

Adding and Appending Allegro Menu Items Through SKILL

Rapid Adoption Kit (RAK)

Product Version: SPB 17.4
February, 2022

Copyright Statement

© 2022 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence and the Cadence logo are registered trademarks of Cadence Design Systems, Inc. All others are the property of their respective holders.

Contents

Purpose	4
Audience.....	4
Overview.....	4
Terms	4
Directory Structure for Allegro Menu SKILL RAK	4
Allegro SKILL APIs for Menus	5
Steps to add menu items to Allegro menu bar	5
Menu customization methods	5
axIUIMenuFind	6
axIUIMenuInsert	7
Adding a Menu to the Allegro Toolbar	8
Adding a Menu Item to an Existing Menu	11
Setup	12
Summary	14
Support	14
Feedback	14

Purpose

This RAK is intended to provide a step-by-step methodology overview and sample files and folders for menu modification to the Allegro Layout suite of tools. This document covers the following methods to customize Allegro menu items:

- Adding a menu to the Allegro toolbar
- Adding a menu item to an existing menu

Audience

Anyone interested in customizing Allegro Layout editor menus, that is, PCB Editor, SiP Layout, or APD+ using Allegro SKILL.

Overview

This RAK is based upon the usage of `CDS_SITE`. The folder structure and contents are automatically understood by Allegro. More information on `CDS_SITE` and its usage can be found in the user manuals and additional application notes.

Terms

- Menu bar: Menu items seen at the top of a window
- Menu item: A menu line; may either be a command, separator, or submenu
- Separator: A horizontal line drawn to visually separate menu items
- Submenu: A pulldown (from the menu bar) or a pull-right (from another submenu)

Directory Structure for Allegro Menu SKILL RAK

- `cds_site_skill_menu` (master `CDS_SITE` directory)
 - `pcb` (the folder automatically recognized by Allegro using `cds_site`)
 - `pcb/skill` (the folder automatically recognized by Allegro containing SKILL files and `allegro.ilinit`)

The RAK database and references can be found in the 'Attachments' and 'Related Solutions' sections below the PDF. This RAK pdf can be searched with the document title on <https://support.cadence.com>.

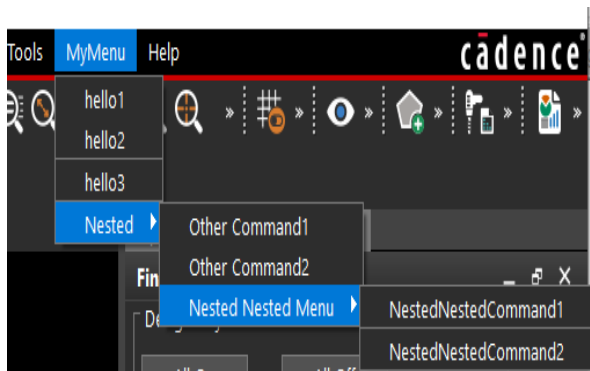
Allegro SKILL APIs for Menus

Steps to add menu items to Allegro menu bar

Steps used to add menu items to the Allegro menu bar are given below:

1. Finding the menu ID by location or name by using `axlUIMenuFind`
2. Building a menu or nested menus, which may have submenus, separators and action menus using `axlUIMenuInsert`. Action menu runs an Allegro command when the menu is selected.

Note: If you want to have your own SKILL function to be called from the menu, you should register the Allegro SKILL function as Allegro command by using `axlCommandRegister`.



```
q = axlUIMenuFind( nil -1)
axlUIMenuInsert(q 'popup "MyMenu")
axlUIMenuInsert(nil "hello1" "echo hello 1" )
axlUIMenuInsert(nil "hello2" "echo hello 2")
axlUIMenuInsert(nil 'separator )
axlUIMenuInsert (nil "hello3" "echo hello 3")
axlUIMenuInsert (nil 'separator )
z = axlUIMenuInsert(nil 'popup "Nested")
```

Menu customization methods

It is recommended to create menu items with the menu trigger. For development purposes, you can create menu items outside of the menu trigger. Various methods of adding menus are given below:

- Provide your own customization menu via `CDS_SITE`.
- Overload your menu customizations on Cadence menus by using the `axlUIMenuRegister` SKILL. This lets you register menu items to be loaded

when Allegro loads a new menu. It is a combination of `axlUIMenuFind` and `axlUIMenuInsert`.

- Register menu Triggers notification through `axlTriggerSet`. `axlTriggerSet` allows an application to register interest in events that occur in Allegro PCB Editor. Typical event used is `'menu`, targeted at application code to modify the menu.

Example: `axlTriggerSet('menu 'MyCustomMenu)`

Other menu commands are given below. Refer to *Allegro SKILL Manual* for details.

- `axlUIMenuDelete`: This deletes a single menu item or submenu based upon what is the current find menu item.
- `axlUIMenuChange`: This changes one or more parameters of an existing menu item.
- `axlUIMenuDump`: This dumps the current menu of the main window to the input argument `menuFile`.
- `axlUIMenuLoad`: This loads the main window menu from `t_menuFile`.

See `<Cadence Installation>/share/pcb/examples/skill/ui/menu.il` for examples on Allegro SKILL APIs for menu.

axlUIMenuFind

`axlUIMenuFind` finds a menu item by location or command.

- **Find by name** finds menu item by command name. This method cannot find menu bar items such as File. When finding by name, you should pass `nil` as the first argument.
- **Find by location** identifies a menu item of the menu bar (`menuId = nil`) or submenu.
 - The location is 0-based. The 0 location is the left or topmost menu item. (Typically, this is the File menu item on the menu bar).
 - A negative number may be used to specify a menu counting from the right side, with `-1` indicating the menu furthest to the left or bottom.

```
axlUIMenuFind(
```

```
x_menuId/nil  
t_cmdName/x_location  
[g_menuOption]  
)
```

`x_menuId` is the return of previous call or nil to search from the menu root.

`x_location` finds item by location.

`t_cmdName` finds item by command name. This may not be just a command but is really a command line. For example, if the command is registered as "echo hello", you must find by "echo hello" and not "echo".

`g_menuOption` makes permitted values as top or bottom. If used with find by command, it returns the top or bottom of the menu where the command exists. Bottom option also indicates to `axlUIMenuInsert` that a new menu item should be appended to the end of the menu.

If used with find by location and the item is a submenu, it returns the top or bottom of that submenu.

axlUIMenuInsert

`axlUIMenuInsert` is the command to insert the menu item that calls Allegro command, separator, submenu to an existing menu. Commonly used ways of using `axlUIMenuInsert` are shown in the following table:

Command that calls Allegro command	Command to add separator	Command to add submenu
<pre>axlUIMenuInsert(x_menuId t_display t_command)</pre>	<pre>axlUIMenuInsert(x_menuId 'separator)</pre>	<pre>axlUIMenuInsert(x_menuId 'popup t_display)</pre>

Adding and Appending Allegro Menu Items Through SKILL: RAK

Let's understand the input parameters of the function `axlUIMenuInsert`

`x_menuId` menu id can be obtained from `axlUIMenuFind` or by creating a submenu via this API. If `nil` the current menu on the menu stack is used/

`t_display` shows text in the menu. Possible values are as follows:

`separator` adds a separator (horizontal line).

`popup` creates a new submenu.

`t_command` is the command to run when the menu is clicked. ~~this is ignored for a 'separator.~~

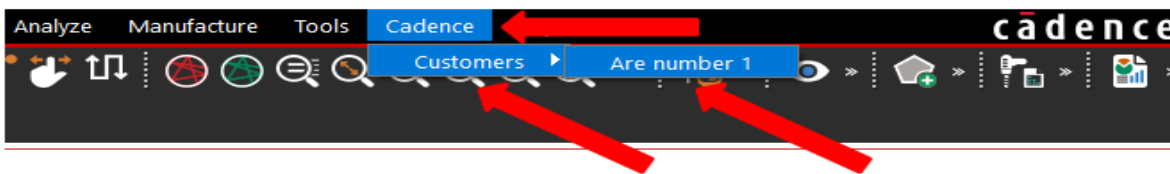
`t_display` This is the display string for the 'popup option.

'end pops the menu stack if creating a menu tree.

`ll_items` is a list of `t_display/t_command` value pairs that instruct this interface to add multiple menu items and submenus in a single call. Both 'separator and 'end options do not have to be a list.

Adding a Menu to the Allegro Toolbar

This example will show how to add a menu named **Cadence** to the existing toolbar. The menu item, **Customers**, will be added to the **Cadence** menu with a submenu item **Are number 1**, which executes the `echo oneTeam` command. Here is a screen capture of the intent.



This is being done using the `cadence.il` SKILL file located at `cds_site_skill_menu/pcb/skill`.

The user-defined name of the function (`CadenceCustomMenu`) needs to be defined at the beginning and end.

Adding and Appending Allegro Menu Items Through SKILL: RAK

```
defun( CadenceCustomMenu (t_menu)
  axlMenuItemFind(nil -1);
  axlMenuItemInsert(nil 'popup "Cadence");
  axlMenuItemInsert(nil 'popup "Customers")
  axlMenuItemInsert(nil "Are number 1" "echo oneTeam")
  axlMenuItemInsert(nil 'end)
  axlMenuItemInsert(nil 'end)
);//
axlTriggerSet('menu 'CadenceCustomMenu);
```

The highlighted two lines, `axlMenuItemFind` and `axlMenuItemInsert`, add the toolbar menu item, Cadence. The preceding word `popup` before the Cadence menu item indicates that there will be a submenu item.

```
defun( CadenceCustomMenu (t_menu)
  axlMenuItemFind(nil -1);
  axlMenuItemInsert(nil 'popup "Cadence");
  axlMenuItemInsert(nil 'popup "Customers")
  axlMenuItemInsert(nil "Are number 1" "echo oneTeam")
  axlMenuItemInsert(nil 'end)
  axlMenuItemInsert(nil 'end)
);//
axlTriggerSet('menu 'CadenceCustomMenu);
```

The next line adds the submenu item, Customers, to the Cadence menu item. Notice that it is preceded by the word `popup` to indicate that there is an additional submenu item being created.

```
defun( CadenceCustomMenu (t_menu)
  axlMenuItemFind(nil -1);
  axlMenuItemInsert(nil 'popup "Cadence");
  axlMenuItemInsert(nil 'popup "Customers")
  axlMenuItemInsert(nil "Are number 1" "echo oneTeam")
  axlMenuItemInsert(nil 'end)
  axlMenuItemInsert(nil 'end)
);//
axlTriggerSet('menu 'CadenceCustomMenu);
```

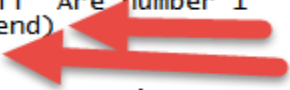
The next line adds the Are number 1 menu item associated with the `echo oneTeam` command.

```
defun( CadenceCustomMenu (t_menu)
  axlMenuItemFind(nil -1);
  axlMenuItemInsert(nil 'popup "Cadence");
  axlMenuItemInsert(nil 'popup "Customers")
  axlMenuItemInsert(nil "Are number 1" "echo oneTeam")
  axlMenuItemInsert(nil 'end)
  axlMenuItemInsert(nil 'end)
);//
axlTriggerSet('menu 'CadenceCustomMenu);
```

The next two lines end the popup menu items, Cadence and Customers.

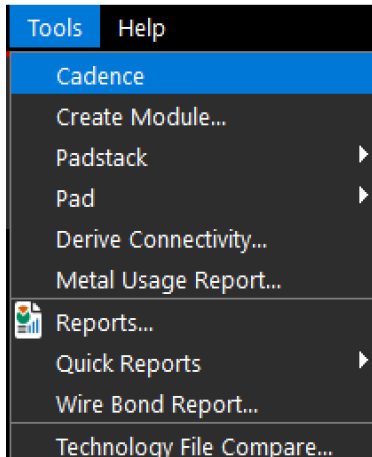
Adding and Appending Allegro Menu Items Through SKILL: RAK

```
defun( CadenceCustomMenu (t_menu)
  axlUIMenuFind(nil -1);
  axlUIMenuInsert(nil 'popup "Cadence");
  axlUIMenuInsert(nil 'popup "Customers");
  axlUIMenuInsert(nil "Are number 1" "echo oneTeam")
  axlUIMenuInsert(nil 'end)
  axlUIMenuInsert(nil 'end)
);//
axlTriggerSet('menu 'CadenceCustomMenu);
```



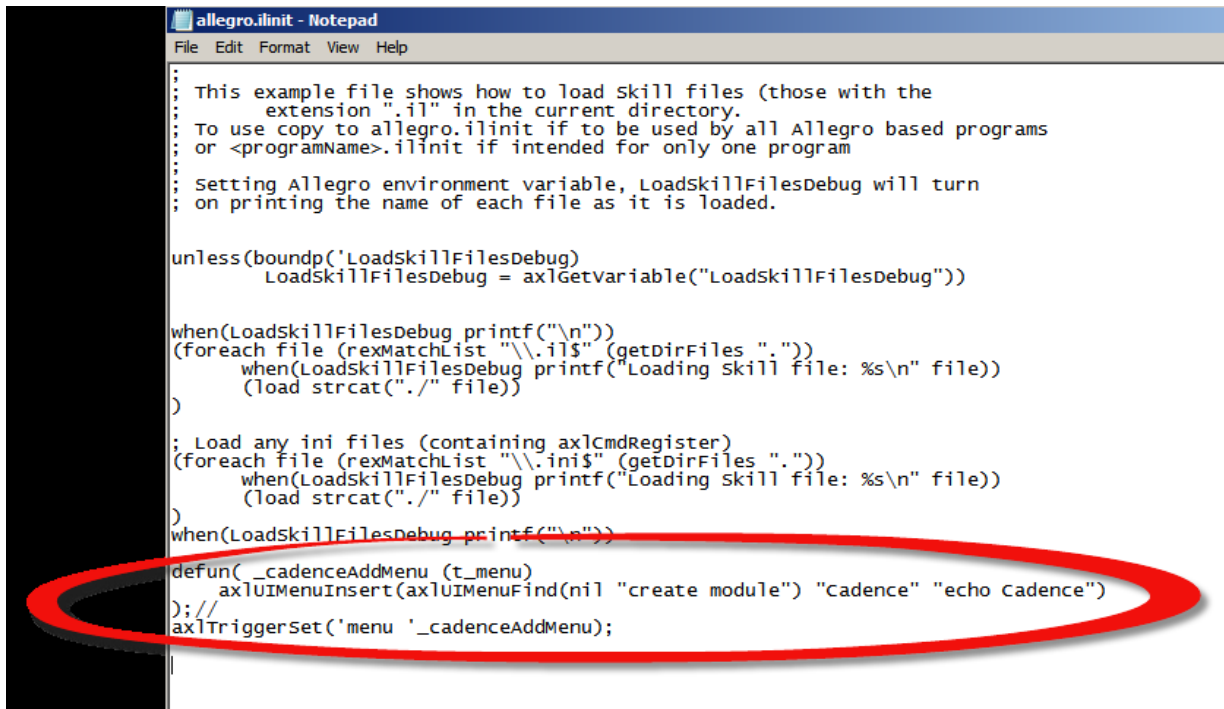
Adding a Menu Item to an Existing Menu

This example will add the **Cadence** menu item to the existing menu item, **Tools**, on the toolbar. Here is a screen capture of the intent.



This is done by modifying the `allegro.ilinit` file to add the **Cadence** menu item. The `allegro.ilinit` file is located at `cds_site_skill_menu/pcb/skill`.

The addition of the **Cadence** menu item, its placement, and the command being executed by the **Cadence** menu item are accomplished with the highlighted portion of the `allegro.ilinit` file in the following image:

A screenshot of a Notepad window titled 'allegro.ilinit - Notepad'. The window shows the contents of the allegro.ilinit file. A red circle highlights the following code block:

```
defun(_cadenceAddMenu (t_menu)
  axlMenuItemInsert(axlMenuItemFind(nil "create module") "cadence" "echo cadence")
);//
axlTriggerSet('menu '_cadenceAddMenu);
```

Adding and Appending Allegro Menu Items Through SKILL: RAK

The user-defined name of the function (`cadenceAddMenu`) needs to be defined at the beginning and end.

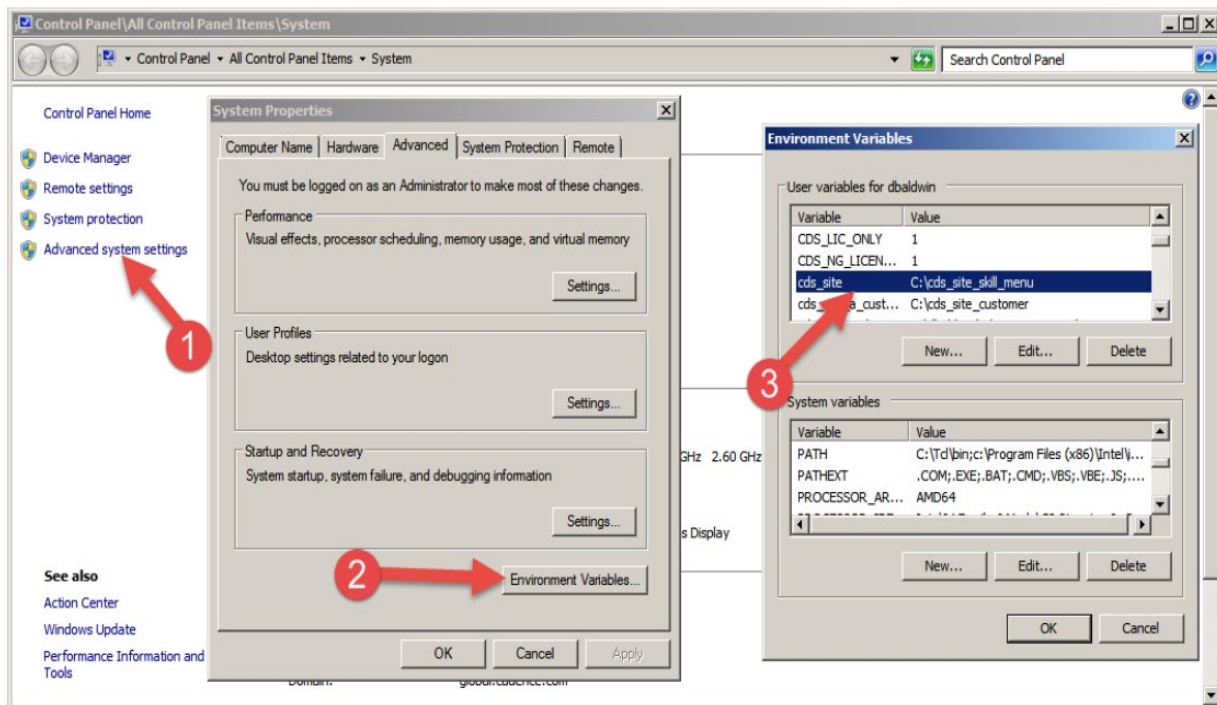
```
defun( _cadenceAddMenu (t_menu)
    axlUIMenuInsert(axlUIMenuFind(nil "create module") "Cadence" "echo cadence")
);//
axlTriggerSet('menu '_cadenceAddMenu);
```

The highlighted line in the following image is to find the `create module` command and insert a menu item, **Cadence**, above it. The **Create Module** menu item (associated with the `create module` command) is located in the **Tools** menu. The **Cadence** menu item is followed by the registered command when selected, which is `echo Cadence` in this case.

```
defun( _cadenceAddMenu (t_menu)
    axlUIMenuInsert(axlUIMenuFind(nil "create module") "Cadence" "echo cadence")
);//
axlTriggerSet('menu '_cadenceAddMenu);
```

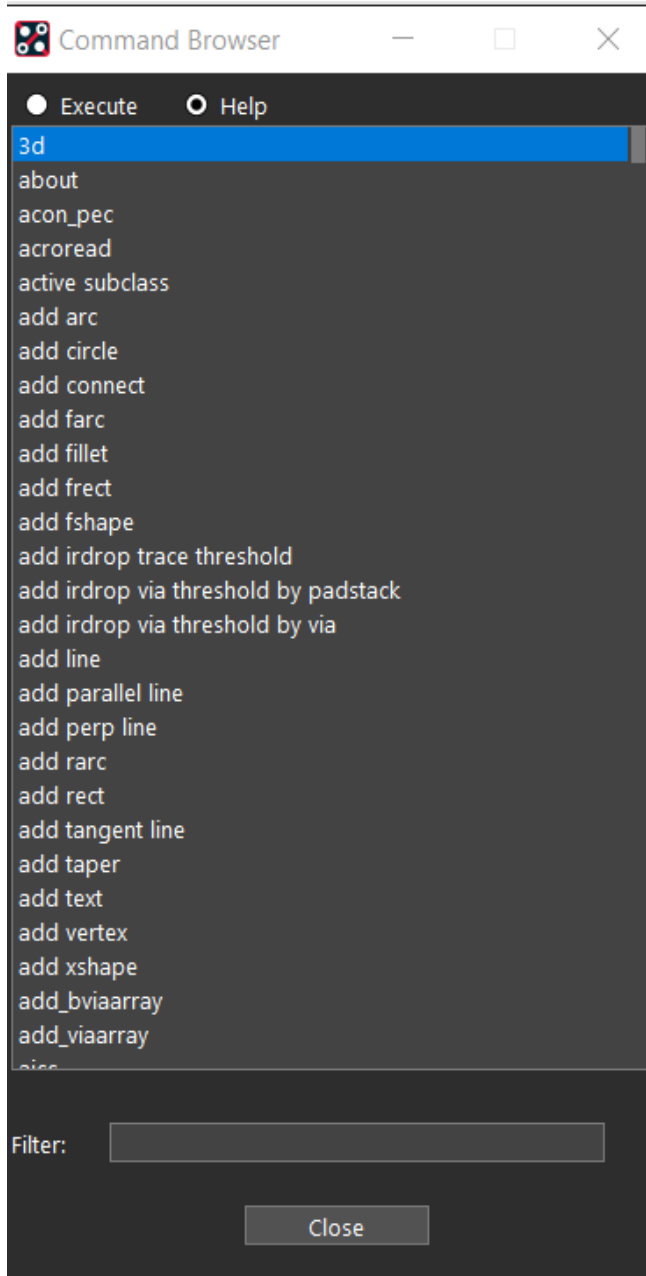
Setup

After downloading and placing `cds_site_skill_menu` in a folder, define a user variable, `CDS_SITE`, with its value as the complete folder location of `cds_site_skill_menu`. Here is a visual aid:



Adding and Appending Allegro Menu Items Through SKILL: RAK

Note: To get a full list of available registered commands in any of the Allegro Layout tools (PCB Editor, APD, or SiP Layout) being used, type `helpcmd` on the command line as shown below:



Register a command for the custom-developed SKILL code. The registered command will show up as an available command in the `helpcmd` output as described above and make custom menu building a breeze.

Adding and Appending Allegro Menu Items Through SKILL: RAK

The axl function for registering a command in a SKILL file is axlCmdRegister ("name of registered command" 'function) as shown below:

```
/* *****  
* DISCLAIMER: The following code is provided for Cadence customers *  
* to use at their own risk. The code may require modification to *  
* satisfy the requirements of any user. The code and any *  
* modifications to the code may not be compatible with current or *  
* future versions of Cadence products. *  
* THE CODE IS PROVIDED "AS IS" AND WITH NO WARRANTIES, INCLUDING *  
* WITHOUT LIMITATION ANY EXPRESS WARRANTIES OR IMPLIED WARRANTIES *  
* OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. *  
* ***** */  
;rename nets for whole design with PC- prefix  
axlCmdRegister("rename all nets" 'sipNetrename) ←  
  
defun( sipNetrename ()  
    let( (netlist oldname newname)  
  
        netlist = axlDBGetDesign()->nets  
  
        foreach( net netlist  
           setq(oldname net->name)  
           setq(newname strcat("PC-" oldname))  
  
                axlRenameNet(oldname newname)  
  
        )  
    );end-let  
);end sipNetrename
```

Summary

Menu customization has been around for a long time. However, the traditional method of updating Cadence-installed menu files is archaic and risky. As menus evolve from release to release with the new functionality and relocation of menu items, menu customization is made easier and permanent with the use of SKILL.

Support

Cadence Learning and Support Portal provides access to support resources, including an extensive knowledge base, access to software updates for Cadence products, and the ability to interact with Cadence Customer Support. Visit <https://support.cadence.com>.

Feedback

Email comments, questions, and suggestions to content_feedback@cadence.com.