test directly on pad	Specifies whether a pin or a via can be chosen as a testpoint.
Allow test directly on trace	Automatically generates a testpoint pad (using the pad specified by the *SMT Testpad* field on the *Padstack Selections* tab, which must be a top-or bottom-side test pad) at the mid-point of a horizontal or vertical trace segment on the net where the testpoint can be added without cline bubbling, and no pin or via pad currently exists. The pad chosen in the *SMT Testpad* field must align with the *Layer* chosen in the Testprep Parameters dialog box. Adding a testpoint directly to a trace eliminates the possibility that auto-generated through-hole test vias violate stub rules and will be attempted after Allegro X PCB Editor encounters an enabled *Allow test directly on pad setting*, but before it encounters an enabled *Allow pin escape insertio*n *setting*. A testpoint is created as an SMT pad entity on external, but not internal, traces on the TOP or BOTTOM layers, as permitted by the *SMT Testpad* field. Allegro X PCB Editor places a testpoint via on grid if possible, or offgrid otherwise. When you use *Manufacture – Testprep – Manual* (`testprep manual` command), if Allegro X PCB Editor identifies a trace rather than a pin or a via, it adds the testpoint. If *Disable cline bubbling* is disabled on the Testprep Parameters dialog box, Allegro X PCB Editor may bubble clines to avoid DRCs. Even if this field is disabled, you can interactively add testpoints to a trace by using *Manufacture – Testprep – Manual*, but if you subsequently use it to delete or swap a testpoint that was added to a trace, or choose *Manufacture – Testprep – Automatic* (`testprep automatic` command) with *Execute Mod*e set to *Overwrite*, Allegro X PCB Editor removes the entire via that was added. You must use *Route – Custom Smooth (*custom smooth command) to manually remove any bubbling that resulted from adding the via, as Allegro X PCB Editor doesn't delete it during overwriting.
Allow pin escape insertion	In conjunction with the *Thru via* field on the *Padstack Selections* tab and the *Via displacement* fields, automatically adds an SMT test pad and a via to a net if no suitable test site exists. Use this option for in-circuit tests to ensure top- or bottom-side access to all surface-mount-technology devices or for routed nets without through-hole vias. Routing is added despite the presence of a NO_PROBE area that may be placed over pins. The actual via is placed outside the NO_PROBE area. When locating a via outside a NO_PROBE area, the *Via displacement: Max* parameter setting controls the distance allowed from the pin to go beyond any NO_PROBE area. Allowing pin escape insertion requires a route keepin.
Test unused pins	Specifies whether pins that do not appear on any net should be designated as testpoints.

Execute mode	Determines the mode in which Allegro X PCB Editor runs testprep after you click *OK*. *Overwrite* removes all existing testpoint designations and physically removes from the board/substrate any element such as testpoint vias added directly to a trace or created by pin- escape insertion (including the routing). If a testpoint via's pad is replaced at the time, the layout editor restores it to the padstack the via had prior to becoming a testpoint. This is the default. *Incremental* does not remove any testpoint designations. Allegro X PCB Editor reports all nets with testpoints and if the *Test method* is *Flood*, analyzes even those nets that already have testpoints. Even though Allegro X PCB Editor does not generate new testpoints for already tested nets, the log file contains information about the testpoints already on the design. Note: If you choose *Incremental* but not *Allow test directly on pin* and *Allow pin escape insertion*, then the `testprep.log` file records existing testpoints.
Via displacement	*Min:* Specifies the minimum distance from the pin or via where the automatically generated testpoint or via can be placed. A value of zero specifies no minimum and that the minimum DRC distance should be used. Note: Allegro X PCB Editor never creates testpoints that violate the DRC rules. *Max:* Specifies the maximum distance from the pin or via that the automatically generated testpoint can be placed.
Generate testpoints	Click to initiate the automatic testprep process.
Parameters	Click to display the Testprep Parameters dialog box.
View log	Click to review the `testprep.log` file, which summarizes the most recent execution of the testprep program. It lists all parameters, net names, and pin numbers for all testpoints. Other statistics are warnings, fails, completions, location (top or bottom), ignores (no test nets), and failure reasons.
Close	Closes the dialog box and saves any changes.
Cancel	Closes the dialog box and discards any changes.

✓ You can also access the Testprep Parameters dialog box by using the Design Parameter Editor. Choose *Setup – Design Parameters* (*prmed* command), then click *Edit testprep parameters* under the *Mfg Applications* tab, or use the `testprep prmed` command.

Related Topics

- testprep automatic
- Generating Testpoints Automatically
- testprep manual
- custom smooth

Generating Test Points Automatically

1. Choose *Manufacture – Testprep – Automatic* (`testprep automatic` command). *The Testprep Automatic dialog box appears.*
2. Choose *Allow test directly on pad* to permit a pin or a via to serve as a test point.
3. Choose *Allow test directly on trace* to allow surface only test pads to be permitted directly on a trace.
4. Choose *Allow pin escape insertion* to automatically add a via to a net if no suitable test site exists.
5. *Choose the Test unused pins field to test unused pins.*
6. Choose whether to run testprep in *Overwrite* mode, which removes all existing test point designations at the beginning of every execution or *Incremental* mode, which does not remove any test point designations.
7. Specify the minimum distance from the pin or via where the automatically generated test point or via can be placed in the Via replacement minimum field. A value of zero specifies there is no minimum, and that the minimum DRC distance should be used.
8. Specify the maximum distance from the pin or via that the automatically generated test point can be placed.
9. *Click Parameters to set parameters as required.*
10. Click *Generate testpoints* to run the automatic test point process.

Related Topics

- testprep automatic
- Testprep Automatic Dialog Box

testprep createfixture

Generates the static FIXTURE_TOP and FIXTURE_BOTTOM subclasses and copies PROBE_TOP and PROBE_BOTTOM subclass information to them. The FIXTURE_TOP and FIXTURE_BOTTOM subclasses maintain the information regardless of what testpoints you add, delete, or move during design revisions, letting you graphically compare the differences between the PCB that represented the fixture and the current design after logic changes.

Related Topics

- Testprep Create Fixture Dialog Box
- Creating FIXTURE Subclass Layers
- Preparing Manufacturing Data

Testprep Create Fixture Dialog Box

- Menu path: *Manufacture – Testprep – Create FIXTURE*

Overwrite existing FIXTURE subclasses	Choose to overwrite existing FIXTURE subclasses with new testpoint locations and testpoint identification text. Circle symbols are superimposed on PROBE subclass triangle symbols that represent testpoint locations.
Create fixture	*Choose to create new* FIXTURE *subclasses.*
Cancel	Closes the dialog box and discards any changes.

Related Topics

- testprep createfixture
- Creating FIXTURE Subclass Layers

Creating FIXTURE Subclass Layers

1. *Choose Manufacture – Testprep – Create FIXTURE.*
2. To overwrite existing FIXTURE subclasses, choose *Overwrite existing FIXTURE subclasses.*
3. To create new FIXTURE subclasses, choose *Create fixture.*

Related Topics

- testprep createfixture
- Testprep Create Fixture Dialog Box

testprep density

Verifies the testpoint density within user-definable unit areas when you enable the *Unit Area Check*, or beneath symbols when you enable *Component Area Check*. You can run the checks simultaneously or separately.

To use the *Unit Area Check,* you specify the maximum number of testpoints allowed per unit area in the *Max testpoints per Unit Area* field. Exceeding this value creates rectangular figures that correspond to the user-defined unit areas and overlay the PROBE_DEN_TOP and PROBE_DEN_BOTTOM subclasses on the MANUFACTURING class. The layout editor automatically creates or clears these subclasses as required to verify the testpoint density within the areas of violation. Based on this data, manufacturing may remove probes from a unit area or use a smaller probe. Executing the *Unit Area Check* saves the settings to the database to synchronize its contents with the settings used during the last execution and to the PROBE_DEN subclasses.

To use the *Component Area Check,* you must attach the TESTPOINT_MAX_DENSITY property to the symbols requiring it, where the associated value specifies the maximum number of testpoints allowed under the symbols. You can add or delete testprep-related properties using *Edit – Properties* (`property edit` command) or *Manufacture – Testprep – Properties* (`testprep properties` command).

When you enable the *Component Area Check,* the layout editor checks the rectangular PLACE_BOUND_TOP/BOTTOM area for the component, which may be non-rectangular if its symbol has a non-orthogonal rotation. A testpoint is deemed to lie beneath a component based on the testpoint location only (not the pad shape), either exactly on or within the area boundary. A rectangle or rotated rectangle representing the PLACE_BOUND area appears on the appropriate PROBE_DEN subclass to flag components that exceed the specified maximum number of testpoints. The *Component Area Check* optionally uses ASSEMBLY data depending on the *Component Representation* setting on the General Parameters tab. ASSEMBLY data is used if it is a SHAPE or RECTANGLE entity, or a single multi-segment LINE entity that forms a closed shape. ASSEMBLY data resembling a rectangle, but actually comprised of four different LINE entities, is not used. Arcs are recognized in a SHAPE or LINE entity.

The TESTPOINT_MAX_DENSITY property on a symbol has no impact on testpoints created when you choose *Manufacture – Testprep – Manual* (`testprep manual` command) or *Manufacture – Testprep – Automatic* (`testprep automatic` command), or if you move or delete vias that happen to be testpoints. Only running a component area density check flags violations. You must change the number of testpoints to meet the specified maximum.

Related Topics

- Testprep Density Check Dialog Box
- Verifying Testpoint Density in a Unit Area Squares
- Verifying Testpoint Density Beneath Components
- testprep manual
- testprep automatic
- property edit
- `testprep properties`
- Preparing Manufacturing Data

Testprep Density Check Dialog Box

- Menu path: *Manufacture – Testprep – Density Check*

Unit Area Check	Choose to verify the testpoint density within user-definable blocks based on the maximum number of testpoints allowed per unit area specified in the *Max testpoints per Unit Area* field.
Unit Area Square Size	Specify the length of a square's side to define a unit area against which to calculate how many probes may legally exist within that unit area. For example, if you enter 100 mils, then the unit area square measures 100 x100 mils. If you enter a measurement other than mils, the layout editor converts it to mils.
Unit Area Square Displacement	Enter the distance from the center point of one unit area square to the next. This value cannot exceed the size of the square itself. Squares must either overlap or abut.
Max Testpoints Per Unit Areas	Enter the greatest number of testpoints that may exist in any unit area.
Component Area Check	Choose to verify the number of testpoints under symbols to which you have attached the TESTPOINT_MAX_DENSITY property. This check evaluates testpoints beneath the component, but on the opposite side to that on which the component/symbol is placed. For example, if a 2000-pin BGA occurs on layer TOP, the layout editor checks for a maximum testpoint allocation on layer BOTTOM, but within that component's place-bound region. The *Component Area Check* optionally uses ASSEMBLY data depending on the *Component Representation* setting on the General Parameters tab. ASSEMBLY data is used if it is a SHAPE or RECTANGLE entity, or a single multi-segment LINE entity that forms a closed shape. ASSEMBLY data resembling a rectangle, but actually comprised of four different LINE entities, is not used. Arcs are recognized in a SHAPE or LINE entity.
Density Check	Runs the chosen checks and logs the results to the `testprep_density.log` file. You can change settings and then re-execute to evaluate their effect.
Close	Closes the dialog box and saves any changes. When you re-open the dialog box, settings from your last session appear.
Viewlog	Click to review the `testprep_density.log` file, which details all unit or component areas containing at least one testpoint. The layout editor flags those that exceed the allowable maximum as violations. Click on any hyperlinked x/y coordinates in the report to center that location in the display.

Related Topics

- testprep density
- Verifying Testpoint Density in a Unit Area Squares
- Verifying Testpoint Density Beneath Components

Verifying Test Point Density in a Unit Area Squares

1. Use *Manufacture – Testprep – Density Check* (`testprep density` command) and enable *Unit Area Check* on the Testprep Density dialog box.
2. Specify the length of a square's side to define a unit area against which to calculate how many probes may legally exist within that unit area in the *Unit Area Square Size* field.
3. Specify the distance from the center point of one unit area square to the next in the *Unit Area Square Displacement* field.
4. Specify the maximum number of test points allowed per unit area in the *Max testpoints per Unit Area* field.
5. Click *Density Check* to execute the unit area check.
6. Review the `testprep_density.log` file, which details all unit areas containing at least one violation and all test points within the area.
7. Examine dispersion of test points on the PROBE_DEN_TOP and PROBE_DEN_BOTTOM subclasses on the MANUFACTURING class.

Related Topics

- testprep density
- Testprep Density Check Dialog Box
- Verifying Testpoint Density Beneath Components

Verifying Test Point Density Beneath Components

1. Use *Edit – Properties* (`property edit` command) or *Manufacture – Testprep – Properties* (testprep properties command) to attach the TESTPOINT_MAX_DENSITY property to symbols as required.
2. To evaluate test points beneath these components, but on the opposite side to that on which the component/symbol is placed, use *Manufacture – Testprep – Density Check* (`testprep density` command) and enable *Component Area Check* on the Testprep Density dialog box.
3. Click *Density Check* to execute the component area check.
4. Review the `testprep_density.log` file, which details all component areas containing at least one violation and all test points within the area.

Related Topics

- testprep density
- Testprep Density Check Dialog Box
- Verifying Test Point Density in a Unit Area Squares
- property edit
- `testprep properties`

testprep fix

Globally sets or resets the status on all testpoint locations on the design. Testpoints are not fixed or unfixed individually.

Related Topics

- Testprep Fix/Unfix Testpoints Dialog Box
- Preparing Manufacturing Data

Testprep Fix/Unfix Testpoints Dialog Box

- Menu path: *Manufacture – Test Prep – Fix/unfix testpoints*

Global testpoint status:	
Fixed	Globally fixes all testpoints, both currently existing and subsequently added to prevent further editing or automatic removal of existing testpoints.
Unfixed	Globally unfixes all testpoints, both currently existing and subsequently added.
OK	Executes the chosen action, closes the dialog box, and saves any changes.

Related Topics

- testprep fix
- Preparing Manufacturing Data

testprep manual

Lets you manually add, delete, or move testpoints and edit testpoint-related properties on nets and symbols.

> ⚠ Manually adding or deleting testpoints sets the *Execute field* on the Testprep Automatic dialog box to *Incremental* to avoid subsequently generating automatic testpoints in *Overwrite* mode and losing manual changes. You must explicitly change the setting to *Overwrite on the Testprep Parameters dialog box.*

Related Topics

- Testprep Query Dialog Box
- Options Tab for the `testprep manual` Command
- Creating a Test Probe
- Deleting a Testpoint from a Pad or Via
- Moving a Testpoint to Another Pad or Via
- Identifying Untested Nets Using Testpoints
- Preparing Manufacturing Data

Testprep Query Dialog Box

The Testprep Query dialog box lets you list the attributes of a chosen pin, via, cline, or test point. It displays the number of existing test points on a net, attached testprep-related properties, and relevant characteristics.

File – Save As	Saves the information in a text file. When you issue this command, Allegro X PCB Editor prompts you for a file name and appends the `.txt` extension.
File – Print	Prints the contents of the window on either UNIX or Windows systems. Use the User Preferences Editor dialog box to set the `print_unix_command` environment variable governing UNIX printing or the `print_nt_extension` environment variable governing Windows printing.
File – Stick	Makes the window remain on screen until you close the window, or the program terminates. Use this option to compare information between two windows.
Close	Closes the dialog box.

Related Topics

- testprep manual
- Options Tab for the `testprep manual` Command
- Creating a Test Probe
- Deleting a Test Point from a Pad or Via
- Moving a Test Point to Another Pad or Via
- Identifying Untested Nets Using Testpoints

Options Tab for the testprep manual Command

- Menu path: *Manufacture – Testprep – Manual*
- Toolbar icon:

Add	Lets you create a new testpoint (probe) designation interactively on a pad or via, according to the parameters in the Testprep Parameters dialog box.
Add (scan and highlight)	Choose to locate untested nets to ensure that all nets to be tested have at least one testpoint. Nets with testpoints or the NO_TEST property are excluded from the search. The net becomes highlighted, and the display zooms in automatically on bounds of the net extents. You can then pick points on the net to which to add testpoints; then right click to display the pop-up menu: *Done:* End scanning. *Oops*: Undoes the action of the last pick. *Next:* Go to next net, accepting any current additions. *Cancel:* Delete any modifications made during this session.
Delete	Removes all existing testpoint designations you choose individually or by window and physically removes from the board/substrate any element such as testpoint vias added directly to a trace or created by pin escape insertion (including the routing). If a testpoint via had its pad replaced at the time, the via is restored to the padstack it had prior to becoming a testpoint.
Swap	Moves a testpoint (probe) designation to another pad or via, or to a new testpoint on a trace, according to the parameters that you set in the Testprep Parameters dialog box.
Query	After choosing a pin, via, or a net (for example, a trace), display information about other testpoints on that net and assigned properties, which appears in the Testprep Query dialog box. Information includes whether a pin or via is a testpoint; if a testpoint via was created directly on a trace, auto inserted, or had its pad replaced; existing testpoints on the net, or the net of the identified pin or via; the existence of NO_TEST and TESTPOINT_QUANTITY net properties; whether any TESTPOINT_QUANTITY property is currently being met; and If a pin was chosen, any existence of the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY property on its parent symbol; and any probe type associated with a testpoint.
	You can also query any testprep density check areas on PROBE_DEN_TOP/BOTTOM layers, which details the number of testpoints within the area and the maximum number allowed.
Parameters	Choose to display the Testprep Parameters Dialog Box.
Properties	Choose to set testprep properties on nets and symbols.

Related Topics

- testprep manual
- Testprep Query Dialog Box
- Creating a Test Probe
- Deleting a Testpoint from a Pad or Via
- Moving a Testpoint to Another Pad or Via
- Identifying Untested Nets Using Testpoints

Creating a Test Probe

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command).
2. Choose *Add.*
3. Choose a pin, via, or point on a trace segment and the testpoint highlights.
   A message identifies the net and side of the design.
4. Confirm the designation by choosing *Next* or Done
   The command adds a test figure with the correct associated text (the net name or the PROBE_NUMBER) at that point.

Related Topics

- testprep manual
- Testprep Query Dialog Box
- Options Tab for the `testprep manual` Command
- Deleting a Testpoint from a Pad or Via
- Moving a Testpoint to Another Pad or Via
- Identifying Untested Nets Using Testpoints

Deleting a Test Point from a Pad or Via

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command).
2. Choose *Delete*.
3. Pick a pin or via location.
   The pin or via and the associated net is highlighted.
4. Click right and choose *Done* or *Next* from the pop-up menu to confirm the location.
   If you checked *Display* in the Testprep Parameters dialog box, the associated text (the net name or the probe number) also appears.

Related Topics

- testprep manual
- Testprep Query Dialog Box
- Options Tab for the `testprep manual` Command
- Creating a Test Probe
- Moving a Test Point to Another Pad or Via
- Identifying Untested Nets Using Test points

Moving a Test Point to Another Pad or Via

1. Choose *Manufacture – Testprep – Manual* (testprep manual command).
2. Click *Swap.*
3. Pick a pin or via location.
   The pin, its associated text and any connect lines/ratsnest lines attached to that point highlight.
4. Pick the location to which to move the test point status from the pin or via.
5. Right-click and choose *Done* or *Next* from the pop-up menu to confirm the location.
   If you checked *Display* in the Testprep Parameters dialog box, the associated text (the net name or the probe number) also disappears.

Related Topics

- testprep manual
- Testprep Query Dialog Box
- Options Tab for the `testprep manual` Command
- Creating a Test Probe
- Deleting a Test Point from a Pad or Via
- Identifying Untested Nets Using Test points

Identifying Untested Nets Using Test Points

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command).
2. Choose *Add (scan and highlight)* on the *Options* tab of the control panel.
   Each untested net highlights for your review.
3. Click an untested net, and choose *Query* to obtain information about it.
4. Choose *Tools – Reports – Testprep* to display the Testprep Report, which lists untested nets.

Related Topics

- testprep manual
- Testprep Query Dialog Box
- Options Tab for the `testprep manual` Command
- Creating a Test Probe
- Deleting a Test Point from a Pad or Via
- Moving a Test Point to Another Pad or Via

testprep ncdrill

Outputs testpoint locations marked as valid to NC files used to drill testpoints in fixtures according to the input parameters you set in the Testprep Parameters Dialog Box, available by choosing *Manufacture – Testprep – Parameters* (`testprep prmed` command).

To test both sides of the design, you can automatically create an NC drill file for each side of the design. The file for the top/surface side is `top_probe.drl`. The file for the bottom/base side is `bottom_probe.drl`. The `probe_tape.log` log file details the number of probes in the top/surface and in the bottom/base.

Access using:

- Menu path: *Manufacture – Testprep – Create NC Drill Data*

This command does not open any dialog boxes.

Related Topics

- testprep prmed
- Preparing Manufacturing Data

testprep prmed

Lets you define the parameters governing the testprep process and determines how Allegro X PCB Editor chooses component pin or via locations as probe sites. Settings you define here determine legal selection points. If untestable nets remain, you then can use *Manufacture – Testprep – Manual* (`testprep manual` command) to create and edit testpoints.

You can enter multiple replacement padstacks, including top- and bottom-side blind vias, for a via using the *Padstack Selections* tab. To populate the *Padstack Selections* tab with predefined parameters, you can also load a Comma Separated Value (`.csv`) file containing your preferred settings; conversely, you can save *Padstack Selections* tab values to a `.csv` file.

To avoid running testprep recursively to optimize test coverage, you can define probe names and spacing combinations on the *Probe Types* tab. The *Probe Types* tab defines largest to smallest spacings, for example:100, 75, 50, etc., correlated to probe types, or names, used in the fixture. The greater the spacing, the more rigid the probe. With tighter spacing (50 mils or less), the probes are thinner and more flexible, which can create fault or structural issues in the fixture bed.

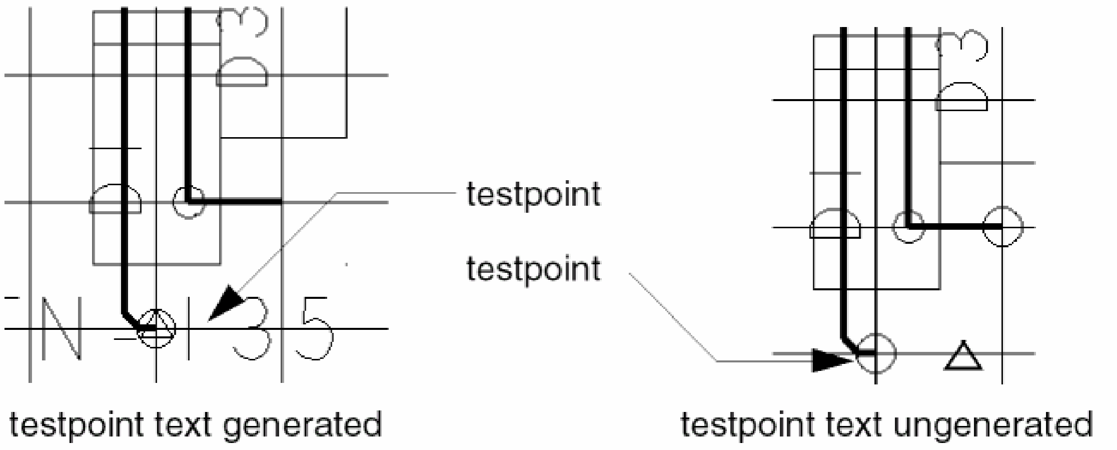
Related Topics

- Testprep Parameters Dialog Box
- Setting Testprep Parameters
- Specifying Replacement Padstacks for New Vias
- Specifying Replacement Padstacks for Existing Vias
- Optimizing Probe Types and Spacing
- testprep manual

Testprep Parameters Dialog Box

- Menu path: *Manufacture – Testprep – Parameters*.

General Parameters tab	
Preferences	Defines the characteristics of pins and vias used as probe sites.
Pin type	Specifies the type of pin that can be chosen for testing: *Input, Output, Any pin, Via*, and *Any pnt.* The first three options specify electrical preferences; the latter two, physical. If no testpoint is found, and *Allow test directly on pin* is enabled, Allegro X PCB Editor attempts to connect a test pad or via to a pin or via. Values are: *Input:* Chooses a pin on an I/O device. *Output:* Chooses a die output pin only. *Any pin:* Choice for input pins to be attempted first, followed by output pins. *Via:* Designates a via only. *Any pnt*: Choose to search for locations using the entire hierarchy shown in the pop-up menu.
Pad stack type	Specifies the type of pad needed as a contact point for the test probe (*SMT Testpad, Thru Via, Either*). You can restrict probing to either SMT pads or through-hole pads, or both by toggling this field. The default is through-hole pads.
Methodology	Defines required testing applications.
Layer	Specifies the side of the design on which to locate the testpoint. Values are *Top/Surface, Bottom/Base*, or *Either*. The default is *Top/Surface. Either:* Allegro X PCB Editor chooses probe points on both sides with a preference given to the *Top/Surface* subclass. *Top/Surface:* Allegro X PCB Editor chooses probe points only on the top side of the design. Specify this layer if you choose a top-layer SMD pad in the *SMT Testpad* field. Otherwise, if the layer specified here is incompatible with that required for a top-layer SMD test pad, this field becomes blank, and a layer setting of *Bottom/Base* takes precedence. *Bottom/Base:* Allegro X PCB Editor chooses probe points only on the bottom side of the design.
Test method	Specifies the number of probe points per net. Values are *Single, Node,* or *Flood*. The default is *Single*. *Single* indicates that one accessible testpoint per net is sufficient (for in-circuit). *Node* indicates Allegro X PCB Editor designates every endpoint on a net (that is, each connect point with only one connection) to reduce bareboard fixture density. When set to *Node,* a testpoint cannot be added to a pin unless it is a node. A pin is a node if it has clines connected to it. *Flood* indicates that Allegro X PCB Editor should designate one testpoint for every pin or via in the net (recommended for bareboard testing).
Bare board test	Specifies whether the board is populated during testing. If checked, any component pin is eligible for testing on either side of the design as long as it has a padstack defined on that side. Enabling this field automatically disables the *Allow Under Component* and *Component Representation* fields for modification; however, the settings themselves are retained. Also allows testpoints on a via if it would otherwise be ineligible, as when it is under a component. If unchecked, component pins can only be tested on the non-component side of the design.
Disable cline bubbling	Prevents bubbling to avoid DRC errors when adding a testpoint via directly on a trace or replacing a via pad while automatically or manually generating testpoints.

Text	Creates text to identify each testpoint. When you highlight or move the testpoint, the associated test highlights and moves with it.
Display	Choose to generate the net name with the testpoint. You can output this graphical data to a hardcopy plot to serve as test documentation. The MANUFACTURING/PROBE_TOP OR PROBE_BOTTOM layers store net name text. To set text sizes for testpoints, use *Add – Text* (add text command) to tailor the *Text Block* field.  testpoint testpoint N 3 5 testpoint text generated testpoint text ungenerated
	net-Alphabetic: Specifies an alphabetic incremental extension displayed on testpoints you add to the same net. The 27th extension is AA. Allegro X PCB Editor does not replace any testpoints you delete from a sequence. For example, if a net has testpoints GND-1, GND-2, and GND-3 and you delete GND-2, the next testpoint is GND-4. The separator is fixed as a dash. *net-Numeric:* Specifies a numeric incremental extension displayed on testpoints you add to the same net. The testpoint net name forms the text root, and Allegro X PCB Editor appends *-1, -2, <n>* for multiple testpoints on a net. Allegro X PCB Editor increments additional extensions to any new testpoints but does not replace any testpoints you delete from a sequence. For example, if a net has testpoints GND-1, GND-2, and GND-3,and you delete GND-2, the next testpoint is GND-4. The separator is fixed as a dash. *stringNumeric:* Specifies an arbitrary string as the root for the text identical for all testpoints created on all nets. Testprep appends an appropriate `1, 2,` *<n>* to uniquely identify all testpoints on the design. With the default string of `TP`, all testpoints on the design are identified with PROBE_ subclass text `TP1, TP2, … TP` *<n>*, where *<n>* approximates the total number of testpoints in the design. You must enable *stringNumeric* to resequence reference designators using *Manufacture – Testprep – Resequence*. ⚠ If you assigned the PROBE_NUMBER property to a net, the property overrides the *Display* options.
Rotation	Specifies the orientation of text labels. You can choose 0, 90, 180, or 270 degrees.
Offset	Specifies the position of the text measured from the pad center in the X and Y directions.
Restrictions	Specifies additional requirements imposed by various testing machines and fixtures for evaluating potential probe sites. As probes bend and become less accurate, you may impose spacing and clearance restrictions.

Test grid	Specifies grid dimensions for the test fixture. Allegro X PCB Editor chooses or inserts testpoints from pads on this grid. A grid value of zero means no grid restriction. The grid origin is the 0,0 point of the layout.
Min pad size	Specifies a minimum pad size for testpoints. If test pads are too small, testpoints may slip off during testing. No pin or via pad that is smaller than this value is chosen as a testpoint.
Allow under component	Specifies whether testpoints may exist beneath components on either side of the board. *Never Top layer only Bottom layer only Either layer*
Component representation	Choose to use ASSEMBLY or PLACE_BOUND data to determine the area that a component covers and the component outline for testpoint-to-component spacing design rule checking on both sides of the board. Note: ASSEMBLY data uses a single entity that must define a contiguous closed shape to determine the area a component covers. The shape cannot comprise discrete line segments joined together.
Padstack Selections tab	Specifies a replacement for every padstack that a via designated as a testpoint uses, which lets you account for multiple via sizes, thru and blind vias, and TOP or BOTTOM side testing.
New Via	This field is read-only. Specifies TOP and BOTTOM padstacks to use when a new via entity is created for the testpoint you are adding.
SMT Testpad	Specifies the surface-mount padstack when adding a testpoint to a TOP or BOTTOM side trace. The ... button displays a file browser that lists available database and library padstacks. The pad type you choose from the list must match that specified in the *Pad stack type* field on the *General Parameters* tab.
Thru via	Specifies the through-hole padstack when adding a testpoint to a TOP or BOTTOM side if you enabled the *Allow pin escape insertion* field (to automatically add a via to a net if no suitable test site exists) on the Testprep Automatic dialog box. This padstack must be a through-hole via with pads defined on all layers. The ... button displays a file browser that lists available database and library padstacks.
TOP Side Testpoint	Defaults to previous settings and may be read-only depending on whether the *Layer* field in the *General Parameters* tab is *Top* or *Either.* If you create a new via entity for the testpoint you are adding, enter a TOP padstack, or left click on the … button to choose a padstack from the design or library from the Select a Library Padstack dialog box that appears. Right-click to display a pop-up menu from which you can choose *Paste* or *Select All*.
BOTTOM Side Testpoint	Defaults to previous settings and may be read-only depending on whether the *Layer* field in the *General Parameters* tab is *Bottom* or *Either.* If you create a new via entity for the testpoint you are adding, enter a BOTTOM padstack, or left click on the … button to choose a padstack from the design or library from the Select a Library Padstack dialog box that appears. Right-click to display a pop-up menu from which you can choose *Paste* or *Select All*.
Enable	Click to use the replacement padstack defined for an existing via designated as a testpoint. Left click to toggle between enabling and disabling this field. Right-click to display a pop-up menu from which you can choose to enable or disable all replacement padstacks. ⚠ A replacement only occurs if you enabled *Replace Vias* in the *Padstack Selections* tab.

Existing Via	This field is read-only and displays the padstack used by an existing via designated as a testpoint. Right-click in a cell under this column to display a pop-up menu from which you can choose to: *Add:* Appends a new row. You can then enter a value in the *Existing Via* field for that row or use the drop-down list to display available padstacks with a pad on the TOP or BOTTOM layers that could be a via padstack. You cannot choose the read-only padstack already listed in the *Existing Via* field in the row above the one you added. *Delete:* Removes the chosen row. ⚠ When an existing via is used as a symbol via whose symbol is mirrored, `-mirrored` appends to the padstack name in the *Existing Via* field. If the existing via is used as both mirrored and non-mirrored, both appear in the *Existing Via* field so you can replace each with different padstacks.
TOP Side Replacement	Choose a padstack to replace the existing padstack when you designate an existing via as a testpoint on the TOP side if the *Layer* field in the *General Parameters* tab is *Top* or *Either*. You may define and enable a replacement padstack here, but a replacement only occurs if you enabled *Replace Vias* in the *Padstack Selections* tab. Right-click to display a pop-up menu from which you can choose *Set all to,* which populates all replacement cells with the contents of the initially chosen cell.
BOTTOM Side Replacement	Choose a padstack to replace the existing padstack when you designate an existing via as a testpoint on the BOTTOM side if the *Layer* field in the *General Parameters* tab is *Bottom* or *Either*. You may define and enable a replacement padstack here, but a replacement only occurs if you enabled *Replace Vias* in the *Padstack Selections* tab. Right-click to display a pop-up menu from which you can choose *Set all to,* which populates all replacement cells with the contents of the initially chosen cell.
Load selections from file	Clears existing settings on the *Padstack Selections* tab and displays a file browser from which you can choose a Comma Separated Value (`.csv`) format file containing predefined settings with which to populate the *Padstack Selections* tab. Each line of the `.csv` file is a separate data record, and a comma separates each field within the record. All records have the same number of fields. Any externally generated file should not have a header row. If the `.csv` file contains a padstack not in the design or the padstack library, or invalid for any other reason, the padstack appears in the *Padstack Selections* tab in red. If the *Enable* column is checked in the `.csv` file, but a TOP or BOTTOM side replacement padstack is invalid for any reason once you load it, the *Enable* column becomes unchecked.
Save selections to file	Displays a file browser from which you can choose a `.csv` file to which to write the current values in the *Padstack Selections* tab.
Load new existing vias	Scans the design for via padstacks missing from the *Padstack Selections* tab *TOP/BOTTOM Side Replacement* fields when you have added via padstacks to a design.

Replace vias	Replaces an existing via that is too small with a larger test via, rather than have the via fail as a testpoint, working in conjunction with the *Thru via* field. When replacing the via, Allegro X PCB Editor automatically examines the surrounding trace for any etch/conductor interference and adjusts the etch/conductor to prevent any DRCs. If DRCs occur, the testpoint is not allowed on the via. Disabling this field disables the lower via replacement settings.
Probe Types tab	Lets you name probe types, specify spacing, and designate symbol figures for each testpoint, and enable or disable these combinations.
Enable	Choose the probe types/spacing combinations that guide automatic and manual testprep as well as resequencing of probe types. If you choose two or more, testprep runs sequentially from the highest to lowest probe type (from 100 to 75, for example). Left click to toggle between enabling and disabling this field. Right-click to display a pop-up menu from which you can choose to enable or disable all probe types and associated settings.
Probe Type	Enter a name for the probe type, which must be a numeric value, such as 100, 75, or 50, for example, that corresponds to a particular probe size to be used. Right-click in a cell under this column to display a pop-up menu from which you can choose to: *Add:* Appends a new row. *Delete:* Removes the chosen row. *Re-sort:* Verify the exact sequence used by the Testprep process after you edit existing probe types or add new ones.
Probe Spacing	Enter the required minimum center-to-center pad spacing for the associated probe type in user units. A zero value means no minimum spacing requirement. Right-click in a cell under this column to display a pop-up menu from which you can choose to: *Add:* Appends a new row. *Delete:* Removes the chosen row. *Re-sort:* Verify the probe spacing sequence the Testprep process uses after you edit existing probe spacing values or add new ones.

Figure	Enter a figure to represent a testpoint, or choose one from the drop-down list. You cannot specify *Circle* because the FIXTURE_TOP and _BOTTOM subclasses created when you choose *Manufacture – Testprep – Create FIXTURE* (testprep createfixture command) use it to show fixture testpoints. Nor can you use the cross (+) because the layout editor uses it to display probe types that are too close on the PROBE_TOP/BOTTOM subclasses. (A message appears to this effect in red at the bottom of the Probe Types tab.) ⚠ Testpoints in designs prior to 15.5 lack probe types, and so use the triangle as a figure, but 15.5 designs do as well if you generate testpoints without defining probe-type settings. When you generate a testpoint with a particular probe type and figure combination, but delete the probe type, the layout editor also uses the triangle. For example, a testpoint has a probe type of 70 and a figure of square, but you delete the probe type setting of 70 while the testpoint still exists. The pre-existing testpoint continues to maintain its probe type 70, as it is an attached attribute, but the table setting that defines the figure to use no longer exists. The probe type 70 with the now missing setting defaults to a triangle as a figure. When the same probe type exists multiple times, but *Probe Spacing* values differ for each one, all instances of that probe type use the same figure. Different figures cannot be associated with the same probe type. Right-click in a cell under this column to display a pop-up menu from which you can choose to: *Add:* Appends a new row. *Delete:* Removes the chosen row. *Re-sort:* Verify the probe type sequence the testprep process uses after you edit existing figures or add new ones.
Load types from file	Clears existing settings on the *Probe Types* tab and displays a file browser from which you can choose a Comma Separated Value (`.csv`) format file containing predefined settings with which to populate the *Probe Types* tab. Each line of the `.csv` file is a separate data record, and a comma separates each field within the record. All records have the same number of fields. The file's first line is the header row, which specifies the names of each field. If the `.csv` file contains a padstack not in the design or the padstack library, or invalid for any other reason, the padstack appears in the *Probe Types* tab in red. If the *Enable* column is checked in the .csv file, but a TOP or BOTTOM side replacement padstack is invalid for any reason once you load it, the *Enable* column becomes unchecked.
Save types to file	Displays a file browser from which you can choose a `.csv` file to which to write the current values in the *Probe Types* tab.
Close	Closes the dialog box and saves any changes.
Cancel	Closes the dialog box and discards any changes.

Related Topics

- testprep prmed
- Setting Testprep Parameters
- Specifying Replacement Padstacks for New Vias
- Specifying Replacement Padstacks for Existing Vias
- Optimizing Probe Types and Spacing

Setting Testprep Parameters

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *General Parameters* tab.
3. Specify the type of pin that can be chosen for testing in the *Pin type* field.
4. Choose the type of pad needed as a contact point for the test probe in the *Pad stack type* field.
5. Specify the side of the design on which to locate the testpoint in the *Layer* field.
6. Specify the type test methodology in the *Test method* field.
7. Choose the *Bare board test* field to populate the board during testing.
8. Choose *Disable cline bubblin*g to prevent bubbling.
9. Choose the *Display* field to generate the net name with the testpoint and one of the following formats in which to generate the text.
   - *net-Alphabetic:* Specifies an alphabetic incremental extension displayed on testpoints you add to the same net.
   - *net-Numeric:* Specifies a numeric incremental extension displayed on testpoints you add to the same net. (default)
   - s*tringNumeric:* Specifies an arbitrary string as the root for the text that is the same for all testpoints created on all nets.
10. Specify the orientation of text labels in the *Rotation* field. You can choose 0, 90, 180, or 270 degrees.
11. *S*pecify the position of the text relative measured from the center of the pad in the X and Y directions in the *Offset* field.
12. In the *Test grid* field, specify grid dimensions for the test fixture. Allegro X PCB Editor chooses or inserts testpoints from pads on this grid. A grid value of zero means that there is no grid restriction. The grid origin is the 0,0 point of the layout.
13. In the *Min pad size* field, specify a minimum pad size for testpoints. If test pads are too small, testpoints may slip off of them during testing. No pin or via pad that is smaller than this value is chosen as a testpoint.
14. In the *Allow under component* field, specify whether testpoints are allowed under components for each side of the board.
15. In the *Component representation* field, choose to use ASSEMBLY or PLACE_BOUND data to determine the area covered by a component and the component outline for the new testpoint-to-component spacing DRC on both sides of the board.
16. Use the *Padstack Selections* tab to specify replacement padstacks for each testpoint via, and the *Probe Types* tab to optimize probe types and testpoint spacing.
17. Click *Close* to save any changes.

Using a .csv File of Predefined Settings to Populate the Padstack Selections Tab

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *Padstack Selections* tab.
3. Click *Load selections from file* to clear existing settings on the *Padstack Selectio*ns tab and display a file browser from which you can choose a Comma Separated Value (`.csv`) format file containing predefined settings with which to populate the *Padstack Selection*s tab.
   The *Padstack Selections* tab populates with the contents of the `.csv` file.

   ⚠ If the `.csv` file contains a padstack not in the design or the padstack library, or invalid for any other reason, the padstack appears in the *Padstack Selections* tab in red.

Saving Padstack Selections Tab Values to a .csv File

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *Padstack Selections* tab.
3. Click *Save selections to file.* A file browser appears from which you can choose a `.csv` file to which to write the current values in the *Padstack Selections* tab. The `.csv` file populates with the contents of the *Padstack Selections* tab.

Related Topics

- testprep prmed
- Testprep Parameters Dialog Box
- Specifying Replacement Padstacks for New Vias
- Specifying Replacement Padstacks for Existing Vias
- Optimizing Probe Types and Spacing
- testprep automatic

Specifying Replacement Padstacks for New Vias

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *Padstack Selections* tab.
3. To create a new via entity when adding a testpoint to a TOP or BOTTOM side trace, in the *New Via* section, do one of the following:
    a. Enter a TOP or BOTTOM side testpoint for the surface-mount padstack in the *TOP* or *BOTTOM Side Testpoint* field for the *SMT Testpad*.
    b. Click the ... button to display a file browser that lists available database and library padstacks. The pad type you choose from the list must match the type specified in the *Pad stack type* field on the General Parameters tab.
4. Right-click to display a pop-up menu from which you can choose *Undo, Cut, Copy, Paste, Delete,* or *Select All*.
5. To create a new via entity when adding a testpoint to a TOP or BOTTOM side if you enabled the *Allow pin escape insertion* field on the Testprep Automatic dialog box, in the *New Via* section, do one of the following:
    a. Enter a TOP or BOTTOM side testpoint for the through-hole padstack in the *TOP* or *BOTTOM Side Testpoint* field for the *Thru Via*. This padstack must be a through-hole via with pads defined on all layers.
    b. Click the ... button to display a file browser that lists available database and library padstacks. The pad type you choose from the list must match the type specified in the *Pad stack type* field on the General Parameters tab.
6. Choose *Replace vias* to substitute an existing via that is too small with a larger test via, rather than have the via fail as a testpoint.
7. Click *Close*.
8. Choose *Manufacture – Testprep – Automatic* (`testprep automatic` command).The Testprep Automatic dialog box appears.
9. Click *Generate testpoints* to run the automatic testpoint process using the modified values.

Related Topics

- testprep prmed
- Testprep Parameters Dialog Box
- Setting Testprep Parameters
- Specifying Replacement Padstacks for Existing Vias
- Optimizing Probe Types and Spacing
- testprep automatic

Specifying Replacement Padstacks for Existing Vias

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *Padstack Selections* tab.
3. Right-click in a cell under the *Existing Via* column, which is read-only and displays the padstack used by an existing via, to display a pop-up menu from which you can choose to:
   a. *Add:* Appends a new row. You can then enter a value in the *Existing Via* column for that row or use the drop-down list that appears to display available padstacks with a pad on the TOP or BOTTOM layers that could be a via padstack. You cannot choose the read-only padstack already listed in the *Existing Via* field in the row above the one you added.
   b. *Delete:* Removes the chosen row.
4. To replace the existing padstack for the via listed in the *Existing Via* column, designated as a testpoint on the TOP or BOTTOM side:
   a. Choose a padstack in the *TOP/BOTTOM Side Replacement* field in the *Existing Via* column.
   b. Click the ... button to display a file browser that lists available database and library padstacks from which you can choose.
5. Right-click to display a pop-up menu from which you can choose *Set all* to populate all replacement cells with the contents of the initially chosen cell.
6. Click *Enable* to use the replacement padstack defined for an existing via designated as a testpoint. Left click to toggle between enabling and disabling this field. Right-click to display a pop-up menu from which you can choose to enable or disable all replacement padstacks.
7. Choose *Replace vias* to substitute an existing via that is too small with a larger test via, rather than have the via fail as a testpoint.
8. Click *Close*.
9. Choose *Manufacture – Testprep – Automatic* (`testprep automatic` command).The Testprep Automatic dialog box appears.
10. Click *Generate testpoints* to run the automatic testpoint process using the modified values.

Related Topics

- testprep prmed
- Testprep Parameters Dialog Box
- Setting Testprep Parameters
- Specifying Replacement Padstacks for New Vias
- Optimizing Probe Types and Spacing
- testprep automatic

Optimizing Probe Types and Spacing

1. Choose *Manufacture – Testprep – Parameters* (`testprep prmed` command). *The Testprep Parameters dialog box appears.*
2. Choose the *Probe/Type Spacing* tab.
3. To populate the *Probe Types* tab with predefined probe type and spacing combinations, (optional) click *Load types from file to* display a file browser from which you can choose a Comma Separated Value (.`csv`) format file that contains the combinations.
4. In the *Probe Type* field, enter a numeric value that corresponds to the probe size to be used, such as 100, 75, or 50, for example. Right-click in a cell under this column to display a pop-up menu from which you can choose to:
   Add: Appends a new row.
   Delete: Removes the chosen row.
   Re-sort: Verify the probe type sequence the Testprep process uses after you edit existing probe types or add new ones.
5. Enter the required minimum center-to-center pad spacing for the associated probe type in user units in the *Probe Spacing* field. A zero value means no minimum spacing requirement. Right-click in a cell under this column to display a pop-up menu from which you can choose to:
   Add: Appends a new row.
   Delete: Removes the chosen row.
   Re-sort: Verify the probe spacing sequence the Testprep process uses after you edit existing probe spacing values or add new ones.
6. Enter a figure to represent a testpoint, or choose one from the drop-down list.

   > ⚠ You cannot specify *Circle* because the FIXTURE_TOP and _BOTTOM subclasses created when you choose *Manufacture – Testprep – Create FIXTURE* (`testprep createfixture` command) use it to show fixture testpoints. Nor can you use the cross (+) because the layout editor uses it to display probe types that are too close on the PROBE_TOP/BOTTOM subclasses. (A message appears to this effect in red at the bottom of the *Probe/Type Spacing* tab.)

   Testpoints in designs prior to 15.5 lack probe types, and so use the triangle as a figure, but 15.5 designs do as well if you generate testpoints without defining probe-type settings. When you generate a testpoint with a particular probe type and figure combination, but delete the probe type, the layout editor also uses the triangle. For example, a testpoint has a probe type of 70 and a figure of square, but you delete the probe type setting of 70 while the testpoint still exists. The pre-existing testpoint continues to maintain its probe type 70, as it is an attached attribute, but the table setting that defines the figure to use no longer exists. The probe type 70 with the now missing setting defaults to a triangle as a figure.

   > ⚠ When the same probe type exists multiple times, but *Probe Spacing* values differ for each one, all instances of that probe type use the same figure. Different figures cannot be associated with the same probe type.

   Right-click in a cell under this column to display a pop-up menu from which you can choose to:
   Add: Appends a new row.
   Delete: Removes the chosen row.

Re-sort: Verify the figure sequence the Testprep process uses after you edit existing figures or add new ones.

7. Choose the probe types and spacing combinations that guide automatic testprep by clicking the *Enable* field. If you choose two or more, testprep runs sequentially from the highest to lowest probe type (from 100 to 75, for example). Left click to toggle between enabling and disabling this field. Right-click to display a pop-up menu from which you can choose to enable or disable all probe types and associated settings.
8. Click *Close*.
9. Choose *Manufacture – Testprep – Automatic* (`testprep automatic` command).The Testprep Automatic dialog box appears.
10. Click *Generate testpoints* to run the automatic testpoint process using the modified values.

Related Topics

- testprep prmed
- Testprep Parameters Dialog Box
- Setting Testprep Parameters
- Specifying Replacement Padstacks for New Vias
- Specifying Replacement Padstacks for Existing Vias
- testprep createfixture
- testprep automatic

testprep properties

Lets you add testprep-related properties to a single net or symbol. You can also display information regarding the other testpoints on that net and assigned properties, which appears in the Testprep Query Dialog Box. Displayed information includes:

- whether a pin or via is a testpoint;
- if a testpoint via was created directly on a trace, auto inserted, or had its pad replaced;
- existing testpoints on the net, or the net of the identified pin or via;
- the existence of NO_TEST and TESTPOINT_QUANTITY net properties;
- whether any TESTPOINT_QUANTITY property is currently being met;
- and if a pin was chosen, any existence of the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY property on its parent symbol.

> ⚠ You can also add or delete testprep-related properties for numerous nets or symbols using *Edit – Properties* (`property edit` command).

Related Topics

- Options Tab for the `testprep properties` Command
- Excluding Nets from Testing
- Adding Multiple Test points to a Net
- Deleting a Testpoint Property from a Net or Symbol
- Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol
- property edit
- Preparing Manufacturing Data

Options Tab for the testprep properties Command

- Menu path: *Manufacture – Testprep – Properties*

Testprep Properties	
Mode	
	Add: Choose to add a property to the net or symbol you choose. *Delete*: Deletes the chosen property from the net or symbol identified by subsequent cursor digitizations. *Query*: After choosing a pin, via, or
Property	
Net	*NO_TEST: Attach this* property to nets that do not require test probes. You can also use the NO_TEST property to apply parameters to different nets by skipping them for a specific pass, changing the
Symbol	TESTPOINT_ALLOW_UNDER Attach this property to a symbol to allow testpoints underneath a component instance of a symbol and override the *Allow under component*field on the Testprep

Related Topics

- testprep properties
- Excluding Nets from Testing
- Adding Multiple Test points to a Net
- Deleting a Test point Property from a Net or Symbol
- Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol
- testprep automatic
- testprep manual
- *testprep density*

Excluding Nets from Testing

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command) and click *Properties* or `choose` *Manufacture – Testprep – Properties* (`testprep properties` command).
2. Choose *Add* as the mode on the *Options Tab* of the control panel.
3. Choose NO_TEST as the property to add.
4. Choose the net to which to add the NO_TEST property.
5. Allegro X PCB Editor issues the command line message:

   Adding property NO_TEST to net N.

Related Topics

- testprep properties
- Options Tab for the `testprep properties` Command
- Adding Multiple Test points to a Net
- Deleting a Test point Property from a Net or Symbol
- Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol
- testprep manual

Adding Multiple Test Points to a Net

You can control the number of test points per net. Power and Ground nets usually require an amount proportional to the number of power and ground pins on the PCB. Special logical nets may also require additional test points.

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command) and click *Properties* or *Manufacture – Testprep – Properties* (`testprep properties` command).
2. Choose *Add* as the mode.
3. Choose TESTPOINT_QUANTITY.
4. Enter the number of test points.
5. Click on the net to which to add the TESTPOINT_QUANTITY property.
6. Allegro X PCB Editor issues the command line message:

   Adding property TESTPOINT_QUANTITY with quantity N to net N.

7. Choose *Query* to verify your changes.

Related Topics

- testprep properties
- Options Tab for the `testprep properties` Command
- Excluding Nets from Testing
- Deleting a Test Point Property from a Net or Symbol
- Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol
- testprep manual

Deleting a Test Point Property from a Net or Symbol

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command) and click *Properties,* or choose *Manufacture – Testprep – Properties* (`testprep properties` command).
2. Choose *Delete* as the mode.
3. Choose the property to be deleted.
4. Choose the net or symbol from which to delete the property.
5. Allegro X PCB Editor issues a command line message:
   Property N removed from N elements.
6. Choose *Query* to verify your changes.

Related Topics

- testprep properties
- Options Tab for the `testprep properties` Command
- Excluding Nets from Testing
- Adding Multiple Test Points to a Net
- Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol
- testprep manual

Adding the TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY Property to a Symbol

1. Choose *Manufacture – Testprep – Manual* (`testprep manual` command) and click *Properties,* or choose *Manufacture – Testprep – Properties* (`testprep properties` command).
2. Choose *Add* as the mode.
3. Choose TESTPOINT_ALLOW_UNDER or TESTPOINT_MAX_DENSITY.
4. Choose the symbol to which to add the property.
5. Allegro X PCB Editor issues one of the following command line messages depending on the property you added:

   Adding property TESTPOINT_ALLOW_UNDER to symbol instance N.

   Adding property TESTPOINT_MAX_DENSITY with maximum N to symbol instance N.
6. Choose *Query* to verify your changes.

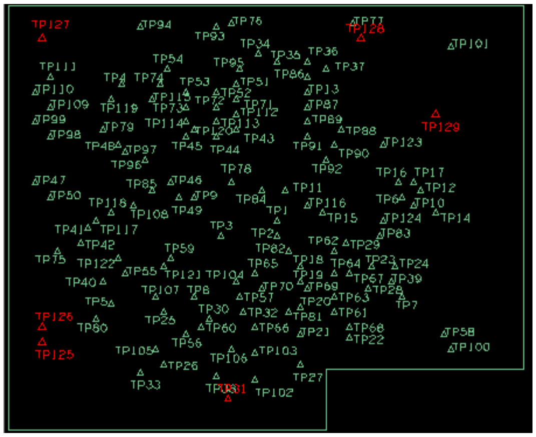
Related Topics

- testprep properties
- Options Tab for the `testprep properties` Command
- Excluding Nets from Testing
- Adding Multiple Test points to a Net
- Deleting a Test point Property from a Net or Symbol
- testprep manual

testprep resequence

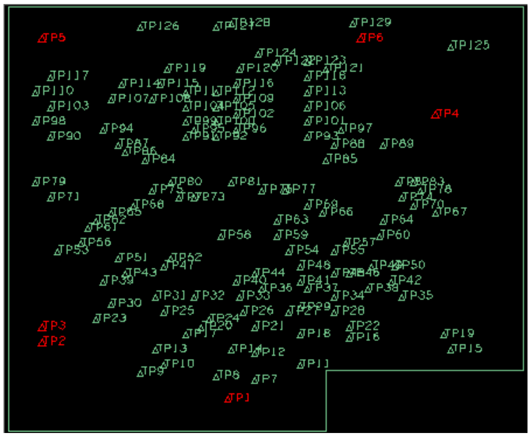
Renames the refdes text of testpoints to ensure a visually sequential appearance, sorted by X/Y location from left to right and bottom to top on each side, starting with the TOP side first and then the BOTTOM side. The following figure shows the refdes text of testpoints prior to resequencing.

Figure -1 Testpoints prior to resequencing



The following figure shows testpoints after resequencing.

Figure -2 Testpoints after resequencing



Probe types may need to be resequenced after you initially create testpoints with particular probe type/spacing combinations enabled and then subsequently change vias that may be testpoints, due to violations. Testpoints may be too close for the probe type. Or a probe type of 100 may be more appropriate for a testpoint initially assigned a probe type of 75 given its proximity to another testpoint, which has since been deleted or moved.

Related Topics

- Testprep Resequence Dialog Box
- Resequencing Reference Designators of Test points
- Resequencing Probe Types
- Preparing Manufacturing Data

Testprep Resequence Dialog Box

- Menu path: *Manufacture – Testprep – Resequence*

Reference Designators	Choose to sort testpoint locations from left to right and bottom to top across the board, first the TOP side and then the BOTTOM side and create the `testprep_resequence.log` log file. ⚠ You must enable *stringNumeric* in the *Display* field in the Testprep Parameters dialog box to resequence reference designators.
Probe Types	Choose to resequence probe types.
Delete Probes Too Close	Only available when you enable *Probe Type*. Choose to remove testpoints’ probe types when they cannot be changed to a valid probe type; that is, a resequencing causes two testpoints to be closer than the probe type specifies. If you do not choose this option, the testpoints remain in the design, but probe types appears as TOO CLOSE in the Testprep report, Testprep Manual Query, or use the query available by choosing *Display – Element* `(show element` command).
Resequence	Click to resequence by reference designators or probe types.
Parameters	Choose to display the Testprep Parameters Dialog Box.
Close	Closes the dialog box and saves any changes.
Viewlog	Click to review the `testprep_resequence.log` file.

Related Topics

- testprep resequence
- Resequencing Reference Designators of Test points
- Resequencing Probe Types
- show element

Resequencing Reference Designators of Test Points

1. Choose *Manufacture – Testprep – Resequence* (`testprep resequence` command).
2. To sort testpoint locations from left to right and bottom to top across the board, first the TOP side and then the BOTTOM side, enable the *Reference Designators* field.

   ⚠ You must enable *stringNumeric* in the *Display* field in the Testprep Parameters Dialog Box to resequence reference designators.

3. To modify settings that govern the testprep process, click *Parameters* to display the Testprep Parameters Dialog Box if necessary.
4. Click *Resequence* to resequence by reference designators.
5. Click *Close* to exit the dialog box and save any changes.
6. Click *View log* to review the `testprep_resequence.log` file.

Related Topics

- testprep resequence
- Testprep Resequence Dialog Box
- Resequencing Probe Types

Resequencing Probe Types

1. Choose *Manufacture – Testprep – Resequence* (`testprep resequence` command)*.*
2. To resequence probe types, enable the *Probe Types* field.
3. Enable the *Delete Probes Too Close* field to remove testpoints' probe types when they cannot be changed to a valid probe type; that is, resequencing causes two testpoints to be closer than the probe type specifies.
   If you do not choose this option, the testpoints remain in the design, but probe types appear as TOO CLOSE in the Testprep report, Testprep Manual Query dialog box, or when you choose *Display – Element (*`show element` command).
4. To modify settings that govern the testprep process, click *Parameters* to display the Testprep Parameters Dialog Box if necessary.
5. Click *Resequence* to resequence probe types.
6. Click *Close* to exit the dialog box and save any changes.
7. Click *View log* to review the `testprep_resequence.log` file.

Related Topics

- testprep resequence
- Testprep Resequence Dialog Box
- Resequencing Reference Designators of Test points

text

The `text` command is used in conjunction with `text edit` and `add text`. It allows you to put a line of text into the design database.

Related Topics

- text edit
- `add text`

text edit

Modifies a text string in a design. For text that is a standalone note, the modification changes the text. If the text is a reference designator, the modification updates the database.

> ⚠ You cannot edit a device label.

Text edit characteristics:

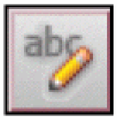
- Highlights the text and displays the text cursor on the first character location of the text string
- Overwrites the existing text
- Lets you choose another text string for editing

This command functions in a pre-selection use model, in which you choose an element first, then right-click and execute the command. It is only available on the right mouse button pop-up menu when you are working in the general edit application mode. Elements ineligible for use with the command generate a warning and are ignored.

> ⚠ If you enable infinite cursor on a Windows machine, the `text edit` command changes infinite cursor to cross cursor.

Access using:

- Menu path: *Edit – Text*
- Toolbar icon:



Text Edit Dialog Box

Enter new text	Creates a new text string.
OK	Replaces existing text with modified text.
Cancel	Closes the dialog box without changes to your design.

Related Topics

- Modifying Text in a Design
- text

Modifying Text in a Design

1. Hover your cursor over a text string. The tool highlights the element and a datatip identifies the current contents of the text string.
2. Right-click and choose *Edit Text* from the pop-up menu.
   The Text Edit dialog box appears.
3. Enter the new text and click *OK.*
   The modified text string replaces the existing text string.

Related Topics

- text edit
- text

thermal_via

The `thermal_via` command lets you create thermal vias in your design. Thermal vias are defined as mechanical symbols (`.bsms`). They are typically used in multi-chip modules in Advanced Package products.

thieving

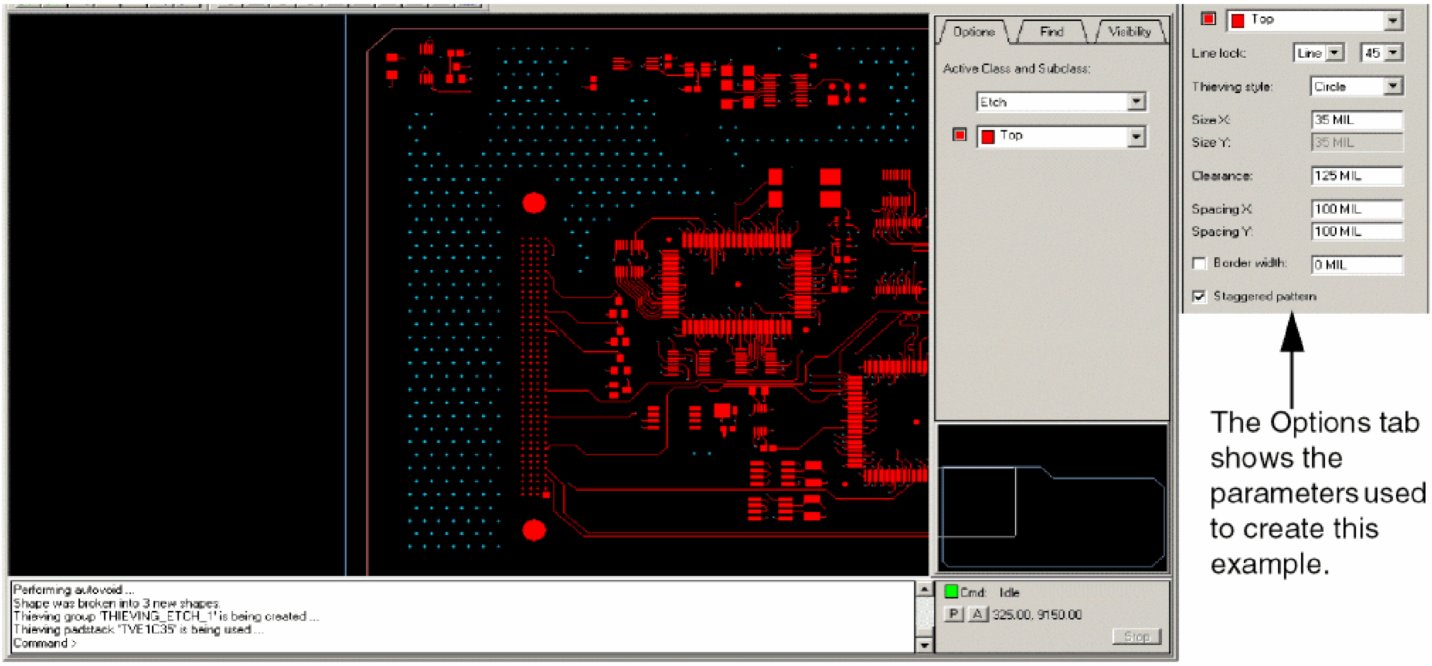
The `thieving` command lets you add a pattern of non-conductive, single-layer figures to areas on the outer layers of a printed circuit board that do not contain copper. You generate the thieving pattern to balance the plating distribution, placing it to avoid interference with the signal quality of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

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

Example

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

Example of a Staggered Thieving Pattern



Line lock: Line 45
Thieving style: Circle
Size X: 35 MIL
Size Y: 35 MIL
Clearance: 125 MIL
Spacing X: 100 MIL
Spacing Y: 100 MIL
□ Border width: 0 MIL
☑ Staggered pattern

```
Performing autovoid ...
Shape was broken into 3 new shapes.
Thieving group 'THIEVING_ETCH_1' is being created ...
Thieving padstack 'TVE1C35' is being used ...
Command >
```

Cmd: Idle
P A 325.00, 9150.00

The Options tab shows the parameters used to create this example.

Related Topics

- Options Tab for the `thieving` Command
- Creating a Thieving Pattern
- reports

Options Tab for the thieving Command

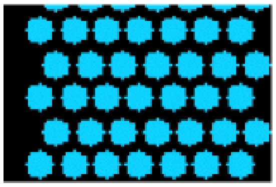
Access using:

- Menu path: *Manufacturing – Thieving*
- Toolbar icon: ▦

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

> ✓ You can also edit thieving parameters by using the *Design Parameter Editor*. Choose *Setup – Design Parameters* (`prmed` command), then click *Edit thieving parameters* under the *Mfg Applications* tab.

Active Class and Subclass	Displays the current parameters and provides a drop-down menu for modifying them. The choices are any subclass of the *ETCH/CONDUCTOR* class and SOLDERMASK_TOP and SOLDERMASK_BOTTOM subclasses of the *Board Geometry* class.
Line Lock	Controls whether the segment type is a straight line or an arc. You can also define the corner angle when a segment changes direction. The choices are *Line*, *Arc* or *Off*, and *45* or *90*.
Thieving Style	Displays the figure style *within the thieving pattern. You can choose Circle or Rectangle or Line.*
Thieving outline	Displays the type of shape for creating thieving outline. You can choose *Shape* or *Rectangle*.The default is *Shape.*
Size X	Specifies the X dimension of the figures. The value must be a positive integer. If you choose *Circle* as the thieving style, *Size X* specifies the diameter. If you choose *Line* and if X is larger in value than Y, the line will be horizontal; X specifying the length and Y specifying the width.
Size Y	Specifies the Y dimension of the figures. The value must be a positive integer. If you choose *Rectangle*, equal values create a square figure. If you choose *Circle* as the thieving style, you cannot edit this field. If you choose *Line* and if Y is larger in value than X, the line will be vertical; Y specifying the length and X specifying the width.
Clearance	Specifies the distance between the thieving pattern and all other objects on the active subclass. The value must be a positive integer.The layout editor uses this value when autovoiding. ⚠ When you generate a thieving pattern, it adheres to route or via keepout boundaries within the thieving outline. If a dynamic shape exists within the outline, the thieving pattern clears around it.
Spacing X	Specifies the center-to-center distance between the figures within the thieving pattern.
Spacing Y	Specifies the Y spacing between the figures. For uniform spacing, set an equal value to *Spacing X*.
Border Width	Specifies the width of the border surrounding the thieving area. The border is optional. The value you enter must be zero or greater.

Staggered Pattern	*Specifies that* every other row appears in an offset pattern, as shown below:  Checked is the default. Deselect it to align the pattern in straight rows and columns.
Clip to Route Keepin	Keeps thieving vias within the route keepin area. Specifies the area for thieving. This area is defined as the intersection of the route keepin area and the thieving boundary area.
All etch layers	Thieving is generated for each positive etch layer of the design. ⚠ If enabled ignores the current settings for *Active Class and Subclass.*
All soldermask layers	Thieving is generated for each soldermask layer of the design. ⚠ If enabled ignores the current settings for *Active Class and Subclass.*
Offset layers	Allows thieving to be generated on all etch/soldermask layers at the same time if these options are selected. The pattern of thieving is offset on adjacent layers.

Related Topics

- thieving
- Creating a Thieving Pattern

Creating a Thieving Pattern

1. Choose *Manufacturing – Thieving* (`thieving` command).

   The *Options* tab changes to display the thieving options. The console window prompt instructs you to enter a thieving outline.

2. Change the parameters in the *Options* tab.

   This step is optional, because you can accept the current settings.

3. Outline the area to fill.

4. Right-click to display the pop-up menu and choose *Done*.

   The layout editor automatically completes the thieving process. The console window prompt displays the following status messages:

   ```
   Validating thieving parameters

   Executing thieving

   Creating voids

   Performing autovoid

   Shape was broken into 8 new shapes.
   ```

   The console window prompt also displays a name for the thieving pattern and the thieving figures, as in the following examples:

   ```
   Thieving group 'THIEVING_ETCH_TOP_1' is being created

   Thieving padstack 'TVE1C35' is being created.
   ```

   The thieving pattern appears in the design.

   If you made a mistake, the console window prompt displays one of the following warning messages:

   ```
   W- WARNING: Not enough spacing in X for the specified size.

   W- WARNING: Not enough spacing in Y for the specified size.
   ```

   Cause: The settings for size, spacing, and a non-staggered pattern result in overlapping pattern figures. The thieving pattern still appears, but the results may not be acceptable.

   Response: Reset the parameters and generate a new thieving pattern.

   ```
   W- WARNING: Spacings don't look right for the specified size with staggering.
   ```

   Cause: The settings for size, spacing, and a staggered pattern may result in overlapping pattern figures. The thieving pattern still appears, but the results may not be acceptable.

   Response: Reset the parameters and generate a new thieving pattern.

   If the layout editor does not generate a thieving pattern, the console window prompt displays one of the following warning messages:

   ```
   W- WARNING: Nothing in the thieving group ... cancelled!

   W- WARNING: A proper outline was not specified for thieving!::
   ```

   Cause: The layout editor was not able to generate the thieving pattern.

   Response: Reset the parameters and generate a new thieving pattern.

⚠ Once you generate a thieving pattern, the layout editor handles the pattern as a group.

⚠ Thieving is not supported on negative plane layers.

Related Topics

- thieving
- Options Tab for the `thieving` Command
- Working with Groups and Modules
- Working with Groups

tiv text in

The `tiv text in` command imports a text file that contains fixed via array data or routing structures, and places the fixed pattern of vias at a specified location. The vias are grouped together into a symbol that moves as a unit and cannot be modified.

The `tiv text in` command also generates a symbol definition, which includes a Design for Assembly (DFA) boundary.
Available when the *Silicon Layout* option is selected in Allegro X Advanced Package Designer.

Related Topics

- tiv text out
- Importing Via Array Data
- Exporting Via Array Data

Via Array Text-In Wizard

Access using

- Menu Path: *Si Layout – TIV Text In*

Via Array Text-In Wizard, Step 1: File Selection Dialog Box

Standard file browser to specify the file containing the via array data.

Via Array Text-In Wizard, Step 2: File Information Dialog Box

Coordinates	Specifies the measurement unit that is used to represent the via data; mils, inches, millimeters, centimeters, or microns. By default, the unit for via data is micron.
Delimiters	Select the column delimiters used in the text-in file being imported.
Tab	Select to specify *tab* as the column delimiter.
Semicolon	Select to specify *semicolon* as the column delimiter.
Comma	Select to specify *comma* as the column delimiter.
Space	Select to specify *space* as the column delimiter.
Other	Select to specify a delimiter other than tab, space, semicolon, or comma. Type in the character you want to use in the box to the right.
Ignore consecutive delimiters	Select to treat consecutive delimiters as one delimiter.
Remove trailing delimiters	Select to remove trailing delimiters from the data.
Back	Click to return to the previous dialog box.
Next	Click to display the next dialog box.
Cancel	Discards all changes and closes the dialog box.

Via Array Text-In Wizard, Step 3: Column Information Dialog Box

Information contained within this dialog box includes the saved grid parameters for the symbol. Editing grid parameters is *not* recommended.

Ignore Row	Select the corresponding column in a row to ignore the information in that row. The selected rows will not be imported.
Padstack	Specifies the padstack type of each via in the array.
X Coordinate	Displays the X coordinate of each via in the array. (Required. You cannot proceed to the next step if no column is denoted for the X coordinate.)
Y Coordinate	Specifies the Y coordinate of each via in the array. (Required. You cannot proceed to the next step if no column is denoted for the Y coordinate)
Rotation	Specifies the value in degrees of each via in the array.
Net Name	Specifies the net names assigned to each via in the array.
Back	Click to return to the previous dialog box.
Next	Click to display the next dialog box.
Cancel	Discards all changes and closes the dialog box.

Via Array Text-In Wizard, Step 4: Final Confirmation Dialog Box

Run purge unused nets on exit	Runs the `purge unused net` command to remove any unused nets from your design. This option is selected by default.
Run derive assignment on exit	Runs the derive assignment command to check the display for unconnected shapes and incomplete netlists and to automatically assign the connections from the existing conductor pattern. This option is selected by default.
Push the new connectivity out to the rest of the design	Propagates new signal names to the connected elements. This is option is not selected by default.
View Log	Displays the log file generated upon creation of the via array.
Back	Steps you back to previous dialog boxes in order to change settings.
Finish	Saves the created array and completes the command session.
Cancel	Discards the created array and completes the command session.

Related Topics

- purge unused nets

Importing Via Array Data

Via array information must be in ASCII text formats since, the *Via Array Text-In Wizard* processes only ASCII text files.

The *Via Array Text-In Wizard* can also import design information that was previously exported using the *Via Array Text-Out Wizard.*

1. Choose *Si Layout – Via Array Text-In Wizard.*
   Alternatively, type `tiv text in` in the command window.
   The *File Selection* dialog box appears.
2. Choose the file that contains the via information.
   The text file opens and appears as a numbered list.
3. Click *Next* to display the *File Information* dialog box.
4. Specify how the text file will be parsed by selecting coordinates and delimiters.
   Each data point needed to create the via array symbol must be in a separate data field (column) in one record (row).
5. Click *Next* to display the *Column Information Dialog Box*.
6. Right-click the heading and choose the correct type of information represented in the column.
7. Click *Next* to display the Final Confirmation Dialog Box.
8. Click *Finish* to create the via array, *Back* to change your settings, or *Cancel* to terminate the wizard without saving the created via array depending on the state of the via creation.

Related Topics

- tiv text in
- tiv text out
- Exporting Via Array Data

tiv text out

The `tiv text out` command creates a text file of fixed via array data.

⚠ Available when *Silicon Layout* option is selected in Allegro X Advanced Package Designer.

The *TIV Text-Out Wizard* presents a series of dialog boxes to guide you through the process of exporting data for a pattern of fixed vias when you run the program.

Related Topics

- tiv text in
- Exporting Via Array Data
- Importing Via Array Data

TIV Text-Out Wizard Dialog Box

Access using

- Command line: `tiv text out`

Column header buttons	Above each column, a heading displays the type of information in the column. To change the order of columns or to assign a blank button an information type, right-click a column header and choose the type of data you want displayed in that column. Available column headers are: • *Remove*: Removes a column from the Export Via output. Once you remove a column, you can only get it back by canceling the operation and starting it over. • *Padstack*: Specifies the padstack type of each via in the array. • *X Coord*: Specifies the *x* coordinate of each padstack. • *Y Coord:* Specifies the *y* coordinate of each padstack. • *Rotation*: Specifies the rotation value in degrees of each via in the array. • *Net Name*: Specifies the net names that are assigned to each via in the array. If the net does not already exist for a via, create it in this column.
Include Column Headers	Specifies whether the column headers are included in the output. (This does not affect the file headers specified in the previous dialog box.)
Duplicate Pin Information	Lets you display or hide repetitive information. For example, there may be a single net with many package pins assigned to it. The net name in this case can be displayed in just the first occurrence or in each subsequent occurrence.
Mirror coordinates in y-axis	Click to display mirrored coordinate values for array vias.
Preview	Click to display the output before it is written to a file.
Sort	Click to sort the package information by up to three criteria in ascending or descending order.
Back	Click to return to the previous dialog box.
OK	Removes from your design all the nets listed in the Delete window, closes the dialog box, and returns the tool to an idle state.
Cancel	Closes the dialog box without changes.

Exporting Via Array Data

Perform the following steps to export via array data:

1. Run `tiv text out` from the command line.
   The *TIV Text-Out Wizard* appears.

   > ⚠ If your design contains only one valid component, this dialog box does not appear.

2. Choose the reference designator of the component you want to export.
3. Click *OK*.
   A standard file browser appears.
4. Name the file in which the data is to be stored, then click *Save*.
   The *TIV Text-Out Wizard Header Information* dialog box appears.
5. Choose the headers that you want included in the exported data file, then click *Next*.
   The *TIV Text-Out Wizard Column Information* dialog box appears.
6. Specify column information according to the description in *TIV Text-Out Wizard Column Information Dialog Box*.
7. Click *OK* when the columns are organized the way you want to write it to a file.

Related Topics

- tiv text out
- Importing Via Array Data

tldelay

The `tldelay` batch command analyzes the etch on a design and estimates etch delay. It generates a generic pin-to-pin delay and net capacitance report. Before executing `tldelay`, specify the characteristics of each etch layer on the Cross-section Editor form. `tldelay [-f netname.lst][-v capacitance] [-m factor][-M A|D][-p] drawingname`

-f netname.lst	Specifies a text file that identifies the nets to be analyzed, one net per line. Without this option, the program analyzes all nets.
-v capacitance	Specifies the value for the via capacitance in picofarads. Without this option, the program ignores the effects of vias.
-m factor	Specifies the value by which the manhattan distance is to be multiplied, to help account for traces backtracking during routing. Typical values range between 1 and 1.3; 1.0 is the default.
-M A|D	Specifies sort order for drivers. If a driver drives multiple receivers, -M A reports values in ascending order and -M D reports them in descending order. -M D is the default.
-p	Causes the program to ignore PINUSE codes. Use this option for diagnostic reasons, when you suspect PINUSE codes have been used incorrectly. When this option is chosen, the program typically outputs a pin-to-pin manhattan-only delay, based on the random order of pins in the drawing.

The `tldelay` command generates a text file in the format used by the `extracta` command. The file is named `drawingname.dly` and is placed in the current working directory.

Related Topics

- `extracta`
- tldphys

tldphys

The `tldphys` batch command uses the information from the `drawingname.dly` file, generated during `tldelay` processing, to create the `physdly.dat` file. This file contains one delay value per receiver pin. When an input/bidirectional pin can receive signals from more than one source, the delay for the first logical driver pin on the network is reported. If you used the -M D option for `tldelay` processing, then the longest delay for each receiver is listed first. If you use the -M A option, then the shortest delay for each receiver is listed first.

```
tldphys drawingname.dly
```

The `tldphys` command is typically executed automatically during `a2vtime` or `a2vsim` processing. To execute it from the command line, use the syntax shown above. The `tldphys` command outputs physical refdes.pin# pin designators into the `physdly.dat` file. This file is used as input for `gphysdly`, which translates the names from physical to logical names.

Backannotation should be performed before executing `tldphys`.

Related Topics

- tldphys
- tldelay

tline calculator

The `tline calculator` command lets you estimate electrical performance for different transmission line structures. The calculator results are for layout reference only. For more precise results, see the simulation tools available with your product.

> ⚠ Available in Allegro Advanced Package Designer when *SiP Layout* option is selected.
>
> Available in Allegro X PCB Editor when *High-Speed* option is selected.

Related Topics

- Transmission Line Calculator Dialog Box
- Using Transmission Line Calculator

Transmission Line Calculator Dialog Box

Access using:

- Menu path: *Analyze – Transmission Line Calculator*

Each transmission line calculator contains five areas: Physical Dimensions, Physical Dimension Units, Line Parameters, Electrical Characteristics, and a cross-sectional view of the interconnect structure.

All Tabs

The following values are common to all the transmission line calculators. The values you specify are retained for the session.

Physical Dimensions	Specifies the physical parameters of the transmission line. The graphic that appears in the dialog box reflects the parameters you set.
Length[L]	Specifies the length of the conductor.
Width[W]	Specifies the width of the conductor.
Height[H]	Specifies the thickness of the dielectric.
Thickness[T]	Specifies the thickness of the conductor.
Physical Dimension Units	Specifies the units for physical dimensions to specify the parameters of the transmission lines
Global physical dimension units	Specifies the units for physical dimensions. The options are: • Nanometer • Micron • Millimeter • Centimeter • Meter • Mills • Inch
Line Parameters	Specifies the dielectric constant and working frequency.
Dielectric	Specifies the dielectric material from the pull-down list.
Dielectric Constant	Displays the *Er* value automatically when you choose the dielectric material. To enter a different value, choose *Enter Custom Er Value* in the *Dielectric* field and enter a value of your choice.
Frequency	*Specifies the signal frequency at which to run the calculations.*

Electrical Characteristics	Displays the calculation results.
Impedance[Z0]	Displays the impedance based on values specified in the *Physical Dimensions* and *Line Parameters* sections. You can change the unit of measurement from the pull-down list.
Electrical Length	Displays the number of full wave lengths that occur within the distance defined by *Length [L]* in the *Physical Dimensions* section. You can change the unit of measurement from the pull-down list.
Propagation Velocity	Displays the speed of the signal expressed in fractions of the speed of light.
Effective Dielectric Constant	Displays the resulting dielectric constant of materials.

The following transmission line calculators have additional values not common to all calculators.

Embedded Microstrip and Asymmetric Stripline Tab

Height [H1] Embedded Microstrip	Specifies the dielectric material thickness that contains a conductor.
Height [H1] Asymmetric Stripline	Specifies the height from the reference plane to the conductor.

FGCPW Tab

Ground Width [WG]	Specifies the width of the return path conductor. You can choose to calculate with or without the reference plane.
Spacing[S]	Specifies the spacing between conductors on the same plane.

Dual Striplines Tab

Spacing[S]	Specifies the spacing between conductors on the Z axis.

Coupled Microstrip, Coupled Stripline, and Dual Stripline Tabs

Even Mode	Calculates the impedance based on even mode switching.
Odd Mode	Calculates the impedance based on odd mode switching.

Related Topics

- tline calculator
- Using Transmission Line Calculator

Using Transmission Line Calculator

1. Choose *Analyze – Transmission Line Calculator* (tline calculator command).
   The Transmission Line Calculators dialog box opens.
2. Choose the transmission line structure you want to use.
3. Enter the physical dimensions and line parameters in the appropriate fields.
4. Click *Calculate*.
   The calculation results appear in the *Electrical Characteristics* section.
5. Click *OK* to close the dialog box.

Related Topics

- tline calculator

toggle

Use the `toggle` command in conjunction with `add line` to reverse the locking sense of the next segment pair being added. For example, if the two segments from the last click to the current cursor positions are horizontal then vertical, the `toggle` command changes them to vertical then horizontal. The `toggle` command can be entered at the console window prompt or from the pop-up menu.

```
toggle
```

Related Topics

- `add line`

toolset

The `toolset` command is active in versions of Cadence tools 12.0 and earlier. It is similar to the `toolswap` command, allowing you to change the product type you are running. This command is not usable in versions of Cadence tools later than 12.0.

```
toolset <product_type>
```

Related Topics

- toolswap

toolswap

The Cadence Product Choices dialog box appears when you launch your Cadence product. It allows you to choose the versions of the product you have a license to run.

When used with the `toolswap` command, it lets you change the product type (tier) of the tool in which you are working, provided you are licensed for those tool sets. For example, you can switch from Allegro PCB Designer to Allegro PCB Venture.

The editor no longer warns you if you are switching your design from the tier where it was last saved to another tier. To display the warning, set the `db_tier_nomsg` environment variable using the *Setup – User Preferences – Drawing* command.

Related Topics

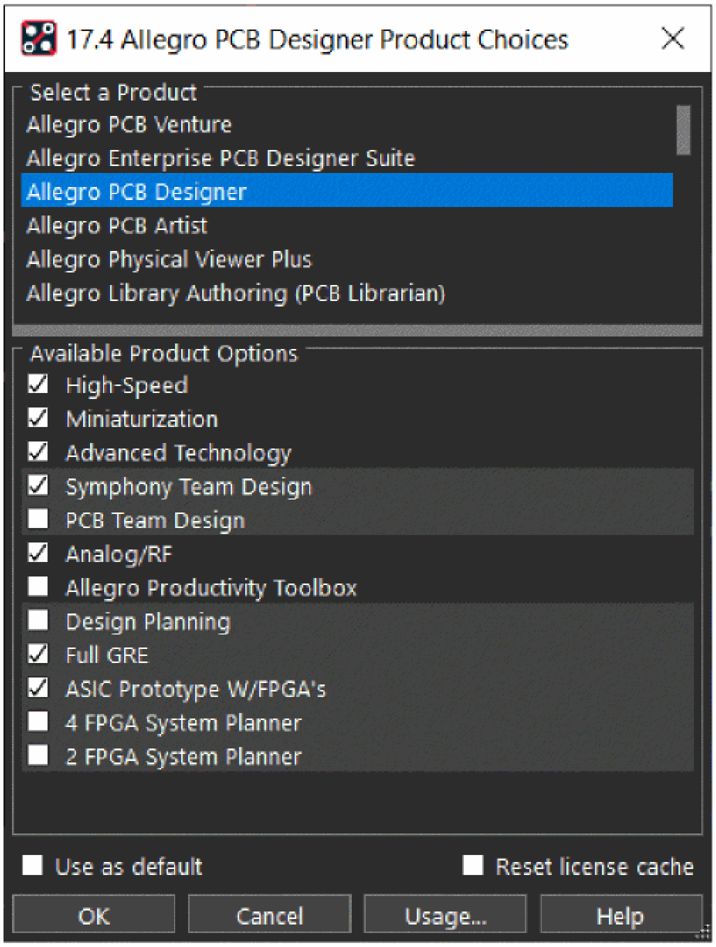
- Cadence Product Choices Dialog Box
- Choosing a Product Type Upon Launch
- Changing Product Options for an Active Design

Cadence Product Choices Dialog Box

Access using:

- Menu path: *File – Change Editor*

The Cadence Product Choices dialog box displays the products you have a license to run. If you are licensed to run Allegro PCB Design products and your licensing supports product options, the dialog box also gives you the option of running one or both of the product options associated with a specific tier. For example, Allegro PCB Designer lets you select either the *High-Speed* option or the *Analog/RF* option. Selecting these options opens versions of PCB Design that contain associated functionality.



17.4 Allegro PCB Designer Product Choices

Select a Product
- Allegro PCB Venture
- Allegro Enterprise PCB Designer Suite
- Allegro PCB Designer
- Allegro PCB Artist
- Allegro Physical Viewer Plus
- Allegro Library Authoring (PCB Librarian)

Available Product Options
- ☑ High-Speed
- ☑ Miniaturization
- ☑ Advanced Technology
- ☑ Symphony Team Design
- □ PCB Team Design
- ☑ Analog/RF
- □ Allegro Productivity Toolbox
- □ Design Planning
- ☑ Full GRE
- ☑ ASIC Prototype W/FPGA's
- □ 4 FPGA System Planner
- □ 2 FPGA System Planner

□ Use as default □ Reset license cache

Related Topics

- toolswap
- Choosing a Product Type Upon Launch
- Changing Product Options for an Active Design

Choosing a Product Type Upon Launch

1. Launch your Cadence product from a command prompt or Start menu of your operating system.
   The Cadence Product Choices dialog box appears.
2. Choose the product type you want to run.
   If your licensing supports product options, they appear in the Product Options section.
3. If you want the same product type to open automatically each time you launch the product, check *Use As Default*.
4. Click *OK*.
   Your Cadence tool opens to the product type you chose.

Related Topics

- toolswap
- Choosing a Product Type Upon Launch
- Changing Product Options for an Active Design

Changing Product Options for an Active Design

1. Start PCB Editor and open a design.
2. To change the current product options settings, run `toolswap`.
   The Cadence Product Choices dialog box appears.
3. Choose the product type you want to run.
4. Click *OK*.
   The user interface is reconfigured to the product type you selected.

Related Topics

- toolswap
- Cadence Product Choices Dialog Box
- Choosing a Product Type Upon Launch

topology pinuse

The `topology pinuse` command lets you change the pin type designation for chosen pins. You can use the dialog box to isolate nets by netname or component (reference designator), or you can choose nets and pins from the design interactively.

Related Topics

- *Logic - Pin Type Dialog Box*
- Changing Pin Type Designation

Logic - Pin Type Dialog Box

Access using:

- Menu path: *Logic – Pin Type*

Selection Area

Use the *Select by Net* or *Select by Component* radio button to search the netlist.

Select by Net	Net Filter - Uses asterisk as wildcard to narrow the search of available nets. Net Selection Window - Shows all netnames as allowed by filter.
Select by Component	Refdes Filter - Uses asterisk as wildcard to narrow the search of available reference designators. Device Filter - Uses asterisk as wildcard to narrow the search of available device names. Sort: Refdes, Device - Choose a selection method. Component Selection Window - Shows Ref Des and component names as allowed by either filter. When you highlight a line, the pin information for the chosen part is added to the *Pin Type Assignment* left side field.

Pin Type Assignment Area

Type Filter	Choose from the pull-down list to limit the search to pins of one specific type, e.g. In, Ground, Bi, and so on.
Refdes-Pin-Type list box (left)	Shows all pins allowed by the filter. Click on a line to move to the right side window, for re assignment.
Refdes-Pin-Type list box (right)	Pins moved to this side acquire the pin type chosen in the *New Pin Type* pull down list.
New Pin Type	All pins moved to the right side window assume this pin type.

Buttons

All	Moves all pin types displayed in a pin type assignment window from one to the other.
Clear	Clears contents of pin type display windows.

Related Topics

- *topology pinuse*
- Changing Pin Type Designation

Changing Pin Type Designation

1. Run `topology pinuse`.
   The Pin Type dialog box appears. You can set pin types from the Pin Type dialog box, from the design, or a combination of the two.
2. Leave the *Type Filter* for the lower left list box set to * to list all pin types. Otherwise, enter a specific pin type or click to choose a specific pin type.
3. In the *New Pin Type* field click the down-arrow to choose a pin type.
4. Choose individual pins or click *All* -> to move pins from the lower left list box to the lower right list box.
   Each pin moved to the right box assumes the type indicated in the *New Pin Type* field.
5. Click *Apply* (optionally, click *Clear*) and repeat the process to assign a different pin type to other pins.

   > ⚠ A pin type that is set overrides any pin types defined in component libraries.

Related Topics

- *topology pinuse*
- Logic - Pin Type Dialog Box

topology template

The `topology template` command displays the Topology Template dialog box that lets you extract a topology into Signal Explorer. Although this functionality is available in Cadence high-speed tools, this topic describes its behavior in Allegro PCB Designer. See Constraint Manager documentation for details on how to extract topologies in other tools.

Related Topics

- *Topology Template Dialog Box*
- Extracting a Topology into Signal Explorer

Topology Template Dialog Box

Access using:

- Menu path: *Tools – Topology Extract*

Use this dialog box to:

- Identify a chosen net as a template.
- Apply the template to a chosen net or a list of chosen nets.

Select Template

Template Name	Enter or browse for a template for a chosen XNet based on the net name. To save the template as a file, use the *Save As* button.
Include Routed Interconnect	Includes traces and vias in the extraction of a net-related object into SigXplorer. This is useful for creating a topology that accurately represents how a net is routed. You cannot apply a topology with trace and via models as an ECSet in Constraint Manager. In SigXplorer, you must choose *Edit – Transform for Constraint Manager*.
XNet Filter	Uses asterisks as wildcards to narrow the search of available XNets.
XNets list box	Shows all XNet names allowed by filter.

Buttons

Browse	Opens a file browse dialog box to import a previously saved topology (`.top` file.).
Save As	Opens a Windows Save As dialog box to save the chosen topology template to a `.top` file.
View	Starts SigXplorer with the chosen template.
OK	Maps the chosen nets to the template and closes the dialog box.

Related Topics

- *topology template*
- Extracting a Topology into Signal Explorer

Extracting a Topology into Signal Explorer

1. Run `topology template`.
2. Click a net in the *XNets list box* to extract the net of the selection. (Leave the *XNet Filter* set to * to list all available XNets or use wildcards to list the target XNet or a restricted range of XNets.)
   The name of the chosen net appears in the *Template Name* field.
3. To save the net as a template file (`*.top`), enter a new template name or browse for an existing template file for the chosen net. To save the template as a file, use the *Save As* button.
4. Check the routed interconnect option to have the interconnect represented by trace and via models during analysis.
   If you do not choose it, interconnect is represented as Tlines during analysis regardless of whether or not the net has been routed.
5. Click *View* to start SigXplorer with the chosen template.
6. Click *OK*.
   The topology is applied to the chosen nets.

Related Topics

- *topology template*
- Topology Template Dialog Box

trapsize

In Allegro-generated scripts, the tool automatically adds a `trapsize <value>` at the start of a script, after every change in zoom factor, and when opening new databases. If you generate scripts manually, place a `trapsize 0` at the start of a script to ensure that it replays independently of a design's zoom level.

The value, in design units, describes the trap box size where to find objects when an object pick is encountered in a script. A value of `0` requires a pick on an object.

```
trapsize 0
```

Related Topics

- `script`

trim segments

The `trim segments` command removes the unwanted line or arc segments that extend beyond the intersection points.

Access using:

- Menu path: *Manufacture – Drafting – Trim Segments*

Related Topics

- *Trimming Unwanted Line or Arc Segments*

Trimming Unwanted Line or Arc Segments

1. Choose *Manufacture – Drafting – Trim Segments* or run the `trim segments` command.
   OR
   Set *General Edit* application mode and select a line or an arc segment. Right-click and choose *Drafting – Trim Segments.*
2. Select a line or an arc segment.
   The selected segment is temporarily highlighted.
3. Select another line or an arc segment that is intersecting with the first segment.
   Both the segments are temporarily highlighted.
4. Select sides of either one or both of the highlighted segments.
   The selected sides are removed.
5. Right-click and choose *Next* to continue or *Done* to complete the operation.

Related Topics

- *trim segments*

tutorial

The `tutorial` command lets you run a product tutorial (if one exists) from the console window prompt.

```
tutorial <product_name>
```

tvision

The Timing Vision (`tvision`) is an environment that allows you to graphically see real-time delay and phase information directly on the routing canvas. This command uses special graphic techniques such as: custom cline coloring, stipple patterns and customized data tip information to define the delay problem in the simplest terms.

You can specify the settings for the timing vision techniques to see the graphics changes to find which nets are affected in the design. The timing vision command does not change the routing, or affect any of the custom color code settings that have been applied to nets, pins, vias, net-groups, etc.

This command provides immediate real time feedback during interactive routing and it also enhances a strategy for resolving timing on large buses or interfaces such as DDRx, PCI-Express, and Is.

⚠ In PCB Editor, this command is available with the High-Speed option only.

Related Topics

- Timing Vision: Options Panel

Timing Vision: Options Panel

Access using:

- Menu path: *Route – Timing Vision*
- Timing Vision

Display Control	Specifies the color/stipple pattern parameters for the timing vision environment.
Style	Specifies the display of clines in the timing vision environment.
Solid	The normal solid filled style.
Striped	A striping pattern to show the selected nets in the timing vision environment. ⚠ If this option is enabled, the stipple pattern option for *Pattern for critical signals* is grayed out.
Colors	Specifies the color code of the nets that are within delay specification or outside the desired range.
Satisfies required timing	Specifies the color code of the nets/rats/clines that meet the delay requirements
Shorter than required value	Specifies the color code of the nets/rats/clines that are more than 5% shorter than the required delay constraints.
A small amount shorter	Specifies the color code of the nets/rats/clines that are less than 5% shorter than the required delay constraint
Longer than required value	Specifies the color code of the nets/rats/clines that are more than 5% longer than the required delay constraints.
A small amount longer	Specifies the color code of the nets/rats/clines that are less than 5% longer than the required delay constraints.
Pattern for critical signals	Specifies a stipple pattern for definition of critical signals in the timing vision environment. You can choose only one pattern for all the critical signals. This pattern is applied to the controlling member of a match group such as a target, the longest member of a match group, and so on.

Timing Mode	Defines the calculation data and delay mode. These choices determine the color code displayed on the canvas.
DRC Timing	Lets you analyze the group of nets, by putting the system into the general delay mode. For example, byte-lanes of a memory system. This option uses Constraint Manager numbers to display the appropriate DRC status/color code for each affected cline on the canvas. The custom data tip shows all delay constraint information.
DRC Phase	Lets you analyze differential clocks, busses, and so on by putting the system into a mode specifically targeted at static phase issues. This option uses Constraint Manager numbers to display the appropriate DRC status/color code for each affected cline on the canvas. The custom data tips shows all phase constraint information.
Smart Timing	Lets you analyze the group of nets, by putting the system into the general delay mode. For example, byte-lanes of a memory system. The custom data tip shows all delay constraint information and uses a two character code to choose the affected nets into compliance.
Smart Phase	Lets you analyze differential clocks, busses, and so on by putting the system into a mode specifically targeted at static phase issues. The custom data tip shows all phase constraint information and uses a two character code to choose the affected nets into compliance.
Off	Turns off the current timing vision color code scheme, restores original color coding and removes the Smart Data entries from the data tips (in the Smart Timing mode). This option preserves the existing Timing Group relationships that were created previously. Reselecting either DRC Timing or Smart Timing re-enables the Timing Vision color code scheme using the existing Timing Group relationships.
Min length % for Smart Goals	Defines the delay distribution when multiple pins/rats are involved in any particular net.
Match Group Selection Mode	Selects and automatically add all members of a match group when some particular net is picked while in the Timing Group command.
Update Goals	Generates/updates goals for the entire design, allows you to work towards a specific solution. Once goals are generated, they are saved into the design until you exit the design. It is recommended to run this command in following scenarios: • The first time Smart Timing mode is used. If it is not run, no goals exist and there will be no color coloring or additional data tip information in Smart Timing mode. • Whenever signals are modified and go beyond their previous Max goal value due to other etch editing functions. If signals are going to stay outside the previous Goal range, then Goals must be updated to adjust other potentially related signal goals. • When additional nets are added to the Timing Group.
Remove Goals	Remove all goals from the design.

Disband	Disbands Timing Groups. The data tips will not show the additional data created by Timing Vision and the Timing Vision color coding is restored to original color scheme.

Running tvision Command

1. Run `tvision` command.
2. Set style as Solid.
3. Set DRC Timing as Timing Mode.
4. Click *Update Goals* to see the color code pattern.
   A Timing Group has been created.
5. Use `add_connect` or `slide` command to modify the data.
6. Right-click and choose *Done* to exit the command.