



Agilent Technologies

**Advanced Design System 2002
Customization and Configuration**

February 2002

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Copyright © 2002, Agilent Technologies. All Rights Reserved.

Acknowledgements

Netscape is a U.S. trademark of Netscape Communications Corporation

Contents

1 Customizing the ADS Environment	
Customizing Environment Variables	1-1
Customizing Configuration Variables	1-2
Configuration File Format	1-3
Configuration Variable Description	1-4
Configuration Variable Expansion	1-6
How a Configuration is Determined	1-6
Special Variables	1-8
Color Definitions, hpeecolor.cfg	1-8
Fill Pattern Configuration, hpeefil.cfg	1-11
Viewing Details of the Current Configuration	1-13
Typical Variables	1-15
Variables in de.cfg, de_sim.cfg	1-15
Variables in the File hpeesof.cfg	1-25
Variables in the File dfilter.cfg	1-26
Variables in the File dsynthesis.cfg	1-27
Variables in the File eeapi.cfg	1-29
Variables in the File hpads.cfg	1-30
Variables in the File hpeesofbrowser.cfg	1-30
Variables in the File hpeesofdds.cfg	1-33
Variables in the File hpeesofhelp.cfg	1-35
Variables in the File hpeesofinstrio.cfg	1-35
Variables in the File momentum.cfg	1-36
Variables in the File smg_ui.cfg	1-36
Variables in the File hpeesofsim.cfg	1-37
2 Customization Examples	
Creating a Custom Menu	2-1
Creating Custom Libraries	2-3
Assisted Customization	2-4
Site-Wide Customization (Assisted)	2-5
User Customization (Assisted)	2-6
Manual Customization	2-7
Site-Wide Customization (Manual)	2-8
User Customization (Manual)	2-8
Project Customization (Manual)	2-9
Advanced Library Customization Techniques	2-10
Site-Wide Customization (Advanced)	2-11
User Customization (Advanced)	2-13

Modifying the List of Available Symbol Names	2-15
Adding Online Help for User-Defined Items	2-17

Index

Chapter 1: Customizing the ADS Environment

The Advanced Design System suite of tools uses a number of configuration files. The files contain definitions for configuration variables and environment variables.

Customizing Environment Variables

Environment variables are usually initialized in a shell start-up file such as *.cshrc* (C-shell) or *.profile* (Bourne-shell) in your home directory. [Table 1-1](#) describes the most common environment variables used by the Advanced Design System. On the PC, these environment variables get set automatically at installation.

Table 1-1. Environment Variables

Name	Default Value	Description
HPEESOF_DIR	/hpeesof	The location where the software has been installed (also called the root location).
LM_LICENSE_FILE	\$HPEESOF_DIR/licenses/license.dat	Specifies the location of the FLEXlm security license files.
PRINTER	lpr	Specifies the printer used by the <i>hpeedit</i> program.

The environment variable *HPEESOF_DIR* is used extensively throughout Advanced Design System. It does not need to be set if Advanced Design System has been installed in the directory */hpeesof*.

Customizing Configuration Variables

The default values of the configuration variables should get you up and running, but they can be customized to better reflect your work environment. By default, the program searches for these configuration files in the following order, and uses the first one found:

- Your current project directory

Define configuration variables here that apply only to this project.

- Your personal directory = $\$HOME/hpeesof/config$

Define configuration variables here that apply to all your projects.

Note: On the PC, $\%HOME\%$ represents the path you specified as the *Home Folder* during installation (by default, $C:\users\default$)

- Your site's customized site directory $\$HPEESOF_DIR/custom/config$

Define configuration variables here that apply to all projects of all users on a site-wide basis. Configuration variables defined here will not be overwritten by installation of subsequent program patches or updates.

- The ADS installation directory $\$HPEESOF_DIR/config$

Define the default installed configuration variables here that apply to all projects of all users. Configuration variables should not be customized here.

Each of the individual Advanced Design System tools has its own configuration file. The supplied configuration files can be found in the directory $\$HPEESOF_DIR/config$ (where $\$HPEESOF_DIR$ represents the complete installation path). The filenames for each of the tools are shown in Table [Table 1-2](#).

Table 1-2. Configuration Filenames

Product/Tool	Filename
Design Environment	de_sim.cfg
Digital Filter	dfilter.cfg
Digital Synthesis	dsynthesis.cfg
Data Displays	hpeesofdds.cfg
Data Sets	hpeesofdss.cfg
Instrument Server	hpeesofinstrio.cfg

Table 1-2. Configuration Filenames (continued)

Product/Tool	Filename
Momentum	momentum.cfg
Browser (Vendor and component parts)	hpeesofbrowser.cfg
Simulation and Synthesis Message Window	hpeesofsess.cfg
Layered A.P.I.	eeapi.cfg
Ptolemy models and MATLAB	hpads.cfg
Colors	hpeecolor.cfg
Fill patterns	hpeeifill.cfg
GUI search paths	hpeesof.cfg
Online help system	hpeesofhelp.cfg
AEL search paths	hpeesofsim.cfg
Library translator	hplibtrans.cfg
Ptolemy	hptolemy.cfg
SMG search path	smg_ui.cfg
Spice to IFF	spctoiff.cfg
Spice translator	spice_xlator.cfg
E-Syn	esyn.cfg
LineCalc	linecalc.cfg
RF Synthesis	rfsynthesis.cfg

Configuration File Format

Configuration variables are defined using standard assignment syntax:

```
<variable>=<value>
```

where *variable* is the configuration variable name and *value* is the string that is assigned to the variable. This *value* can be a directory, search path, numeric value, or other value defined by the program. For example, the location of the preferences could be set by:

```
PREFERENCES_DIR=${$HPEESOF_DIR}/de/defaults
```

where *\$HPEESOF_DIR* is an environment variable representing the complete installation path.

Note the following details about file format, illustrated in the accompanying example:

- Any line starting with # is a comment line and is ignored
- Blank lines are ignored
- Blank spaces to the left or right of the equal sign are ignored
- Empty assignments are acceptable (as shown by *env_var_2*)

Example

```
#
# This is a sample configuration file
#
env_var_1 = first_value
env_var_1a = value-1a
#
env_var_2 =
env_var_3 = 17
env_var_4 = $SYSTEM/%HOME/ex4
env_var_4a = {$HPEESOF_DIR}/de/defaults/{%env_var_1a}.ex4a
```

References to environment variables *\$name* and ADS configuration variables *%name* can appear in any order; they are resolved inside each ADS program. A reference to an environment variable or configuration variable name may be enclosed in braces {} when their use is not followed by a punctuation character. The braces are not included in an expanded value.

Configuration Variable Description

ADS configuration variables are identified by a case-insensitive string of 1 to 31 alphanumeric characters (an underscore is allowed). Variables that are longer than 31 characters are automatically truncated when the configuration file is read. When a variable is set from a configuration file, the line

```
<variable>=<value>
```

is parsed and *variable* is set to the *value*.

A configuration variable can contain a reference. A reference may be enclosed in braces and the text is preceded by either a \$ or % (see [Table 1-3](#)). For example,

```
PREFERENCES_DIR={$HPEESOF_DIR}/de/defaults
```

assigns the string *\$HPEESOF_DIR/de/defaults* to the configuration variable *PREFERENCES_DIR*, and *\$HPEESOF_DIR* is defined in a start-up file.

Table 1-3. ADS Configuration Variables

Configuration variable	Description
\$name	The configuration variable is replaced by the text string assigned to <i>name</i> in the environment.
%name	The configuration variable is replaced by the text string assigned to <i>name</i> in the ADS configuration file. Certain ADS variables have a pre-defined meaning in a given program. All of the special ADS configuration variables that are recognized are given in the shipped <code>{\$HPEESOF_DIR}/config/<app-name>.cfg</code> file.

Most ADS configuration variables are assigned a single value. This text can represent a *number*, *string*, *file*, or *path* as in the following examples.

```
STATUS_DISP = 0
EESTATUS_LOG_FILE = Off
HPEESOF_KEY = $HPEESOF_DIR/licenses/hpeesof.key
PREFERENCES_DIR={ $HPEESOF_DIR }/de/defaults
```

Note that when the *file* is outside of a project configuration, it is necessary to include the full path with the filename to ensure that the file is found.

Some ADS configuration variables accept a *path* assignment. A *path* is a list of one or more directories, where each directory is separated by a colon.

```
variable=directory:directory: ....
```

These variables can represent search paths or load paths, depending on the variable and the program. Ordering of the list is significant. When treated as a search path, the list is only scanned until the item needed is found. For loading, the entire list is sequentially examined with the last directory usually taking precedence.

The documentation for each specific variable indicates the type of value allowed. Some variables may have a limited type, such as a range of numbers or list of specific strings.

Configuration Variable Expansion

Configuration variable expansion refers to the process of replacing all references and variables that make up the configuration variable with their text equivalents until the complete value of the configuration variable is known. (There are no limits to the levels of referencing you can use.) Note that this expansion is done internally by the program; the file that contains the variable assignments is not modified.

For example, assume that the environment variable *HPEESOF_DIR* is set to */hpeesof/mysite* and the following two lines exist in the file *de_sim.cfg*.

```
PROJECT3 = de
SYSTEM_CUSTOM_DE_SYMBOLS = {$HPEESOF_DIR}/custom/{%PROJECT3}/symbols
```

After expansion within hpeesof tool:

```
SYSTEM_CUSTOM_DE_SYMBOLS = /hpeesof/mysite/custom/de/symbols
```

As in the environment, you may redefine and add your own ADS variables in any ADS configuration files to help manage system- and user-specific configurations. However, a given ADS program only uses certain expanded ADS configuration variables, but the unexpanded value could contain any number of user-defined and/or ADS variables.

How a Configuration is Determined

Each configuration is distinguished by its name. Each named configuration contains all of the variables needed for describing that particular setup. At the topmost level is the configuration file named *hpeesof.cfg*, which is associated with all ADS programs. Other named configurations are associated with specific programs.

Each named configuration is determined by the location and content of any related configuration files. For example, the complete *hpeesof* configuration could consist of:

- The file *hpeesof.cfg*, located in the install directory *\$HPEESOF_DIR/config*
- An edited copy of this same file, located in the customized site directory *\$HPEESOF_DIR/custom/config*
- An edited copy of this same file located under your home directory *\$HOME/hpeesof/config*
- An edited copy of this same file located in a particular project directory

As noted earlier, configurations in the user directory take precedence over the installation directory, and configurations in the project directory take precedence over the user directory. This is because the search order for configuration information is: project directory, user directory, site directory, and installation directory.

Note that variables in each successive file override any previously set value. You can even put two lines in the same configuration that assign values to the same variable, as in:

```
MY_VAR1 = entry_a
MY_VAR1 = entry_b
```

When expanded, MY_VAR1 will always have the value of *entry_b*.

Table 1-4 shows a list of named configurations used in the programs.

Table 1-4. Named Configurations

Activity	Configuration
Design entry	hpeesof
Fill patterns	hpeefill
Online Help	hpeesofhelp
Colors	hpeecolor
Simulation	hpeesof
	de
	de_sim
	comms_ckt (Communications Design Suite only)
Status/Synthesis Messages	hpeesof
	eestatus
Data Displays	hpeesof
	eehelp
	hpeefill
	hpeecolor
	eegraph

Special Variables

There are two configuration files that have a different syntax than the others: *hpeecolor* and *hpeefill*. Their formats are presented here. Note that the naming convention and loading of system-wide, site-wide, home directory, and project-specific configurations still apply. Only the internal file format differs.

Color Definitions, *hpeecolor.cfg*

The design windows (Schematic and Layout), Data Display windows, and plotters use the colors defined in the *hpeecolor* configuration. By default, these colors are read from *{\$HPEESOF_DIR}/config/hpeecolor.cfg*, but can be overridden at the site, user, or project level (any of which can have an *hpeecolor* file).

The *hpeecolor* file format differs from the *variable=value* syntax. Each line contains these major fields:

- RGB color values
- Color name
- Plotter pen number

The syntax for each line is as follows:

```
<Red #> <Green #> <Blue #> : <Color name> : <Pen number>
```

The following example illustrates the format of a typical color definition file:

R	G	B	X11 color name	Plotter pen number
0	0	0	: black	: 1
255	0	0	: red	: 2
0	255	0	: green	: 3
0	0	255	: blue	: 4
255	255	255	: white	: 5
128	64	0	: salmon	: 6
			: salmon	: 7

Colors are described in the first and/or second fields of this file. You can enter three RGB color values (in the range 0-255) in the first field and/or you can enter the X color name from the RGB color database (found in */usr/lib/X11/rgb.txt* for Motif or */usr/openwin/lib/rgb.txt* for Sun's Open Windows) in the second field. If both are entered, the RGB values take precedence, and the color name is ignored.

The last field specifies which pen number to use for hard-copy output when performing HPGL plotting. An entry in this field is required and pen numbers may be recycled and repeated.

By default, the current implementation uses shared server colors from the RGB color database and uses no privately allocated color cells for X Window displays. This is done to simplify the color specification and to promote sharing of colors. Sharing of color cells can only happen if two clients allocate read-only color cells with the same X color name or RGB values.

Important On UNIX, we strongly recommend that you only use the names given in the RGB color database instead of explicit RGB values; on the PC, only RGB values are supported.

Because of differences in screen hardware, the same RGB values may generate different colors on different hardware. By using names from the color database, you are more likely to get colors close to the ones requested across different hardware platforms.

It is also important to note that values corresponding to color names are not fixed. Therefore, they may not be exactly the same across all platforms or correspond to an exact RGB value.

Note Some platforms have limited color resources and some platforms may use non-shared colors, resulting in color exhaustion problems. When ADS programs run into these types of color problems, the solution is to reduce the number of requested colors until color conflicts stop occurring. As an example, truncating the list of colors in *hpeecolor.cfg* to 64 colors may be a good start.

An example *hpeesof* configuration file is shown:

```
# =====
# Example hpeesof Color Configuration File:
# =====
#   R   G   B   :   Color Name   :   Plotter Pen
#   --- --- ---   :   -----   :   -
#   0   0   0   :   black       :       1
#  255  0   0   :   red         :       2
#   0  255  0   :   green      :       3
#   0   0  255  :   blue       :       4
#  255 255  0   :   yellow    :       5
#  255  0  255  :   magenta   :       6
#   0  255 255  :   cyan      :       7
#  255 255 255  :   white     :       8
# Either RGB values or standard X Color Names, or both,
# may be specified for color entries. Note, if both RGB
# and Color Name values are specified, the RGB values take
# precedence.
# =====
: black : 1
: red : 2
: yellow : 3
: green : 4
: cyan : 5
: blue : 6
: magenta : 7
: gray : 8
: white : 1
```

If the ADS configuration file cannot be found or the *hpeesof* configuration does not contain an HPEESOF_COLOR variable (which determines the location of the ADS color definition file), a set of ten basic default server colors are automatically loaded as follows:

1 = black	5 = cyan	9 = white
2 = red	6 = blue	10 = medium blue
3 = yellow	7 = magenta	
4 = green	8 = gray	

Fill Pattern Configuration, hpeefil.cfg

Besides colors, the design windows (Schematic and Layout), Data Display windows, and plotters also use the fill patterns defined by the *hpeefil* configuration. By default, these patterns are read from *{\$HPEESOF_DIR}/config/hpeefil.cfg*, but can be overwritten at the site, user, or project level (any of which can have an *hpeefil* file).

The *hpeefil* file format differs from the *variable=value* syntax. Each line contains two major fields.

- Path and filename of the fill pattern
- HPGL/2-pattern-specification

The syntax for each line is as follows:

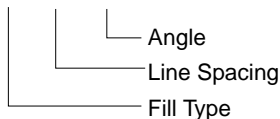
```
<Bitmap_file_location> : <Fill type> <Line spacing> <Angle>
```

where

- Fill type—is the pattern inside an object's border
- Line spacing—is the distance between the hatched lines in plotter units (where a plotter unit is typically 1/72-inch).
- Angle—is the angle of hatch lines in the fill pattern, expressed in degrees (for fill type 3 or 4 only).

The following example illustrates the format of a typical fill pattern definition file:

```
/fill/pat/path/pat1.pattern : 3 50 15
/fill/pat/path/pat2.pattern : 3 50 30
/fill/pat/path/pat3.pattern : 4 60 45
/fill/pat/path/pat4.pattern : 4 60 90
```



Fill patterns use standard X-bitmap files that can be easily created with the *X bitmap* utility. The first field specifies the path to the X-bitmap file, which is used for pattern fills, while the remaining fields are the HPGL/2 fill pattern specification.

Fill type description	Fill type number
Solid bidirectional	1
Solid unidirectional	2
Hatched	3
Cross hatched	4

Note Solid bidirectional fill is faster than solid unidirectional fill, but is not supported by all plotters.

An *hpeesof* configuration file is shown in the following example:

```
# =====
# hpeesof Fill Pattern Configuration File
# =====
# Screen Output      : Hardcopy Output
# X Bitmap Filename  : Type Spacing Angle
# -----
# path/pattern_file1 : 3      70      15
# path/pattern_file2 : 4      50      45
# path/pattern_file3 : 4      45      90
# =====
#
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat1.pattern : 1 0 0
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat1.pattern : 3 50 30
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat2.pattern : 3 50 45
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat3.pattern : 3 50 90
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat4.pattern : 4 50 15
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat5.pattern : 4 50 30
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat6.pattern : 4 50 45
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat7.pattern : 4 50 90
# ${HPEESOF_DIR}/lib/PDE/fill_patterns/pat8.pattern : 3 70 15
```

If the ADS configuration file cannot be found or the *hpeesof* configuration does not contain an `HPEESOF_FILL` variable (which determines the location of the ADS fill definition file), a single, unidirectional, solid fill pattern is automatically loaded.

Viewing Details of the Current Configuration

The *Configuration Explorer* enables you to search, view, save to file, and print the current settings of ADS configuration files. You can browse the configuration files in the following locations:

- Installation directory
- Site-wide customization directory
- User directory
- Current project

To launch the Configuration Explorer:

From the ADS Main window, choose **Options > Configuration Explorer**.

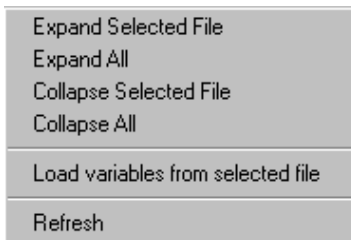
To view the properties of a given file:

1. Select the desired configuration directory from the drop-down list.
2. Click the desired filename. The file location, size, date, and permissions are displayed in the Status pane, as well as the location of any additional files by that same name.

```
File Name : C:/users/default/hpeesof/config/de_sim.cfg
File Size : 924
File Date : Jun 4 , 18:34
File Permissions : -rw-r--
Other de_sim.cfg files :
  C:\users\default\hpeesof\config\de_sim.cfg <<<
  C:\ADS2001\custom\config\de_sim.cfg
  C:\ADS2001\config\de_sim.cfg
```

To access the pop-up menu:

Position the pointer over any filename and click right.



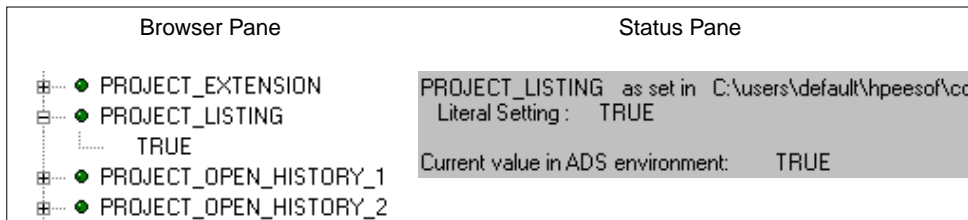
- Use the uppermost group of commands to quickly expand and collapse the tree.
- Use *Load variables from selected file* to explicitly load into memory changes you have made to a configuration file since launching ADS.
- Use *Refresh* to view changes you have made to a configuration file since launching ADS.

To view the list of variables in a given file:

Click once on the plus sign (expand) in front of the filename to display the list.

To view the current setting of a given variable:

- Click once on the variable name to display the value in the Status pane
- or
- Click once on the plus sign in front of the variable to display its value in the browser pane



To search for a specific variable name and/or value:

1. Select **Variable** in the Search pane.
2. Optionally, refine your search by selecting **Variable Name** and/or **Variable Value**.
3. Optionally, refine your search by selecting **Exact Match**.
4. Type the search string in the Search for field and click **Search**. The results are displayed in the Status pane.

To search for a specific filename:

1. Select **File** in the Search pane.
2. Optionally, refine your search by selecting **Exact Match**.

3. Type the search string in the Search for field and click **Search**. The results are displayed in the Status pane.

To retain the results of all listings and searches:

Select the **Keep history** option in the Status pane. Note that to retain the variable value listings, you must display the values in the *Status* pane, not the *Browser* pane. To clear the history click **Clear**.

To print the contents of the Status pane:

Click **Print**. The contents are sent to your default printer.

To save the contents of the Status panel to a file:

1. Click **Save Status**. A dialog box appears prompting you for a filename.
2. Change paths as desired, supply a filename, and click **OK**.

This information may be especially helpful when discussing configuration issues with Technical Support.

Typical Variables

The remaining variables use the *variable=value* syntax.

Note The Design Kit variables are documented in the *Design Kit Installation and Setup* manual.

Variables in de.cfg, de_sim.cfg

A number of resources related to the user interface and simulation can be customized in the file *de_sim.cfg*. This file can be customized for individual projects, all projects, etc., as described in, “[Customizing Configuration Variables](#)” on page 1-2. You will find a default copy of this file, with a limited number of variables in it, in the following locations:

- In your */hpeesof/config* directory under \$HOME (on UNIX) or *c:\users\default* (on PC, or whatever directory you chose instead during installation)
- Every project directory you create

You can add to this file (or modify, if they currently exist in the file) all variables described in [Table 1-5](#).

Note The variables described in the table are defined in either *de.cfg* or *de_sim.cfg* (under *\$HPEESOF_DIR/config*), however when you customize any of these variables, it must be done in *de_sim.cfg*. To minimize typing and ensure accuracy of the variable names, copy the variables you want to customize from one or both of these files to the appropriate *de_sim.cfg* file and modify them there.

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg*

Variable, description, and example	Value(s)
FILE_FORMAT Specifies the format for writing design files. Use ASCII for portability and upgrades; binary is faster and uses less disk space. Example: FILE_FORMAT=ascii	ascii binary
LAYERS_PATH Specifies the layer file search path. Example: LAYERS_PATH=./:\${HPEESOF_DIR}/de/defaults	Path
PREFERENCES_PATH Specifies the preference file search path. Example: PREFERENCES_PATH=./:\${HPEESOF_DIR}/de/defaults	Path
LOCAL_AEL Specifies the project-specific AEL files to be automatically loaded each time when a project is opened. It should be set only in the <i>de_sim.cfg</i> file located in the project directories. These AEL filenames may have absolute, relative, or no path specified, but since the files are generally located in the project directory, a relative path would be used beginning with a dot and slash (<i>./</i>). An AEL file that is already in the <i>networks</i> subdirectory will be automatically loaded. Any other AEL files that you would like included should be placed here. Example: LOCAL_AEL= my_file	Filenames (without extension) or Path
USER_AEL Specifies the user-specific AEL files to be automatically loaded once at startup. It should be set only in the <i>de_sim.cfg</i> file located in the user's personal customized directory <i>\$HOME/hpeesof/config</i> . If you specify a path, rather than filenames, it cannot be a relative path; it must be an absolute path, or you can use an environment variable. A typical use would include <i>\$HOME</i> to reference files in the user's home directory. Example: USER_AEL= \$HOME/hpeesof/MyLibrary/ael/	Filenames (optional extension) or Path

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
<p>USER_DSN_PATH Specifies the directory (or directories) separated by a colon (:) where user-specific design files are located. It should be set only in the <i>de_sim.cfg</i> file located in the user's home directory. This variable cannot specify a relative path; it must be an absolute path, or you can use an environment variable. A typical use would include \$HOME to reference files in the user's home directory. Note: this variable should be used in conjunction with USER_AEL. Example: USER_DSN_PATH = \$HOME/hpeesof/MyLibrary/designs/</p>	Path
<p>SITE_AEL Specifies the site-specific AEL files to be automatically loaded once at startup. It is typically set in the <i>de_sim.cfg</i> file located in \$HPEESOF_DIR/custom/config. If you specify a path, rather than filenames, it cannot be a relative path; it must be an absolute path, or you can use an environment variable. Example: SITE_AEL= {\$ADS_DESIGNS}/library1/ael/ Example: SITE_AEL=Q:\ads_designs\library1\ael\</p>	Filenames (optional extension) or Path
<p>SITE_DSN_PATH Specifies the directory (or directories) separated by a colon (:) where design files for site-wide use are located. It is typically set in the <i>de_sim.cfg</i> file located in \$HPEESOF_DIR/custom/config. This variable cannot specify a relative path; it must be an absolute path, or you can use an environment variable. Note: this variable should be used in conjunction with SITE_AEL. Example: SITE_DSN_PATH = { \$ADS_DESIGNS}/library1/designs/ Example: SITE_DSN_PATH=Q:\ads_designs\library1\designs\</p>	Path
<p>HPEESOF_LANGUAGE Specifies the language used by the PDE program. Example: HPEESOF_LANGUAGE=english</p>	Language name
<p>ENGLISH_MSG_EXT Specifies the language of the program messages. Example: english_MSG_EXT=eng</p>	English = eng German = ger French = fra
<p>ENGLISH_MSG_FONT Specifies the font supporting the selected local language. Example: ENGLISH_MSG_FONT=9x15 bold</p>	European = 9x15 Japanese = 8x16
<p>PDE_MSG_FILE Specifies the full path to the PDE message directory file. Example: PDE_MSG_FILE= {\$HPEESOF_DIR}/msgs/english/PDE/PDE.msg</p>	Path
<p>NETLIST_FILE_NAME Specifies the netlist filename. Example: NETLIST_FILE_NAME =netlist.log</p>	Filename

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
EGSLIST_FILE_NAME Specifies the filename for the EGS translator option. Example: EGSLIST_FILE_NAME = ./egslist.log	Filename
EGS_OPTION_FILE: Specifies the path to the EGS translator option files. Example: EGS_OPTION_FILE = ./:\${HPEESOF_DIR}/config	Path
GDSII_OPTION_FILE Specifies the path to the GDSII option files. Example: GDSII_OPTION_FILE=./:\${HPEESOF_DIR}/config	Path
MASK_OPTION_FILE Specifies the path to the mask translator option files. Example: MASK_OPTION_FILE=./:\${HPEESOF_DIR}/config	Path
IGES_OPTION_FILE Specifies the path to the IGES translator option files. Example: IGES_OPTION_FILE=./:\${HPEESOF_DIR}/config	Path
SPICE_OPTION_FILE Specifies the search path for the Spice files. Example: SPICE_OPTION_FILE=./:\${HPEESOF_DIR}/config	Path
IFF_OPTION_FILE Specifies the search path for the IFF option files. Example: IFF_OPTION_FILE=./:\${HPEESOF_DIR}/config	Path
DIALOG_TIME_OUT Specifies the time before the copyright or information dialog box is automatically dismissed. Example: DIALOG_TIME_OUT=10000	Time in milliseconds
BALLOON_HELP_TIMEOUT Time taken for balloon help to appear when icon is selected. Example: BALLOON_HELP_TIMEOUT = 600	Time in milliseconds
BALLOON_HELP_DISMISSAL_TIMEOUT Time taken for balloon help to disappear when icon is selected. Example: BALLOON_HELP_DISMISSAL_TIMEOUT = 2500	Time in milliseconds
BALLOON_HELP Sets whether balloon help exists or not. Example: BALLOON_HELP = TRUE	TRUE or FALSE
DOUBLE_CLICKS_TIME_OUT Specifies the time interval in which a double-click is registered. Example: DOUBLE_CLICKS_TIME_OUT=200	Time in milliseconds

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
QUERY_PROCESS_STATUS Specifies how often PDE should query the child process status. Example: QUERY_PROCESS_STATUS=1	Time in seconds
MAX_PROCESS_READY_TIMEOUT Defines when PDE should quit waiting for the child process to be ready after it's spawned Example: MAX_PROCESS_READY_TIMEOUT=300	Time in seconds
BOM_LIST Specifies the list of report types presented in the Bill of Materials dialog box. Example: BOM_LIST=HP EEsof (HP EEsof format)/HPEESOF_bom	Report types
PARTS_LIST Specifies the list of report types presented in the Parts List dialog box. Example: PARTS_LIST=HP EEsof (HP EEsof format)/HPEESOF_parts	Report types
BOM_ITEM Specifies the spacing allocated for item name field in the Bill of Materials dialog box. Example: BOM_ITEM=30	Integer
BOM_QTY Specifies the spacing allocated for quantity field in the Bill of Materials dialog box. Example: BOM_QTY=7	Integer
BOM_DESC Specifies the spacing allocated for description field in the Bill of Materials dialog box. Example: BOM_DESC=50	Integer
WEB_BROWSER Web browser Example: WEB_BROWSER = netscape	Browser name
TEXT_EDITOR_PROGRAM_NAME Text editor Example: TEXT_EDITOR_PROGRAM_NAME=hpeesofedit	Editor name
INVOKE_DDS_ON_BOOTUP Program Preference. Example: INVOKE_DDS_ON_BOOTUP=FALSE	TRUE or FALSE
INVOKE_LIB_SERVER_ON_BOOTUP Program Preference. Example: INVOKE_LIB_SERVER_ON_BOOTUP = FALSE	TRUE or FALSE

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
FILE_OPEN_HISTORY_COUNT Maximum number of files to be shown at the end of the File Menu in the Schematic and Layout windows. Example: FILE_OPEN_HISTORY_COUNT = 4	Integer
PROJECT_OPEN_HISTORY_COUNT Maximum number of projects to be listed at the end of the File Menu in the Schematic and Layout windows. Example: PROJECT_OPEN_HISTORY_COUNT = 4	Integer
DESIGN_LIST_COUNT Maximum number of designs to appear at the end of the Windows Menu in the Schematic and Layout windows. Example: DESIGN_LIST_COUNT = 9	Integer
SCHEMATIC_COMPONENT_PALETTE_LIST_COUNT Maximum number of palettes displayed in the component palette drop-down list on the Schematic window toolbar. Example: SCHEMATIC_COMPONENT_PALETTE_LIST_COUNT = 16	Integer
LAYOUT_COMPONENT_PALETTE_LIST_COUNT Maximum number of palettes displayed in the component palette drop-down list on the Layout window toolbar. Example: LAYOUT_COMPONENT_PALETTE_LIST_COUNT = 16	Integer
SCHEMATIC_COMPONENT_PALETTE_LIST_WIDTH Width of the component palette drop-down list on the Schematic window toolbar. Example: SCHEMATIC_COMPONENT_PALETTE_LIST_WIDTH = 16	Integer
LAYOUT_COMPONENT_PALETTE_LIST_WIDTH Width of the component palette drop-down list on the Layout window toolbar. Example: LAYOUT_COMPONENT_PALETTE_LIST_WIDTH = 16	Integer
SCHEMATIC_COMPONENT_HISTORY_LIST_COUNT Maximum number of components displayed in the component history drop-down list on the Schematic window toolbar. Example: SCHEMATIC_COMPONENT_HISTORY_LIST_COUNT = 10	Integer
LAYOUT_COMPONENT_HISTORY_LIST_COUNT Maximum number of components displayed in the component history drop-down list on the Layout window toolbar. Example: LAYOUT_COMPONENT_HISTORY_LIST_COUNT = 10	Integer
SCHEMATIC_COMPONENT_HISTORY_LIST_WIDTH Width of the component history drop-down list on the Schematic window toolbar. Example: SCHEMATIC_COMPONENT_HISTORY_LIST_WIDTH = 7	Integer

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
<p>LAYOUT_COMPONENT_HISTORY_LIST_WIDTH Width of the component history drop-down list on the Layout window toolbar. Example: LAYOUT_COMPONENT_HISTORY_LIST_WIDTH = 7</p>	Integer
<p>LAYER_LIST_COUNT Maximum number of layers to be shown in the layer list drop-down list on the Layout window toolbar. Example: LAYER_LIST_COUNT = 10</p>	Integer
<p>LAYER_LIST_WIDTH Width of the drop-down layer list on the Layout window toolbar. Example: LAYER_WIDTH = 14</p>	Integer
<p>SIMULATION_HOST_LIST Remote simulation host names Example: SIMULATION_HOST_LIST = machine.company.com</p>	Host names
<p>MAIN_WINDOW_X_LOC, MAIN_WINDOW_Y_LOC The default location of the Main window. The upper left corner of the window is positioned at the specified X,Y screen coordinates, where 0,0 is the upper left corner. Example: MAIN_WINDOW_X_LOC = 0 Example: MAIN_WINDOW_Y_LOC = 0</p>	Integer
<p>MAIN_WINDOW_WIDTH, MAIN_WINDOW_HEIGHT The default size of the Main window in pixels (window can still be re-sized manually). Example: MAIN_WINDOW_WIDTH = 365 Example: MAIN_WINDOW_HEIGHT = 340</p>	Integer Min. Width = 340 Min. Height = 300
<p>SCHEMATIC_WINDOW_X_LOC, SCHEMATIC_WINDOW_Y_LOC The default location of the Schematic window. The upper left corner of the window is positioned at the specified X,Y screen coordinates, where 0,0 is the upper left corner. Example: SCHEMATIC_WINDOW_X_LOC = 260 Example: SCHEMATIC_WINDOW_Y_LOC = 0</p>	Integer
<p>SCHEMATIC_WINDOW_WIDTH, SCHEMATIC_WINDOW_HEIGHT The default size of the Schematic window in pixels (window can still be re-sized manually). Example: SCHEMATIC_WINDOW_WIDTH = 780 Example: SCHEMATIC_WINDOW_HEIGHT = 530</p>	Integer Min. Width = 760 Min. Height = 500

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
<p>LAYOUT_WINDOW_X_LOC, LAYOUT_WINDOW_Y_LOC</p> <p>The default location of the Layout window. The upper left corner of the window is positioned at the specified X,Y screen coordinates, where 0,0 is the upper left corner.</p> <p>Example: LAYOUT_WINDOW_X_LOC = 260 Example: LAYOUT_WINDOW_Y_LOC = 95</p>	Integer
<p>LAYOUT_WINDOW_WIDTH, LAYOUT_WINDOW_HEIGHT</p> <p>The default size of the Layout window in pixels (window can still be re-sized manually).</p> <p>Example: LAYOUT_WINDOW_WIDTH = 780 Example: LAYOUT_WINDOW_HEIGHT = 530</p>	Integer Min. Width = 760 Min. Height = 500
<p>SAVE_PROJECT_STATE</p> <p>Sets a default to be used for any project for which no <i>save_project_state.ael</i> file exists. This file is created/modified when you change projects or exit the program and the <i>Save Project State on Exit</i> option is on. When on (TRUE), the status of every currently open Schematic and Layout window will be saved and subsequently restored, the next time you open that project. Note that only windows with named designs are saved; untitled windows are not saved.</p> <p>Example: SAVE_PROJECT_STATE = TRUE</p>	TRUE, FALSE
<p>COMMAND_DIALOG_WIDTH</p> <p>Sets the width, in pixels, of the Command Line dialog box (<i>Options > Command Line</i> in the Main window). The default is 400; a wider setting may be helpful for viewing lengthy AEL functions.</p> <p>Example: COMMAND_DIALOG_WIDTH = 550</p>	Integer
<p>DATA_FILES</p> <p>Specifies the search path for data files.</p> <p>Example: DATA_FILES = ./data:{\$HOME}/hpeesof/my_datafiles</p>	Path
<p>OPEN_PROJECT_AUTO_SCHEMATIC</p> <p>Sets a default for all projects. When TRUE, an empty Schematic window is automatically displayed when you create a project.</p> <p>Example: OPEN_PROJECT_AUTO_SCHEMATIC = TRUE</p>	TRUE, FALSE
<p>OPEN_PROJECT_AUTO_LAYOUT</p> <p>Sets a default for all projects. When TRUE, an empty Layout window is automatically displayed when you create a project.</p> <p>Example: OPEN_PROJECT_AUTO_LAYOUT = TRUE</p>	TRUE, FALSE
<p>SCHEMATIC_ANALOG_DEFAULT_PALETTE</p> <p>Sets the specified palette as the default palette for Analog/RF designs in the Schematic window.</p> <p>Example: SCHEMATIC_ANALOG_DEFAULT_PALETTE=Devices-MOS</p>	Standard or custom palette name, exactly as it appears in program.

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
<p>LAYOUT_ANALOG_DEFAULT_PALETTE Sets the specified palette as the default palette for Analog/RF designs in the Layout window. Example: LAYOUT_ANALOG_DEFAULT_PALETTE=TLines-Stripline</p>	<p>Standard or custom palette name, exactly as it appears in program.</p>
<p>SCHEMATIC_DSP_DEFAULT_PALETTE Sets the specified palette as the default palette for DSP designs in the Schematic window. Example: SCHEMATIC_DSP_DEFAULT_PALETTE=Timed Linear</p>	<p>Standard or custom palette name, exactly as it appears in program.</p>
<p>LAYOUT_DSP_DEFAULT_PALETTE Sets the specified palette as the default palette for DSP designs in the Layout window. Example: LAYOUT_DSP_DEFAULT_PALETTE=Block Text Fonts</p>	<p>Standard or custom palette name, exactly as it appears in program.</p>
<p>OPEN_DDS_AFTER_SIM Establishes a default state for whether or not to automatically open a Data Display window upon completion of a simulation. This setting can be overridden for any given simulation using the Simulation Setup dialog box.</p> <p>Note: If this variable is not defined, or is defined but set to something other than TRUE or FALSE, the initial state for Analog/RF designs is TRUE and for DSP designs is FALSE.</p> <p>Example: OPEN_DDS_AFTER_SIM=TRUE</p>	<p>TRUE or FALSE</p>
<p>CONFIG_EXPLORER_VARS Controls which variables are listed when you launch the Configuration Explorer. This list is delimited by semicolons. Each variable name is preceded by one of the following designations, as well as an underscore character: E_ For <i>Environment</i> variables Example: CONFIG_EXPLORER_VARS=E_HPEESOF_DIR C_ For <i>Configuration</i> variables Example: CONFIG_EXPLORER_VARS=C_SITE_AEL S_ For <i>System</i> variables (PC only) Example: CONFIG_EXPLORER_VARS=S_TEMP</p>	<p>E_<VariableName>, C_<VariableName>, S_<VariableName></p>
<p>CONFIG_EXPLORER_CMP_VARS Controls whether or not the Configuration Explorer evaluates the value of every variable and compares it against the current value in the environment. If the setting of a given variable is the same as the current value in the environment, it is preceded by a green dot; if the values are different, it is preceded by a yellow dot. (Note: Setting this variable to <i>yes</i> is helpful for troubleshooting configuration problems but takes your system considerable time to perform the evaluations.) The default value is <i>no</i>.</p> <p>Example: CONFIG_EXPLORER_CMP_VARS=yes</p>	<p>Yes or No</p>

Table 1-5. Variables in the Files *de.cfg* and *de_sim.cfg* (continued)

Variable, description, and example	Value(s)
MAKELIB_USER_AEL_PATH Complete path for .ael files (associated with .dsn files) added to a user library using the custom library tool. Multiple libraries can be defined by specifying a semicolon-delimited list. Example: MAKELIB_USER_AEL_PATH={\$HOME}/hpeesof/circuit/ael	Path
MAKELIB_USER_DSN_PATH Complete path for .dsn files added to a user library using the custom library tool. Multiple libraries can be defined by specifying a semicolon-delimited list. Example: MAKELIB_USER_DSN_PATH={\$HOME}/hpeesof/circuit/networks	Path
MAKELIB_USER_DATA_FILES Complete path for data files referenced by designs added to a user library using the custom library tool. Example: MAKELIB_USER_DATA_FILES={\$HOME}/hpeesof/datafiles	Path
MAKELIB_USER_LIBRARY_NAMES The library name(s) that are used if no Library Name was specified in the Design Parameters dialog box. Multiple libraries can be defined by specifying a semicolon-delimited list. The members of this list must have a one-to-one correspondence with the lists set for the three aforementioned variables. Example: MAKELIB_USER_LIBRARY_NAMES=User Library	Default or custom library name
MAKELIB_SITE_AEL_PATH Complete path for .ael files (associated with .dsn files) added to a site library using the custom library tool. Multiple libraries can be defined by specifying a semicolon-delimited list. Example: MAKELIB_SITE_AEL_PATH={\$HPEESOF_DIR}/custom/circuit/ael	Path
MAKELIB_SITE_DSN_PATH Complete path for .dsn files added to a site library using the custom library tool. Multiple libraries can be defined by specifying a semicolon-delimited list. Example: MAKELIB_SITE_DSN_PATH={\$HPEESOF_DIR}/custom/circuit/networks	Path
MAKELIB_SITE_DATA_FILES Complete path for data files referenced by designs added to a site library using the custom library tool. Example: MAKELIB_SITE_DATA_FILES={\$HPEESOF_DIR}/custom/datafiles	Path
MAKELIB_SITE_LIBRARY_NAMES The library name(s) that are used if no Library Name was specified in the Design Parameters dialog box. Multiple libraries can be defined by specifying a semicolon-delimited list. The members of this list must have a one-to-one correspondence with the lists set for the three aforementioned variables. Example: MAKELIB_SITE_LIBRARY_NAMES=Site Library	Default or custom library name

Variables in the File *hpeesof.cfg*

All programs that are part of the ADS suite read the *hpeesof* configuration file.

Table 1-6. Variables in the File *hpeesof.cfg*

Variable, description, and example	Value(s)
HPEESOF_LM_LICENSE_FILE Complete path to FLEXlm <i>license.dat</i> file. Example: HPEESOF_LM_LICENSE_FILE = {\$HPEESOF_DIR}/license/license.dat	Filename
HPEESOF_COLOR Complete path to system-wide <i>hpeecolor</i> configuration file. Example: HPEESOF_COLOR={\$HPEESOF_DIR}/config/eecolor.cfg	Filename
HPEESOF_FILL Complete path to system-wide <i>hpeefill</i> configuration file. Example: HPEESOF_FILL={\$HPEESOF_DIR}/config/eefill.cfg	Filename
HPEESOF_FONT_FILE Directory where font index files are located. Example: HPEESOF_FONT_FILE={\$HPEESOF_DIR}/config	Directory
HPEESOF_OCCIDENTAL_FILE Complete path to specify European foreign language fonts. (For future use.) Example: HPEESOF_OCCIDENTAL_FILE= {\$HPEESOF_DIR}/config/hersh.oc	Filename
HPEESOF_ORIENTAL_FILE Complete path to specify Asian foreign language fonts. (For future use.) Example: HPEESOF_ORIENTAL_FILE={\$HPEESOF_DIR}/config/hersh.or	Filename
HPEESOF_LANGUAGE Specifies the default language. Example: HPEESOF_LANGUAGE=english	Language name

Variables in the File `dfilter.cfg`

This configuration file is used by the Digital Filter tool.

Table 1-7. Variables in the File `dfilter.cfg`

Variable, description, and example	Value(s)
DF_AEL_PATH The AEL path for Digital Filter designs. Example: <code>DF_AEL_PATH=</code> <code>./:\$HOME/hpeesof/dfilter/ael:\$HPEESOF_DIR/custom/dfilter/ael:\$HPEESOF_DIR</code> <code>/dfilter/ael</code>	Path
DFUI_DATA_FILE Data file, only for Digital Filter UI Example: <code>DFUI_DATA_FILE = \$HPEESOF_DIR/dfilter/ael/firdemoui.ael</code>	Filename
DFUI_SPEC_HEADING_COLOR Foreground color for spec heading data, only for Digital Filter UI Example: <code>DFUI_SPEC_HEADING_COLOR = BLUE</code>	Color name
DFUI_SPEC_EDITABLE_COLOR Foreground color for spec editable data, only for Digital Filter UI Example: <code>DFUI_SPEC_EDITABLE_COLOR = BLACK</code>	Color name
DFUI_SPEC_NOEDITABLE_COLOR Foreground color for spec non-editable data, only for Digital Filter UI Example: <code>DFUI_SPEC_NOEDITABLE_COLOR = GRAY</code>	Color name
DFUI_SPEC_CELL_WIDTH Column width for spec data, only for Digital Filter UI Example: <code>DFUI_SPEC_CELL_WIDTH = 7</code>	Integer
DF_DDS_IDEAL_TRACE_COLOR Trace color index to be used in hpeesof data display server Example: <code>DF_DDS_IDEAL_TRACE_COLOR = 1</code>	Integer
DF_DDS_SCALED_TRACE_COLOR Trace color index to be used in hpeesof data display server Example: <code>DF_DDS_SCALED_TRACE_COLOR = 3</code>	Integer
DF_DDS_SPEC_TRACE_COLOR Trace color index to be used in hpeesof data display server Example: <code>DF_DDS_SPEC_TRACE_COLOR = 4</code>	Integer

Variables in the File *dsynthesis.cfg*

This configuration file is used by the overall DSP Synthesis product.

Table 1-8. Variables in the File *dsynthesis.cfg*

Variable, description, and example	Value(s)
DSUI_BIND_COLOR Foreground color for spec bind data. Example: DSUI_BIND_COLOR = VIOLET_RED	Color name
DSUI_DESIGN_SPACE_BACKGROUND_COLOR Background color for spec design space data. Example: DSUI_DESIGN_SPACE_BACKGROUND_COLOR = IVORY1	Color name
DSUI_ESTIMATE_RECOMMENDED_COLOR Foreground color for spec estimate recommended data. Example: DSUI_ESTIMATE_RECOMMENDED_COLOR = BLUE	Color name
DSUI_ESTIMATE_NON_RECOMMENDED_COLOR Foreground color for spec estimate non recommended data. Example: DSUI_ESTIMATE_NON_RECOMMENDED_COLOR = BLACK	Color name
DSUI_ESTIMATE_BACKGROUND_COLOR Background color for spec estimate data. Example: DSUI_ESTIMATE_BACKGROUND_COLOR = IVORY1	Color name
DSUI_SCHEDULE_COLOR Foreground color for spec schedule data. Example: DSUI_SCHEDULE_COLOR = GREEN	Color name
DSUI_SPEC_HEADING_COLOR Foreground color for spec heading data. Example: DSUI_SPEC_HEADING_COLOR = BLUE	Color name
DSUI_SPEC_EDITABLE_COLOR Foreground color for spec editable data. Example: DSUI_SPEC_EDITABLE_COLOR = BLACK	Color name
DSUI_SPEC_NOEDITABLE_COLOR Foreground color for spec non-editable data. Example: DSUI_SPEC_NOEDITABLE_COLOR = GRAY	Color name
DSUI_SYNTHESIS_RECOMMENDED_COLOR Foreground color for spec synthesis recommended data. Example: DSUI_SYNTHESIS_RECOMMENDED_COLOR = RED	Color name
DSUI_SYNTHESIS_NON_RECOMMENDED_COLOR Foreground color for spec synthesis non recommended data. Example: DSUI_SYNTHESIS_NON_RECOMMENDED_COLOR = MAGENTA	Color name

Table 1-8. Variables in the File *dsynthesis.cfg* (continued)

Variable, description, and example	Value(s)
DSUI_SYNTHESIS_BACKGROUND_COLOR Background color for spec synthesis data. Example: DSUI_SYNTHESIS_BACKGROUND_COLOR = IVORY1	Color name
DSPDB_AEL_FILES AEL files for the DSP database (<i>blank</i> is allowed). Example: DSPDB_AEL_FILES =	Path
DSPDB_FILES Defines the mapping libraries which will be loaded by the DSP synthesis and HDL generator tool suite. The HPTolemyLib mapping file must be loaded at all times and should not be removed from this list; it should remain first in the list. Other mapping libraries can be removed if they are not needed for the user's designs. Example: DSPDB_FILES =HPTolemyLib:HPLib:HPLib-Std_Logic_Arith:HPLib-DesignWare:LSI500k-HPLib:LSI500k-HPLib-Std_Logic_Arith:LSI500k-HPLib-DesignWare:Xilinx4000e-HPLib:Xilinx4000e-HPLib-Std_Logic_Arith:Xilinx4000e-HPLib-DesignWare	Mapping Library names
DSUI_DEFAULT_HDL_LANG Default language for HDL generation dialogs. Example: DSUI_DEFAULT_HDL_LANG=VHDL	Verilog or VHDL
DSUI_DEFAULT_RTL_LANG Default language for RTL generation dialogs Example: DSUI_DEFAULT_RTL_LANG=VHDL	Verilog or VHDL
HDLGEN_VHDL_GEN_UTIL_FILES Designates the Utility Files for numeric standard based VHDL. Example: HDLGEN_VHDL_GEN_UTIL_FILES = hp_arithutils.vhd:hp_utils.vhd	Filenames
HDLGEN_VHDL_SYN_UTIL_FILES Designates the Utility Files for standard logic arithmetic based VHDL. Example: HDLGEN_VHDL_SYN_UTIL_FILES = hp_arithutils_syn.vhd:hp_utils_syn.vhd:hplib_syn.vhd	Filenames

Variables in the File *eeapi.cfg*

This configuration file is used by the layered Application Programming Interface (API).

Table 1-9. Variables in the File *eeapi.cfg*

Variable, description, and example	Value(s)
API_BALLOON_HELP_TIMEOUT Time taken for balloon help to appear when icon is selected. Example: API_BALLOON_HELP_TIMEOUT = 600	Time in milliseconds
API_BALLOON_HELP Sets whether balloon help exists or not. Example: API_BALLOON_HELP = TRUE	TRUE or FALSE
API_BALLOON_HELP_DISMISSAL_TIMEOUT Time taken for balloon help to disappear when icon is selected. Example: API_BALLOON_HELP_DISMISSAL_TIMEOUT = 6000	Time in milliseconds
API_DOUBLE_CLICKS_TIMEOUT The time between two clicks required for them to be recognized as a unit, or double-click. Example: API_DOUBLE_CLICKS_TIMEOUT = 150	Time in milliseconds
API_NO_TEAR_OFF_MENU Turn on/off Motif's tear-off menu Example: API_NO_TEAR_OFF_MENU = 0	0=allow tear-off menu 1= do not allow tear-off menu

Variables in the File `hpads.cfg`

This configuration file is used by PTOLEMY for locating user-compiled models and the location of MATLAB software.

Table 1-10. Variables in the File `hpads.cfg`

Variable, description, and example	Value(s)
HPTOLEMY_MOD_PATH Controls where Agilent Ptolemy will look for models. You must reference the HPTOLEMY_MODEL_PATH variable here so that user-compiled models are found. If ADS is started with the -b option, then \$HPEESOF_DIR/hptolemy/beta will be appended to load beta models. Example: HPTOLEMY_MOD_PATH=hptolemy:\$HOME/hpeesof/hptolemy:\$HPTOLEMY_MODEL_PATH:\$HPEESOF_DIR/hptolemy	Path
MATLAB The MATLAB variable should point to the root of your Matlab 5 installation (directories bin and extern should be there). Example: MATLAB=/usr/local/matlab	Directory
MATLABCMD If the command to invoke Matlab is not <code>matlab</code> , you'll need to set the MATLABCMD variable. See the documentation for engOpen in the Matlab Application Programming Interface Guide for more details on what you can do here. This variable is ignored on Windows. Example: MATLABCMD="matlab -c licensefile"	Command

Variables in the File `hpeesofbrowser.cfg`

This configuration file is used by the Browse and Search dialog box.

Table 1-11. Variables in the File `hpeesofbrowser.cfg`

Variable, description, and example	Value(s)
HPEESOF_BROWSER_WINDOW_X_LOCATION The default X coordinate of the Component Library window relative to the upper left corner of the screen. Example: HPEESOF_BROWSER_WINDOW_X_LOCATION = 100	Integer
HPEESOF_BROWSER_WINDOW_Y_LOCATION The default Y coordinate of the Component Library window relative to the upper left corner of the screen. Example: HPEESOF_BROWSER_WINDOW_Y_LOCATION = 100	Integer
HPEESOF_BROWSER_WINDOW_WIDTH The default width of the Component Library window. Example: HPEESOF_BROWSER_WINDOW_WIDTH = -1	Integer

Table 1-11. Variables in the File *hpeesofbrowser.cfg* (continued)

Variable, description, and example	Value(s)
<p>HPEESOF_BROWSER_WINDOW_HEIGHT The default height of the Component Library window. Example: HPEESOF_BROWSER_WINDOW_HEIGHT = -1</p>	Integer
<p>HPEESOF_BROWSER_TIME_OUT The default time-out for warning/information messages. Example: HPEESOF_BROWSER_TIME_OUT = 10000</p>	Time in milliseconds
<p>HPEESOF_BROWSER_AEL_PATH Path for the browser AEL files. Example: HPEESOF_BROWSER_AEL_PATH = \$HOME/hpeesof/hpeesofbrowser/ael:\$HPEESOF_DIR/hpeesofbrowser/ael</p>	Path
<p>HPEESOF_BROWSER_AEL Top level AEL file Example: HPEESOF_BROWSER_AEL = lbb_main</p>	Filename
<p>HPEESOF_BROWSER_PATH Path for the control, records and Device Libraries Binary File. Path can only have System variables. Example: HPEESOF_BROWSER_PATH = \$HOME/hpeesof/hpeesofbrowser/records;\$COMPL_DIR/ComponentLibs/records</p>	Path
<p>HPTOLEMY_BROWSER_PATH Agilent Ptolemy related control, record and IDF files. All the control files and the IDF files in these directories are read to create the Agilent Ptolemy related libraries and sub-libraries Example: HPTOLEMY_BROWSER_PATH = \$HOME/hpeesof/hptolemy/records;\$HPEESOF_DIR/hpeesof/hptolemy/records;\$HPTOLEMY_MODEL_UI</p>	Path
<p>HPEESOF_BROWSER_CONTROL_FILE Device Library Control File: defines the device libraries to be included in the Browser Tree. Example: HPEESOF_BROWSER_CONTROL_FILE = hpeesof_lib.ctl</p>	Filename
<p>HPEESOF_BROWSER_DEFAULT_BITMAP Main Group Bitmap File in the tree nodes (<i>blank</i> is allowed). Example: HPEESOF_BROWSER_DEFAULT_BITMAP =</p>	Bitmap filename
<p>HPEESOF_BROWSER_DEVICE_LIBRARY_FILE Device Libraries Item definition binary file. Example: HPEESOF_BROWSER_DEVICE_LIBRARY_FILE = hpeesof_lib.idf</p>	Filename
<p>HPEESOF_BROWSER_LIBRARIES_FIELD_WIDTH The default width of the component library tree structure. Example: HPEESOF_BROWSER_LIBRARIES_FIELD_WIDTH = 40</p>	Integer

Table 1-11. Variables in the File *hpeesofbrowser.cfg* (continued)

Variable, description, and example	Value(s)
HPEESOF_BROWSER_PARTS_DISPLAY_WIDTH The default width of the Component column. (This column is always visible.) Example: HPEESOF_BROWSER_PARTS_DISPLAY_WIDTH = 50	Integer
HPEESOF_BROWSER_FIELD_NUM_ROWS The default number of rows of components listed for any selected sub-library. Example: HPEESOF_BROWSER_FIELD_NUM_ROWS = 25	Integer
HPEESOF_BROWSER_COMPONENT_DESCRIPTION_VISIBLE The default visibility status of the Description column. Example: HPEESOF_BROWSER_COMPONENT_DESCRIPTION_VISIBLE = 1	0 = False 1= True
HPEESOF_BROWSER_PARTS_DESCRIPTION_WIDTH The default width of the Description column. Example: HPEESOF_BROWSER_PARTS_DESCRIPTION_WIDTH = 20	Integer
HPEESOF_BROWSER_COMPONENT_VENDOR_VISIBLE The default visibility status of the Vendor column. Example: HPEESOF_BROWSER_COMPONENT_VENDOR_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_VENDOR_WIDTH The default width of the Vendor column. Example: HPEESOF_BROWSER_PARTS_VENDOR_WIDTH = 10	Integer
HPEESOF_BROWSER_COMPONENT_LIBNAME_VISIBLE The default visibility status of the Library column. Example: HPEESOF_BROWSER_COMPONENT_LIBNAME_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_LIBNAME_WIDTH The default width of the Library column. Example: HPEESOF_BROWSER_PARTS_LIBNAME_WIDTH = 20	Integer
HPEESOF_BROWSER_COMPONENT_PLACEMENT_VISIBLE The default visibility status of the Placement column. Example: HPEESOF_BROWSER_COMPONENT_PLACEMENT_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_PLACEMENT_WIDTH The default width of the Placement column. Example: HPEESOF_BROWSER_PARTS_PLACEMENT_WIDTH = 15	Integer
HPEESOF_BROWSER_COMPONENT_WEBSITE_VISIBLE The default visibility status of the Website column. Example: HPEESOF_BROWSER_COMPONENT_WEBSITE_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_WEBSITE_WIDTH The default width of the Website column. Example: HPEESOF_BROWSER_PARTS_WEBSITE_WIDTH = 20	Integer

Table 1-11. Variables in the File *hpeesofbrowser.cfg* (continued)

Variable, description, and example	Value(s)
HPEESOF_BROWSER_COMPONENT_AVAILABILITY_VISIBLE The default visibility status of the Availability column. Example: HPEESOF_BROWSER_COMPONENT_AVAILABILITY_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_AVAILABILITY_WIDTH The default width of the Availability column. Example: HPEESOF_BROWSER_PARTS_AVAILABILITY_WIDTH = 10	Integer
HPEESOF_BROWSER_COMPONENT_LICENSE_VISIBLE The default visibility status of the License column. Example: HPEESOF_BROWSER_COMPONENT_LICENSE_VISIBLE = 0	0 = False 1= True
HPEESOF_BROWSER_PARTS_LICENSE_WIDTH The default width of the License column. Example: HPEESOF_BROWSER_PARTS_LICENSE_WIDTH = 10	Integer

Variables in the File *hpeesofdds.cfg*

This configuration file is used to configure data displays.

Table 1-12. Variables in the File *hpeesofdds.cfg*

Variable, description, and example	Value(s)
AEL_PATH Top level AEL file Example: AEL_PATH=\$HOME/hpeesof/instrio/ael/:\$HPEESOF_DIR/custom/instrio/ael /:\$HPEESOF_DIR/instrio/ael	Path
DDS_DEFAULTS_FILE Users default defaults file. This file gets read at start-up and written at shutdown with the current set of defaults. Example: DDS_DEFAULTS_FILE = \$HOME/hpeesof/config/ddsdefaults.ael	Filename
DDS_WIN_FOREGROUND Window foreground color Example: DDS_WIN_FOREGROUND = 0	Integer
DDS_WIN_BACKGROUND Window background color Example: DDS_WIN_BACKGROUND = 8	Integer
DDS_WIN_HEIGHT Window height Example: DDS_WIN_HEIGHT = 510	Integer

Table 1-12. Variables in the File *hpeesofdds.cfg* (continued)

Variable, description, and example	Value(s)
DDS_WIN_WIDTH Window width Example: DDS_WIN_WIDTH = 655	Integer
DDS_HIGHLITE_LINE_TYPE System-wide highlight line type Example: DDS_HIGHLITE_LINE_TYPE = 0 [= solid]	Integer
DDS_HIGHLITE_LINE_WIDTH System-wide highlight line width Example: DDS_HIGHLITE_LINE_WIDTH = 56 [= 4 points]	Integer
DDS_HIGHLITE_LINE_COLOR System-wide highlight line color Example: DDS_HIGHLITE_LINE_COLOR = 0 [= black]	Integer
DDS_DRAWING_UNITS_PER_INCH Default graphical object characteristic: Internal database resolution Example: DDS_DRAWING_UNITS_PER_INCH = 1000	Integer
DDS_PAGE_WIDTH Default graphical object characteristic: Plotting area width Example: DDS_PAGE_WIDTH = 33000	Integer
DDS_PAGE_HEIGHT Default graphical object characteristic: Plotting area is height Example: DDS_PAGE_HEIGHT = 21000	Integer
DDS_DEFAULT_LINE_WIDTH Default graphical object characteristic: Plotting line width Example: DDS_DEFAULT_LINE_WIDTH = 20	Integer
DDS_LEFT_CLICK_SENSITIVITY A left-mouse-button press-and-hold drag must move this many pixels or else it is considered a left click Example: DDS_LEFT_CLICK_SENSITIVITY= 2	Integer
DDS_DEFAULT_PLOT_WIDTH Default width of new plots Example: DDS_DEFAULT_PLOT_WIDTH = 3750	Integer
DDS_DEFAULT_PLOT_HEIGHT Default height of new plots Example: DDS_DEFAULT_PLOT_HEIGHT = 2318	Integer

Table 1-12. Variables in the File *hpeesofdds.cfg* (continued)

Variable, description, and example	Value(s)
DDS_DEFAULT_FONT Default font Example: DDS_DEFAULT_FONT = HersheyRomanNarrow	Font size, in points
DDS_NUMBER_OF_TRACE_COLORS Number of trace colors Example: DDS_NUMBER_OF_TRACE_COLORS = 16	Integer
DDS_TRACEn_COLOR n th color, where $n = 1$ to DDS_NUMBER_OF_TRACE_COLORS Example: DDS_TRACE16_COLOR = 90	Integer

Variables in the File *hpeesofhelp.cfg*

This configuration file is used to locate the online documentation.

Table 1-13. Variables in the File *hpeesofhelp.cfg*

Variable, description, and example	Value(s)
TOPLEVEL_DOC Top level Index document location. Example: TOPLEVEL_DOC=\$HPEESOF_DIR/doc	Directory

Variables in the File *hpeesofinstrio.cfg*

This configuration file is used by the Instrument Server.

Table 1-14. Variables in the File *hpeesofinstrio.cfg*

Variable, description, and example	Value(s)
AEL_PATH Top level AEL file Example: AEL_PATH=\$HOME/hpeesof/instrio/ael/:\$HPEESOF_DIR/custom/instrio/ael /:\$HPEESOF_DIR/instrio/ael:/	Path

Variables in the File *momentum.cfg*

This configuration file is used by Momentum.

Table 1-15. Variables in the File *momentum.cfg*

Variable, description, and example	Value(s)
SUPL_GF_DATABASE Location of supplied Momentum Substrate files. Example: SUPL_GF_DATABASE = {\$HPEESOF_DIR}/momentum/lib	Directory
SITE_GF_DATABASE Location of user Momentum Substrate files. Example: SITE_GF_DATABASE = {\$HOME}	Directory
LOCL_GF_DATABASE Location of project Momentum Substrate files. Example: LOCL_GF_DATABASE = ./	Directory
MOM_SLM_PATH Path to search for Momentum Substrate file that is saved with design files. Example: MOM_SLM_PATH = ./networks:{\$SUPL_GF_DATABASE}	Path
MOMDDS_FILE Location of default template file for Momentum Data Display. Example: MOMDDS_FILE= {\$HPEESOF_DIR}/sess/ael/momdds.ael	Filename

Variables in the File *smg_ui.cfg*

This configuration file is used by the SPICE Model Generator.

Table 1-16. Variables in the File *smg_ui.cfg*

Variable, description, and example	Value(s)
SMG_AEL_PATH Top level AEL file Example: SMG_AEL_PATH=={\$HOME}/hpeesof/smg/ael:{\$HPEESOF_DIR}/smg/ael:{\$HPEESOF_DIR}/ael_smg_ui	Path

Variables in the File hpeesofsim.cfg

This configuration file is used by the simulators.

Table 1-17. Variables in the File *hpeesofsim.cfg*

Variable, description, and example	Value(s)
RF_CW_PRIORITY Controls which simulator license takes precedence, Circuit_Linear or System_Freq. If not set (the default) or set to 'y' the System_Freq license takes precedence. If set to 'n' the Circuit_Linear license takes precedence. RF_CW_PRIORITY = n	n (no), y (yes)

Chapter 2: Customization Examples

The topics listed below represent some of the ways in which you can customize the Advanced Design System environment.

- “Creating a Custom Menu” on page 2-1
- “Creating Custom Libraries” on page 2-3
- “Advanced Library Customization Techniques” on page 2-10
- “Modifying the List of Available Symbol Names” on page 2-15
- “Adding Online Help for User-Defined Items” on page 2-17

Many of the topics presented here refer to the variables `%HOME%` and `%HPEESOF_DIR%`. On the PC these variables have the following meaning:

- `%HOME%`—the path you specified as the *Home* folder during installation (`C:\users\default` by default)
- `%HPEESOF_DIR%`—the path you specified as your *Program* folder during installation (`C:\ADS2002` by default)

Creating a Custom Menu

You can add custom menus to the ADS Main window and Schematic and Layout windows by modifying the `.ael` file `usermenu.ael`.

Note This procedure applies only to user-level custom menus. The number of slots available for custom menus is limited; you must use care so that you do not wipe out another user’s menu.

To add a custom menu:

1. Under `$HOME/hpeesof`, create the additional directories `/de/ael`.
2. Copy the file `usermenu.ael` from `$HPEESOF_DIR/de/ael` to the new directory `$HOME/hpeesof/de/ael` and ensure you have write permission for this file.
3. Using any text editor, open the file `usermenu.ael`.
4. The file `usermenu.ael` contains sample code that can be uncommented to help you learn how to use the functions available for defining user menus. Before

proceeding, review the descriptions of the following functions in the AEL manual: *check_user_menu()*, *set_user_menu_label()*, and *add_menu()*. After you understand how to use the sample code, replace it with your own code.

The function *app_add_user_menus()* is the function in which you will add your custom code. The sample code provided includes the following function calls:

```
// to get the internal name of the first free user menu
decl freeMenuName = app_find_empty_user_menu(winType);

// uses the menu's internal name to set the label on a free user menu
set_user_menu_label("my first menu", freeMenuName);

// uses the menu's internal name to add a menu pick on a free user menu
add_menu( "menuPickA", "my_menu_cb", freeMenuName );
```

Uncomment these lines in your file by removing the two slashes at the beginning of the three executable lines, as shown above. Do not delete the slashes from the comment lines.

5. Save the file.

Note If this code is used as is, it will initialize the first user menu, unless your system already has user menus defined. If your system does not have any user menus defined by other applications, and you want to see what would happen if it did, you can uncomment the test function *prefill_menus()* and the call to it, both provided in *usermenu.ael*. This will initialize the first two user menus, so that a call to *app_find_empty_user_menu()* will return "User3" as the first available user menu.

6. Restart ADS to see the change.

Creating Custom Libraries

As with other aspects of ADS customization, you can create custom libraries at several levels: for site-wide use, for all your projects, and for individual projects. The procedures are presented in three different groupings, based on the following use models:

- Assisted customization for the average user, done through the user interface (*Tools > Custom Library*), for a small number of designs
- Manual customization, typically done by a librarian, using the recommended directory structure, for a large number of designs
- Manual customization, typically done by a librarian, using directories other than the recommended directories (*Advanced*)

If creating libraries via the user interface method, before defining a Library Name—through the Design Parameters dialog box—for any given design, please review the following guidelines, as assigning a library name in this manner is not always recommended:

- Basic—Do *not* use the Library Name field in the Design Parameters dialog box. Create a library via the user interface method. Each process (*Create > User Part* or *Create > Site Part*) results in a single library. By default the library name is *User Library* or *Site Library*. Note that because all designs are in a single library, the entire library must be loaded or not loaded.
- Intermediate—Do use the Library Name field in the Design Parameters dialog box to categorize designs so that they appear in the library browser under those categories (for example, *My Resistors*, *My Capacitors*). Create a library via the user interface method. Each process (*Create > User Part* or *Create > Site Part*) results in a single library. Note that because all designs are in a single library, the entire library must be loaded or not loaded.
- Advanced—Do *not* use the Library Name field in the Design Parameters dialog box. Set the *MAKELIB_** configuration variables to define multiple libraries. Create libraries via the user interface method (*Create > User Part* or *Create > Site Part*). Each time you add a part you will be prompted to select one of your pre-defined libraries. In this manner, the loading of individual libraries can be enabled or disabled.

Assisted Customization

The procedures in this section are geared toward individuals working with a small number of designs, because this method is based on adding designs to a library one at a time. Once the libraries have been created, either restart ADS to see the changes or use the *Configuration Explorer* to explicitly read the modified *de_sim.cfg* file for the current session.

Note Because this method is based on *copying* designs from their source location to a library location, if the source design is subsequently modified, you must repeat the process so that the design gets copied to the library again.

Keep in mind the following limitations when creating libraries in this manner:

- Only the current design is copied to the library when using the *Create > User Part* or *Create > Site Part* commands. If the design is hierarchical, you must open or push into each subnetwork and repeat the process.

Hint To easily see the designs comprising a hierarchical design, from the ADS Main window, choose *View > Design Hierarchies*.

- Some files required to successfully recreate the design in its library location are not copied to the library. Those files that are not automatically copied to the library must be explicitly copied to ensure the integrity of the design:

Files Copied to the Library
<design>.dsn—The design file
<design>.ael—The component definition AEL file
<design>.atf—The compiled component definition AEL file
<design>_art.ael—The layout artwork AEL file generated by the Graphical Cell Compiler, where applicable
<design>_art.atf—The compiled layout artwork (GCC) AEL file, where applicable
<datafile>.ext—Data files referenced by current design [†]

Files Not Copied to the Library

Custom symbols stored in a separate .dsn file. The workaround is to open the .dsn file containing the symbol and add it to the library using the appropriate create part command.

Referenced fixed artwork stored in a separate .dsn file. The workaround is to open the .dsn file containing the artwork and add it to the library using the appropriate create part command.

Artwork macros stored in a separate file, other than those created by the Graphical Cell Compiler. The workaround is to manually copy the .ael file containing the artwork to the library (*\$HPEESOF_DIR/custom/circuit/ael* for Site or *\$HOME/hpeesof/circuit/ael* for User).

†

- If a design being copied to the library references a data file that is not in the project's /data directory—but is specified using a path, absolute or relative—the file is not copied. Thus for site libraries, you must make sure all users on all workstations can access the file(s) in this location.
- If your design contains a DAC that references a discrete MDIF file that lists multiple data files, only the directly referenced file will be copied to the library. The individual files listed in the MDIF file must be manually copied to the library (*\$HPEESOF_DIR/custom/datafiles* for Site or *\$HOME/hpeesof/datafiles* for User).

Site-Wide Customization (Assisted)

Copying designs to and removing designs from the site library requires write permission in that directory (by default, *\$HPEESOF_DIR/custom/circuit*).

To copy a specific design to a site library making it available for all users:

1. From the design you want to copy to the site library, choose **Tools > Custom Library > Create > Site Part**.
2. Repeat this process for each design you want to be part of a site library.

To remove a specific design from a library:

Choose **Tools > Custom Library > Delete > Site Part** and select the design you want to remove from the library.

Hint If the design you select to remove from the library references any data files, you are prompted to choose to keep or remove the data files.

To prevent an entire library of designs from being loaded when you start ADS:

Choose **Tools > Custom Library > Disable > Site Library**, and where applicable, select the library.

To enable the loading of a library that you previously disabled:

Choose **Tools > Custom Library > Enable > Site Library**, and where applicable, select the library.

Notes:

- When you use *Create > Site Part*, the design and related files are copied to a number of default directories under *\$HPEESOF_DIR/custom/circuit*.
- When you use *Enable > Site Library*, the appropriate variable (in the file *\$HPEESOF_DIR/custom/config/de_sim.cfg*) is updated so that the library is loaded the next time you start ADS. This command is used to enable libraries previously disabled.
- Site libraries created using the default procedure just presented will always be loaded by all users. To allow users the freedom to enable/disable loading of these libraries, the following variables can be set in *\$HPEESOF_DIR/custom/config/de_sim.cfg*:

```
MAKELIB_USER_AEL_PATH
MAKELIB_USER_DSN_PATH
MAKELIB_USER_LIBRARY_NAMES *See note below
MAKELIB_USER_DATA_FILES
```

*Note: This variable is only used if no Library Name was specified in the Design Parameters dialog box, for any given design. For details on using these variables, refer to [Table 1-5 in Chapter 1, Customizing the ADS Environment](#).

User Customization (Assisted)

To copy a specific design to a local library making it available for all your projects:

1. From the design you want to copy to the library, choose **Tools > Custom Library > Create > User Part**.
2. Repeat this process for each design you want to be part of a user library.

To remove a specific design from a library:

Choose **Tools > Custom Library > Delete > User Part**, and select the design you want to remove from the library.

To prevent an entire library of designs from being loaded when you start ADS:

Choose **Tools > Custom Library > Disable > User Library**, and where applicable, select the library.

Manual Customization

This section is geared toward the librarian, or individual, working with a large number of designs, and working with the recommended directory structure.

For the two manual methods, each case involves the following basic steps, though the details vary:

- Creating the directories shown, if they do not exist
- Copying the *.dsn* and *.ael* files (you want made available) to these directories
- Modifying the search paths of specific variables to look in those directories

Note Do not include line breaks of any kind when setting variables; let the lines break wherever your text editor breaks them.

When customizing for any of the scenarios described in the manual methods, wherever you see *<product_area>*, replace it with one of the following:

- *de*—if you are customizing both Analog/RF and Agilent Ptolemy libraries
- *circuit*—if you are customizing Analog/RF libraries only
- *hptolemy*—if you are customizing Agilent Ptolemy libraries only

Site-Wide Customization (Manual)

To make a specific design available for all users:

1. Copy the *.dsn* files to *\$HPEESOF_DIR/custom/<product_area>/symbols/* directory. Example:

Copy *mydesign1.dsn* and *mydesign2.dsn* to *\$HPEESOF_DIR/custom/de/symbols/*

2. Copy the *.ael* files to the *\$HPEESOF_DIR/custom/<product_area>/ael/* directory. Example:

Copy *mydesign1.ael* and *mydesign2.ael* to *\$HPEESOF_DIR/custom/de/ael/*

3. Edit or create a file *\$HPEESOF_DIR/custom/config/de_sim.cfg* and set *SITE_AEL* equal to each of the *.ael* filenames (separated by semicolons or colons). Example:

```
SITE_AEL=mydesign1;mydesign2
```

Note If you want all *.ael* files in the directory to be loaded, a shortcut is to set the *SITE_AEL* variable to the directory. For example: *SITE_AEL=\$HPEESOF_DIR/custom/de/ael/*. Be sure to include the trailing slash.

User Customization (Manual)

To make designs available for all your projects:

1. Copy the *.dsn* files to *\$HOME/hpeesof/<product_area>/symbols/* directory. Example:

Copy *mydesign1.dsn* and *mydesign2.dsn* to *\$HOME/hpeesof/de/symbols/*

2. Copy the *.ael* files to *\$HOME/hpeesof/<product_area>/ael/* directory. Example:

Copy *mydesign.ael* and *mydesign2.ael* to *\$HOME/hpeesof/de/ael/*

3. Open *\$HOME/hpeesof/config/de_sim.cfg* and set *USER_AEL* equal to each of the *.ael* file names (separated by semicolons or colons). Example:

```
USER_AEL=mydesign1;mydesign2
```

Project Customization (Manual)

To make designs available in multiple—but not all—projects:

1. Copy the *.dsn* files to *\$HOME/hpeesof/<product_area>/symbols/* directory.

Example:

Copy *mydesign1.dsn* and *mydesign2.dsn* to *\$HOME/hpeesof/de/symbols/*

2. Copy the *.ael* files to *\$HOME/hpeesof/<product_area>/ael/* directory.

Example:

Copy *mydesign.ael* and *mydesign2.ael* to *\$HOME/hpeesof/de/ael/*

3. For each project, open *<project>/de_sim.cfg* and set LOCAL_AEL to each of the *.ael* filenames (separated by semicolons or colons). Example:

```
LOCAL_AEL=mydesign1;mydesign2
```

Advanced Library Customization Techniques

We recommend storing library designs for site-wide use in the following locations:

Note The *custom* directory shown here is created during installation, but you must create the product-specific directories (beneath it) yourself.

	Analog/RF	Digital Signal Processing
.dsn files	\$HPEESOF_DIR/custom/circuit/symbols/ or \$HOME/hpeesof/circuit/symbols/	\$HPEESOF_DIR/custom/hptolemy/symbols/ or \$HOME/hpeesof/hptolemy/symbols/
.ael files	\$HPEESOF_DIR/custom/circuit/ael/ or \$HOME/hpeesof/circuit/ael/	\$HPEESOF_DIR/custom/hptolemy/ael/ or \$HOME/hpeesof/hptolemy/ael/

If for some reason you cannot or do not want to store your files in the these directories and would rather locate your files elsewhere, you can.

1. Copy the *.dsn* files to any directory. Example:

Q:\ads_designs\library1\designs

2. Copy the *.ael* files to any directory. Example:

Q:\ads_designs\library1\ael

Choose one of the following customization schemes:

- Site-Wide basis (all users at the site)
- User basis (all projects for a single user)

Note There is no way to do this for project-level customization at this time.

Site-Wide Customization (Advanced)

Edit the `$HPEESOF_DIR/custom/config/de_sim.cfg` file as follows:

1. The `SITE_AEL` variable needs to be set to load your `.ael` files. Typically you would set it to point to the directory containing your `.ael` files. Example:

```
SITE_AEL=Q:\ads_designs\library1\ael\      (Loads all .ael files in directory)
```

Be sure to include the trailing slash. Note that you cannot not use relative paths; you must use either an absolute path or an environment variable.

Alternatively, you can list specific files that you want to load. Example:

```
SITE_AEL=Q:\ads_designs\library1\ael\myfile
```

2. Set the `SITE_DSN_PATH` variable equal to the directory you copied the `.dsn` files to earlier. Example:

```
SITE_DSN_PATH=Q:\ads_designs\library1\designs\      (absolute path)      or
```

```
SITE_DSN_PATH={$ADS_DESIGN}\library1\designs\      (environment variable)
```

Important If you set the `SITE_AEL` variable (in step 1) to a path, rather than a list of individual designs, you are done; if you listed designs individually, continue with the next two steps.

Edit the `$HPEESOF_DIR/custom/config/de_sim.cfg` file as follows:

1. Note that every user will already have the variable `AEL_PATH` set in their `$HOME/hpeesof/config/de_sim.cfg` file. It will be set to one of the following variables, based on the choice they made in the setup dialog box the first time they started ADS:

```
{%ANALOG_RF_AEL_PATH}  
{%DSP_AEL_PATH}  
{%BOTH_ANALOG_RF_FIRST_AEL_PATH}  
{%BOTH_DSP_FIRST_AEL_PATH}
```

Note The setting chosen on startup can be changed at any time from within ADS from the Main Window through *Options > Advanced Design System Setup*.

The default definitions of these four variables can be found in *\$HPEESOF_DIR/config/de_sim.cfg*.

Copy all four variables from *\$HPEESOF_DIR/config/de_sim.cfg* and add them to the *\$HPEESOF_DIR/custom/config/de_sim.cfg* file.

2. Modify each variable to add the path—that you copied the *.ael* files to earlier—to the front of the existing path. Example:

```
ANALOG_RF_AEL_PATH=Q:\ads_designs\library1\ael:{%HOME_CIRC...
DSP_AEL_PATH=Q:\ads_designs\library1\ael:{%HOME_HPTOL...
BOTH_ANALOG_RF_FIRST_AEL_PATH=Q:\ads_designs\library1\ael:{%HOME...
BOTH_DSP_FIRST_AEL_PATH=Q:\ads_designs\library1\ael:{%HOME_HP...
```

Now, even if a user changes their setup (from within ADS) to start with a different design type—thereby updating their local *AEL_PATH* variable—the designs can still be found because the search path is provided for all cases.

Example:

The *AEL_PATH* variable that is set based on your choice of design type:

\$HOME/hpeesof/config/de_sim.cfg **→** ***AEL_PATH={%ANALOG_RF_AEL_PATH}***

The variable your *AEL_PATH* variable points to, after being copied and modified:

\$HPEESOF_DIR/custom/config/de_sim.cfg

└ ***ANALOG_RF_AEL_PATH=Q:\ads_designs\library1\ael:{%HOME_CIRCUIT_AEL...***

User Customization (Advanced)

Edit the `$HOME/hpeesof/config/de_sim.cfg` file as follows:

1. Set the `USER_AEL` variable equal to the design filenames (separated by semicolons or colons). Example:

```
USER_AEL=mydesign1;mydesign2
```

Note If you want all `.ael` files in the directory to be loaded, a shortcut is to set the `USER_AEL` variable to the directory. For example:
`USER_AEL=$HOME/hpeesof/MyLibrary/ael/`. Be sure to include the trailing slash.

2. Set the `USER_DSN_PATH` variable equal to the directory you copied the `.dsn` files to earlier. Example:

```
USER_DSN_PATH=${HOME}/hpeesof/MyLibrary/Designs/
```

Note If you set the `USER_AEL` variable (in step 1) to a path, rather than a list of individual designs, you are done; if you listed designs individually, continue with the next step.

3. Add the path—you copied your `.ael` files to earlier—to the front of the current `AEL_PATH` definition. Example:

```
AEL_PATH=${HOME}/hpeesof/MyLibrary/ael;{%BOTH_ANALOG_RF_FIRST_AEL_PATH}
```

Note If you have customized the `AEL_PATH` path variable and you change the design type startup setting through *Options > Advanced Design System Setup*, in the Main window and click OK, this variable will be overwritten and you will lose your customized path. If you need to change your setup, you will either have to change it manually in your `de_sim.cfg` file, or re-establish your customized path after changing it through the dialog box.

Example

The steps that follow can be used as a shortcut for creating a library from a single project.

1. Create a project, such as *library_prj*.
2. Place in *library_prj/networks* only those parts you are interested in using directly.
3. Modify the variables `USER_AEL` and `USER_DSN_PATH`, shown in the example below, in the file `$HOME/hpeesof/config/de_sim.cfg`. Example:

```
USER_AEL = /eesof/users/kclarke/library_prj/networks/  
USER_DSN_PATH = /eesof/users/kclarke/library_prj/networks/
```


Modifying the List of Available Symbol Names

To assign a symbol to a schematic, you must supply a *Symbol Name* in the Design Definitions dialog box (*File > Design Parameters*). You can type a symbol name or you can select one from the default list of commonly used symbols. You can modify this list and add names of other supplied symbols as well as the names of custom symbols you have created.

To modify the list of names, you modify the AEL function *set_design_choices* in the file shown next, in accordance with its intended use (*circuit* vs. *hptolemy*):

- */circuit/ael/geminidsndef.ael*
- */de/ael/spdsndef.ael*

To modify this file for individual use, modify a local copy of it; to modify this file for site use, modify a copy of this file on a shared drive. We recommend that you maintain the established directory structure for storing and referring to *.ael* files. Note that the *custom* directory shown below is created during installation, but you must create the product-specific directories (beneath it) yourself.

Use this directory structure on UNIX:

UNIX	
Individual Use	<i>\$HOME/hpeesof/circuit/ael</i> <i>\$HOME/hpeesof/hptolemy/ael</i>
Site Use	<i>\$HPEESOF_DIR/custom/circuit/ael</i> <i>\$HPEESOF_DIR/custom/hptolemy/ael</i>

Use this directory structure on the PC:

PC*	
Individual Use	<i>%HOME%/hpeesof/circuit/ael</i> <i>%HOME%/hpeesof/hptolemy/ael</i>
Site Use	<i>%HPEESOF_DIR%/custom/circuit/ael</i> <i>%HPEESOF_DIR%/custom/hptolemy/ael</i>
* <i>%HOME%</i> represents the path you specified as the <i>Home Folder</i> during installation (<i>C:\users\default</i> by default); <i>%HPEESOF_DIR%</i> represents the path you specified as your <i>Program Folder</i> during installation (<i>C:\AdvDesSys</i>).	

Note If you use a directory other than one of those shown in the tables for storing the modified file (*gemini.ael* or *spdsndef.ael*), then you must declare the variable `USER_AEL` and provide the search path. The *custom* directory shown here is created during installation, but you must create the product-specific directories (beneath it) yourself.

To take advantage of the modified version of the *gemini.ael* or *spdsndef.ael* file—if stored in a directory other than one of the defaults:

1. Using any text editor, open the file *\$HOME/hpeesof/config/de_sim.cfg*.
2. Add the variable `USER_AEL` and set it equal to the path you have chosen for your modified file.
3. Save the file.

Once you modify *geminidsndef.ael* or *spdsndef.ael*, you must also modify the variable `USE_RELATIVE_AEL_PATHS` (regardless of the location of the modified file). This should be done in the file *\$HOME/hpeesof/config/de_sim.cfg* and you must set the variable equal to `TRUE` (`USE_RELATIVE_AEL_PATHS =TRUE`). For information on modifying search paths, refer to the section, “[Modifying Search Paths](#)” on page 10-18 in [Chapter 10, Working with Symbols](#), in the *User’s Guide*.

Once you have created the directory structure, copy the *geminidsndef.ael* (circuit) or *spdsndef.ael* (hptolemy) file from the installation directory to the newly created local */ael* directory.

To modify the list of symbol names appearing for selection:

1. Using any text editor, open the copied *geminidsndef.ael* or *spdsndef.ael* file in the local */ael* directory.
2. Locate the heading

```
/* symbol list: DMDESIGN_SYMBOL_CHOICES */
```

followed by the function

```
set_design_choices
```

Notice that the function name is followed by a list of symbol names, each surrounded by quotation marks, separated by commas (including sequential numbering and commas), all enclosed in parentheses.

3. Add the desired names to the list using the same format.

4. Save the file.

After you have completed this process, you will be able to select your custom symbol from the drop-down *Symbol Name* list (in the Design Definitions dialog box).

If the program is currently running, you need to restart the program to see these changes take effect.

Adding Online Help for User-Defined Items

When you create your own items, you can also create documentation for them and make that documentation available online. The starting point of this procedure assumes you have already created the items.

Note This procedure requires the ability to write to the installation directory.

1. Create a directory, for example *my_items*, under *\$HPEESOF_DIR/doc* for storing your new documentation files (where *\$HPEESOF_DIR* represents your complete installation path).
2. Change the write permission to this directory as needed, based on the users who will create documentation for the new items.
3. Create documentation for the new items using the HTML format (one file for each item) and move (or copy) the files to the new directory. While filenames can be any name you choose—except for the required *.htm* extension—we recommend using *<item_name>.htm*.
4. Create a text file, for example *my_items_help.txt*, containing pointers to the new items, using the following syntax (note the spaces before and after the first exclamation point and before the final exclamation point):

```
item_name ! <path>/<filename> !
```

where

item_name is the element name you used in the AEL *create_item()* function

path is the directory you created below *\$HPEESOF_DIR/doc* to store the HTML files you created (for example, *my_items*)

This file should contain one line for each new item you created. An example line is shown next.

```
PIPAD ! my_items/pipad.htm !
```

5. The file that contains the pointers for help on all components is named *hshpeesofsim_index* and can be found in *\$HPEESOF_DIR/doc*. Create a copy of this file (as a safeguard), for example, *hshpeesofsim_index.sav*.
6. Using any text editor, add the contents of the help file you created (in this example, *my_items_help.txt*), to the bottom of the *hshpeesofsim_index* file.

Optionally, on UNIX, you can use the *cat* command (instead of a text editor) and concatenate the files:

```
cat my_items_help.txt >> hshpeesofsim_index
```

The help file now contains pointers to the documentation you created.

7. Save the file (no extension) and restart ADS to verify your changes.

The following figure illustrates the directory structure for the user-defined files.

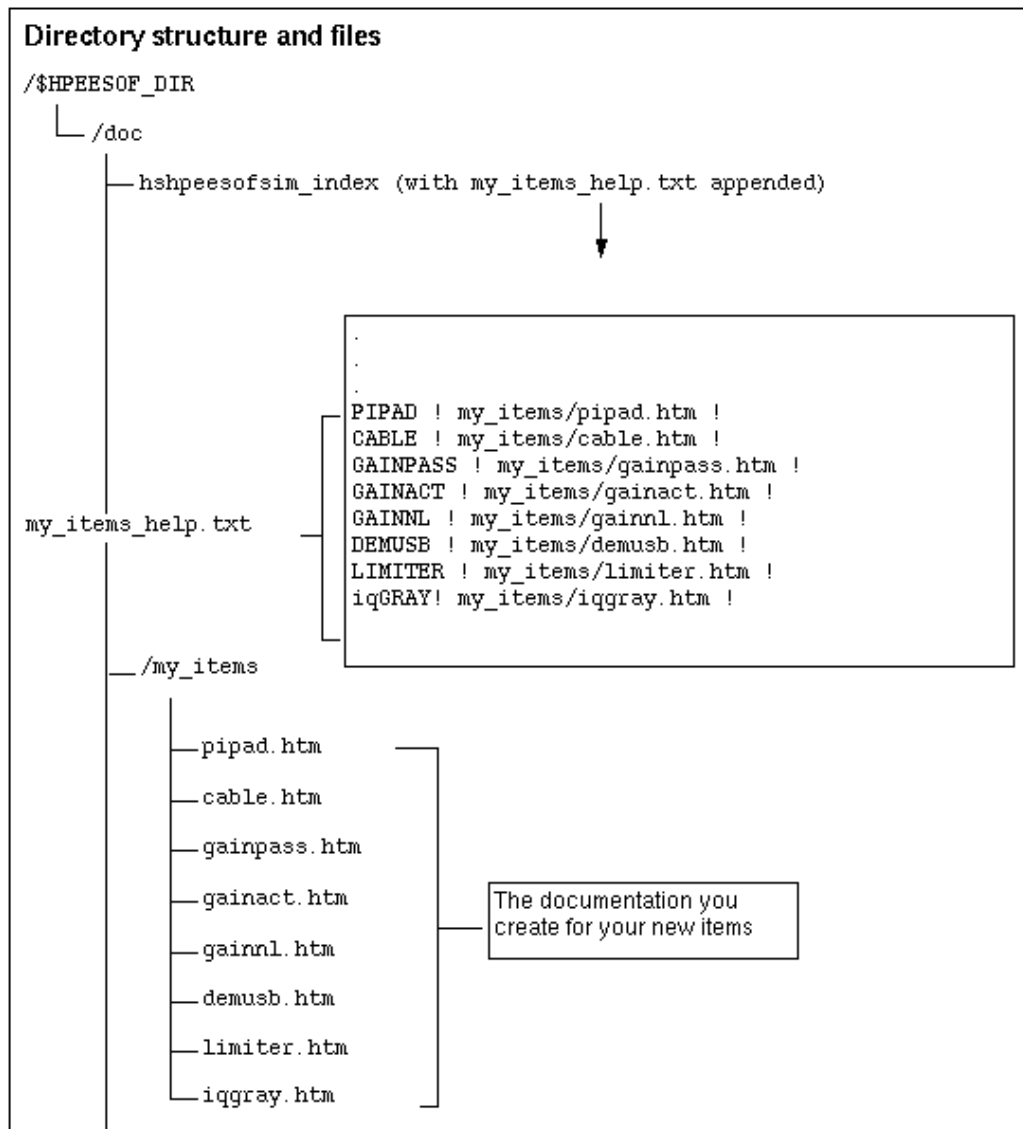


Figure 2-1. Directory Structure for User-Defined Help

Index

C

- Configuration Explorer description, 1-13
- configuration files
 - colors, 1-8
 - de.cfg, 1-15
 - de_sim.cfg (primary ADS Design Environment file), 1-15
 - dfilter.cfg (Digital Filter), 1-26
 - dsynthesis.cfg (DSP Synthesis), 1-27
 - eeapi.cfg (layered API), 1-29
 - fill patterns, 1-11
 - hpads.cfg (HP Ptolemy, MATLAB), 1-30
 - hpeesof.cfg (top-level file), 1-25
 - hpeesofbrowser.cfg (Library Browse/Search), 1-30
 - hpeesofdds.cfg (Data Display), 1-33
 - hpeesofhelp.cfg (online documentation), 1-35
 - hpeesofinstrio.cfg (instrument input/output), 1-35
 - hpeesofsim.cfg, 1-37
 - momentum.cfg (Momentum), 1-36
 - smg_ui.cfg (SPICE Model Generator), 1-36
 - viewing current settings, 1-13
- configuration variables
 - expansion, 1-6
 - file format, 1-3
 - overview, 1-2
 - syntax, 1-4

D

- design history, File menu, 1-20

E

- environment variables
 - used for customizing ADS, 1-1

L

- libraries, creating custom, 2-3

M

- menus, creating custom, 2-1

P

- project history, File menu, 1-20

S

- search paths
 - customizing
 - for data files, 1-22
 - layers, 1-16
 - preferences, 1-16
- simulation
 - remote machine name, 1-21
- symbol names
 - customizing the list of, 2-15

T

- text editor, changing default, 1-19

U

- user-defined models
 - creating online help for, 2-17

V

- variables
 - configuration
 - expansion, 1-6
 - file format, 1-3
 - overview, 1-2
 - syntax, 1-4
 - environment, 1-1

W

- window positions, changing defaults
 - Component Library, 1-30
 - Layout, 1-22
 - Main, 1-21
 - Schematic, 1-21

