U Commands

Product Version 23.1
September 2023

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

Contents

1

U Commands

uiresources	unalias	unassign
unbutton	undo	unfix
unlock symbol	unmark fanout	unmiter_by_pick
unplace component	unrats all	unrats component
unrats net	unrats outside partition	unset
update_rf_drcs	update codesign die	update codesign pkg
update package	update pcell_symbols	uprev
use altsym		

uiresources

Internal command.

unalias

The `unalias` command lets you delete aliases for commands and function keys.

```
unalias <
user-defined name
> <
command to execute
>
```

```
unalias <
Fkey
>
<command to execute
>
```

command to execute	Specifies the command from which you want to remove the alias. Multiple commands can be closed within quotation marks ("") and separated by a semicolon (;).
user–defined name	Specifies the alias name that you want to be remove as a shortcut for a command.
Fkey	Specifies the function key to be remove as a shortcut for a command.

Examples

This section provides examples of the `unalias` command.

- `unalias db dbdoctor`
  The above example deletes the alias `db` for the command `dbdoctor`
- `unalias pecl "class package geometry; drawedit -menuload place"`
  The above example deletes one alias `pecl` for multiple commands `class package geometry` and `drawedit -menuload place`
- `unalias F2 shell`
  The above example deletes the function key `F2` for the command `shell`

Related Topics

- alias

unassign

Internal command.

unbutton

The `unbutton` command removes the association of the mouse button and the command.

```
unbutton|[modifier]|[wheel]|[wheel_up]|[wheel_down]|[action to execute]
```

modifier	removes the association of buttons with or without `Shift` and `Control` keys or a combination of both. Modifiers are S (`Shift` key), C (`Control` key), and SC (`Shift` and `Control`) and are case insensitive.
wheel	Specifies upward or downward mouse wheel movement if the `wheel_up` and `wheel_down` arguments are not specified.
wheel_up	Specifies an upward mouse wheel movement. Defining this argument suppresses the upward mouse movement of the `wheel` argument.
wheel_down	Specifies a downward mouse wheel movement. Defining this argument suppresses the downward mouse movement of the `wheel` argument.
action to execute	Specifies the action to execute when the mouse rolls up or down.

Example

```
unbutton Cwheel_down zoom in
```

The above command removes the zoom in action when you press the Control key and roll the mouse wheel down.

Related Topics

- button

undo

The `undo` command reverses the results of the most recent action after it is complete or those of a series of actions when you repeat this command. Undo-enabled commands are used to edit physical database entities, such as lines, vias, shapes, voids, pins, components, and so on.

When you click the down arrow key on the *Undo* toolbar icon, a history of commands used in the current session displays, which lists the most recent actions that can be reversed. The *Undo* toolbar icon is grayed out when no commands are active. The following parameters cannot be used with the `undo` command:

- Database parameters, such as size, units, and accuracy
- Application parameters such as those related to artwork
- Form states such as the last selected directory in a file browser
- Display-related parameters, such as use of a cross-hair or an infinite cursor

Access using:

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

Undoing the Most Recent Actions

To undo the most recent actions:

- Choose *Edit – Undo*.
  Alternatively, you can also click the Undo button on the toolbar.
  The most recent action is reversed.

You can repeat step one as many times as required to undo other actions in their reverse order of execution.

Related Topics

- Redo

Configuring Undo Actions

You can change the number of commands that appear in the history and the amount of memory used to store it. To configure the number of commands in undo history:

1. Choose *Setup – User Preferences*.
2. Choose *Undo* from the *Categories* section
3. Specify the values in the *undo_depth* and *max_undo_memory* fields.

When you exceed the number of commands specified in *undo_depth*, the tool deletes commands from the end of the history. The higher the *undo_depth* value you set, the more memory the system uses.

Related Topics

- Redo

unfix

The `unfix` command removes the FIXED property from chosen elements allowing unrestricted interactive and automatic edits. The objects can then be moved or deleted; the automatic router can rip up connections in the net; and glossing on the net may occur.

Valid elements are:

- Components
- Symbols
- Nets
- Pins
- Vias
- Clines
- Lines
- Shapes

Access using:

- Menu Path: Right-click - *Unfix*
- Toolbar Icon:

Related Topics

- fix
- FIXED

Removing FIXED Property from Objects

You can remove the FIXED property from an object to allow altering by any subsequent autorouter operation. This command functions in the pre-selection use model, in which you select an object with the FIXED property first, then right-click and execute the command. Objects not allowed to be used with the command generate a warning and are ignored.

1. Hover your cursor over an element or draw a window around the objects from which you want to remove the FIXED property and allow them to be modified.
   The tool highlights the element and a datatip identifies its name.
2. Right-click and choose *Unfix* from the pop-up menu to automatically launch the command. Alternatively, you can also type `unfix` in the Command window. A message similar to the following appears in the console window for each selected element from which the tool removed the FIXED property to allow modification:

   Property FIXED removed from element <variable>: <variable>.

You can also unfix multiple objects in the design. To do this, draw a window around the objects you want to unfix, right-click and choose *Unfix All*.

Related Topics

- fix
- FIXED

unlock symbol

The `unlock symbol` command unlocks the selected instances of a symbol by removing the `LOCKED` property from the instances.

Select a symbol instance and right-click to access the command.

> ⚠ Unlocking a symbol instance for editing might lead to potential issues, such as manufacturing and connectivity errors in the design, problems with collision detection in DFM/DFA, and more. To prevent any unwanted or accidental edits, you can unlock the symbol instance, make the required changes, and lock it again immediately.

Related Topics

- lock symbol

unmark fanout

The `unmark fanout` command disassociates clines and vias from their respective component symbol instances, when a design containing fanouts, created with Specctra or third-party tools, is read into the board or the layout editor.

Unmarked fanouts comprise clines and vias connected to a pin, but are not associated with the component symbol instance. When unmarking, only existing fanouts can be selected, and all unmarked clines and vias appear dimmed. You can identify fanouts by using the mark fanout command.

Valid objects are:

- Pins
- Clines
- Vias
- wires
- Wire bond fingers (APD)

> ⚠ Wire bond fingers attached to a design are treated as marked fanouts and are associated with components.

Access Using:

- Menu Path:
  - In PCB Editor: *Route – Convert Fanout – Unmark*
  - In APD: *Route – Via Structure – Convert Fanout – Unmark*

Disassociating Clines and Vias with Component Symbol Instances

This command functions in both the noun-verb (pre-selection) mode and verb-noun mode. In the pre-selection use model, you choose an element first, then choose the command from menu.

1. Choose *Setup – Application Mode – Etch Edit* to access the etchedit application mode. Alternatively, you can also type `unmark fanout` in the Command window.
2. Hover your cursor over an element or draw a window around the elements to unmark as fanouts. The tool highlights the element and a datatip identifies its name.
3. Right-click and choose *Unmark* from the pop-up menu to automatically launch the command and unmark the fanouts.

Related Topics

- unmark fanout

unmiter_by_pick

The `unmiter_by_pick` command lets you remove 45° wire corners and change them to 90° corners.

Access using:

- Menu Path: *Route – UnMiter by Pick*

Removing 45 Degree Wire Corners

To remove 45° wire corners:

1. Chose *Route – UnMiter by Pick*
   Alternatively, you can also type `unmiter_by_pick` in the Command window.
2. Select a net or a group of nets.
   The 45° wire corners are removed.
3. Right-click and choose *Done*.

unplace component

The `unplace component` command returns a placed symbol to the *Placement List* in the *Placement* dialog box. The symbol is not deleted and remains in the database. It is available to be placed again.

Valid objects:

Symbols

Unplacing Components

This command functions in the pre-selection use model, in which you choose an object first, then right-click and execute the command from the pop-up menu. Objects that are not allowed to be used with the command generate a warning and are ignored.

1. Hover your cursor over a symbol or select a group of symbols. The tool highlights the object and a datatip identifies its name.
2. Right-click and choose *Unplace component* from the pop-up menu to automatically launch the command.
   Alternatively, you can also type `unplace component` in the command window.
   The symbol appears in the *Placement List* in the Placement dialog box and not in the design.

Related Topics

- *Placement*

unrats all

The `unrats all` command hides all ratsnest lines in your design.

Access using:

- Menu Path: *Display – Blank Rats – All*
- Toolbar Icon:

Hiding All Ratsnest Lines

1. Choose *Display – Blank Rats – All*.
   Alternatively, you can also run `unrats all` in the command window. All ratsnest lines in the design disappear.
2. Run *View – Refresh* to clean up the appearance of your design.

unrats component

The `unrats component` command hides visible ratsnest lines to pins on an individual component or a group of components in a design. Click to select the components or select the appropriate symbol name or symbol list from the Find by Name section of the Find filter.

Access using:

- Menu Path: *Display – Blank Rats – Component*

Hiding Ratsnest Lines Connected to Component Pins

1. Choose *Display – Blank Rats – Component*.
   Alternatively, you can also type `unrats component` in the command window.
2. Click the components on the canvas.
   All ratsnest lines to pins on the components that you select disappear.
   Optionally, you can extend your selection by right-clicking the canvas and choosing *Refdes List* or *Refdes Name* from the pop-up menu.

unrats net

The `unrats net` command hides visible ratsnest lines to pins on an individual net or a group of nets in a design. To select the nets to be invisible, select the pins on the appropriate net or select the appropriate net name or net list from the Find by Name section of the Find filter.

Access using:

- Menu Path: *Display – Blank Rats – Net*

Hiding Ratsnest Lines to Pins on Nets

To hide visible ratsnest lines to pins on an individual net or a group of nets, follow these steps:

1. Choose *Display – Blank Rats – Net*.
   Alternatively, you can also type `unrats net` in the command window.
2. Click the nets on the canvas.
   All ratsnest lines to pins on the nets that you select are removed.
   Optionally, you can extend your selection by Net by right-clicking the canvas and choosing *Net List* or *Net Name* from the pop-up menu.

unrats outside partition

The `unrats outside partition` command hides all ratsnest lines outside the active partition when working with the Design Partition feature.

> ⚠ This command is available only with the Design Partition option.

Access using:

- Menu Path: *Display – Blank Rats – Outside Partition*

Hiding Ratsnest Lines Outside Partition

1. Open the partitioned design (`.dpf` or `.dps`).
   Ensure that the Design Partition feature running enabled.
2. Choose *Display – Blank Rats – Outside Partition*.
   All ratsnest lines in the design disappear.
3. Run *View – Refresh* to clean up the appearance of your design.

Related Topics

- unrats net
- unrats all
- unplace component

unset

The `unset` command returns an environment variable setting to its previous value. You can also unset environment variables interactively in the User Preferences Editor.

```
unset <variable_name>
```

`variable_name`	Specify the name of the environment variable to unset.

Example

```
unset pcb_cursor
```

The above command unsets the `pcb_cursor` environment variable.

Related Topics

- enved

update_rf_drcs

⚠ This is a depreciated command.

The `update_rf_drcs` command adds DRC marker on user-schedule violations of a net. You need to explicitly run this command because this is a user-defined DRC in SiP Layout, and is not included in the default DRC check.

The `update_rf_drcs` command is used to mark topology violations that are defined using the `USER_SCHEDULE` property on a net. On running the front-to-back flow, the information about the net topology, defined using the `USER_SCHEDULE` property, is transferred to the physical layout for the SiP. After routing, running the `update_rf_drcs` command places the DRC markers on the pins where the topology defined using `USER_SCHEDULE` property is violated.

Syntax

```
update_rf_drcs
```

update codesign die

Not documented for this release.

update codesign pkg

Not documented for this release.

update package

Internal command.

update pcell_symbols

The update pcell_symbols command modifies the pcell symbol.

⚠ This is a depreciated command.

```
update pcell_symbols (ALL/SELECTED) (PROP_MODIFIED/IRREGULAR_EDITED/ALL)
```

Parameter	Description
`ALL|SELECTED`	• ALL: Updates all the symbols • SELECTED: Updates the selected symbols.
`PROP_MODIFIED|IRREGULAR_EDITED|ALL`	• PROP_MOIDIFIED: Updates the footprint with the modified property value. • IRREGULAR_EDITED: Updates the footprint with the property values that have not been changed, effectively undoing the irregular edit. • ALL: Regenerates the footprint as per the current property values.

Example

```
update_pcell_symbols PROP_MODIFIED
```

⚠ Though SiP RF Architect provides support for modifying shapes by changing the shapes by hand directly in the layout, this method is not recommended.

uprev

The `uprev` batch command takes a design database from its current version to the latest version of the tool.

```
uprev

Layout, drawing, or symbol file name (*.brd):

Output layout, drawing, or symbol file name (*.brd):
```

input_file	The name of the database you want to uprev. The default is `brd`.
output_file	The name of the database after the uprev. Giving an output name that is different than the input name prevents the input database from being destroyed.
-version	Prints the version.

Updating a Design Database

1. Run uprev from your operating system command prompt.
   If you type the command name without arguments, you are prompted for the input and output file names.
2. Enter the appropriate file name and press Return/Enter.
   The design is uprevved to the latest tool version.

The `uprev` command will produce a log file output_db.log that reports information and any error messages that have been reported.

use altsym

The `use altsym` lets you choose an alternate symbol for one or all symbols in a design.

Available only in the *Placement Edit* application mode, this command functions in a pre-selection use model, in which you choose a symbol or a group of symbols first, then right-click to display a list of valid alternate symbols.

Valid objects are:

- Components (in this context, refers to the database element SYMBOL_INSTANCE)
  You can choose a particular alternate symbol to use in place of a component, globally or by selection, if you previously attached the ALT_SYMBOLS property type to the components using the schematic-capture tools, such as Allegro Design Entry HDL or Allegro Design Entry CIS. The ALT_SYMBOLS property defines an alternate package symbol that can be substituted for the primary package symbol. When using a third-party schematic, in the device file, assign the ALT_SYMBOLS property to components by specifying a PACKAGEPROP property record in the device file.
  If any alternate symbols are defined for one or several selected symbol instances of the same type, when you right-click, the following popup menus display, each of the menus expands into the list of available symbols with which you can replace the original symbols with.
- **Selected Instances**: Displays a list of alternate symbols. The chosen alternate symbol replaces the currently selected symbol instances. If the symbol definition for the alternate symbol cannot be found, the original symbol instance remains intact.
- **All instances**: the chosen alternate symbol replaces multiple symbols of the same type as the preselected symbol instances. Any symbol instances which cannot be replaced with the alternate symbol remain intact.

If any alternate symbols are defined for one or several selected symbol instances of different types, on right-clicking, each symbol name displays. Each of those names then expands into the available alternate symbols.

Replacing a Single or Multiple Symbol Instances of the Same Type

To replace a single or multiple instances of a symbol of same type, follow these steps:

1. Choose *Setup – Application Mode – Placement Edit* to access the placement edit application mode.
   Alternatively, you can also type `use altsym` in the command window.
2. Click one or more symbols you want to replace.
3. Do one of the following:
   - Right-click and choose *Alternate Symbol – Selected Instances* to replace a single instance of a symbol.
     A list of all valid alternate symbols appears.
   - Right-click and choose *Alternate Symbol – All Instances* to replace multipe instances of a symbol.
     A list of all component types in the selection set appears, along with a list of all the valid alternate symbols.
4. Choose an alternate symbol from the list.
   A confirmation dialog box appears with select options to preserve or rip up etch.
5. Click *Yes* to rip up etch or *No* to preserve it.
   The command console window displays a message similar to the following:

   Replaced <*number*> instance(s) of symbol <*name*> with alternate symbol <*name*>

   The chosen alternate symbol replaces the currently selected symbol instances.

Replacing Multiple Symbol Instances of Different Types

If you want to replace multiple instances of a symbol that are of different types, follow these steps:

1. Choose *Setup – Application Mode – Placement Edit* to access the placement application mode.
   Alternatively, you can also type `use altsym` in the command window.
2. Select a group of symbols to be replaced with alternates.
3. Right-click to display each symbol name, which expands into the available alternate symbols from which you choose a replacement.
   A confirmation dialog box appears where you specify whether to preserve or rip up etch/conductor.
4. Click *Yes* to rip up etch/conductor or *No* to preserve it.
   The command console window displays the following message:

   Replaced <*number*> instance(s) of symbol <*name*> with alternate symbol <*name*>