

Title: PCB Editor Environment

Product: OrCAD / Allegro PCB Editor

Summary: It explains the whole PCB Editor environment include site environment.

Author/Date: Beate Wilke / 24.03.2009

Table of Contents

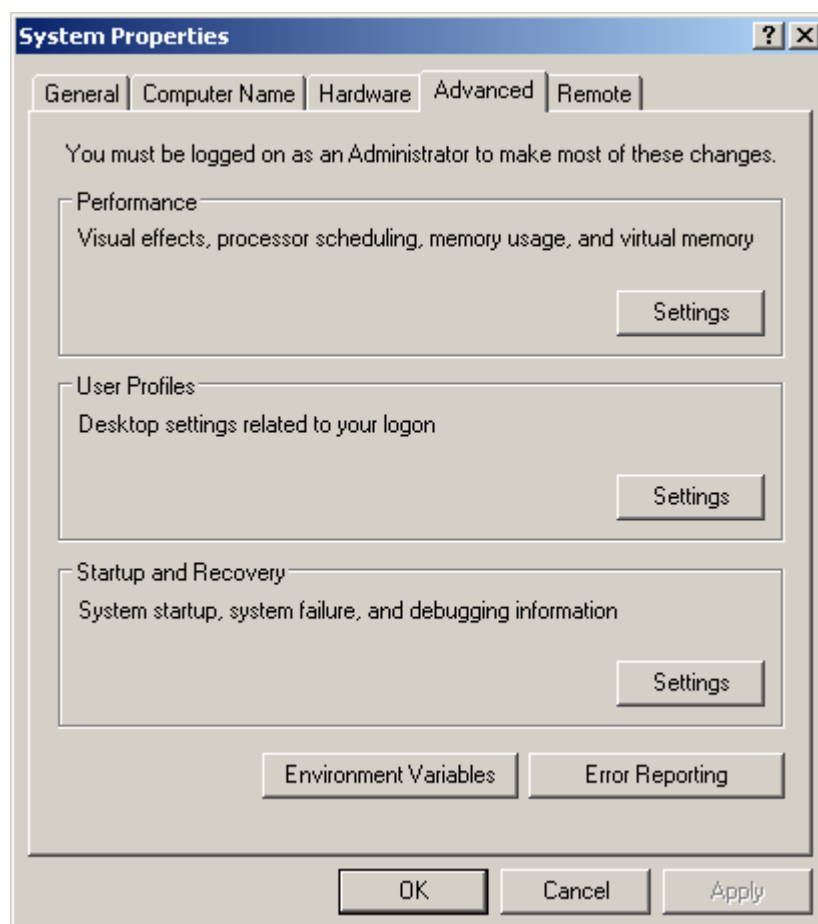
1	PCB Editor Settings.....	2
1.1	HOME Variable.....	2
1.2	PCBENV.....	4
1.3	Path Definitions in the ENV File.....	4
1.4	Funckey.....	6
1.5	Allegro Sub Directories for Artwork and Reports.....	7
1.6	Skill and Allegro.ilinit.....	8
2	CDS_SITE.....	9
2.1	Read Order.....	9
2.2	Site Content.....	9
2.3	Site structure.....	9
3	CDSROOT - Working with different Software Versions.....	10
3.1	Cadence Switch Release.....	10
3.2	Batch Files.....	10

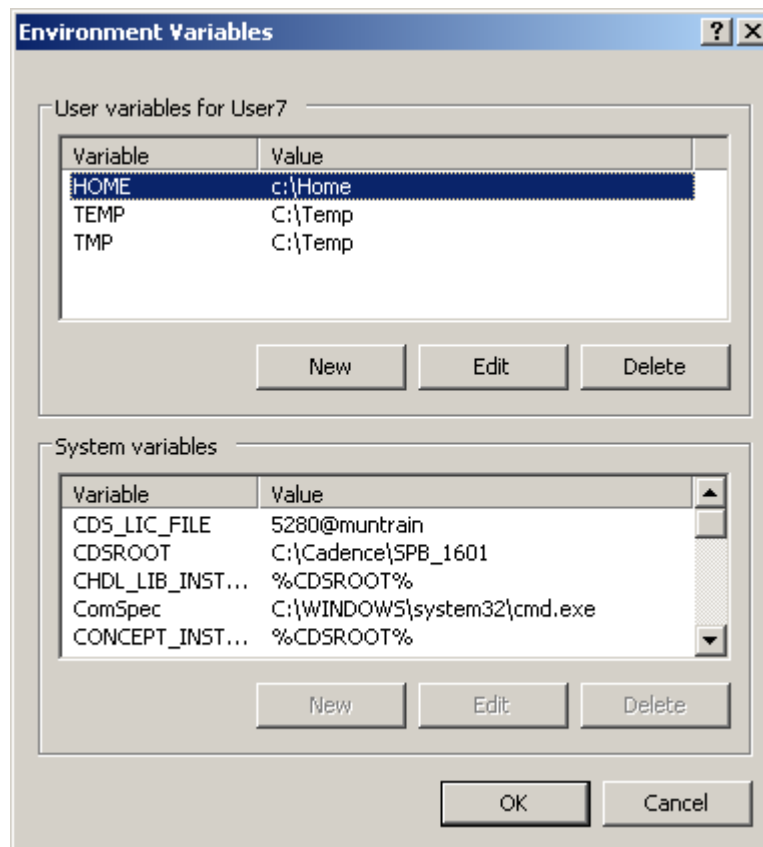
1 PCB Editor Settings

1.1 HOME Variable

HOME is a system variable that defines the location for your PCB Editor environment settings. Every user needs his own HOME folder.

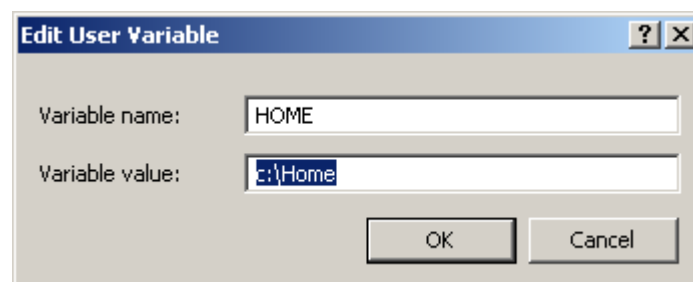
To change the HOME variable go to Start -> Settings -> Control Panel -> System -> Advanced -> Environment Variables.





Home is a user variable. If you set HOME as user Variable, every user needs his own variable. If only one users works on the system you can also define it as system variable.

If you want to change the variable select it and click on change.



You can change the value to all existing folder where the user has full privileges.

If you work with 2 different versions for example 15.7 and 16.01 it would be good to have two different HOME folders. The binary code of 15.7 and 16.01 board files and symbols is different. If you have saved a board file or symbol with 16.01 you can't open it with 15.7. The HOME folder contains all path definitions to symbol and footprint libraries and a list of the recent designs.

1.2 PCBENV

Pcbenv is a folder in the HOME folder and contains the PCB Editor environment. These are the possible files:

Env	This is the Environment file. It is read by PCB Editor at the time the tool is executed. It contains individual user settings such as aliases and function key definitions, library paths to access files on the system, and system variables used by PCB Editor to find the software.
allegro.ini	This file keeps track of the path where your working file is located. It keeps track of the size and location of the main tool window. DO NOT EDIT THIS FILE if you are having problems with PCB Editor. This file can be deleted as a form of troubleshooting. It will be recreated automatically the next time you start PCB Editor.
allegro.mru	This file records a list of the most recently used board files. DO NOT EDIT THIS FILE!
allegro.geo	This file remembers where the forms last came up and places the same type of form in the same location. DO NOT EDIT THIS FILE!
myfavorites.txt	This file contains which class/subclass(es) to be displayed in the MyFavorites folder of the Color Dialog form.
pad_designer.geo	This records the Pad Designer form location and size.
pad_designer.mru	This records a list of most recently used padstacks and paths.

1.3 Path Definitions in the ENV File

The env file contains all path definitions which are different from the default system settings. When you change these setting in PCB Editor by using Setup -> User Preferences -> Design_paths or config_paths the changes are written in the env file. You can also change the env file by open it with a text editor, it's written in ASCII. For new users it can be good to have a basic env file which contains all important settings.

Here is an overview of all possible path settings. The 2 important one are padpath and psmppath. If you work with your own component library you have to set them first.

CATEGORY: design_paths

Dclpath	Search path for decoupling capacitor list files (.dcf).
Dfacnspath	Search path for dfa constraints spreadsheet files (dfa).
Ldfpath	Search path for Library definition file (.ldf).
Modulepath	Search path for design reuse modules (.mdd).
Padpath	Search path for library padstacks (.pad).

Parampath	Search path for parameter files (.prm). These allow reuse of physical design data option settings like text, visibility and grid settings.
pcell_lib_path	Search path for pcell component implementation (.il .ile).
psmpath	Search path for library symbols (.psm .osm .bsm .ssm, fsm).
Techpath	Search path for technology files (.tech).
Tilepath	Search path for reusable die pin tiles (.til).
topology_template_path	Search path for topology template files (.top).

CATEGORY: config_paths

Aptpath	Search path for aperture flash files -- this is obsolete with .fsm support (.bsm).
Artpath	Search path for artwork aperture files (.txt).
Clippath	Search path for sub-drawing files (.clp).
Devpath	Search path for library devices (.txt).
Dfaauditpath	Search path for DFA Audit (.arl .rle).
Materialpath	Search path to locate materials.dat (Allegro) or mcmmat.dat (APD) (.dat).
Ncdpath	Search path for NC Drill files (.txt).
Scriptpath	Search path for scripts.
Textpath	Search path for extracta command files (.txt).
Viewpath	Search path for visibility schema files (.color).
wizard_template_path	Search path for Allegro templates (.brd .dra).
xtalk_table_path	Search path for Cross talk tables (.xtb).

1.4 Funckey

In the PCB Editor training you will learn how to define temporary Funckeyes. To get a permanent definition you have to define the funckeyes in the env file. Here some examples for Funckey definitions:

```
funckey Esc cancel
funckey + subclass -+
funckey - subclass --
funckey " " subclass -+
```

```
funckey e 'settoggle edit etchedit generaleedit; $edit; echo "Toggle EDIT Mode"
```

```
funckey w ""add connect"; echo "ADD Connect"
funckey s 'settoggle hug Hug Shove ; options bubble_space $hug preferred'
funckey d 'settoggle direction 90 45 Off; options lock_direction $direction; echo "Toggle Direction"
```

```
funckey q ""show element";echo "Query Element"
funckey a 'settoggle rats "rats all" "unrats all"; $rats; echo "Toggle Airlines"
```

```
funckey u undo
```

```
funckey l ""define lyrstack"; echo "Define Lyrstack"
funckey c ""cmgr"; echo "Constraints Manager"
funckey k ""dfa_spreadsheet"; echo "Design for Assembly Spreadsheet"
funckey v ""status"; echo "Status Manager"
funckey b ""prmed"; echo "Parameter Edit"
funckey n ""enved"; echo "Environment Edit"
```

```
funckey p "place manual; setwindow form.plc_manual; FORM plc_manual hide; setwindow form.find; FORM find name_type Comp; setwindow form.plc_manual"
```

```
funckey P "place manual; setwindow form.plc_manual; setwindow form.find; FORM find name_type Comp; setwindow form.plc_manual"
```

```
funckey r "iangle 90"
funckey m "pop mirror"
```

```
funckey g grid toggle
```

```
funckey , show measure
```

```
funckey . drc update
```

```
funckey x pick grid
funckey y pick grid
```

```
funckey g 'grid toggle; redraw; echo "Toggle Grid"
```

```
funckey i "zoom in"
funckey o "zoom out"
```

```

funckey f "zoom fit"
funckey z "zoom center"
funckey < "setwindow worldview; trapsize 3285"
funckey h hilight
funckey j dehilight

```

```

funckey S shadow toggle

```

Every user can have its own set of Funckeyes. Normally the System Administrator defines a set of Funckeyes for all users to get a unique handling of the tool.

1.5 Allegro Sub Directories for Artwork and Reports

In the User Preferences you have a category file_management. This category helps to define a file structure.

ads_autosaverevs	Enables file versioning for AUTOSAVE database files. Value <n> = number of versions you want maintained. Default is no versioning.
ads_boardrevs	Enables file versioning for allegro layouts (.brd) and symbol (*.sm) files. Value <n> = number of versions you want maintained. Default is 1 version.
ads_logrevs	Enables file versioning for Allegro log files. Value <n> = number of versions you want maintained.
ads_sdart	The subdirectory to which artwork files should be written.
ads_sdlog	The subdirectory to which log files should be written.
ads_sdplot	The subdirectory to which plot files should be written.
ads_sdreport	The subdirectory to which report files should be written.
ads_textrevs	Enables file versioning of allegro files which are not .brd *.sm or .log Value <n> = number of versions you want maintained.
journal_nobuffer	By default, the journal file has buffered output to improve performance when writing the file over a network. This option changes the output to unbuffered. The only advantage of unbuffered output is if a program crashes more lines are written to the journal file.
journal_prefix	Prefix appended to the program's journal file name. Two special prefix keys are available; "user" substitutes user's login and "host" which substitutes system name. Prefix must be a legal file name (no / or :). Example journal_prefix = user for login "me" would create an allegro journal file of me_allegro.jrl
temp	or TMP, <directory path with a large quantity of free disk space> This variable causes allegro processes to use the specified directory for temporary storage of data files. Most often, artwork requires more temporary space than the /tmp directory contains.

1.6 Skill and Allegro.ilinit

Skill is the PCB Editor internal programming language. You can define custom specific menus and functions with Skill.

At start up, Allegro PCB Editor looks for the allegro.ilinit file. This file contains the location of the SKILL directory and loads the SKILL commands for use. The allegro.ilinit file is located in <HOME>/pcbenv folder or <CDS_SITE>/pcb.

Example for an allegro.ilinit file:

```
load "./skill/ict.il"  
load "./skill/ass.il"  
load "./skill/cds2f.il"
```


2 CDS_SITE

CDS_SITE is a variable on system level. It allows the administrator to define standard setting and paths for all users. CDS_SITE normally is located to a server.

Note: With CDS_Site you can administrate only PCB Editor not DE CIS.

2.1 Read Order

Allegro PCB Editor searches for setting in the following order:



2.2 Site Content

With SITE definition you can administrate nearly all settings. Here are some examples:

- Path to symbols and padstacks
- Funckey
- Sub directories for artwork and reports
- Path to Scripts
- Path to Views
- Skill Routines

2.3 Site structure

A possible structure can be:

```
<cds_site>/pcb /    skill/*.il  
                    script/*.scr  
                    site.env  
                    allegro.ilinit
```

Note: The information in site.env is equal to the env file in the HOME path. In site.env are the global settings in pcbenv/env are the user settings.

3 CDSROOT - Working with different Software Versions

CDSROOT is the system variable which defines the install-dir you use to start the tools. During installation this variable automatically is set. If you only work with one software version you didn't take care of it. But if you work with 2 or more software versions it's important to know which value it just set. You can modify CDSROOT manually, with Cadence Switch Release or use Batch Files.

3.1 Cadence Switch Release

Cadence Switch Release sets CDSROOT and all necessary Path settings. You start it by selecting Start -> Programs -> SPBxxx -> Cadence Switch Release. Use always Switch Release of the highest software version.

Note: Cadence Switch Release does not change the HOME variable.

3.2 Batch Files

Batch Files are short script file which sets variables and starts programs. The Batch File creates a local environment to start a tool. For example you work with 15.7 and have a test installation for 16.01 it would be the best to set all variable to 15.7 and use Batch Files to start 16.01. In this environment you have no risk to open and save a 15.7 board file with 16.01 because you have total different settings.

Example:

```
echo off
set CDSROOT=D:\SPB_16.01
set HOME=E:\HOME\1601

Path=
C:\WINDOWS\system32;
C:\WINDOWS;
C:\WINDOWS\System32\Wbem;
%CDSROOT%\tools\PSpice\Library;
%CDSROOT%\tools\Capture;
%CDSROOT%\tools\bin;
%CDSROOT%\tools\pcb\bin;
%CDSROOT%\tools\specctra\bin;
%CDSROOT%\tools\libutil\bin;
%CDSROOT%\tools\fet\bin;
%CDSROOT%\tools\dfll\bin;
%CDSROOT%\tools\PSpice
```

```
call allegro
```

Note: The path setting must be one row. It's divided into different rows for a better overview. With a Batch File you can set different HOME variables for different tools or versions.