cadence®

N Commands

Product Version 23.1
September 2023

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

Contents

1

N Commands

na2 import	ncdrill customization	ncdrill legend
ncdrill param	nc drill report	ncroute
nctape	nctape_full	net
net_properties	net delay report	netin
netin param	net list in	net logic
netout	netrev	net schedule
net short	new	next
noappmode	nographic	

na2 import

The `na2 import` command imports an `.spd2` (Cadence® Sigrity™ Unified Package Designer or UPD) or `.na2` (Encore BGA) file from a third-party tool to Allegro® Advanced Package Designer. After the import, you can view a log file containing the data and any errors or warnings that may have occurred. The tool uses default values for elements available in `.mcm` files that do not have equivalents in the imported file. Data from an imported file that does not have equivalent mapping in APD is lost, and a warning is issued in the log file.

ⓘ This command is available only with Allegro Advanced Package Designer.

Related Topics

- SPD2/NA2 Format
- Importing Third-Party Encore BGA (NA2) or Unified Package Designer (SPD2) File

Importing Third-Party Encore BGA (NA2) or Unified Package Designer (SPD2) File

You can use the SPD2/NA2 third-party translator to either translate or verify the third-party files.

To import SPD2/NA2 files to Allegro X Advanced Package Designer, do the following steps:

1. Choose *File – Import – SPD2/NA2* or run `na2 import` from the Command window prompt.
   The SPD2/NA2 Import dialog box appears.
2. Enter the name of the source file or browse to its location in the *Source File field*.
   The file path appears and the New MCM File field automatically fills in.
   Check the option Perform syntax check of source file only [no database modification] and click *Import* to run the check. This will not modify the database or import any files but the results will appear in the command window prompt.
3. Choose the data you want to import and other options that are available.
4. Click *Import* to start the conversion process.
   A progress meter appears indicating the status of the import.
   A pop-up message appears warning that data translation may not be identical between the databases.
5. Click View Log to see the data conversions and any errors or warnings that may have occurred during import.
6. Click *Close* to close the dialog box.
   The tool saves the imported file.

Related Topics

- na2 import

SPD2/NA2 Import Dialog Box

Access Using

- Menu path: *File – Import – SPD2/NA2*

Design File Data	
Source File	Enter the name of the source file or browse to its location.
New MCM file	Populates automatically with the name of the destination file after the Source File field fills in. You can replace the default name if it is not the same as the *Source File* name.
Perform syntax check of source file only [no database modification]	Click the box to have the tool evaluate the imported file for any potential problems before importing.
Import Data	
Logical connectivity	Check to import the net information. The default is checked.
Padstack definitions	Check to import the padstack definitions. If not checked and you choose to import placed components, vias, or bond fingers, the tool assumes that the necessary padstack definitions already exist in the pad library. ⚠ Upon import of an SPD2/NA2 file, the tool automatically modifies the padstack definitions used for bond fingers so that they face east when that is not their defined orientation. This makes the wire bond tools work properly with the definitions.
Physical constraints	Check to import the physical and spacing rules. Refer to the log file for a list of constraint mapping from `.spd2`/`.na2` to `.mcm`. The default is checked.
Dies	Check to import all die components. The default is checked.
BGAs	Check to import all BGA components. The default is checked.

Discretes	Check to import all discrete components. The default is checked.
Plating bar	Check to import the plating bar component. The tool handles the plating bar as a regular line on the specified conductor layer of the design, so you need to create an APD plating bar component after importing. The default is unchecked.
Wire bonds	Check to import the wirebonds that cause the creation of default wire groups. The tool assigns the wirebonds to these groups based on their physical characteristics. The default is checked.
Package routing	Check to import all package routing components (clines, vias, plating traces, fillets). The default is checked.
Shapes/Planes	Check to import all shape objects (power and ground rings, planes, degassed shapes). Initially all shapes are static. The default is checked.
Etch back	Check to import all etch-back traces.
For new designs, cross-section information will be read. For incremental updates, cross-section data match between NA2/SPD2 and MCM databases.	
Options	
Post-process cleanup [Derive connectivity]	Check to have the tool check for errors in connectivity between clines and their endpoints and fix these errors. The default is checked.
Purge unused nets	Check to delete nets not referenced by objects in the database. The default is unchecked.
Batch DRC update	Check to have the tool perform a highly recommended DRC update on the entire design after importing completes. The default is unchecked.
Import	Click to import the NA2 or SPD2 data based on the specified settings.
Close	Click to close the dialog box and save the changes, if applicable.
View Log	Click to display the log file to see all information read from the source file including any errors, missing data, or warnings. If errors occurred during importing, the log file automatically appears.
Help	Displays help for this command.

ncdrill customization

The `ncdrill customization` command opens the *Drill Customization* spreadsheet that you use to manage drill symbol information at the design level, adding or customizing drill tolerances, symbols, or characters before generating drill legends or NC drill files.

Drill Customization Spreadsheet

The *Drill Customization* spreadsheet automatically generates drill symbol figures and characters, resets values to original design or library intent, and detects and corrects duplicate drill symbols.

You can assign positive and negative tolerance for drill holes to accommodate designs where separate applications use the same hole size, so the hole tolerance requirements vary as a result. For example, you might use a 0.125 hole as a connector mounting hole and also as a tooling hole for the board.

Initially, padstack-defined information populates the *Drill Customizatio*n spreadsheet. Overrides you make here appear in blue and overwrite information in the padstack when you click *OK* and exit the spreadsheet.

> ⓘ Customization changes are saved directly to design padstacks when you exit the Drill Customization spreadsheet. If you refresh design padstacks by using *Place – Update Symbols* with *Update Symbol Padstacks* enabled, or *Tools – Padstack – Refresh (*refresh symbol command), drill customizable data will not be updated or refreshed unless you enable the *Reset Customizable Drill Data* field on the *Update Symbols* dialog box.

The spreadsheet divides hole information into eight sections, each sorted by the *Size X* field in ascending order:

- Plated Circle Drills
- Non-Plated Circle Drills
- Plated Non-Standard Circle Drills
- Non-Plated Non-Standard Circle Drills
- Plated Oval Slots
- Non-Plated Oval Slots
- Plated Rectangle Slots
- Non-Plated Rectangle Slots

Related Topics

- Preparing Manufacturing Data
- Reviewing Drill Information Using Library Drill Report
- Setting All Drill Parameters to the Same Value
- Clearing Drill Symbol Characters

Drill Customization Dialog Box

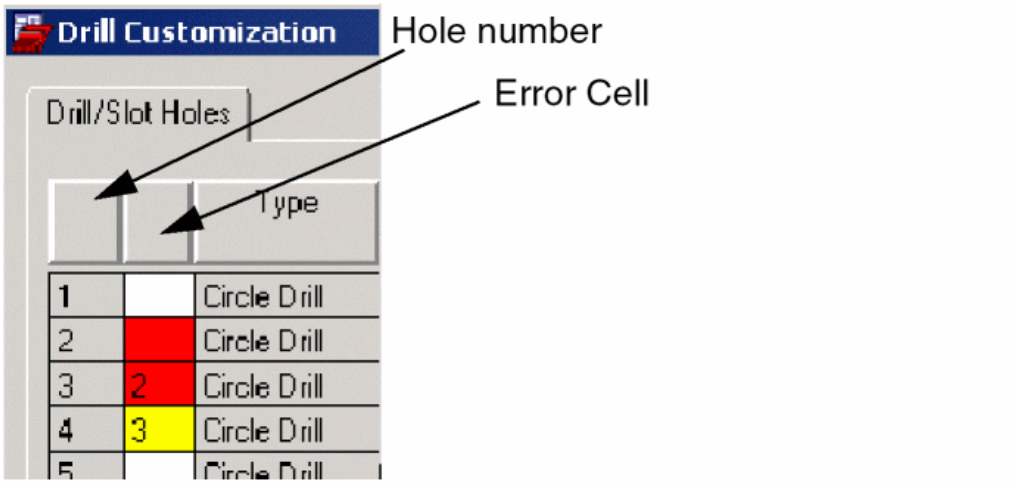
Access Using:

- Menu path: *Manufacture – NC – Drill Customization*
- Toolbar icon:

Drill/Slot Holes

#	Lists the drill holes. If you right-click and choose *List associated padstacks* from the pop-up menu, the padstack name associated with that drill hole appears in the console window prompt. For example: `Padstacks associated with hole # 1...VIA30 Padstacks associated with hole # 5...MTG120`
Type	Specifies a hole type of *Circle Drill* for circular holes; or *Oval Slot* and *Rectangle Slot* for non-circular slot holes. The field is read only and defaults from the definition created in Padstack Designer.
Size X	Specifies the hole drill size in the padstack's unit of measurement. The field is read only and defaults from the definition created in Padstack Designer.
Size Y	Specifies the hole drill size in the padstack's unit of measurement if you chose *Oval* or *Rectangle Slot* as a *Hole Type.* The field is read only and defaults from the definition created in Padstack Designer.
Tolerance +/-	Specifies positive and negative tolerance for each padstack hole size. (The Drill Legend displays these tolerance values in one column, as *+<value>/-<value>.* Each NC Drill output file header, where the drill tools and sizes display, contains any defined positive and negative tolerances.) Right-click and choose *Set all + Tolerance values to* or *Set all - Tolerance values to* for entering the same value in all cells in their respective columns.

Symbol Figure	Specifies the geometric shape that identifies each hole size (Null, rectangle, square, circle, octagon, cross, diamond, oblong, hexagon X, hexagon Y, or triangle) if you chose a *Hole Type* of *Circle Drill*. Otherwise, the field is read only and defaults from the definition created in Padstack Designer of *Oblong X, Oblong Y, or Rectangle* if *Oval Slot* or *Rectangle Slot* appears in the *Hole Type* field. You must specify the same figure for all holes of the same size and plating type. Right-click and choose *Set all Symbol Figures to* for specifying the same figure in each cell.
Symbol Characters	Specifies up to three optional, printable characters to define a drill size and its respective symbol code. The character height fits the given width and height of this symbol. Right-click and choose *Clear all Symbol Characters Strings* to empty all cell content.
Symbol Size X	Specifies the symbol size in the padstack's unit of measurement. You can only enter a value if you chose a *Hole Type* of *Circle Drill*. If you chose *Oval* or *Rectangle Slot* as a *Hole Type,* the field is read only and its value defaults from the *Size X* field.
Symbol Size Y	Specifies the symbol size in the padstack's unit of measurement. You can only enter a value if you chose a *Hole Type* of *Circle Drill*. If you chose *Oval* or *Rectangle Slot* as a *Hole Type,* the field is read only and its value defaults from the *Size Y* field.
Plating	Specifies the type of plating: *Plated, Non-Plated,* and *Optional*.
Non-standard	Read only display of the drill manufacturing method, which defaults from the padstack definition of *Laser, Plasma, Punch, Wet/Dry Etching, Conductive Ink Formation, Photo Imaging,* or *Other*. Blank if non-standard drilling does not apply. NC Drill output filenames appear as `<design name>_<type><n>.drl`, where *<type>* is `laser, plasma, punch, or other`. For example, `<design name>_laser1.drl`.
Quantity	Indicates the number of instances of the hole definition in the design.

Validate	Flags duplicate hole definitions or those with identical *Symbol Characters, Symbol Figure, Symbol Size X,* and *Symbol Size* Y fields in the error cell, which turns red, for the first detected hole. Error cells for subsequent holes with duplicate symbols turn red and display the number of the first hole with the same symbol. Yellow in the error cell flags holes whose entire hole definition is identical. Holes flagged in yellow can subsequently be merged into one using Merge. Holes flagged in red, however, must have their symbol definitions changed manually to make them unique.
	 
	When no errors occur, the following message appears in the console window prompt:
	`Validating`
	`No validation errors detected.`

Merge	Combines drills with common definitions, except quantity, into one entry. The *Quantity* field for the first duplicate hole updates with the total number of duplicate holes when multiple identical hole definitions merge into one definition.
Reset to design	Discards any changes made in the current session and resets the information to that currently in the design padstacks.

Reset to library	Discards any changes made in the current session, and resets information to that currently in the library padstacks. If a library padstack is not found, a warning message appears, and the information from the design padstack is used.
Auto generate symbols	Clears any existing symbol definitions for drill and slot holes, and automatically generates new ones, which you can modify on the spreadsheet. The first 11 drill holes use the cross, square, hexagon x, hexagon y, octagon, diamond, triangle, oblong x,oblong y, rectangle, and circle drill figures. Subsequent holes use the drill characters A-Z, AA, AB … AZ, BA, BB … BZ, etc. Oval slot holes use OA … OZ, etc; rectangle slots, RA … RZ. For circular drill holes, the *Symbol Size* in both *X* and *Y* is the actual hole size. For slot holes, *Symbol Size X* and *Y* remain the size of the slot hole.
Write Report File	Saves the output to a file using the Comma Separated Value (`.csv`) format, or the HTML format. By saving reports in a `.csv` format, which is a Microsoft Excel-compatible ASCII text data table, you can open them directly in spreadsheet programs such as Microsoft Excel or import them via its Text Import Wizard. Each line of the 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. You can view web-ready reports within the editor by saving reports in HTML If you choose the `.csv` format, the filename is: `drill_customization.rpt`. If you choose the `.html` format, the filename is: `drill_customization.html`.
Total Quantity	Displays the total number of each hole definition.
OK	Applies any customization changes directly to design padstacks, if you click *Yes* on the confirmer dialog box that appears, and closes the spreadsheet.
Cancel	Discards changes and closes the spreadsheet.
Library drill report	Click to display a read-only spreadsheet detailing the drill information for all available library padstacks.

Related Topics

- Setting All Drill Parameters to the Same Value
- Clearing Drill Symbol Characters

Reviewing Drill Information Using Library Drill Report

Use the Library Drill Report read-only spreadsheet to review drill information found in all available library padstacks. Field names duplicate those in the *Drill Customization* spreadsheet with the exception of the *Padstack* column.

Using this report, you can assess if your padstack hole definition is used elsewhere in the library, and ensure other drill information (such as drill symbols, for example) is synchronized.

Sorting the Spreadsheet

Sorting allows you to quickly determine and assess various aspects of the elements reported. For example, you can quickly determine what Laser drills exist in the padstack library by sorting on the *Non-standard Drill* column. Sort by the contents of the *Symbol Figure* column to assess if any holes already use a *Triangle* figure as part of the drill symbol definition. To discover whether the character M is currently used by any padstack as part of the drill symbol definition, sort using the *Symbol Characters* column.

To sort the spreadsheet, do the following steps:

1. Right-click any spreadsheet cell in the column to be used for sorting
2. Choose *Sort from* from the popup menu

Saving the Report in CSV or HTML Format

You can save the report either as a CSV (Comma Separated Value) file or an HTML file.

To save the report either as CSV or HTML, do the following steps:

1. Click *Write Report File*
2. Choose either of the following formats:
    a. Choose the `.csv` format to save the report as a CSV file.
    The report will be saved with the filename `library_drill.rpt`.
    By saving reports in a `.csv` format, which is a Microsoft Excel-compatible ASCII text data table, you can open them directly in spreadsheet programs such as Microsoft Excel or import them via its Text Import Wizard. Each line of the 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.
    b. Choose the `.html` format to save the report as an CSV file.
    The report will be saved with the filename `library_drill.html`.

You can view web-ready reports within the editor by saving reports in HTML.

3. Click *Close* to exit the report.

Related Topics

- ncdrill customization
- Clearing Drill Symbol Characters

Setting All Drill Parameters to the Same Value

You can synchronize all cell values in Drill Customization by setting information in all column cells to the same value.

To set information in all column cells to the same value, do the following:

1. Right-click a cell in the column with the desired value.
   If the desired value does not already exist in a column, enter it.
2. Choose the *Set all <cell name>* popup.

Related Topics

- ncdrill customization
- Drill Customization Dialog Box

Clearing Drill Symbol Characters

Clear the value in the Symbol Characters column to clear drill symbol characters.

To clear all values in the Symbol Characters column, do the following:

1. Right-click any cell in the *Symbol Characters* column.
2. Choose *Clear all Symbol Characters Strings* popup item.

Related Topics

- ncdrill customization
- Drill Customization Dialog Box
- Reviewing Drill Information Using Library Drill Report

ncdrill legend

The `ncdrill legend` command creates different types of drill legend tables, which sort hole sizes and map drill figures or text symbols to each drill bit size. A text table added to the layout includes the figure, hole size, hole-plating, and quantity for each drill size on the drawing. The required number of subclasses for blind or buried designs automatically generate in one execution to account for multiple drilling operations and design changes.

Related Topics

- Drill Legend Dialog Box
- Generating a Drill Legend Table
- Preparing Manufacturing Data

Types of Drill Legends

Drill legends are presented as a text table added to the layout. The table includes the figure, hole size, hole-plating, and quantity for each drill size on the drawing.

In addition to a *Layer Pair* or a *By Layer* type of drill legend, you can enable the *Include Backdrill* and *Include C-Bore* options to create backdrill and counter bore legends. (*Layer Pair* and *By Layer* legends are mutually exclusive. Generating one type removes the other if it exists in the design.)

For each required *Layer Pair* drill legend, an NCLEGEND-*<L1>*-*<L2>* subclass automatically generates whether subclasses are visible or not, where *<L1>* and *<L2>* are the layer numbers of the drilled layers. Each subclass includes all holes for that layer pair. Slot hole figures display at the true hole geometry and size, including user-specified characters. Tolerance values display in one column, as *+<value>/-<value>.*

For *By Layer* drill legends, an NCLEGEND-BL-*<L1>*-*<L2>* subclass generates on the MANUFACTURING class, where *-BL* indicates *By Layer* drilling and groups legend graphics as DRILL_LEGEND_BL_*<L1>*_*<L2>*.

For backdrilling legends, an NCBACKDRILL-*<L1>*-*<L2>* subclass generates on the MANUFACTURING class and groups legend graphics as DRILL_LEGEND_BD_*<L1>*_*<L2>*, in which *<L1>* indicates the from side layer number; *<L2>*, the to layer.

The counter bore/counter sink are based on which side of the board the pins are placed. For counter bore/counter sink legends, an NCCOUNTERDRILL-*<L1>* subclass generates on the MANUFACTURING class and groups legend graphics as DRILL_LEGEND_CT_*<L1>*, in which *<L1>* indicates layer number.

Legend Type	Subclass	Group Naming
Layer Pair	NCLEGEND-*<L1>*-*<L2>*	DRILL_LEGEND_*<L1>*_*<L2>*
By Layer	NCLEGEND-BL-*<L1>*-*<L2>*	DRILL_LEGEND_BL_*<L1>*_*<L2>*
Backdrill	NCBACKDRILL-*<L1>*-*<L2>*	DRILL_LEGEND_BD_*<L1>*_*<L2>*
C-Bore	NCCOUNTERDRILL-*<L1>*-	DRILL_LEGEND_CT_*<L1>*

⚠ The NCLEGEND subclass combines the former NCDRILL_LEGEND and NCDRILL_FIGURE subclasses for multiple layer drills and is automatically visible when generated. For single layer drills, drill figures are still created on the MANUFACTURING / NCDRILL_FIGURE class and is used in IPF output.

Drill Legend data is not updated dynamically. Changes to the database that involve the addition or subtraction of drills require regeneration of the legends.

When you create a drill size that references more than one set of tolerances at the padstack level, the drill legend can separately output the drill data for a padstack with the same drill size and plating but different tolerances. For example, a 0.035 mil drill size may require a tolerance of +/- 0.001 and +/-0 .002.

Related Topics

- Generating a Drill Legend Table
- Preparing Manufacturing Data

Drill Legend Dialog Box

To generate drill, backdrill, and counter bore legends on your board, use this dialog box to specify legend values.

Access Using:

- Menu path: *Manufacture – NC – Drill Legend*
- Toolbar icon:

Template File	Indicates the template to use to create the drill legend. Click the browse button to locate existing templates.
Output Unit	Outputs the drill legend data in units that differ from those in the design.
Library	Lets you view template files that are available via the NCDPATH variable that you set in *User Preferences – Paths – Config*.
Legend Title	Indicates the legend title specified in the `.dlt` drill legend template file.
Drill	Indicates the drill legend title specified in the `.dlt` drill legend template file, which you can modify here. Drilled layers for each drill legend subclass are visually identifiable as a result. If the string `$lay_nams$` or `$lay_nums$` appears within the title string, the layer names or numbers are respectively substituted in the title for each generated legend table. For example, TOP to BOTTOM or 1 to 4, respectively.
Backdrill	Indicates the backdrill legend title specified in the `.dlt` drill legend template file, which you can modify here, if you choose to generate a backdrill legend (optional).
C-Bore	Indicates the counter bore/counter sink legend title specified in the `.dlt` drill legend template file, which you can modify here, if you choose to generate a drill legend (optional).
Hole Sorting Method	

By Hole Size	Defines how to sort hole sizes in the legend. *Ascending*: Lists hole sizes from largest to smallest in the legend. *Descending*: Lists hole sizes from smallest to largest in the legend.
By Plating Status	Indicates whether to list plated or non-plated holes first in the legend. *Plated First*: Lists plated holes first in the legend. *Non-plated First*: Lists non-plated holes first in the legend.
Legends	Each execution of this command generates one of the following types of legend, depending on which you have chosen. *Layer Pair* and *By Layer* legends are mutually exclusive. Generating one type removes the other if it exists in the design.
Layer Pair	Choose to generate drill legends that represent holes to be drilled according to combinations of layer pairs. For example, for a four-layer board using thru via technology, this option represents the via that spans layers one through four as existing on the layer pair "1-4."
By Layer	Choose to generate multiple drill legends that represent each hole to be drilled for each via as existing between one entry layer and one exit layer, typically used to meet microvia technology requirements. For example, for a four-layer board, this option represents the via that spans layers one through four as existing on layer "1-2," layer "2-3," and layer "3-4" and displays it in three different drill legends. Note that no output appears for layer "1-4." An <*n*> layer board therefore always has one fewer drill legend outputs than the total number of layers, or <*n-1*>, because a hole starting on one layer has to at least appear on the next layer as well, and a hole never appears on only one layer.
Include Backdrill	Choose to generate backdrill size, must-not-cut-layer, maximum drill depth, and manufacturing stub length in addition to *Layer Pair* or *By Layer* drill legends, depending on your choice.
Include C-Bore	Choose to generate drill legends for counter bore/counter sink structure, depending on your choice.
Other Options	
Drill Legend Columns	Choose to display drill legend columns, depending on your choice. *Tolerance drill:* Includes tolerance for both circular drills and slots *Tolerance travel:* Includes tolerance for a slot along the path *Tool size:* Includes drill tool size or drill bit name *Rotation:* Includes rotation for the square drills and slots *Non-standard type*: Includes non-standard drill types such as laser, punch, and so on defined in the Padstack Editor

Display total slot/drill count	Choose to display total number of slots or drills.
Separate slots from drills	Choose to generate legends for drills separately.
Suppress tolerance column if all values are 0's	Choose to not display tolerance column if all values are 0.
Suppress tool size column if all values are empty	Choose to not display tool size column if all values are nil.
Suppress rotation column if all values are 0's	Choose to not display rotation column if all values are 0.
OK	Click to create the drill legend. The drill character as defined in the padstack for that hole is drawn over each hole in the design.
Cancel	Click to close the dialog box without generating a drill legend.

Related Topics

- ncdrill legend

Generating a Drill Legend Table

You generate drill, backdrill, and counter bore legends on your board.

To create drill legends, do the following:

1. Run the `ncdrill legend` command.
2. Complete the Drill Legend Dialog Box.
3. Click *OK*.
   The drill symbol characters or figures, as defined in the padstack for each hole, are drawn at the hole location in the design. A dynamic rectangle attaches to the cursor, representing the largest drill legend table.
4. Choose the location to place the drill legend with your mouse. A single placement prompt puts all drill legend tables on all drill legend subclasses. You can change those placements that do not meet your requirements and move the legend as a group if the Find Filter is set to Group, as the NC Drill Legend constitutes a group object in the database. When you run subsequent Drill Legend outputs, the current location of an existing drill legend is re-used even if had been moved previously.
   The MANUFACTURING class and NCLEGEND subclass now control the drill figure visibility.

Related Topics

- ncdrill legend
- Types of Drill Legends
- Preparing Manufacturing Data

ncdrill param

The `ncdrill param` command displays the *NC Parameters* dialog box where you define the operating characteristics for numerically controlled routing and NC Route output files in a parameter text file, which specifies the drill coordinate data format.

Related Topics

- Defining Operating Characteristics for NC Routing

NC Parameters Dialog Box

Use this dialog box to define characteristics for NC route output files.

Access Using

- Menu Path: *Manufacture – NC – NC Parameters*
- Toolbar icon:

Parameter File	Enter the name and path of the file to which to save the NC parameters when you exit this dialog box. The default path is your current working directory. Click **...** to open a file browser from which you can choose an existing file.
Output File	

Header	Specifies one or more ASCII headers in the output file with maximum of 1024 characters. For example, if you enter *Record 1 Record 2 Record 3* here, the following appears in the output file:
	
	`HEADER: Record 1 Record 2 Record 3`
	
	If you enter sequential new lines here as follows:
	
	`HEADER: Record 1`
	`HEADER: Record 2`
	`HEADER: Record 3`
	
	The following appears in the output file:
	
	`HEADER: Record 1`
	`HEADER: Record 2`
	`HEADER: Record 3`

Leader	Specifies the leader length.The default is *12*.

Code	Specifies the output format. The default format is *ASCII.*
Excellon format	
Format	Identifies the coordinate data in the output NC Drill file. The integer that you enter to the left of the decimal point sets the number of digits displayed before the decimal point. The integer that you enter to the right of the decimal point sets the number of digits displayed after the decimal point. The default value is *2.3*.
Offset X,Y	Identifies the offset from the drawing origin for the coordinate data in the output files specified in the *File Name* fields in this dialog box. The default *0,0*.
Coordinates	Specifies whether the output coordinates as incremental or absolute. The default setting is *Absolute*.
Output Units	Specifies output units as English or Metric. The default is *English*.
Trailing Zero Suppression	Choose to eliminate trailing zeros in the output coordinate data. By default, this option is unchecked.
Leading Zero Suppression	Choose to eliminate leading zeros in the output coordinate data. By default, this option is unchecked.
Equal Coord Suppression	Specifies whether equal coordinate data is suppressed. By default, this option is unchecked.
Enhanced Excellon format	Choose to generate a header in NC Drill and NC Route output files that more fully uses Excellon commands. The header starts with M48 and ends with % and lists tool specifications, the appropriate INCH/METRIC command appears, and LZ/TZ as required for padding the leading or trailing zeros in the data section. The `Tnn` tool-diameter specification codes expand to a `TnnC.xxx` format to specify the required router bit size.
Close	Click to exit the dialog box and save changes to local parameter file named `nc_param.txt`, which is either created or updated with your additional modifications, even if you specified a user-defined, non-local parameter file with which to initially populate the *NC Parameters* dialog box parameters. The non-local parameter file remains unchanged, as the local `nc_param.txt` parameter file drives the numerically controlled routing and NC Route processes.
Cancel	Click to close the dialog box without saving changes.

Defining Operating Characteristics for NC Routing

You can define or update NC (numerically controlled) parameters either by changing and updating the NC Parameters dialog box or by reading in a parameter file.

You can read in a parameter file using any filename in any directory, allowing a customer site, for example, to create a number of pre-defined standard parameter files available for general use. This user-defined, non-local parameter file can be initially read in to populate the *NC Parameters* dialog box parameters. You can interactively modify these parameters before closing the *NC Parameters* dialog box.

To define the parameters, do the following:

1. Choose *Manufacture – NC – NC Parameters*.
   The NC Parameters dialog box appears.
2. Specify the parameters.
   When you close the *NC Parameters* dialog box, a local parameter file named `nc_param.txt` is then either created, or updated with your modifications, leaving the non-local file unchanged.
   The local `nc_param.txt` parameter file drives the numerically controlled routing and NC Route processes.
   After setting the parameters, generate output for NC routers.

Related Topics

- ncdrill param
- Preparing Manufacturing Data

nc drill report

An obsolete command.

ncroute

The `ncroute` command generates output for an NC router based on the parameters you set in the NC Parameters Dialog Box using the ncdrill param command. The output is an ASCII file in Excellon Format with an `.rou` extension. This command enables you to:

- Use any number of cuts to define a design outline
- Specify the direction of each cut
- Offset the cut from the design outline
- Specify different line widths
- Generate the output for routing oval and rectangle slot holes
- Generate separate output files for plated/non-plated routing

Syntax

`ncroute [-q][-v][-o] [-version] <design_name>`

-q	Enables quiet mode (no status messages display as the command executes —default).
-v	Enables verbose mode (the editor displays database status messages in the window as the command executes).
-o	Lets you define an output name. Otherwise the default output filename is *<design_name>*`.rou`. If separate route files are specified for plated/non-plated routing then two rou file are generated with *_plated"* appended to the route file name for the plated output.
-n *<alt_name>*	Lets you define an alternative design name in output files. The default is the *<design>* name.
-version	Prints the version.
<design_name>	Specifies the name of the design for which you are running the output.

The NC Route output supports multiple router bit tool sizes, based on different width lines found on the NCROUTE_PATH and NCROUTE_PLATED subclasses. To specify varying widths for your cutting paths, you must generate a text file called `ncroutebits.txt` that specifies route bit sizes in design units. This file cross-references router tool diameters and Excellon tool codes. Each line of

the file contains one diameter followed by a space and a tool code. If this file does not exist NC Route auto generates `ncroutebits_auto.txt`.

> ⚠ The existing `ncroutebits.txt` is not usable because it specifies the route bit sizes in NC Route output units. You need to create new `ncroutebits.txt` that specifies route bit sizes in design units.

When NC Route routes oval and rectangle slot holes, an appropriate tool is chosen from `ncroutebits.txt` using the following guidelines:

- Rectangle Slot: a tool size smaller than the minimum dimension of the rectangle must exist to route a rectangle path with appropriate Excellon tool compensation.
- Oval Slot: a tool size that exactly matches the minor dimension of the oval must exist to route the oval as a single line path. Otherwise, a tool size smaller than the minor dimension must exist to route an oval path with appropriate Excellon tool compensation.

If you choose *Enhanced Excellon format* on the *NC Parameters* dialog box, NC Route generates a header in its output file that more fully uses the Excellon commands. The header starts with M48 and ends with % and lists tool specifications, the appropriate INCH/METRIC command appears, and LZ/TZ as required for padding the leading or trailing zeros in the data section. The `Tnn` tool-diameter specification codes expand to a `TnnC.xxx` format to specify the required router bit size.

You can run this command from the Command window or as a batch command from an operating system prompt.

Related Topics

- Creating a Design Profile Routing File
- Preparing Manufacturing Data

NC Route Dialog Box

Use this dialog box to generate the design profile routing file.

Access Using

- Menu path: *Manufacture – NC – NC Route*

File Name	Specifies the name of the design for which you are running the output.
Route Feedrate	Specifies in inches per second the routing tool's speed.
Separate files for plated/non-plated routing	Choose to generate two output files for plated/non-plated routing with "_plated" appended to the route file name for the plated output.
Route	Click to generate the output files.
NC Parameters	Click to access the NC Parameters Dialog Box.
Close	Click to save changes and exit the dialog box.
Cancel	Click to close the dialog box without saving changes.
View Log	Click to view the `ncroute.log` file.

Creating a Design Profile Routing File

Before starting, ensure that the *nc_param.txt* file is defined by *NCDPATH* for controlling additional ncroute settings. If the file is not found, Cadence defaults will be used.

Before creating the output file, you must define route parameters and create cutting paths.

To define route parameters do the following:

1. Open the NC Parameters dialog box.
2. Complete the following fields in the NC Parameters dialog box:
    - *Format*
    - *Offset X, Y*
    - *Coordinates*
    - *Output Units*
    - *Trailing Zero Suppression*
    - *Leading Zero Suppression*
3. Click *Close*.

To set up subclasses, do the following:

1. Choose *Setup – Subclasses*.
2. In the *Define Subclass* dialog box, choose *BOARD GEOMETRY*.
3. In the *Define Non-Etch/Conductor Subclass* dialog box, *NCROUTE_PLATED* in the *New Subclass* field.
4. In the *Define Subclass* dialog box, click *OK*. Both dialog boxes close.
5. For creating non-plated board routing file, in the Options tab, change the class to *BOARD GEOMETRY* and the subclass to *NCROUTE_PATH*.
6. For creating plated board routing file, in the Options tab, change the class to *BOARD GEOMETRY* and the subclass to *NCROUTE_PLATED.*

Add a cutting path using any of the options from the *Add* menu, which can be *Line*, *3pt Arc*, or *Arc w/ Radius*. If you have cutting paths of different widths, create the `ncroutebits.txt` file.

To generate the design profile routing file, do the following:

1. Choose *Manufacture – NC – NC Route*.

You can also run this as a batch command using the ncroute.

1. Check the `ncroute.log` and the `extract.log` files.
2. Transfer the ASCII output file (it has an `.rou` extension) to your machine.

Related Topics

- ncroute
- NC Parameters Dialog Box
- add line
- add arc
- add rarc
- Preparing Manufacturing Data

nctape

The `nctape` command locates the `nc_param.txt` file via NCDPATH to control additional output-file settings and creates one or more customized NC drill files (`.drl`) in batch mode.

⚠ You can change the default file extension of `.drl` for NC drill output filenames by setting the `ext_drill` environment variable in the User Preferences Editor, available by choosing *Setup – User Preferences* (enved command).

Syntax

`nctape [-q] [-v] [-o] [-s <scale_value>] [-version] [-n <outfile>] [-l] [-b] <board_name>`

-q	Enables quiet mode, in which no messages display as the command executes. Enabled by default.
-v	Enables verbose mode, in which database status messages display as the command executes.
-o	Reduces the total drill head travel path to increase efficiency.
-s *<scale>*	Scales the X,Y drill locations by a user-defined value. If you include the `-s` switch and a numeric value when you execute `nctape`, all drill locations in the output drill files are multiplied by this value. The new coordinates round off to the same accuracy as the original units of the drawing. For example, using a scale factor of `1.25`, a drill located at 1034, 1051scales to 1293, 1314. These values convert to the drill file units defined in the NC Drill Parameters dialog box. If your drawing requires greater accuracy, increase the number of decimal places for the drawing before executing `nctape`.
-version	Prints the version.
<boardname>	Specifies the name of the design for which you are running the output.
-n *<outfile>*	Overrides the default design name when naming the output file. The *TAPE-FILE* parameter in `nc_param.txt` (used in generating a base `.drl` name and output directory), however, will in turn override even *-n.*

-l	Generates multiple drill output files that represent each hole to drill for each via as existing between one entry layer and one exit layer, typically used to meet microvia technology requirements. For example, for a four-layer board, this option represents the via that spans layers one through four as existing on layer "1-2," layer "2-3," and layer "3-4" and displays it in three different drill output files. Note that no output appears for layer "1-4." An <*n*> layer board therefore always has one fewer drill output files than the total number of layers, or <*n-1*>, because a hole starting on one layer has to at least appear on the next layer as well, and a hole never appears on only one layer.
-b	Generates drill output files for backdrilling, a manufacturing process driven by high-speed requirements on net-based objects. In backdrilling, the conductive path of the unused section of the plated hole is drilled out to a controlled depth. The secondary drill diameter must be larger than the primary drill to ensure removal of all deposited metal. For more information, see the *Backdrilling* chapter in the *Preparing Manufacturing Data* user guide in your documentation set.
-c	Generate drill output for counterdrilling

When you execute the command, by default it generates NC drill files that use *Layer Pair* type drilling, even if the `nc_param.txt` file being read specifies *By Layer* type drilling, or backdrilling. To specify *By Layer* type drilling, or include backdrilling, you must enter the command line options of `-l` and `-b`, respectively, as detailed in the Syntax section.

The generated files use the following name convention:

```
<name>-<l1>-<l2>-<type>-<plate><-len>.drl
```

where

<name>	Specifies the base name of the *FILE* parameter specified in `nc_param.txt`.
<l1>-<l2>	Specifies the numbers of the drill start layer (*<l1>*) and the drill end layer (*<l2>*).
<type>	Specifies a non-standard drill–either laser, plasma, punch, or other–as defined in the Padstack Designer, which outputs as *<name>*`-laser`, *<name>*`-plasma`, *<name>*`-punch`, and *<name>*`-other`.
<plate>	When you generate separate files for plated and non-plated holes, filenames for the latter include `np`.

<len>	Specifies length if single file is break into multiple files due to exceeding file LENGTH specified in `nc_param.txt`

Example

A 6-layer board with a *TAPE-FILE* parameter of `/home/xyz/drill.drl` in `nc_param.txt` outputs the following drill files to the `/home/xyz` directory:

```
drill-1-6-laser-np.drl
drill-1-6-plasma.drl
drill-bd-top-<l2>.drl  (backdrilling)
drill-bd-bottom-<l2>.drl
drill-bl-<l1>-<l2> (By Layer drilling)
```

Related Topics

- nctape_full
- Preparing Manufacturing Data

Generating Drill File Output in Batch Mode

Before generating drill file output, update the NC Parameters dialog box or create `nc_param.txt` file in the current working directory. If needed, create `nc_tools.txt` and `nc_exclude.txt` files.

To generate the drill file output, do the following:

1. From the operating system prompt, run the `nctape` command.

The output files are created using the name you specified in the *Parameter File* field of the NC Parameters dialog box.

1. Verify the results using the `explot` batch command.
   Using an output file from the `nctape` command (the `.drl` extension is optional), `explot` generates two files, `outputfile.plt` and `outputfile.ctl`. You can use the `outputfile.plt` file to drive a penplotter.

Related Topics

- NC Parameters Dialog Box
- Defining Operating Characteristics for NC Routing
- Generating Drill File Output
- explot
- Preparing Manufacturing Data

nctape_full

The `nctape_full` command generates customized NC drill output files that reflect the parameters you set in the NC Parameters Dialog Box using the ncdrill param command.

If you choose *Enhanced Excellon format* on the *NC Parameters* dialog box, NC Drill generates a header in its output file that more fully uses the Excellon commands. The header starts with M48 and ends with % and lists tool specifications, the appropriate INCH/METRIC command appears, and LZ/TZ as required for padding the leading or trailing zeros in the data section. The `Tnn` tool-diameter specification codes expand to a `TnnC.xxx` format to specify the required router bit size.

In output files generated for same-size holes with identical plating but different tolerances, a different tool code is used.

NC Drill output only applies to (circular) drill holes; use the ncroute command for slot holes. You can also run this command in batch mode using the nctape command.

Related Topics

- Generating Drill File Output
- Generating an NC Drill File in Enhanced Excellon Format

NC Drill Dialog Box

Use this dialog box to create the output drill files.

Access Using

- Menu path: *Manufacture – NC – NC Drill*

Root File Name	Enter a base filename for output text files before any appended extension. If you do not specify a filename extension, `.drl` is the default, which you can change by setting the `ext_drill` environment variable in the User Preferences Editor, available by choosing *Setup – User Preferences* (enved command).The filename defaults to *<design name>*`-<`*l1*`>-<`*l2*`>.drl`, where <*l1*> and <*l2*> are the two drilled layers. When you choose to generate separate files for plated and non-plated holes, non-plated filenames are *<design name>*`-np-<`*l1*`>-<`*l2*`>.drl`. Holes defined on the Padstack Designer with *Non-standard Drill* types of *Laser, Plasma, Punch, and Other* output to separate files named *<design name>*`-laser`, *<design name>*`-plasma`, *<design name>*`-punch`, *<design name>*`-other`. For example, for non-plated holes that use non-standard drilling, the filename is *<design name>*`-laser-np-1-2.drl`. The naming conventions for backdrill files are `drill-bd-top-<`*l2*`>.drl` and `drill-bd-bottom-<`*l2*`>.drl`; for By Layer files, `drill-bl-<`*l1*`>-<`*l2*`>`.
Scale Factor	Indicates the value by which all drill locations in the output drill files are multiplied, to scale the X and Y drill locations.
Tool Sequence	Specifies whether the tool sequence starts with the smallest drill size and increases, or with the largest drill size and decreases. The default is *Increasing.*

Auto Tool Select	Inserts `Tnn` tool-select codes into the data portion of the Excellon-format output file, instead of M00 stop codes for manual tool changing. `Tnn` codes automatically generate in sequence (for example, `T01, T02, … Tnn`). `Tnn` tool-diameter specification codes only append to the tool-select code in the header portion of the Excellon-format output file, expanding to a `TnnC.xxx` format to specify the required router bit size, if you enabled *Enhanced Excellon Format* in the *NC Parameters* dialog box. For example, T01C.045 specifies that Tool 1 has a 45-mil diameter. You can also opt to associate specific tool sizes with specific `Tnn` tool codes in an `nc_tools.txt` file, which is used if it exists; otherwise, NC Drill automatically determines the appropriate tools the design needs and assigns tool codes to them. An `nc_tools_auto.txt` file is created for reference. A warning message appears when an `nc_tools.txt` file is not found.
Separate Files for Plated/Non-plated holes	Choose to generate separate files for plated and non-plated holes.
Repeat Codes	Specifies whether your drill supports repeat codes. Enabled by default.
Optimize Drill Head Travel	Choose to optimize drill travel on the NC Drill output files.
Drilling	Each execution of this command generates one of the following drill output files, depending on which you have chosen. *Layer Pair* and *By Layer* drill-output files are mutually exclusive. Generating one type removes the other if it exists in the design.
Layer Pair	Choose to generate drill output files that represent the holes to drill according to combinations of layer pairs. For example, for a four-layer board using thru via technology, this option represents the via that spans layers one through four as existing on the layer pair "1-4."

By Layer	Choose to generate multiple drill output files that represent each hole to drill for each via as existing between one entry layer and one exit layer, typically used to meet microvia technology requirements. For example, for a four-layer board, this option represents the via that spans layers one through four as existing on layer "1-2," layer "2-3," and layer "3-4" and displays it in three different drill output files. Note that no output appears for layer "1-4." An <*n*> layer board therefore always has one fewer drill output files than the total number of layers, or <*n-1*>, because a hole starting on one layer has to at least appear on the next layer as well, and a hole never appears on only one layer.
Include backdrill	Choose to include backdrill information
Include counterdrill	Choose to include counterdrill information
Drill	Click to generate output files
NC Parameters	Click to access the NC Parameters Dialog Box dialog box.
Close	Click to save changes and exit the dialog box.
Cancel	Click to close the dialog box without generating an output file.
View Log	Click to view details concerning the `nctape.log` file, including tool number information.

Related Topics

- Generating an NC Drill File in Enhanced Excellon Format
- NC Parameters Dialog Box

Generating Drill File Output

Before generating drill file out, update the NC Parameters dialog box and, if needed, create `nc_tools.txt` and `nc_exclude.txt` files. The output files are created using the name you specified in the *Parameter File* field of the NC Parameters dialog box.

To create the drill file output, do the following:

⚠ You can also create the drill files in the batch mode using the `nctape` command.

1. Choose *Manufacture – NC – NC Drill*
2. In the *NC Drill* dialog box, enter the *Scale Factor*.
3. Click *Drill*.
   The tool creates the output files using the name you specified in the *Parameter File* field of the NC Parameters dialog box.
   The following message appears in the console window prompt:

   nctape completed successfully - use Viewlog to review the log file.
4. Click *Close*.
5. Verify the results using the `explot` command.

Using an output file from the `nctape` command (the `.drl` extension is optional), `explot` generates two files, `outputfile.plt` and `outputfile.ctl`. You can use the `outputfile.plt` file to drive a penplotter.

Related Topics

- nctape_full
- NC Parameters Dialog Box
- Generating Drill File Output in Batch Mode
- explot

Generating an NC Drill File in Enhanced Excellon Format

You can also generate NC Drill output in the enhanced Excellon format by doing the following steps:

1. Choose *Manufacture – NC – NC Drill*.
2. Click *NC Parameters*. The NC Parameters dialog box appears.
3. Enable *Enhanced Excellon format* to generate a header in the NC Drill and NC Route output files that uses Excellon commands to a greater extent. The header starts with M48, lists the appropriate units (INCH or METRIC), and the T*nn* tool-diameter specification codes expand to T*nn*C.xxx format to specify the required router bit size and end with %.
4. Click *Close* to save the settings.
5. Enable *Auto Tool Select* in the NC Drill dialog box to insert T*nn* tool-select codes into the data portion of the Excellon-format output file, instead of M00 stop codes for manual tool changing. Tnn codes automatically generate in sequence (for example, T01, T02, ... T*nn*).
6. Click *Drill.*
   You can also associate specific tool sizes with specific T*nn* tool codes in an `nc_tools.txt` file, which is used if it exists; otherwise, NC Drill automatically determines the appropriate tools the design needs and assigns tool codes to them. An `nc_tools_auto.txt` file is created for reference. A warning message appears when an `nc_tools.txt` file is not found in the `ncdrill.log`.

Related Topics

- nctape_full
- NC Drill Dialog Box
- NC Parameters Dialog Box
- Generating Drill File Output in Batch Mode

net

The `net` command is used in conjunction with the following and any other commands requiring selection of a specific net:

- `net logic` to edit nets
- `property edit` to locate nets
- `show element` command to display information on the named selection

Syntax

`net <net_name>`

<net_name>	Specify the name of the net that you want to edit, locate, or display information.

Related Topics

- Net Logic
- property edit
- show element
- Show Element Dialog Box
- Show Properties Dialog Box
- Edit Property Dialog Box

net_properties

The `net_properties` command launches Constraint Manager and displays a worksheet of general net properties. You can use this worksheet to find and apply general net properties. You can also customize the worksheet to meet your needs. Once you save the customized view in the Constraint Manager, it becomes your default view.

The worksheet of net properties is called the *All* worksheet, which is located in the *Net: General Properties* workbook in the Constraint Manager.

The *All* worksheet contains the following general net properties:

- VOLTAGE
- WEIGHT
- NO_RAT
- ROUTE_PRIORITY
- FIXED
- NO_ROUTE
- NO_RIPUP
- NO PIN ESCAPE
- NO_TEST
- TESTPOINT_QUANTITY
- PROBE_NUMBER
- NO_GLOSS
- SHIELD_NET
- SHIELD_TYPE

Access Using:

- Menu path: *Edit – Net Properties*

Related Topics

- Allegro Constraint Manager User Guide

net delay report

The `net delay report` command calculates timing delays in picoseconds on a net-by-net basis and then output the results to a Net Delay Report. This report provides valuable information for designs in various stages of completion:

- Following net assignment
  Generating a delay report at this stage provides a rough estimate of delay values by calculating the distance of ratsnest lines in your design. You can use the results of the report to consider alternate assignments if, for example, delay value is unacceptably high.
- After performing `route feasibility`
  The results of the delay report at this stage is based on the route feasibility lines determined by the routing schedule. Delay values are more accurate at this point than following net assignment.
- Upon completion of routing
  Provides final verification of the delay value of each routed net in your design, based on connect lines (clines).

The net delay report supports and lists all pin pair delays for multi-chip designs.

Access Using

- Menu Path: *Tools – Net Delay Report*

Related Topics

- Generating a Net Delay Report

Net Delay Report Operating Parameters and Structure

Be aware of the following conditions when you run `net delay report`:

- All clines, including bond wires, must have a line width greater than 0.
- Power and ground net assignments are not calculated if attached to property VOLTAGE
- The followed design elements are ignored
  - Pins and clines not assigned to a named net (a "dummy net")
  - Cline segments connected to a plating bar
  - Dangling cline segments and clines with property FILLET
  - Shapes
  - Vias
- Thicknesses and z-dimensions are based on the design s layer stack-up and material parameters, such as dielectric constant values and conductivity.
- Reports are for use with models, simulations, and analysis techniques developed for version 15.1 and up

The net delay report contains a header section and a data section. The header section records when the report was generated, and the name and location of the current design. The data section is made up of columns that provide the following information:

Status	The type of net being analyzed. • *Routed:* the calculated delay value based on a fully routed net • *Rat:* the calculated delay value based on a ratsnest line • *Feas_line:* the calculated delay value based on a route feasibility line
Net	Net names, sorted in alphanumeric order
Delay	Calculated delay value in picoseconds
Connection	The reference designator and pin number of the die pin; the reference designator and pin number of the BGA ball.

Generating a Net Delay Report

To generate a net delay report, do the following:

1. Choose *Tools – Net Delay Report.*
2. Enter a file name and location for the report.
   The default name is *net_delay.rpt*.
3. Click *OK* to generate the report.
   A progress meter appears while the tool generates the report. The meter lists the net being processed and the percentage of completion.

   ⚠ The report generation might take a long time if your design is dense.If the report pauses on one net for an extended period, this may indicate that a power or ground net is missing the VOLTAGE property. Click the *Stop* beneath the command console to stop the report, add the property, and run the report again.

Example

```
Delay Report
============
Design Name : /hm/taylor/testcases/15.1/delay_rpt/delay_rpt_routed.mcm
Date/Time   : Aug  7 18:07:40 2003
```

Status	Net	Delay (ps)	Connection	
Rat	SIG_DIE-1_TO_BGA-A23	26.683567	DIE.1	BGA.A23
Rat	SIG_DIE-1_TO_BGA-A23	0.000000	DIE.1	BGA.I18
Rat	SIG_DIE-3_TO_BGA-A21	25.298473	DIE.3	BGA.A21
Rat	SIG_DIE-3_TO_BGA-A21	0.000000	DIE.3	BGA.I17
Routed	SIG_DIE-4_TO_BGA-A20	18.941618	DIE.4	BGA.A20
Rat	SIG_DIE-5_TO_BGA-A19	24.166313	DIE.5	BGA.A19
Rat	SIG_DIE-5_TO_BGA-A19	0.000000	DIE.5	BGA.I16

Routed	SIG_DIE-6_TO_BGA-B17	15.536806	DIE.6	BGA.B17
Rat	SIG_DIE-7_TO_BGA-A17	23.323999	DIE.7	BGA.A17
Rat	SIG_DIE-7_TO_BGA-A17	0.000000	DIE.7	BGA.I15

Related Topics

- net delay report

netin

The `netin` command displays the Import Logic dialog box, where you load the logic for your design into the design's database and establish the operating characteristics for the netrev utility. The `netin` command also assigns any extra functions or gates to packages/parts.

Syntax

```
netin [-aA|-bB|-cC|-dD|-eE|-g|-s|-v|-x|-y<number>] [z] netlist [input_drawing] [output_drawing]
```

-a or -A	Creates an A size drawing.
-b or -B	Creates a B size drawing.
-c or -C	Creates a C size drawing.
-d or -D	Creates a D size drawing.
-e or -E	Creates an E size drawing.
-f	Reserved for Cadence internal use only.
-g	Runs gate assignments.
-h	Searches and updates device STEP mapping data. ⚠ Logic imports automatically imports STEP mapping data. Ensure to set *step_facet_path* and *step_mapping_path* environment variables before logic import.
-s	Supersedes all logical data with the new netlist.
-v	Prevents creation of the device log.
-x	Corresponds to the *Allow etch/conductor removal during ECO* field in the Import Logic dialog box. Use the `x` switch to rip up etch/conductor during an ECO.

-y <number>	Corresponds to the *Place changed components* in the Import Logic dialog box. Use this switch as follows: `-y1` = *Always* `-y2` = *If same symbol* `-y3` = *Never*
-z	Ignores FIXED property when updating the design.
File Names	
`netlist`	Specifies the name of the input file.
`<input_drawing>`	Specifies the drawing name in which to load the netlist. Type the name without the `.brd` file extension.
`<output_drawing>`	Specifies the drawing name in which to store the resulting design data. If you do not enter an output drawing name, `netin` overwrites the input design file.

⚠ If you do not specify either the input or the output drawing, `netin` runs in a *Syntax check only* mode and writes the syntax check messages to the log file, making no changes to the drawing. This is a fast method to check for syntax errors in the netlist file.

⚠ You do not need to type the `.txt` file name extension. If you enter an option, or a group of options, precede them with a single dash.

For device, symbol, and padstack filenames, Allegro X PCB Editor always converts the names into lowercase when locating files to insert into a board. This can create a problem between UNIX and Windows because UNIX files are case-sensitive; Windows files are not. To prevent such potential issues, create these files using lowercase naming conventions only. For example, Allegro X PCB Editor can find a file named `7400A.txt` on Windows, but the file must be named `7400a.txt` for UNIX.

You can run this command from the console window prompt or as a batch command from an operating system prompt.

⚠ The `netin` command is identical to `netin param`.

Related Topics

- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Checking netlist Syntax
- Loading Logic Data
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode
- netin param
- netrev
- Transferring Logic Design Data

Import Logic Dialog Box

Use this dialog box to load the logic for your design into the design's database and establish the operating characteristics for the `netrev` utility.

Logic is derived natively (that is, from a Cadence front-end source) or from a third-party netlist. Choose the appropriate tab to set the parameters for loading logic into your design.

Access Using

- Menu path: *File – Import – Logic*

Import Logic Dialog Box — Cadence Tab		
Branding	Identifies the logic format of the files being loaded.	
Import logic type	Choose the type of logic you want to load. Initially, this field displays the logic type associated with the active design.	
Place changed component	Specifies whether any components that have been changed are placed in the design. An ECO can result in a reference designator applying to a different type of device in the schematic than the device in the layout. This selection tells the tool what to do when you load this new logic into the design's database. If the design has not been placed or routed, the new transfer files simply replace the original design database. These are the choices:	
	Always	(Default) Replaces all components in the layout with the new components from the Packager according to their reference designators. This option lets the tool replace one type of component with an entirely different type of component.
	Never	Replaces no components in the layout with the new components from the Packager. You must make the changes interactively.
	If same symbol	Replaces the components in the layout with the new components from the Packager according to reference designator, only if the replacement component matches the package/part symbol of the component in the layout.

HDL Constraint Manager Enabled Flow options	Determines how electrical constraints are imported by way of `pstcm*.dat` files. This option is available only when you choose *HDL-Concept* for *Import logic type*. These are the choices:	
	Import changes only	Imports only changes to electrical constraints in the schematic. After initially importing the logic, choose this option for all subsequent logic imports.
	Overwrite current constraints	Modifies all electrical constraints in the design with the constraints in the schematic. Only when you first import logic to a design should you choose this option. Only attributes defined in the schematic will be modified.
Show constraint difference report	Displays the constraint differences between the source and destination designs.	
Allow etch removal during ECO	Specifies what happens to etch/conductor that connects to a pin when an ECO removes that pin from a net. ◦ If you check this option, the tool rips up the etch/conductor from a removed pin to the closest T connection or pin. This option saves you time. ◦ If you do not check this option, you rip up the etch/conductor interactively.	
Ignore FIXED property	Allows the command to replace and delete symbols, rip up etch/conductor, and make other changes even if elements of your design are fixed (are assigned the FIXED property).	
Create user-defined properties	Check this box to allow the creation of property definitions from the netlist.	
Create PCB XML from input data	Creates an XML file of the schematic for the current board. The file created is `boardname_sch.xml`. You can view this file when you start up the PCBCompare tool.	

Design Compare	Press this button to start up the PCBCompare tool. PCB Compare displays the schematic file (XML) on the left and the XML file of the board on the right of the main window.
Rename existing refdes	(Appears only in Allegro SI) Specifies whether you are mapping reference designators in the imported netlist to the reference designators in the design database. Brings up the RefDes Mapping dialog box after the netlist is imported.
Import directory	Specifies the location of your `pst*.dat` files. By default, the path is your current working directory, indicated by a period.

Import Logic Dialog Box — Other Tab	
Import netlist	Identifies the name of the third-party netlist being loaded. For information regarding proper netlist syntax, punctuation, and guidelines for writing a netlist, see the *Transferring Logic Design Data user guide in your documentation set.*
Syntax check only	Determines whether the syntax of the input file should modify the physical layout of the design. ◦ If you check this option, `netin` reads the netlist and creates a log file listing any syntax errors in the file. The `netin` program does not load the netlist data into your drawing and uses new design syntax rules. For example, the name size value is taken from *allegro_long_name_size* environment variable. If you have created many device files by hand or if you created your netlist manually, run `netin` with this option selected until it runs error-free. ◦ If you do not check this option, `netin` first performs a syntax check using existing design syntax rules. For example, the name size value is taken from *Long name size* parameter in the Design Parameter Editor. If your netlist is free of errors, proceeds to load the netlist. If an automated system created the netlist, you do not need to run a separate syntax check. By default, this option is not selected (`netin` loads the netlist).

Supersede all logical data	Determines whether `netin` deletes all existing logical data before loading the new netlist. ◦ If you check this option, it replaces all existing logic with the new netlist. ◦ If you do not check this option, it adds the new netlist to the existing logic. By default: ◦ In PCB Editor, this option is not selected. ◦ In Allegro SI, this option is selected.
Reuse device files for existing components	Specifies that `netin` reuse component definitions for components that already exist in the design instead of re-importing them. As a result, changes to these device files will not be seen on refreshing the design.
Append device file log	Determines whether `netin` appends the device file log. ◦ If you check this option, `netin` writes all messages created by the device file parser as it parses the device files to the `netin.log` file. If you have created any new device files for this design, this parameter lets you save the device file parser messages so you can check them. ◦ If you do not check this option, `netin` does not append the device file messages to the log. If all device files are standard library files, you do not have to use this option. By default, this option is not selected (no append).
Allow etch removal during ECO	Specifies what happens to etch/conductor that connects to a pin when an ECO removes that pin from a net. ◦ If you check this option, the tool rips up the etch/conductor from a removed pin to the closest T connection or pin. This option saves you time. ◦ If you do not check this option, you rip up the etch/conductor interactively.
Ignore FIXED property	Allows the command to replace and delete symbols, rip up etch/conductor, and make other changes even if elements of your design are fixed (are assigned the FIXED property).

Rename existing refdes	(Appears only in Allegro SI) Specifies whether you are mapping reference designators in the imported netlist to the reference designators in the design database. Brings up the RefDes Mapping dialog box after the netlist is imported.

Related Topics

- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Checking netlist Syntax
- Loading Logic Data
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode

RefDes Mapping Dialog Box

Use this dialog box to map reference designators from the netlist you are importing to the reference designators in the design database. This is available only in Allegro SI.

RefDes Mapping	
RefDes Filter	Searches on part data by reference designator.
Device Filter	Searches on part data by device name.
Package filter	Searches on part data by package name.
Board RefDes	The left-hand box lists the devices in the design database.
Imported RefDes	The right-hand box lists the devices in the netlist you are importing.
Match	Maps the highlighted item in the *Imported RefDes* list to the highlighted item in the *Board RefDes* list.
RefDes Match	Lists the device mappings from the imported netlist (in the left columns) to the design database (in the right columns).
OK	Starts the mapping process.
Cancel	Stops the mapping process, but imported data remains in the board.

Related Topics

- netin
- Importing Third-Party Logic
- Checking netlist Syntax
- Loading Logic Data
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode
- Transferring Logic Design Data

Importing Native (Cadence) Logic

To import Cadence logic, do the following:

1. Choose *File – Import – Logic*.
2. Complete the Cadence tab of the Import Logic dialog box.
3. Click *Import Cadence*.
4. (Allegro SI only) If you selected the *Rename existing refdes* option, map reference designators in the RefDes Mapping dialog box.
5. When the import is complete, click *Close*.
   The `netrev` utility reads and compiles the netlist design logic, updates the active board/substrate, and then creates the `netrev.lst` file, which appears in a window. It also creates the `eco.txt` file, which contains all the changes to a database that result from loading the schematic logic.

Related Topics

- netin
- Import Logic Dialog Box
- Checking netlist Syntax
- Loading Logic Data
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode
- Transferring Logic Design Data

Importing Third-Party Logic

To import non-Cadence or third-party logic, do the following:

1. Choose *File – Import – Logic*.
   The Import Logic dialog box appears.
2. Complete the Other tab in the Import Logic dialog box.
3. Click *Import Other*.
4. (Allegro SI only) If you selected the *Rename existing refdes* option, map reference designators in the RefDes Mapping dialog box.
5. When the import is complete, click *Close*.
   The `netin` command reads and compiles the netlist and generates the `netin.log` file.

Related Topics

- netin
- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Loading Logic Data
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode
- Transferring Logic Design Data

Checking netlist Syntax

You can look for syntax errors in both the netlists and the device files specified in a netlist.

1. Choose *File – Import – Logic*.
   The Import Logic dialog box appears.
2. Choose Third party in the Netlist Type field.
3. Enter the netlist `filename.txt` in the *Import netlist* field.
   If the netlist is not in the current working directory, specify a complete path.
4. Choose *Syntax check only*.
5. Choose *Append device file log*.
   This selection is optional. netin generates a log file, named `netin.log`, that contains a copy of the netlist and syntax errors. If you choose *Append device file log*, the log file also contains a copy of the device files that are in the netlist and the syntax errors in the device files.
6. Click *OK*.
   `Netin` reads and compiles the netlist and generates the `netin.log` file.
   The following example shows the contents of a `netin.log` file after netin has checked syntax. In this example a comma was omitted from the list of reference designators that was continued on another line. `netin` assumed that the reference designator, Y29, was a package name.

   ```
   (NETLIST)

   (FOR DRAWING: /Allegro/APD_test/dfa.brd)

   (Thu Nov 19 12:28:54 1992)

   $PACKAGES

   CAPCK05 ! 'CAPACITOR-1' ; C2 C4 C6 C8

   CAPCK05 ! 'CAPACITOR-2' ; C1 C3 C5 C7

   CONN10 ! CONNECTOR ; J1 J2 J3

   DIP14 ! 74F02 ; N20 N29

   DIP14 ! 74F74 ; N05 N08 N11 N14 N17 N23 T11

   Y29

   ^
   ```

ERROR: Expected '!' before device, line ignored.

DIP16 ! 74F138 ; N26

Related Topics

- netin
- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Running the Netin Command in Incremental Mode
- Running the Netin Command in Supersede Mode

Loading Logic Data

After you check the syntax on the netlist, you load the logic data to create a database for the current layout.

To load logic, do the following steps:

1. Run the `netin` command.
   The Import Logic dialog box appears with the netlist you specified when you checked the syntax.
2. Turn off *Syntax check only*.
   The other options are for updating the layout after an ECO. Do not use them now.
3. Click *OK*.
   Netin reads and compiles the netlist design logic, then creates a design database for the current layout and a log file named `netin.log`.

Related Topics

- netin
- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Running the Netin Command in Supersede Mode

Running the Netin Command in Incremental Mode

To run netin in incremental mode, do the following:

1. Choose *File – Import – Logic*.
   The Import Logic dialog box appears.
2. Enter the netlist `filename.txt` in the *Import netlist* field.

   ⚠ Check the syntax of your netlist before loading the changes.

3. Be sure *Supersede all logical data* is turned off.
4. Choose *Append device file log.*
   This is optional. Use it to write a copy of the device files specified in the netlist in the `netin.log` file. These copies of device files help you when the `netin.log` file contains warning or error messages about device files.
5. Place a changed component.
6. Choose *Allow etch removal during ECO*.
   This is optional. Use it to rip up the etch/conductor in the layout on nets that you delete with entries in the netlist. If you do not choose it, the etch/conductor remains in the layout.
7. Click *OK.*

Related Topics

- netin
- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Checking netlist Syntax

Running the Netin Command in Supersede Mode

Do the following to prepare running netin in supersede mode:

1. Backannotate the current layout to the schematic before you generate the new netlist from the schematic.
2. Make sure that all ECL terminators in the layout are also in the schematic before you generate this new netlist.
3. Check the properties in the device files to make sure they contain an accurate description of the device.
4. Generate the netlist.
5. Check the netlist.
   If it is a $FUNCTIONS netlist, be sure that the pin designators in the $NETS section include reference designators, function designators, and pin numbers.

To run netin in the supersede mode, do the following:

1. Choose *File – Import – Logic*.
   The Import Logic dialog box appears.
2. Enter the name of the netlist file (with a `.txt` extension) in the *Import netlist* field.

   ⚠ Check the syntax of your netlist before loading the changes.

3. Choose *Supersede all logical data*.
   This specifies that netin run in supersede mode.
   Choose *Append device file log*.This is optional. Use it to write a copy of the device files specified in the netlist in the `netin.log` file. These copies of device files help you when the `netin.log` file contains warning or error messages about device files.
4. Choose *Allow etch removal during ECO*.
   This is optional. Use it to rip up the etch/conductor in the layout on nets that you delete with entries in the netlist. If you do not choose it, the etch/conductor remains in the layout.
5. Choose one of the following options in *Place changed component*

Always	Replaces all deleted components with the new component that has the reference designator of the deleted component. This is the default selection.

If same symbol	Replaces all deleted components with the new components that have the reference designator of the deleted component but only if the replacement component uses the same package symbol as the deleted component.
Never	Does not replace components with new components. When you choose Never you must place the new components.

6. Click *OK.*
   The database is updated with the changes from the netlist.

Related Topics

- netin
- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Checking netlist Syntax
- Loading Logic Data

netin param

The `netin param` command displays the Import Logic dialog box, where you load the logic for your design into the design's database and establish the operating characteristics for the netrev utility. The `netin param` command also assigns any extra functions or gates to packages/parts.

Syntax

```
netin param[-aA|-bB|-cC|-dD|-eE|-g|-s|-v|-x|-y<number>] [z] netlist [input_drawing] [output_drawing]
```

-a or -A	Creates an A size drawing.
-b or -B	Creates a B size drawing.
-c or -C	Creates a C size drawing.
-d or -D	Creates a D size drawing.
-e or -E	Creates an E size drawing.
-f	Reserved for Cadence internal use only.
-g	Runs gate assignments.
-h	Searches and updates device STEP mapping data. ⚠ Logic imports automatically imports STEP mapping data. Ensure to set *step_facet_path* and *step_mapping_path* environment variables before logic import.
-s	Supersedes all logical data with the new netlist.
-v	Prevents creation of the device log.
-x	Corresponds to the *Allow etch/conductor removal during ECO* field in the Import Logic dialog box. Use the `x` switch to rip up etch/conductor during an ECO.

-y <number>	Corresponds to the *Place changed components* in the Import Logic dialog box. Use this switch as follows: `-y1` = *Always* `-y2` = *If same symbol* `-y3` = *Never*
-z	Ignores FIXED property when updating the design.
Filenames	
`netlist`	Specifies the name of the input file.
`<input_drawing>`	Specifies the drawing name in which to load the netlist. Type the name without the `.brd` file extension.
`<output_drawing>`	Specifies the drawing name in which to store the resulting design data. If you do not enter an output drawing name, `netin` overwrites the input design file.

⚠ If you do not specify either the input or the output drawing, `netin param` runs in a *Syntax check only* mode and writes the syntax check messages to the log file, making no changes to the drawing. This is a fast method to check for syntax errors in the netlist file.

⚠ You do not need to type the `.txt` file name extension. If you enter an option, or a group of options, precede them with a single dash.

For device, symbol, and padstack filenames, Allegro X PCB Editor always converts the names into lowercase when locating files to insert into a board. This can create a problem between UNIX and Windows because UNIX files are case-sensitive; Windows files are not. To prevent such potential issues, create these files using lowercase naming conventions only. For example, Allegro X PCB Editor can find a file named `7400A.txt` on Windows, but the file must be named `7400a.txt` for UNIX.

You can run this command from the console window prompt or as a batch command from an operating system prompt.

⚠ The `netin param` command is identical to `netin`.

Access Using

- Menu path: *File – Import – Logic*

Related Topics

- Import Logic Dialog Box
- RefDes Mapping Dialog Box
- Importing Native (Cadence) Logic
- Importing Third-Party Logic
- Checking netlist Syntax
- Loading Logic Data
- Running the Netin C*ommand* in Incremental Mode
- Running the Netin *Command* in Supersede Mode
- netin
- netrev
- Transferring Logic Design Data

net list in

The `net list in` command creates and assigns nets for components after you have created a BGA package and die. In the Netlist-In wizard, you can:

- Generate pin assignments and connectivity by importing an ASCII spreadsheet of net information
- Manipulate the spreadsheet information in the Netlist-In wizard to modify individual net values
- Place columns of data in a standard format

Prerequisites

- If the netlist information is in a spreadsheet, convert it to ASCII text format because the Netlist-In wizard processes ASCII text files.
- Make sure that the reference designators in the netlist file match placed components in the design.

Related Topics

- Routing the Design
- Defining Connectivity Automatically

Netlist-In Wizard

Use this wizard to create and assign nets for components for BGA package and die.

Access Using

- Menu path: *File – Import – Netlist-In Wizard*

The Netlist-In wizard flags the error when a pin is assigned to multiple nets in the file. The offending line is highlighted and you must check one of the lines as *Ignore* to continue.

Netlist-In Wizard, Step 2: File Information Dialog Box

In this step of the wizard, specify how to delimit the netlist data.

You cannot edit the file in the text window.

Delimiters	Defines the delimiter that your spreadsheet is using to separate columns of data. *Tab* is the default.
Ignore consecutive delimiters	Specifies that consecutive delimiter characters be considered one character.
Remove trailing delimiters	Specifies that the tool delete trailing, consecutive delimiter characters from the data.

Netlist-In Wizard, Step 3: Net Information Dialog Box

In this step of the wizard, specify net information. The required columns cannot be labeled with *Ignore*.

Empty net name indicates continuation of previous net	Click this button so that when APD finds empty net name fields, it recognizes that the current line is a continuation of the last line entry for the net. This is the default setting.

Empty net name indicates dummy net assignment	Click this button so that when APD finds empty net name fields, it assigns the pin to a dummy net.
Empty net name indicates error (Blank net names not allowed)	Click this button so that when APD finds an empty net name field, it treats this as an error. The cell is highlighted and the net name column is set to "ignore" to preserve data integrity. You can ignore the offending row or choose another option for how to treat the empty cell.
Ignore Rows	Designates the rows of data that the Netlist-In Wizard does not import. To choose rows to be ignored, click on the box in the corresponding row to place an **x** in it.
If other column headings are incorrect, open a pop-up menu over the heading and choose the correct type of information displayed in the column.	
These are the column headings you can choose:	
Ignore	Data in this column is for reference only and is not imported by the Netlist-In Wizard.
Pin Number	(Required) Pin numbers of the pins in the BGA or die.
Mixed Case Pin Number	Allows you to import and export mixed-case names of the object, for example, from LEF/DEF or OpenAccess.
Pin Name	Specifies the logical pin name that is different from the physical pin number.
Net Name	(Required) Net names assigned to each pin. If the net does not already exist for a pin, you can create it in this column.
Mixed Case Net Name	Allows you to import and export mixed-case names of the nets, for example, from LEF/DEF or OpenAccess.
Ref Des	(Required) Reference designator of the BGA or die whose pins are listed in the corresponding *Pin Number* column. These components must have been previously placed in the design.

Net Prop Name	Property names of the nets. The Net Prop Name and Net Prop Value columns must be present in matched pairs in the file. The prop name column indicates the name of a property which should be created on the netlisted in the net name column, while the prop value column gives the value the property should be set to. One example would be a prop name of *VOLTAGE* with a value of `0.0v` to be created on a net to indicate it is a ground net.
Net Prop Value	Property values of the nets.

⚠ You cannot choose a type of information that is already used in another heading. To switch headings, you may have to first choose *Ignore* for one heading before changing another heading. For example, if you want to swap the *Net Name* and *Ref Des* headings, first change *Net Name* to *Ignore*, then change *Ref Des* to *Net Name,* and then *Ignore* to *Ref Des*.

Change any cell by clicking on it and entering new data.

Defining Connectivity Automatically

⚠ Make sure that the reference designators in the netlist file match placed components in the design.

To define connectivity automatically, do the following steps:

1. If the netlist information is in a spreadsheet, convert it to ASCII text format in a spreadsheet program.
   Note that the ASCII file is tab-delimited by default. But you can also specify any other character to separate the columns, in addition to the standard separators: Space, Semicolon (;), and Comma (,). The file can contain columns for the fields Pin Number, Mixed-Case Pin Number, Pin Name, Net Name, Mixed-Case Net Name, RefDes, Net Prop Name, and Net Prop Value. The order of the columns is not important. The fields Pin Name, Net Name, and RefDes must be present in the file.

   ⚠ You can list the RefDes-Pin Name pairs for a particular Net Name in one line or you can list them in separate lines. The tool imports only 12 RefDes-Pin Name pairs listed in one line. If a net contains more than 12 RefDes-Pin Name pairs, list them in separate lines to ensure a line has no more than 12 pairs.

2. Choose *Generate – Netlist-In Wizard*.
3. In the Netlist In Wizard dialog box, choose the ASCII file containing the netlist information.
4. Complete the Netlist-In Wizard Step 2– Delimiters dialog box.
5. Complete the Netlist-In Wizard Step 3– Net Information dialog box.
6. Click *Finish* to import the information from the text file and create a logical database.

Related Topics

- net list in

net logic

The `net logic` command interactively creates and edits nets in a design. The Options tab controls the editing functions for this command.

⚠ By default, this command is disabled to prevent accidental changes in logic. To enable it, run the `enved` command, choose the *Misc* category, and enable the *logic_edit_enabled* preference.

Access Using

- Menu path: *Logic – Net Logic*

Related Topics

- Editing Nets
- Placing the Elements

Creating a Net

⚠ If you are using Allegro Design Entry HDL or Allegro System Architect, you may not be able to backannotate your design if you edit the netlist.

Ensure *logic_edit_enabled* under Misc is enabled before running this task.

To create a new net, do the following:

1. Choose *Logic – Net Logic.*
2. In the Options tab, click *Create*.
3. Enter a new net name in the pop-up window and click *OK*.
   If it is accepted, the new name appears as the selected net in the list of net names. (Messages in the command console display success/failure status for each action you perform.)
   The new net is immediately available for pin assignment.

Editing Nets

You can perform many functions with nets. You can assign nets to pins or deassign nets from pins. You can also rename or remove a net.

Ensure *logic_edit_enabled* under Misc is enabled before running the tasks in this topic.

Assigning Net to Pin

To assign a net to a pin, do the following:

1. Choose *Logic – Net Logic.*
2. In the Options tab, make sure that *Assign* is selected.
3. Choose a net. For details, see Selecting a Net.
4. If you want to identify the net to which a pin is attached, choose *Identify* from the pop-up menu.
5. Choose a pin. For details, see Selecting a Pin.
   The selected net is assigned to the pin. If the pin already had a net assigned to it, the selected net replaces it.

De-assigning a Net from Pin

To remove an attached net from a pin, do the following:

1. Run the `net logic` command.
2. In the *Options* tab, enable *Deassign*.
3. If you want to delete etch/conductor connected to the pin you are deassigning, click *Ripup Etch/Conductor*. Etch/conductor is deleted back to the next pin or junction.
4. Choose a net. For details, see Selecting a Net.
5. If you want to identify the net to which a pin is attached, choose *Identify* from the pop-up menu.
6. Choose a pin. For details, see Selecting a Pin.
   The selected net is removed from the pin, and the pin is attached to a dummy net.

Renaming a Net

To change the name of a net, do the following:

1. Run the `net logic` command.
2. Choose a net. For details, see Selecting a Net.
3. In the *Options* tab, click *Rename*.
4. Enter the new net name in the pop-up window and click *OK*.
   If it is accepted, the new name appears as the selected net in the list of net names. The old name no longer exists. (Messages in the command console display success/failure status for each action you perform.) The renamed net is immediately available for further editing.

Removing a Net

To remove a net from your design, do the following:

1. Run the `net logic` command.
2. Choose a net. For details, see Selecting a Net below.
3. In the *Options* tab, click *Remove*.
   A confirmation message appears. Text within the confirmation window depends on whether the net you want to remove contains pin or shape assignments.
4. Click *Yes* to complete the action. –or– *No* to close the confirmation window without removing the net.
   If you choose *Yes*, the pins are assigned to a dummy net.

Selecting a Net

To choose a net, do one of the following:

- Entering the net name in the *Net* field
- Selecting the net name from the scroll list
- Picking a net-carrying object in the design window
- Entering `net <`*name*`>` at the console window prompt

Selecting a Pin

To choose a pin, do one of the following:

- Entering the pin name <*refdes.pin_number*> in the *Pin* field of the Options tab

- Picking a pin in the design window
- Entering `refdes pin <`*pin_name*`>` at the console window prompt

Related Topics

- net logic

Selecting a Net

To choose a net, do one of the following:

- Entering the net name in the *Net* field
- Selecting the net name from the scroll list
- Picking a net-carrying object in the design window
- Entering `net <`*name*`>` at the console window prompt

Selecting a Pin

To choose a pin, do one of the following:

- Entering the pin name <*refdes.pin_number*> in the *Pin* field of the Options tab
- Picking a pin in the design window
- Entering `refdes pin` <*pin_name*> at the console window prompt

netout

The `netout` command displays the Netout dialog box for generating a netlist output file that contains pin and net properties for the current design.

Access Using

- Menu Path: *File – Export – Netlist with Properties*

Generating a Netlist with Properties Output File

Do the following steps to open the Netout dialog box for generating a netlist output file that contains pin and net properties for the current design:

1. Choose *File – Export – Netlist w/Properties*.
2. Enter the name of the netlist output file.

You do not need to add the `.txt` extension. The default name is `netlist.txt`.

1. Click *OK*.

netrev

The `netrev` batch command reads the *`pst*.dat`* files generated by Packager-XL into a physical design to create a fully assigned database. You can also run this utility through the *Import Logic* dialog box using the `netin` command.

Syntax

`netrev [<arguments>] [<input_board>] [<output_board>]`

[-e]	Reports footprints missing from the design as errors. Used for library development. If you do not use this argument, `netrev` reports missing footprints as warnings, unless you have set *netrev_missing_footprints* environment variable, which reports any footprint warnings as errors and causes logic import to fail.
[-d]	Disable DRC checking.
[-i <directory>]	(Optional) Indicates where to find the `.pst` files. If the given project file uses the `-proj` option, `netrev` looks in the directory specified by the "view_packager" global directive in the project file. If you do not specify a project file, `netrev` looks in the current working directory. The `-i` argument overrides the project file. The `-proj` option is described in the CPM control Options section of this table.
[-n]	Generates a new board as you import the logic. With this argument, you are not importing logic into an existing board; instead, you are creating a new board with `netrev`. You specify the name of the new board in the <*input_board*> variable, described at the end of this table.
[-t]	Generates an XML file `u_sch.xml` for design compare. This is based on the contents of `.pst` files.
[-x]	Rips up connecting etch/conductor when ripping up components during an ECO.

[-r 1|2|3]	Rip up options available when rip up is enabled (-x):
	`r1`: Deletes the first etch-segment only.
	`r2`: Retains bondwires for wirebonded components, but deletes connecting etch beyond the bondfingers.
	`r3`: Both 1 and 2, retain bondwires and delete the first etch segment beyond the bondfingers
[-y 1|2|3|4|5]	Indicates how symbols are replaced if ripped up during an ECO:
	`y1`: Always replaces the symbol (default).
	`y2`: Replaces the symbol only if it is the same symbol as was there originally.
	`y3`: Never replaces the symbol.
	`y4`: Does not rip up the symbol; instead, unassigns it. This has the effect of turning it into a non-logic- bearing component.
	`y5`: Replace symbol even if there is change in component and symbol definitions, if the pins match.
[-z]	Ignores fixed properties. This allows the program to replace and delete symbols and rip up etch/conductor if they are fixed in the design.
[-h]	Searches and updates STEP mapping data for devices. ⚠ Logic imports automatically imports STEP mapping data. Ensure to set *step_facet_path* and *step_mapping_path* environment variables before logic import.
[-t]	Writes out a copy of the input netlist in Cadence PCB XML format, which then can be used within the PCBCompare netlist comparison tool. The name of file is <u_sch.xml>. The contents of file are based on the contents of the .pst files.
[-u]	Creates schematic user-defined properties in the PCB Editor. Default is to not create property definitions for those properties not already defined in the design.

SCALD-Only Arguments		
[-g]	Imports SCALD-based Packager files.	
[-s <schematic_ path>]	Defines the path to the schematic directory.	
[-p<layout path>]	Path to the input board.	
CPM control Options		
[-l]	Prevents changing to the pstDirectory if running with the `-proj` argument.	
[-proj <project_file>]	The HDL project file. This typically has with a `.cpm` extension.	
[-f <file>]	Writes constraint difference report in design difference format.	
[-q <file>]	Writes constraint difference report in design difference format. If there are conflicts or the designs are corrupt, launches constraint difference report viewer to show differences in the design.	
[-o]	Overwrites current electrical constraints if you are using a Constraint Manager-enabled flow. The default (no `-o`) is to import changes only. In case version mismatch is detected: [-1] Overrides a changes-only error and update the design with the reported constraint changes. [-2] Overrides a changes-only error and update the design without the reported constraint changes.	
[-v]	Launches constraint difference report viewer to show differences in design difference format. Report will be written to the temporary file.	
[-w <file>]	Launches constraint difference report viewer to show differences in design difference format. Report will be written to the file specified.	

[-b <file>]	Specifies the whitelist file. The whitelist file contains only those properties that are being reported in the constraint difference report. ⚠ Whitelist file is defined only when output board is different from the input board. It is recommended to delete output board after running netrev.
Board Names	
[<input_board>]	Specifies the design file that `netrev` is updating with logic. You can enter a design with any of these extensions: `.brd`, `.mcm`, `.dra`, or `.mdd`.
[<output_board>]	Specifies the file where the resulting design data is stored. If you do not enter an output design name, `netrev` overwrites the input design file.
Obsolete Options (for backward compatibility but ignored)	
[-m]	Enables migration mode. Use this only if migrating a SCALD design to HDL. Must be used with the `-5` HDL argument. When you use this argument, `netrev` attempts to migrate a SCALD design to HDL using `.pst` files. The layout must match the schematic for migration to be successful. If `netrev` finds design changes, the process stops with an error.
[-5]	Indicates the default HDL mode.
[-c]	Composer logic mode. Used if `.pst` files were generated from Composer.
[-g]	Packager files are SCALD based.
[-s<schematic path>]	The path to the schematic directory (Scald only)
[-p<layout path>]	The path to the input board (Scald only).
Import Logic Files(pst files)	
Modern(Single file flow)	`pstdedb.cdsz` ⚠ Do not delete this file, if present.

Design Entry HDL/Allegro System Architect Constraint Manager enabled (5 file flow) flow	New design: `pstchip.dat,pstxprt.dat, pstxnet.dat,pstcmdb.dat` ECO: `pstchip.dat,pstxprt.dat, pstxnet.dat,pstcmdb.dat,pstcmbc.dat` Optional dml file: `pstdmlmodels.dat`
Traditional (3 file flow)	`pstchip.dat,pstxprt.dat,pstxnet.dat`
OrCAD Traditional (3 file flow)	Base: `pstchip.dat,pstxprt.dat,pstxnet.dat` NetGroup option file: `pstngdefs.dcf`

In a Constraint Manager-enabled flow, `netrev` creates Xnets in the design based on information in the `pstcmdb.dat` file. When *File – Export Physical* executes in Design Entry HDL or Allegro System Architect, Constraint Manager connected to Design Entry HDL or Allegro System Architect reads the SIGNAL_MODEL property on discretes and creates Xnets. (For a net to be recognized as an Xnet, discrete devices dividing the net into segments must have a signal model associated with it.) The SIGNAL_MODEL property from the `chips.prt` and phys_prt.dat files pass to `pstchip.dat`, and on individual instances, to `pstxprt.dat`.

⚠ An error occurs when the signal model referenced by the SIGNAL_MODEL property is missing, and as a result, the Xnet cannot be maintained, thereby losing electrical constraints. You can change this error to a warning using the `netrev_model_warning` environment variable.

- If Allegro X PCB Editor finds no signal model, EspiceDevice models auto-generate based on the default value of the SIGNAL_MODEL property for 2-pin discrete components.
- If PCB Editor finds no SIGNAL_MODEL property or no auto-generation occurs, it copies constraints from the Xnet to individual member nets, addressing pin pairs appropriately.
- If PCB Editor finds no SIGNAL_MODEL property on a discrete component, it deletes the existing Xnet, if any, from the board.
- If PCB Editor finds a SIGNAL_MODEL property with a user-defined value (and not prefixed with DEFAULT), but not the signal model in the given path, it deletes the Xnet from the `.brd` file, if it already exists.

The Allegro PCB Performance Option 220 has a static Xnet model. In 15.2, Xnets can be updated inside `netrev` when running from Design Schematic HDL in the Constraint Manager enabled mode. Once you open the design in Performance Option, the Xnets become static until you either open the

design in a 600-series product, or update it from the schematic due to an ECO.

Example

```
netrev  -i <packaged folder path> -t -y 3 -z -v <output board path>
```

In this example, `netrev` command reads packaged logic data and writes data into PCB XML format. During the process, the command compares package symbols and generates a constraint difference report to show differences in design difference format.

Related Topics

- netin
- Import Logic Dialog Box
- Transferring Logic Design Data

net schedule

The `net schedule` command interactively schedules or unschedules the order in which pins route in a particular net. You can schedule the entire net, partially schedule multiple sections of a net (subschedules), or insert Tpoints (ratsnest T) into a net. You can create subschedule connect points on subschedules to control where a subschedule connects to the remaining net. Once you schedule a net, Allegro X PCB Editor refers to this schedule as a user schedule.

Related Topics

- Scheduling a Net Interactively
- Partially Scheduling a Net
- Scheduling a Net with Tpoints

Net Schedule Pop-up Menu

Access Using

- Menu path: *Logic – Net Schedule*
- Toolbar icon:
  

Done	Saves your scheduling choices and exits the `net schedule` command.
Cancel	Cancels `net schedule` without scheduling the net.
Oops	Cancels the most recent action.
Insert T	Inserts a branch to the remaining pins in the net.
Unschedule Pin	Unschedules a pin without unscheduling the entire net.
Create Subschedule	Creates a partial schedule of the chosen pins and Tpoints and places a connection point on the last chosen pin or Tpoint. You can create any number of subschedules in any net.
Delete Subschedule	Deletes a chosen subschedule and removes the diamond from any connection points.
Add Subschedule Connection Point	Defines additional connection points to a subschedule.These become the only valid connection points between the subschedule and the remaining net.
Delete Subschedule Connection Point	Deletes an existing subschedule connection point.
Finish	Saves your scheduling choices and lets you schedule a new net.
Unschedule Net	Unschedules the entire net.

Related Topics

- Partially Scheduling a Net
- Scheduling a Net with Tpoints

Partially Scheduling a Net

You can partially schedule a net when you want a specific connection order for a portion of the net. Choose pins, multiple series of pins, or Tpoints in a net to specify the order in which to connect to other pins in the net. You can create subschedule connect points on a pin or Tpoint of a subschedule to designate where to join the remaining net connections. If you do not specify any subschedule connect points, any pin within the subschedule is a legal connection point during net routing. Multiple, partial schedules (subschedules) can exist in a net.

Creating a Subschedule with a Connect Point

A subschedule is a subset of pins in a net that have a specific connection order. The connect point on the subschedule determines where subschedules must connect to the other pins in the net. You can create as many subschedules as you want in a net.

1. Schedule the net interactively.
2. When you have connected all the pins in the subschedule, right-click and choose *Create Subschedule* from the pop-up menu.
   By default, the last pin scheduled becomes the connect point to the subschedule and is marked with a diamond.
3. Repeat this procedure to create additional subschedules in the same net.
   or
4. Right-click and choose *Done* from the pop-up menu to save your scheduling choices and exit the command or *Finish* from the pop-up menu to save your scheduling choices and schedule a new net.

Creating Additional Subschedule Connect Points

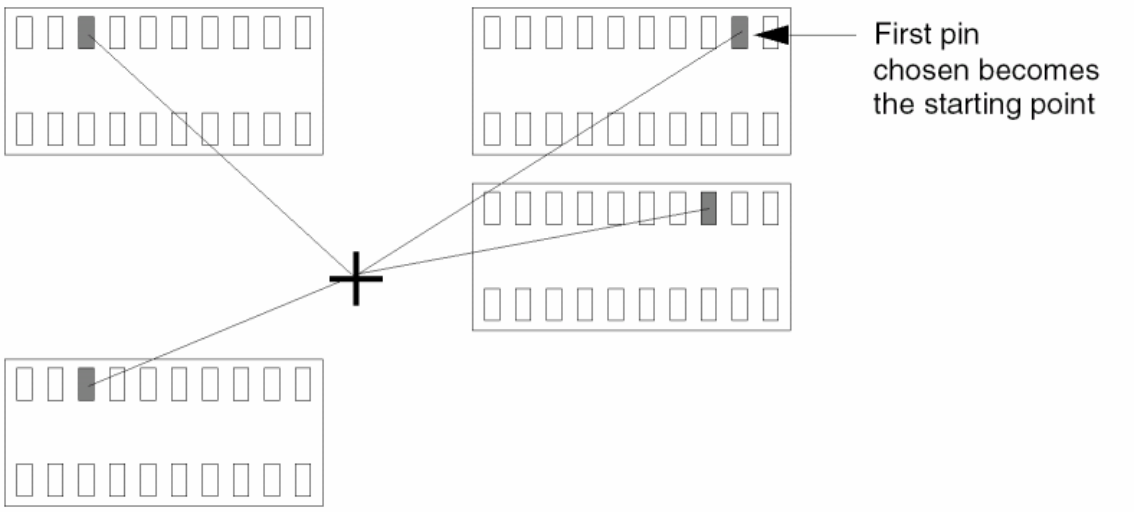
1. Right-click and choose *Add Subschedule Connection Point* from the pop-up menu.
   The next pin you pick becomes another connection point.

Deleting a Subschedule Connect Point

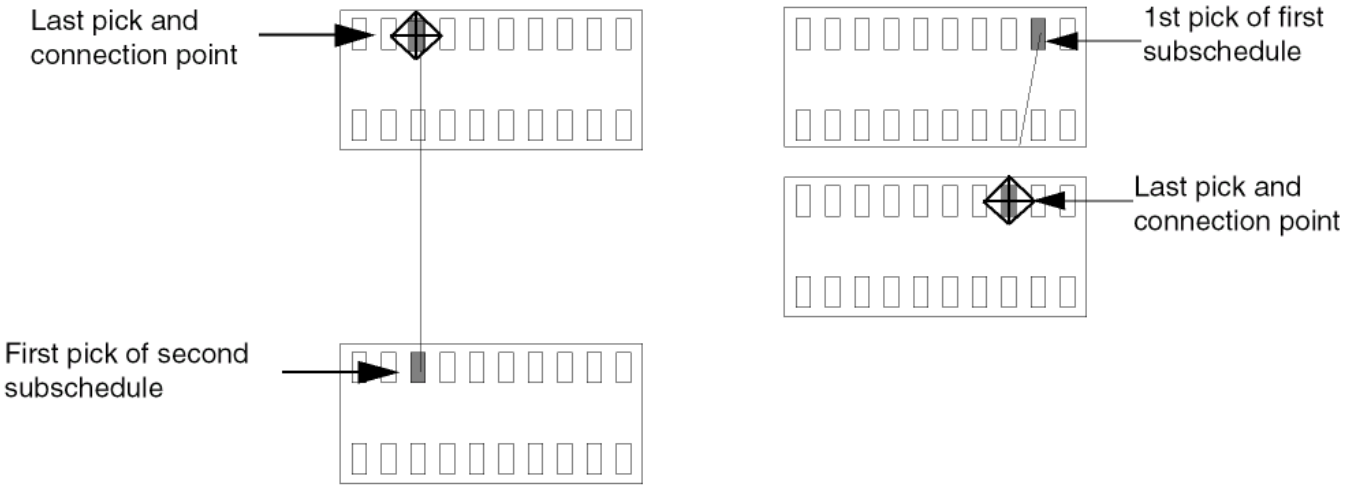
1. *R*ight-click and choose *Delete Subschedule Connection Point* from the pop-up menu.
   The next pin you pick becomes the deleted connect point.

The following examples illustrate creating a subschedule with and without subschedule connect points.

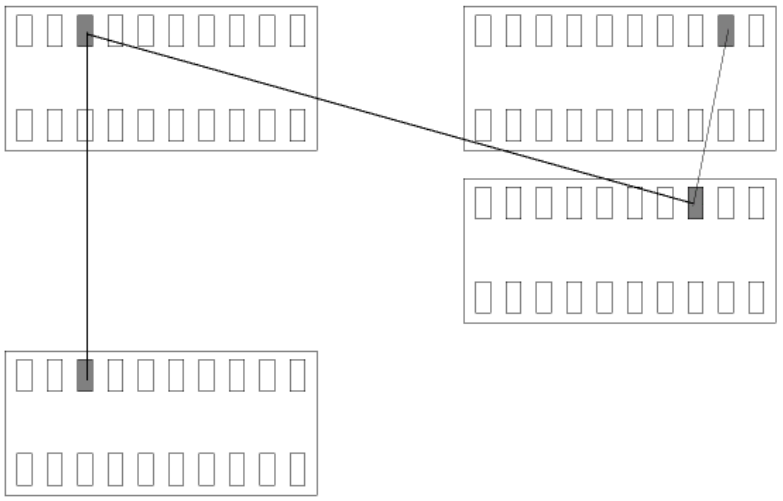
Ratsnest Display While Creating a Subschedule



Subschedule Connection Points



After Subschedule Completion



Related Topics

- net schedule
- Scheduling a Net Interactively

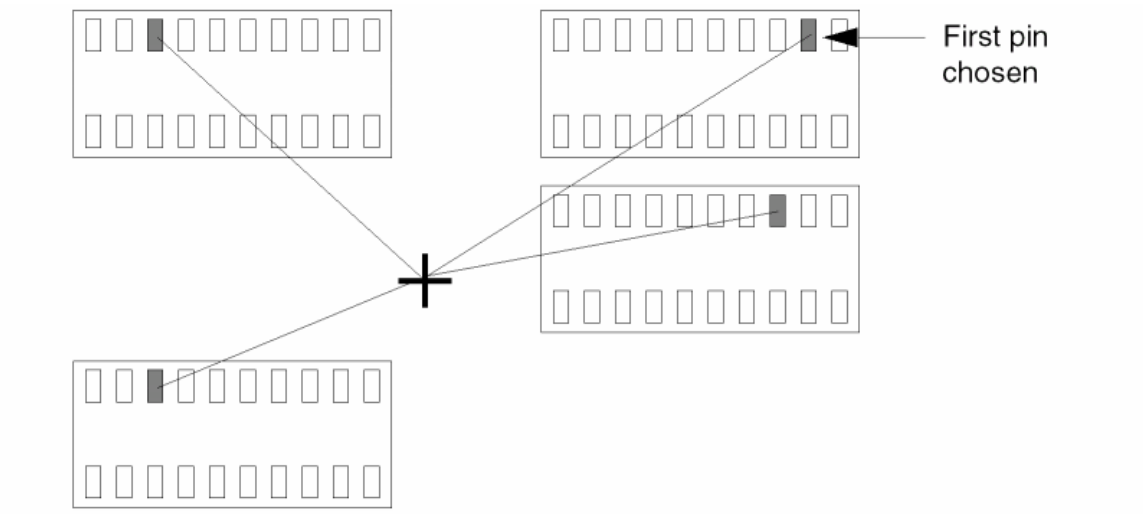
Scheduling a Net Interactively

To schedule a net interactively, do the following:

1. Choose *Logic – Net Schedule*.
   The console window prompt displays the following message:
   ```
   Pick to select a net to schedule.
   ```
2. Click on an object to start scheduling.
   The name of the net appears in the console window prompt.
   The current route order displays when you choose the first object.

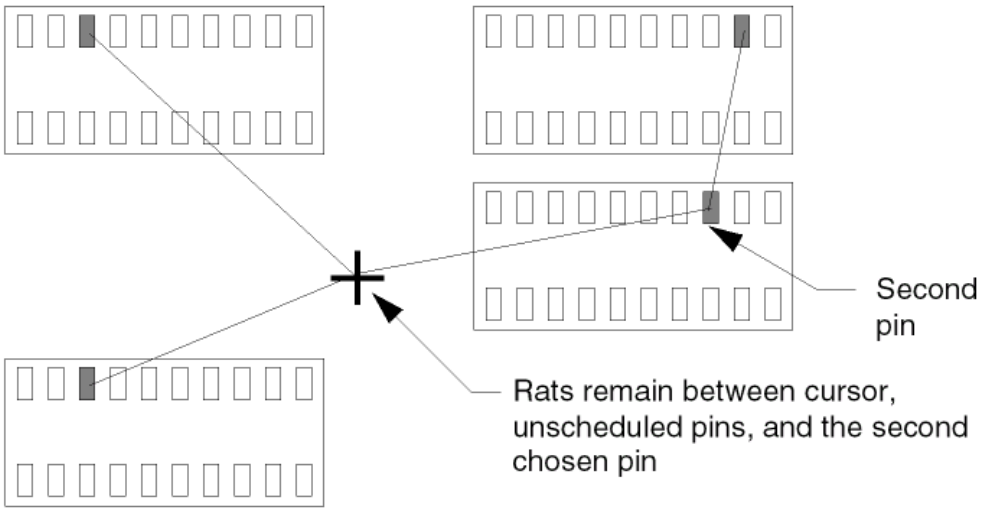
> ⚠ If the ratsnest display is disabled, you will not see the ratsnest connections on the net to schedule unless you click on a pin or Tpoint. You must choose a pin or Tpoint to establish the starting point of the schedule.

Ratsnest Display and Choosing First Pin



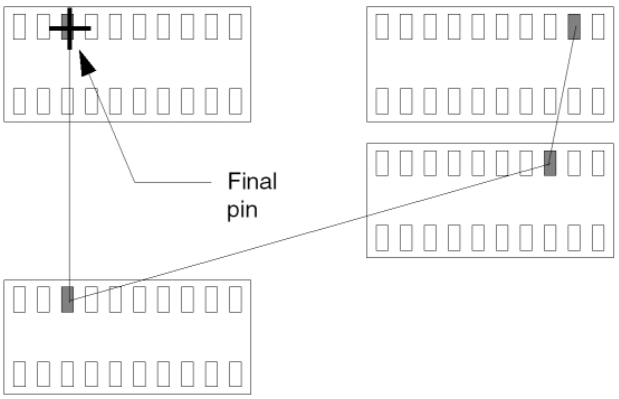
3. Choose the second pin in the net that you want to route.
   A rat appears between the first two pins you chose. Ratsnest lines remain between the cursor, any remaining unscheduled pins in the net, and the last pin you chose.

Choosing Second Pin



4. Choose the remaining pins in the net in the order in which you want them to route.

Choosing Final Pin



5. After choosing the final pin, you must choose *Done* or *Finish* on the pop-up menu. The rescheduled order displays when you finish.

You can partially schedule a net when you want a specific connection order for a portion of the net. Choose pins, multiple series of pins, or Tpoints in a net to specify the order in which to connect to other pins in the net. You can create subschedule connect points on a pin or Tpoint of a subschedule to designate where to join the remaining net connections. If you do not specify any subschedule connect points, any pin within the subschedule is a legal connection point during net routing. Multiple, partial schedules (subschedules) can exist in a net.

Creating a Subschedule with a Connect Point

A subschedule is a subset of pins in a net that have a specific connection order. The connect point on the subschedule determines where subschedules must connect to the other pins in the net. You can create as many subschedules as you want in a net.

1. Schedule the net interactively.
2. When you have connected all the pins in the subschedule, right-click and choose *Create Subschedule* from the pop-up menu.
   By default, the last pin scheduled becomes the connect point to the subschedule and is marked with a diamond.
3. Repeat this procedure to create additional subschedules in the same net.
   or
4. Right-click and choose *Done* from the pop-up menu to save your scheduling choices and exit the command or *Finish* from the pop-up menu to save your scheduling choices and schedule a new net.

Creating Additional Subschedule Connect Points

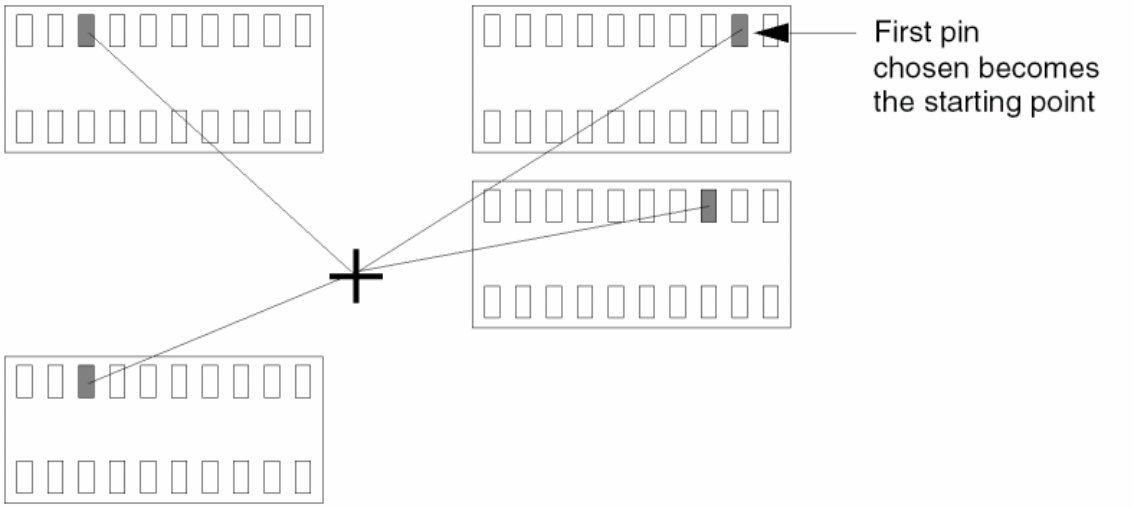
1. Right-click and choose *Add Subschedule Connection Point* from the pop-up menu.
   The next pin you pick becomes another connection point.

Deleting a Subschedule Connect Point

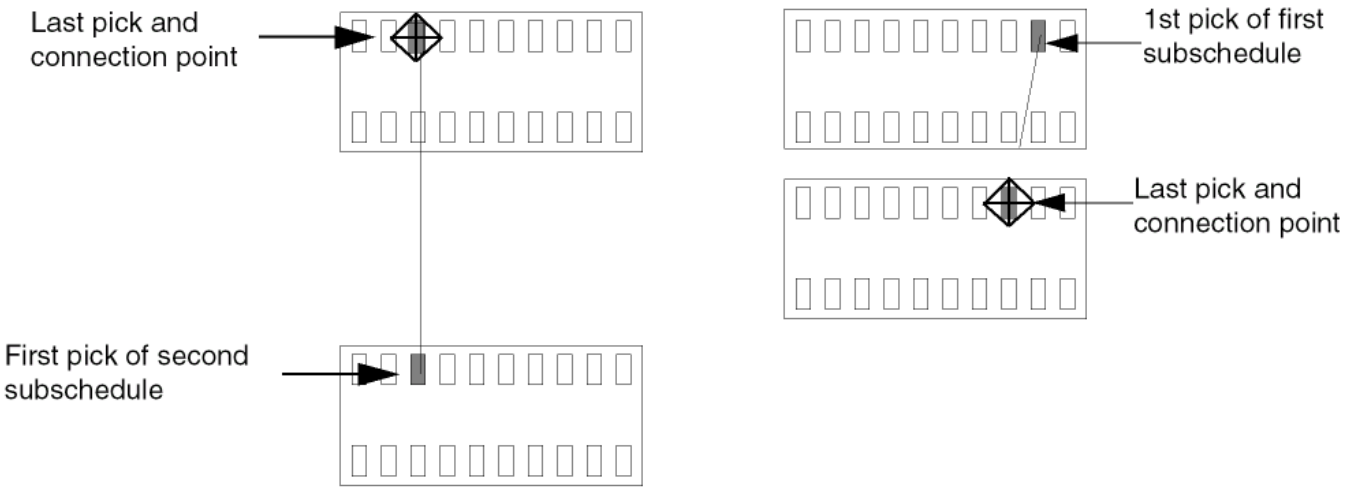
1. *R*ight-click and choose *Delete Subschedule Connection Point* from the pop-up menu.
   The next pin you pick becomes the deleted connect point.

The following examples illustrate creating a subschedule with and without subschedule connect points.

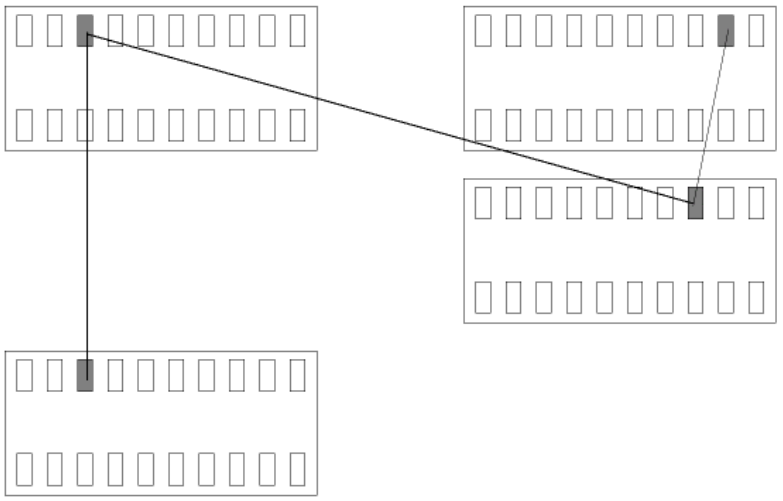
Ratsnest Display While Creating a Subschedule



Subschedule Connection Points



After Subschedule Completion



Related Topics

- net schedule

Scheduling a Net with Tpoints

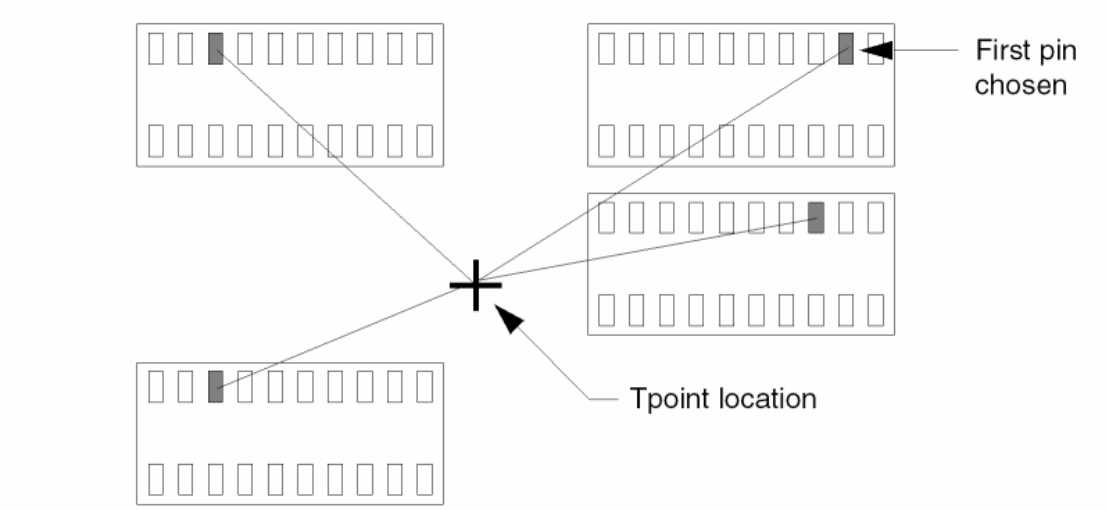
A Tpoint (also called a ratsnest T) is a point in the physical layout of a net that indicates the signal path splits into multiple paths. For illustrations of this task, see "Example of Creating Branches from a Tpoint" below.

1. Run the `net schedule` command.
   The console window prompt displays the following message:
   `Pick to select a net to schedule.`
2. Choose the pin you want to be the source pin for the net you are scheduling.
   The name of the net appears in the console window prompt.
   As you move the cursor away from the pin, a ratsnest displays from the cursor to every unscheduled pin in the net.
3. On the pop-up menu, choose Insert T .
4. Click where you want to specify the Tpoint which is the source for branching to the remaining pins in the net.
   The Tpoint is named automatically. Also, the rat appears between the first pin you chose and the Tpoint. Ratsnest lines remain between the cursor and any remaining pins on the net.
5. For each pin in the net, in the order you want the pins to be routed:
   a. Click the Tpoint.
   b. Click the pin to create a branch from the Tpoint.
6. Choose Done or *Finish* on the pop-up menu.

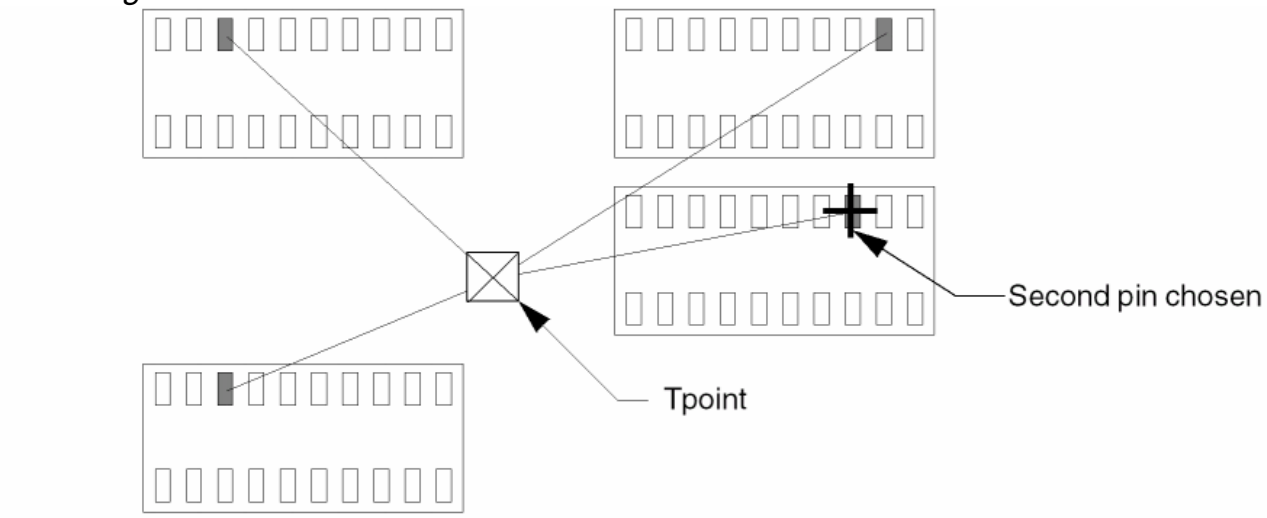
Example of Creating Branches from a Tpoint

The following illustrates how to connect pins to a Tpoint.

Choosing Source Pin and Tpoint Location



Choosing Second Pin



Moving a Tpoint

1. Run `move`.
2. On the Find filter, make sure *Rat Ts* is turned on.
3. Choose the Tpoint, which is indicated by a diamond.
   All the ratsnest lines pass through the Tpoint rubber band as you move your cursor.
4. Click to place the Tpoint in a new location.
5. Choose Done on the pop-up menu.

⚠ Apply the FIXED_T_TOLERANCE property with tolerance value to control the location of a rat T for auto-routing.

You can automatically optimize the location of all Tpoints in your design based on physical criteria. For further details, see the optimize_ts command.

Deleting a Tpoint

You can delete a Tpoint in a number of ways:

- Use the *Unschedule* option of the `net schedule` command, described below.
- For a net imported from a third-party utility, use the netin command to import a netlist that does not include the Tpoint in the $SCHEDULE subsection of the file or that unschedules the net in the $UNSCHEDULE subsection.

Both of these subsections appear in the $NETS section of a netlist.

For additional information, see the *Transferring Logic Design Data user guide in your documentation set.*

Unscheduling a Net

1. Run the `net schedule` command.
2. Click on an object in the net you want to unschedule.
3. Right-click and choose Unschedule Net from the pop-up menu.
4. Choose Done on the pop-up menu.

Unscheduling Nets by Window

1. Run the `net schedule` command.
2. Right-click and choose *Unschedule Net* from the pop-up menu.
3. Using the left mouse button, draw a box around the nets you want to unschedule.
4. Right-click and choose *Done* from the pop-up menu.

Related Topics

- net schedule
- Net Schedule Pop-up Menu
- Scheduling a Net Interactively

net short

The `net short` command shorts two or more nets together by adding the NET_SHORT property on an object.

Available only in the *General edit* and *Etch edit* application modes, this command functions in a pre-selection use model, in which you choose the object first, then right-click and execute the command.

Prior to using this command, enable relevant object types in the *Find* filter. Valid elements are pins, ias, and static shapes.

⚠ The command reports an error for dynamic shapes.

Related Topics

- Shorting Two or More Nets

Net short Pop-up Menu

Oops	Roll-back the selection of objects
Cancel	Cancels `net short` without adding the property
Complete Net Short	Applies NET_SHORT property to the selected object and exits the command

Shorting Two or More Nets

To short two or more nets, do the following:

1. Set application mode as *General edit* or *Etch edit.*
2. Select Pins, Vias, or Shapes in the *Find* filter.
3. Hover your cursor over the selected element.
   The tool highlights the element.
4. Right-click and choose *Net Short* form the pop-up menu or type `net short` in the command window.
   The following message appears in the command window.

   Pick net(s) to be added for net shorting. When finished, select Complete Net Short from the mouse popup menu.
5. Click in the canvas or enter the net name in the command window to select the net for shorting.
6. Repeat the previous step to select more nets.
7. Right-click and choose *Complete Net Short* from the pop-up menu options.
   The command exits and assigns the NET_SHORT property to the selected object. The value of the property is displayed in the command window.

   Property NET_SHORT added to 1 element(s).

   Pin at 11.0000,114.0000 shorted to XSIG010192:CON_5V_SB_TOP_O:5V_SB_O_FIL.

Related Topics

- net short

new

The `new` command sets up a new drawing.

Related Topics

- layout wizard
- New Drawing Configuration Dialog Box
- Creating a New Drawing

New Drawing Dialog Box

Use this dialog box to specify details of a new design.

Access Using:

- Menu path: *File – New*
- Toolbar icon:
  

Drawing Name	Lets you specify a name for your drawing.
Browse	Choose an existing design to use as the basis for a new design from the file browser that appears.
Drawing Type	Lets you choose a drawing type. See the table below for descriptions of the drawing types and the location of additional information on creating them.
Template	Click to choose a template containing default design information as mandated by corporate standards that you can use as the basis for a new design. To specify the search path for template databases, set an environment variable *wizard_template_path* in the User Preferences Editor with value as the CDS_SITE directory location. For example, `wizard_template_path = C:/CDS_SITE/SITE/pcb/new_templates` If a template file named `new_default.<ext>` (for example, `new_default.brd` or `new_default.dra`) exists in this path, this file is used as the starting template if a template database is not explicitly specified. If you choose Board as a *Drawing Type*, the template contains default design (`.brd`) information; if Package, Mechanical, Format, Shape, or Flash symbol, the template contains default symbol (`.dra`) information. The *Template* button is disabled for the Board wizard and the Package Symbol wizard.
OK	Click to create a new drawing if you are using Allegro X PCB Editor or display the New Drawing Configuration dialog box if you are using APD.
Cancel	Closes the New Drawing dialog box without creating a drawing.

Drawing Type	Description	Location of Creation Instructions
Board (PCB Editor) Package/multi-chip (APD)	Creates a design file—either a board file (`.brd`) or a multichip module file (`.mcm`). A design file represents the drawing database where you perform such tasks as component placement and routing. You can create a design file manually or use the Design Wizard or Board Wizard.	The Getting Started with Physical Design user guide in your documentation set explains how to create a design from creating the file to importing logic, defining design rules and layers, placing, and routing to manufacturing.
Board wizard (PCB Editor) Package/multichip wizard (APD)	Provide an easy way for you to prototype a new design. The wizard is designed either to help beginning users create a design, or for experienced users who want a quick way to perform routine setup procedures as a foundation for a more complex design database.	The `layout wizard` command in the *Allegro PCB and Package Physical Layout Command Reference.*
Module (PCB Editor) Module Definition (APD)	Creates a design element that is made up of various physical entities. The tool appends the `.mdd` extension to the file name that you specify.	Working with Modules in *The Placing the Elements* user guide in your documentation set.
Package Symbol Wizard (PCB Editor/APD)	Assists novice users with creating a simple package symbol, or experienced designers who want a quick way to create a base package symbol that they can modify into a more complex symbol.	The Package Symbol Wizard in the *Allegro PCB and Package Physical Layout Command Reference*.
Package Symbol (PCB Editor/APD)	Creates a symbol file. The editor saves these databases as files with the `.dra` extension. This invokes the Symbol Editor, from which you can create various types of symbols listed as follows:	The Defining and Developing Libraries user guide in your documentation set.

	• Package/part Creates a new component symbol such as an IC or a discrete. When you save package/part symbols to the symbol library, the tool appends the `.psm` extension to the file name that you specify.	The Defining and Developing Librariesuser guide in your documentation set.
	• Mechanical Creates a drawing symbol such as a card edge connector or a board/design outline. When you save mechanical symbols to the symbol library, the tool appends the `.bsm` extension to the file name that you specify.	The Defining and Developing Librariesuser guide in your documentation set.
	• Format Creates a drawing symbol such as a legend or a company logo. When you save format symbols to the symbol library, the tool appends the `.osm` extension to the file name that you specify.	The Defining and Developing Librariesuser guide in your documentation set.
	• Shape Creates a drawing symbol such as a special shape for a padstack. When you save mechanical symbols to the symbol library, the tool appends the `.ssm` extension to the file name that you specify.	The Defining and Developing Librariesuser guide in your documentation set.
	• Flash Creates a thermal relief symbol. When you save flash symbols to the symbol library, the tool appends the file name that you specify with the `.fsm` extension. The character limit for flash names is 30. See the Limits section in Chapter 2 of the *Getting Started with Physical Design* user guide in your documentation set.	The Defining and Developing Librariesuser guide in your documentation set.

Related Topics

- layout wizard
- package symbol wizard
- Defining and Developing Libraries
- Creating a New Drawing

New Drawing Configuration Dialog Box

Opening a new design in manual planner design mode or as a module definition displays the New Drawing Configuration dialog box, from which you can choose the package type you want to create: flip-chip (chip-up or chip-down) or wire bond (chip-up or chip-down).

⚠ This dialog box is available only in Allegro X Advanced Package Designer.

For information on overriding the default settings you accept in the New Drawing Configuration dialog box, see the *Getting Started with Physical Design* user guide in your documentation set.

Creating a New Drawing

1. Choose *File – New* to open the New Drawing dialog box.
2. Enter a drawing name.
3. Click *Browse* to choose an existing design to use as the basis for a new design from the file browser that appears, and click OK.
   –or–
   Click *Template* to choose a template containing default design information as the basis for a new design from the library browser that displays, and click OK.
4. Choose the *Drawing Type* as described in the previous table.
   The New Drawing Configuration dialog box appears only if you are using APD. For Allegro X PCB Editor, go to step 4.

   ⚠ For Allegro X Advanced Package Designer, if you enter a file extension for a drawing name that differs from the normal extension of the drawing type you selected, the extension automatically changes to the extension associated with the drawing type. For example, if you enter `newdesign.mcm`, but selected the drawing type *Tile*, the drawing name changes to `newdesign.til`.

5. Choose the package configuration and accept the new drawing default parameters.
   You can also set your drawing parameters manually, as described in *the Getting Started with Physical Design user guide in your documentation set.*
6. Click OK to close the dialog box.
   After defining the drawing options, you are ready to add an outline from the symbols library, or create a new outline. The drawing displays in the Design Window, according to the values in the *Design Parameter Editor* tabs .
7. Choose *File – Save* to save your new settings.
8. Follow the appropriate set of instructions for creating this design as described in the previous table.

Related Topics

- new
- prmed
- save
- Getting Started with Physical Design

next

The `next` command is available on the pop-up menu when an interactive command is active. The *Next* command executes the command selections already made during the current interactive command and loops to the beginning of the command, ready for the selection of another of the same element. For example, if you are adding lines using the *`add line`* command, after you draw one line, you can choose *Next* and draw another line.

noappmode

The `noappmode` command exits from the current application mode and returns to a menu-driven editing mode, or verb-noun use model, in which you choose a command, then the design element.

Access Using:

- Menu path: *Setup – Application Mode – None*

Exiting Application Mode

In application mode, you first select a design element and then choose an option from the pop-up menu. If you are in one of the application modes and want to come out of the current application mode, do the following:

1. Choose *Setup – Application Mode – None*
   The status bar shows *None*.

Related Topics

- Application Modes, the Pre-Select and the Post-Select Use Models

nographic

The `nographic` command runs the tool in a non-graphic mode. On UNIX it requires an X display. This switch can also be typed `-nograph`.