DesignPad Text Editor User Manual

for the HDL Designer Series™

Software Version 2.72

June, 2011

Contractor/manufacturer is:
Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/
Send Feedback on Documentation: supportnet.mentor.com/user/feedback_form.cfm

Table of Contents

Chapter 1
About DesignPad

This chapter provides an overview of the DesignPad key features and user interface.

Introduction

DesignPad is a powerful multi-platform language-sensitive text editor designed to integrate with HDL design flows and to complement the functionality available in the main HDL Designer Series (HDS) tools.

Standard Features

DesignPad supports all normal text editing procedures including file creation, read-only or edit modes, print and HTML export. Advanced features include multiple undo or redo operations, indenting, case changing and automatic keyword completion.

Search and navigation features include find, replace, goto and bookmark commands.

Line numbers or whitespace can be optionally displayed and a separate report pane used for error reporting.

Language support includes automatic syntax highlighting, checking and comment insertion for C++, Tcl, PSL, Verilog, VHDL and SPICE. In addition, language templates are provided for fast insertion and expansion of Verilog or VHDL code constructs.

The code browser window displays the structure of the code in the active window as a hierarchical tree and allows rapid navigation through large blocks of code.

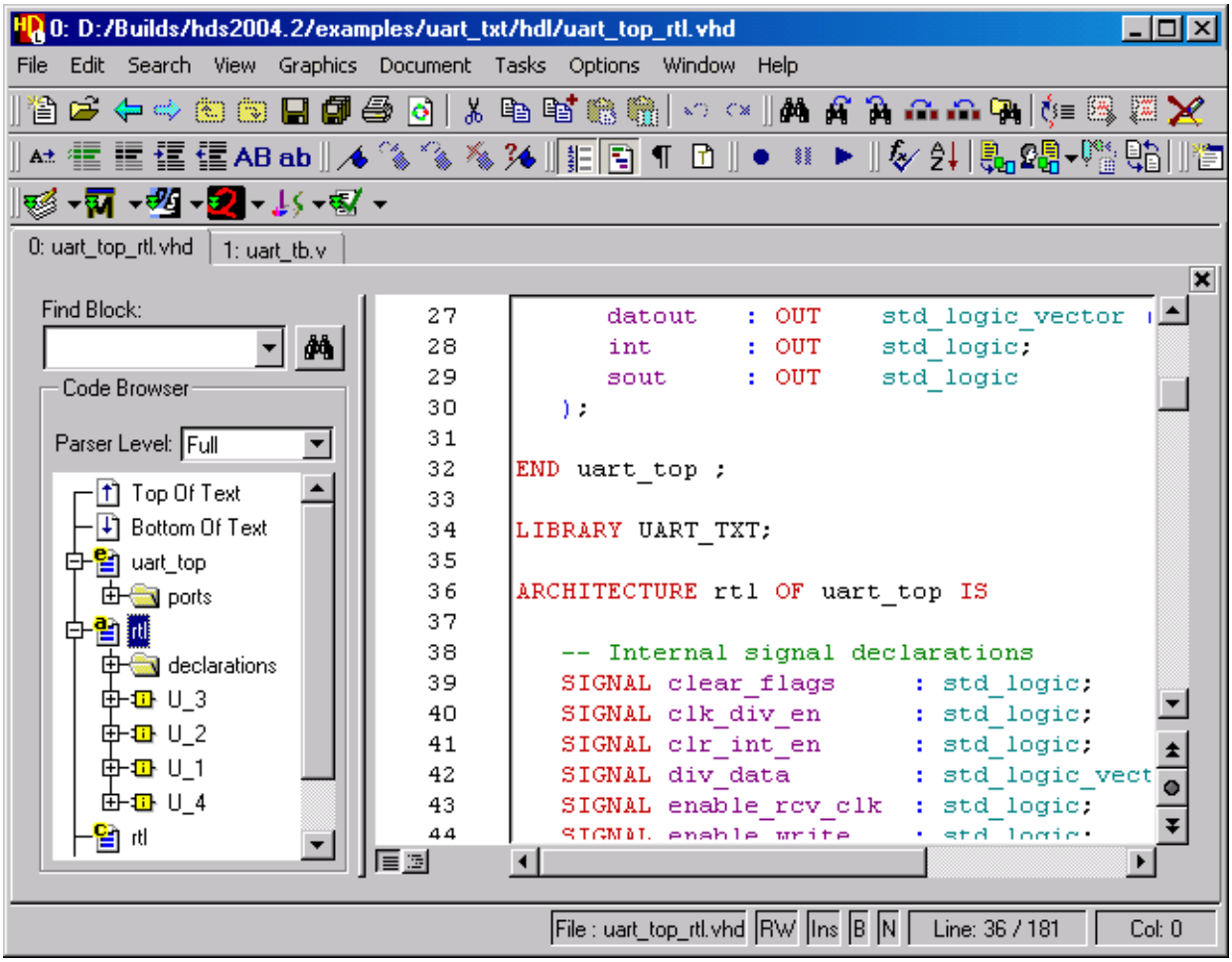
An outline mode can be used to selectively expand or collapse code blocks within a large files and a column select mode is also supported. The editor also includes a file comparison utility and supports user-defined keystroke macros.

Integration with HDS Tools

DesignPad is the default tool used for creating, editing or viewing HDL or plain text views in HDL Detective, HDL Author and HDL Designer.

DesignPad is invoked when you use the Design Content Creation wizard or when you open down (or double-click) on a recognized HDL or text file. You can also invoke a DesignPad editor window by using the  button in the main HDS design manager window.

The following picture shows a VHDL file (*uart_top_rtl.vhd*) displayed in the active tab and a Verilog file (*uart_tb.v*) open in a separate editor window tab:



Note

Refer to "Introduction" on page 5 for information about the default toolbars, status bar and window scrolling controls.

If the main HDS design manager is hidden below other windows, it can be popped to the top by choosing **Design Manager** from the DesignPad **File** menu.

Design Content Creation

You can create a new HDS source view file by using the  button, Ctrl+N shortcut or choosing **Design File** from the **New** cascade in the **File** menu.

The Design Content Creation Wizard is displayed prompting you to specify a file type, choose a location for the file and enter a new view name.

For information about the Design Content Creation Wizard, refer to the “Design Views” chapter in the *HDL Designer Series User Manual* or click the Help button on the wizard.

HDL text views edited in DesignPad are automatically updated in the HDS design explorer windows. You can open up or down into related text or graphical views and cross-reference between design units or from error messages issued by the HDL parser.

You can create a new untitled plain text view by choosing **Plain Text File** from the **New** cascade in the **File** menu.

You can also create a graphical test bench for the view which is currently active in DesignPad by choosing the **Graphical Test Bench** option.

Note

This option is inactive if the active view open in DesignPad is not a HDS source view.

Use the Help button on the dialog box or refer to the *Graphical Editors User Manual* for more information about graphic editor views and creating graphical test benches.

Component Instantiation

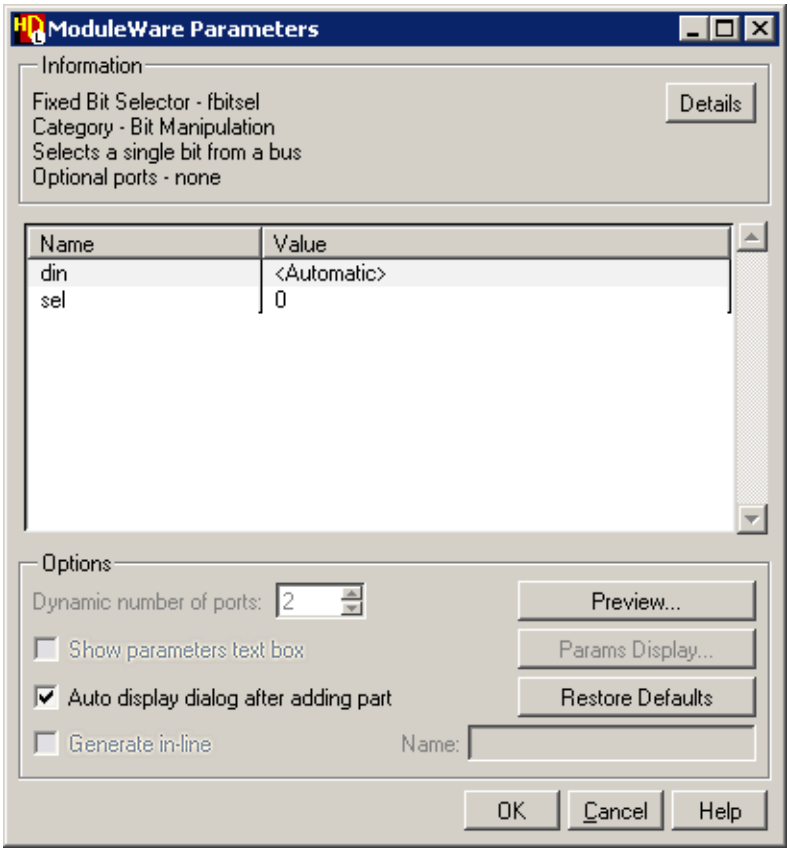
You can instantiate a HDS or ModuleWare component in DesignPad by copying or dragging and dropping from a design explorer window or from the component browser which is displayed when you use the  button or choose **Component Browser** from the **Window** menu.

You can drag and drop a component from the component browser by selecting a component, holding down the left mouse button and dragging it into DesignPad.

Use the  icon on the component browser or refer to “Using the Component Browser” in the *HDL Designer Series User Manual* for more information about the component browser.

The required component instantiation statements are inserted into the file and if automatic indenting is set, the indent level is set to match the location in the file.

On instantiating a ModuleWare component the ModuleWare Parameters dialog is displayed. Refer to “Editing ModuleWare Properties” in the *Graphical Editors User Manual*.



ModuleWare component code is only generated when choosing **Run Through Components** from the **Generate** cascade of the **Tasks** menu or using the button from the DesignPad toolbar. When running a flow ModuleWare component code is also generated.i.e ModelSim flow.

Note that inline generation is not yet supported for ModuleWare components.

Graphical Rendering

You can render any VHDL or Verilog HDL text view as a temporary or saved graphical view which can be viewed using the HDL Designer Series graphical editors.

Note

These commands are available when DesignPad is integrated with HDL Designer, HDL Detective or HDL Assistant. However, you cannot render graphics using a HDL Author license.

Refer to the “Graphical Rendering” chapter in the *Graphical Editors User Manual* for information about these features.

Converting to Graphics

You can create a graphical view from the active HDL declaration by using the  button or by choosing **Convert To Graphics** from the **Graphics** menu to display the Convert to Graphics wizard.

You can set preferences for converting HDL to Graphics by choosing **HDL2Graphics** from the **Options** menu to display the HDL2Graphics Options dialog box.

Use the 

button on the dialog box or refer to the “Setting Convert to Graphics Options” section in the *Graphical Editors User Manual* for information about these options.

Showing as Graphics

In case of opening a non Hierarchy Data Model (HDM) file - that is to say a HDL file that does not belong to a library -the Show as Graphics drop-down palette will be available.

You can render the active HDL view as a temporary graphical view by using the  button.

A pulldown palette on the button allows you to choose a block diagram  , IBD view  , state diagram  or flow chart  view.

Tracing to Graphics

You can trace a line of HDL text to the associated graphics, when the active HDL text file has a corresponding graphics view. For example, when you are viewing the HDL generated from a graphics file.

You can display the graphics view by using the  button or by choosing **Trace to Graphics** from the **Graphics** or popup menu. The graphics file is opened and centered on the graphics corresponding to the HDL text under the cursor.

Visualizing Code

The Visualization feature enables you to convert your HDL text views into graphical views known as visualization views to which you can apply only non-logical edits; that is to say, you can only make layout changes to your visualizations thus preserving your HDL source code intact.

To visualize your code do one of the following:

- Select a graphical view type from the Visualize Code drop-down palette
- Select a view from the **Visualize Code** cascade of the **Graphics** menu.

The newly generated view is added in the Files pane of the design explorer, under the *Visualization* folder.

You can revert from the visualization window to the DesignPad through the Open HDL Source button. Likewise, you can revert from the DesignPad to the visualization window using the Visualize Code button in the DesignPad toolbar (the name of the button will be Open As in this case). Thus, you can easily switch back and forth between the visualization view and its source code.

After opening the HDL source code of the visualization view in the DesignPad, if you make any changes in the source code, the overlay is added to the Open As button and the button name is changed to Update, thus indicating that the corresponding visualization view is out-of-date.

Tasks

DesignPad supports direct access to HDS tasks from the Tasks toolbar and **Tasks** menu.

Default tasks are provided to **Generate** HDL for any instantiated graphical views and to run the **ModelSim**, **LeonardoSpectrum** or **Precision Synthesis** flows. You can also **Set Generate Always** or **Set Compile Always** to force HDL generation or compilation.

Refer to the “Tasks, Tools and Flows” chapter in the *HDL Designer Series User Manual* for more information about tasks.

Version Management

If a version management interface in enabled in the HDS preferences, an additional Version Management toolbar in available and the **Version Management** cascade menu is available from the **File** menu.

You can access the Version Management Settings dialog box which allows you to enable a version management interface and set other default options by choosing **Version Management** from the **Options** menu.

Use the 

button on the Settings dialog box or refer to the “Version Management” chapter in the *HDL Designer Series User Manual* for information about using version management.

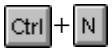
Chapter 2
How to Use DesignPad

This chapter describes procedures for using DesignPad.

Working with Files

You can access commands for working with files from the Standard toolbar or the **File** menu.

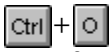
Creating a New File

You can create a new source view file by using the  button,  shortcut or choosing **New** from the **File** menu.

You can also open a new untitled text file by choosing **Plain Text File** from the **File** menu.

Opening and Closing Files

DesignPad is opened in edit mode when you open a text view from the design manager or in read-only mode when you view a generated HDL file.

You can also open a file from DesignPad by using the  button,  shortcut or by choosing **Open** from the **File** menu to display the standard file browser for your workstation. Note that you can use the *Files of type* filter in the dialog box to select from any of the file types for the currently supported language types.

Opening a Selected Pathname

If a text file pathname is referenced in the active text file you can open the file in a new DesignPad tab by selecting the full pathname and choosing **Open Selected** from the **File** menu.

Note

DesignPad supports the use of *HyperText Transfer Protocol (http)* paths to enable you to open selected Web pages directly from within text files.

Opening Up or Down

You can use the  button or choose **Open Down** from the popup menu to open down into a HDL text or graphical view when an instance is selected in the code browser or in the body of a HDL text file. If the current view is a generated HDL text view, a popup list allows you to choose the source or generated view. If the child view is described by a text file, it is opened in a new DesignPad tab.

You can use the  button to open up when the parent view is a graphical symbol, block diagram or IBD view.

Opening a Recently Opened File

You can open a recently accessed file by selecting **Recent Files** from the **File** menu. The cascade menu shows a list of recently opened files which can be re-opened for viewing or editing.

Closing a File

Choose **Close** from the **File** menu or use the button to close the current file. You are prompted whether to save the file if there are any unsaved edits. You can close all the open files by repeating the procedure until all the file tabs have been closed.

Saving a File

Use the button, Ctrl + S shortcut keys or choose **Save** from the **File** menu to save the file in the active tab.

Alternatively, you can choose **Save as** to display the Save File dialog box for your window system. You can optionally navigate to a different location or enter a new name into the *File name* text entry box. A file type filter on the dialog box allows you to view existing files of the supported language types in the destination directory. For example:

```
Verilog files (*.vo,*.v,*.vlg,*.verilog,*.vlog)
VHDL Files (*.vhd,*.vho,*.vhdl)
Tcl Scripts (*.tcl,*.tk,*.itcl,*.itk)
C/C++ sources (*.c,*.h,*.cpp,*.hpp)
XML Files (*.xml,*.htt,*.htl)
All Files (*.*, *.*)
```

Saving All Files

You can save all of the unsaved files open in DesignPad by choosing **Save All** from the **File** menu or using the Ctrl + Shift + S shortcut keys.

Reverting to the Last Saved File

You can revert to the last saved version of the active file by choosing **Revert to last save** from the **File** menu.

Read-Only Mode

DesignPad indicates read-only mode by showing the words (read-only) after the filename on the menu title bar.

If you have write access to the file, you can check the mode by setting or unsetting the **Readonly** option in the **File** menu.

Printing and Exporting Files

Printer page layout, print and export commands are available from the **File** menu.

Page Layout

You can setup the page layout for printing by choosing **Page Setup** from the **File** menu to display the Page Setup dialog box for your window system.

You can choose from a pull down list of paper sizes. On Windows, these are the paper sizes supported on your current printer. On UNIX systems, a list of standard sizes are available or you can define the width and height for a custom page size (which should correspond to a paper size supported by your system printer).

You can set the horizontal and vertical margins at the left, top, bottom and right of the page and set portrait or landscape orientation.

Note

Note that the GNU *enscript* utility is used to print text files on UNIX systems and you can pass extra arguments to *enscript* by entering them in the Page Setup dialog box.

Printing a File

You can print the active file by using the  button,  shortcut keys or choosing **Print** from the **File** menu. The standard print dialog box for your window system is displayed.

On Windows, you can choose from a list available printers and set properties for the selected printer. On UNIX systems you can choose to specify the print command (which initially defaults to *lp -c*) or choose from a pulldown list of recently used print commands.

You can choose to print the whole file, selected text or a specified page range or you can choose to print to a file.

Exporting a HTML File

You can export the active file as a HTML file by using the  button or choosing **Document and Visualize** from the **File** menu. The Document and Visualize dialog box is displayed. Choose to create a website. You can specify a different filename or navigate to a new location using the standard directory browser features supported by your window system.

Click the OK button to confirm the dialog box and write out the HTML file.

Editing Operations

Standard editing commands are available from Standard or Edit toolbars and the **Edit** menu.

Undo and Redo

You can use the button, Ctrl+Z shortcut keys or choose **Undo** from the **Edit** menu to cancel the last action you have performed. In general, any insert or delete editing operations can be undone including those performed by a replace or sort command.

You can reverse the undo action by using the button, Ctrl+Y shortcut keys or choosing **Redo** from the **Edit** menu.

Cut, Copy, Append, Paste and Delete

You can cut, copy, append and paste text to or from the clipboard using the following toolbar buttons, shortcut keys and menu commands:

Table 2-1. Cut, Copy, Append, Paste and Delete Commands

Toolbar	Shortcut	Menu
	Ctrl+X	Cut
	Ctrl+C	Copy
	Ctrl+Shift+C	Append
	Ctrl+V	Paste
	Ctrl+Shift+V	Paste Column

Note

You can select and copy columns when Column Select Mode is enabled as described in "Using Column Select Mode" on page 36.

You can delete text using the Del or ←BS keys. Either key deletes the selected text. Additional shortcuts are defined using the Ctrl and Shift keys:

Del	Delete selected text or next character if nothing selected
←BS	Delete selected text or previous character if nothing selected
Ctrl+Del	Delete to start of next word
Ctrl+←BS	Delete to start of previous word
Ctrl+Shift+Del	Delete to the end of the line

You can also copy or move selected text by holding down the right mouse button and dragging it to a new location within the file. When you release the mouse button a popup menu is displayed for you to choose **Copy Here** or **Move Here** and drop the text at the new location.

Selecting Text

You can select text with mouse by pressing and holding the left mouse button down at the start of the text, dragging the cursor to the end of the required selection and then releasing the button.

Alternatively, you can click the left button at the start of the text and hold the Shift key down. Then click the left mouse button at the end of the text to complete the selection.

You can select text using the keyboard by using the cursor arrow keys to move the cursor to the start of the text you want to select. Then press and hold the Shift key while using the arrow keys to move the cursor to the end of the selection.

You can select a word by moving the cursor over the word and double-clicking the left mouse button.

You can select the current line by using the F8 shortcut key or choosing **Line** from the **Select** cascade of the **Edit** menu.

You can select the current text block by choosing **Block** from the **Select** cascade of the **Edit** menu.

You can select the whole document by using the Ctrl + A shortcut keys or choosing **All** from the **Select** cascade of the **Edit** menu.

The following shortcuts are also available to extend the current selection:

Shift + →	Extend selection one character to the right
Shift + ←	Extend selection one character to the left
Ctrl + Shift + →	Extend selection to end of word
Ctrl + Shift + ←	Extend selection to beginning of word
Shift + ↑	Extend select to line above
Shift + ↓	Extend selection to line below
Shift + Home	Extend select to beginning of line
Shift + End	Extend selection to end of line

Clearing Highlighted Text

You can clear all highlighted text by using the  button or choosing **Clear Highlights** from the **Edit** menu.

Inserting a Text File

You can insert the contents of a specified text file by choosing **Insert File as Text** from the **Edit** menu.

The Open File dialog box for your window system allows you to browse for the required file. When you select a text file and confirm the dialog box its contents are imported at the current cursor position.

Commenting Lines

You can enter the comment characters for the active language (for example, // in a Verilog file or -- in a VHDL file) by using the  button, Ctrl+K shortcut keys or by choosing **Comment Selected/Line** from the **Comments** cascade of the **Edit** or popup menu. The comment characters are added before the selected text or at the start of the current line if nothing is selected.

You can add comment characters at the start of all lines in the current text block by choosing **Comment Block** from the **Comments** cascade of the **Edit** or popup menu.

You can remove comment characters from the current line (or from the selected lines) by using the  button, Ctrl+Shift+K shortcut keys or by choosing **UnComment** from the **Comments** cascade of the **Edit** or popup menu.

Changing Indents

You can set defaults for the indents and spacing used by each language type by setting preferences as described in "Setting Preferences" on page 57.

Note

The default preferences are set to insert a specified number of spaces when you use the  (tab) key instead of inserting explicit tab characters.

You can increase the indent used for the selected text or the current line by using the  button or by choosing **Increase Indent Selected/Line** from the **Change Indent** cascade of the **Edit** or popup menu.

You can decrease the indent used for the selected text or the current line by using the  button or by choosing **Decrease Indent Selected/Line** from the **Change Indent** cascade of the **Edit** or popup menu.

You can increase or decrease the indent used for the current text block by choosing **Increase Indent Block** or **Decrease Indent Block** from the **Change Indent** cascade of the **Edit** or popup menu.

You can apply automatic indenting to the selected text (or all text in the document if nothing is selected) by choosing **Auto-Indent Selected/All** from the **Change Indent** cascade of the **Edit** or popup menu.

You can apply automatic indenting to the current text block by choosing **Auto-Indent Block** from the **Change Indent** cascade of the **Edit** or popup menu.

If the selected text includes indents using tabs, you can convert the tabs to spaces by choosing **Convert tabs to spaces** from the **Edit** menu.

Caution

To prevent improper auto-indentation in Blocks, it is recommended to put the open parenthesis in the same line of the statement that opens this block. Also, put the closed parenthesis and a semicolon ");" in a separate line to close the block so that auto-indentation in Blocks works properly .

Note

You can display the spaces and tabs in your document by choosing **Show Whitespace** from the **View** menu.

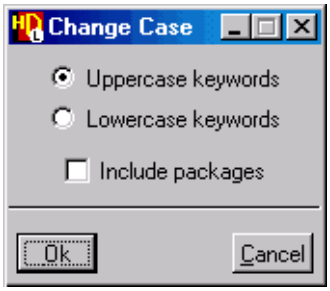
Changing Case

You can change the selected text from lower case to upper case characters by using the AB button, Ctrl+Shift+U shortcut keys or choosing **Make Uppercase** from the **Change Case** cascade of the **Edit** or popup menu.

You can change the selected text from upper case to lower case characters by using the ab button, Ctrl+U shortcut keys or choosing **Make Lowercase** from the **Change Case** cascade of the **Edit** or popup menu.

You can change the selected text (in any case) to use initial capitals for each word by choosing **Make Initial Capitals** from the **Change Case** cascade of the **Edit** or popup menu.

You can change the case of all keywords in the active file by choosing **Keywords** from the he **Change Case** cascade of the **Edit** or popup menu to display the Change Case dialog box:



The dialog box allows you to make all keywords upper case or lower case. You can optionally choose to apply the changed case in VHDL packages.

Completing Keywords, Symbols and Signals

You can automatically complete a language keyword by partially typing the required word and using the  button from the toolbar, Ctrl + spacebar shortcut keys or choosing **Complete Keyword** from the **Edit** or popup menu. You can also use Ctrl + I shortcut keys or choosing **Complete Symbol** from the **Edit** or popup menu to automatically complete symbol and signal names.

The language templates are loaded if they are not already loaded and if a single matching keyword for the current language is found, it is automatically completed. If more than one keyword is found (or there is a corresponding language construct), the choices are listed in a popup window. For example, the keywords *case*, *casex* and *casez* are listed if you type *ca* in a Verilog file. You can insert a keyword (or language construct) by clicking on its entry in the list. Refer to "Using Language Templates" on page 29 for more information about language templates.

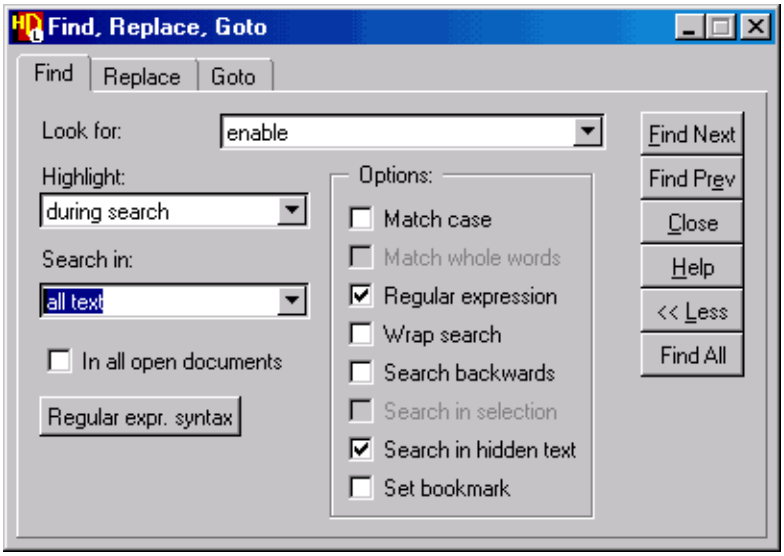
You can automatically complete a signal, port or variable name by partially typing the required name and choosing **Complete Symbol** from the **Edit** menu. This command allows you to quickly insert any matching signal, port or variable name that already exists in the code browser.

Search and Navigation

You can access commands for finding or replacing strings and for navigating through a document from the Search toolbar or the **Search** menu.

Finding and Replacing Text

Use the button, Ctrl + F shortcut keys or choose **Find** from the **Search** menu to display the **Find** tab of the Find, Replace, Goto dialog box.



You can enter a simple search string and use the Find Next or Find Prev buttons to search in the forward or reverse directions.

Alternatively, you can press the More >> button to reveal additional search options that you can use to refine and extend the search.

For example, you can control how the search string is highlighted when found or search in specific areas of a text file by choosing from the pulldown list. For example, you could limit the search to the configuration or architecture code blocks in a VHDL file.

You can also apply the search to all open documents as well as the active file or use the Find All button to find all matching occurrences of the search string.

If you set **Match case** in the dialog box, a case-sensitive search is performed. For example, if you specified the string *Clk*, occurrences of *CLK* or clk would not be found. However, words containing *Clk* (such as *SysClk*) would be found.

You can also choose **Match whole words** to find only occurrences where the text string is a complete word.

Note

The **Match whole word** option is not available when **Regular Expression** is set.

Find normally operates on simple text strings. However, if **Regular expression** is set, you can search for a regular or class expression instead of a text string. For example, the regular expression \<*Clk_* would find all strings starting with the characters *Clk_*.

When this option is set, you can access a full list of supported regular expressions by using the Regular expr. syntax button.

If you set **Wrap search**, the search repeats once all occurrences have been found. If unset, a "*Search string not found*" message indicates when all occurrences have been visited.

If you set **Search backwards** the direction when you use the Find Next button is reversed. However, the Find Prev button always searches backwards.

If there is any text selected in the active file, you can set **Search in selection** to limit the search to the selected text.

The search is normally limited to visible text. However, you can set **Search in hidden text** to search in hidden text. For example, when Outline mode is set. If the search string is found in a hidden text block, it is automatically expanded.

You can also choose **Set bookmark** to automatically set a bookmark (with the class *search*) when the search string is found.

Refer to "Using Bookmarks" on page 25 for more information about bookmarks.

You can use the button, F3 shortcut key or choose **Find Next** from the **Search** menu to select the next occurrence of a text string using the last settings in the dialog box.

Similarly, you can use the button, Shift+F3 shortcut keys or choose **Find Previous** from the **Search** menu to select the previous occurrence of a text string.

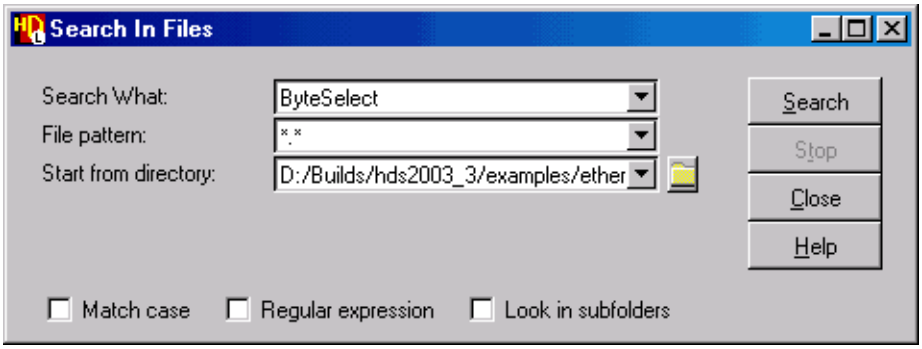
Looking for a Word

You can move the cursor to the next occurrence of a word or string by placing the cursor in the word or selecting the required string and using the button, Ctrl+F3 shortcut keys or choosing **Look Word Next** from the **Search** or popup menu.

You can move the cursor to the previous occurrence of the current word or selected string by using the button, Ctrl+Shift+F3 shortcut keys or by choosing **Look Word Previous** from the **Search** or popup menu.

Finding Text in Files

Use the button or choose **Find in Files** from the **Search** menu to display the Search In Files dialog box.



You can search for a simple text string or set the Regular expression option and enter the search string as a regular expression.

You can also set the **Match case** option to search for string that match the case of the specified simple search string.

You can search files that match a specified pattern. The following filters are available from a pulldown list to search files using the extensions recognized by the supported language types:

VHDL	*.vhd *.vhdl
Verilog	*.v *.vlog
Tcl	*.tcl *.tk *.itcl *.itk
C	*.c *.h *.cpp *.hpp
XML	*.xml *.htt *htl
All Files	*.*

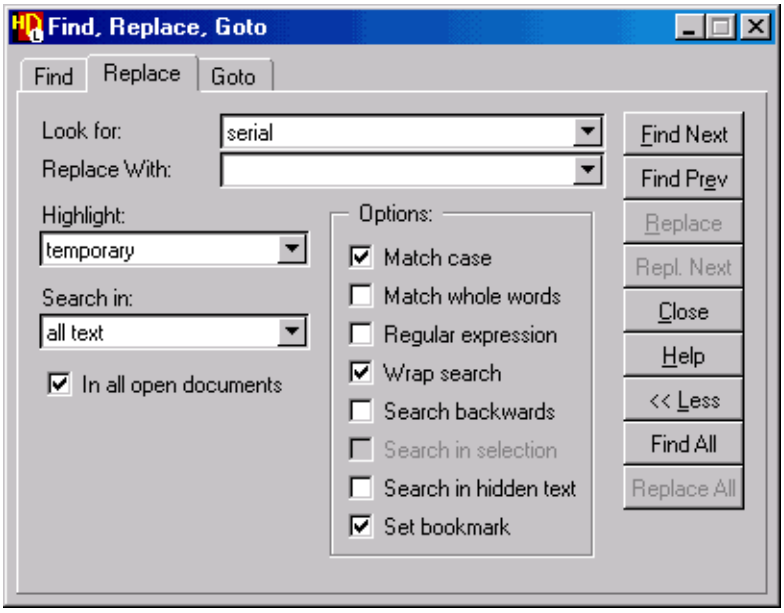
You can edit the file pattern by adding space separated file patterns. For example, you could add *.vlg* and *.verilog* to the Verilog patterns or enter a more specific match such as **_register.vhd*.

You can specify the starting directory for the search and set **Look in subfolders** option to search below the specified directory.

Replacing a Text String

Use the Ctrl + H shortcut keys or choose **Replace** from the **Search** menu to display the **Replace** tab of the Find, Replace, Goto dialog box.

This tab provides similar facilities to the **Find** tab described in "Finding and Replacing Text" on page 20 but with an additional entry box for you to specify the replacement text string.



You can use the Replace button to replace the selected string, the Repl Next button to replace the next occurrence of the search string or the Replace All button to replace all occurrences.

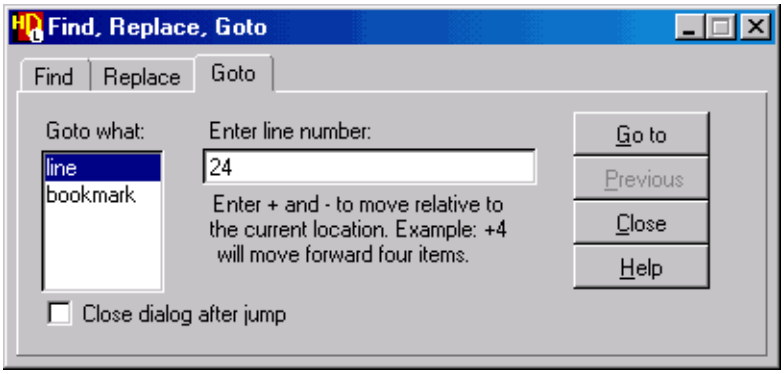
Using Go To Commands

A number of Go To commands are available from the **Search** menu.

Go To a Line or Bookmark

You can go to a specified line or bookmark by using the Ctrl + G shortcut keys of choosing **Go To** from the **Search** menu.

The **Goto** tab of the Find, Replace, Goto dialog box is displayed:



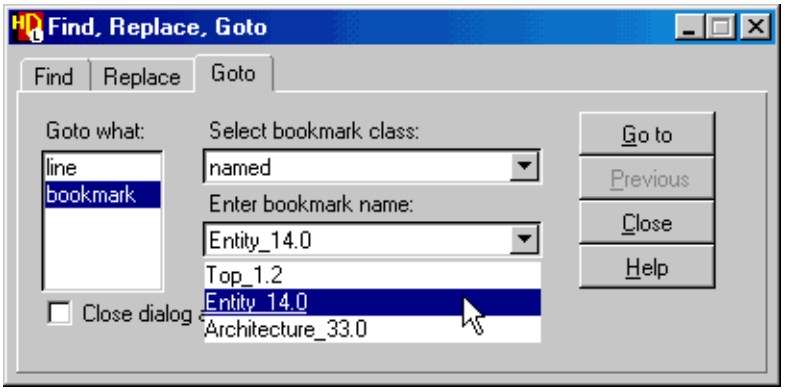
You can specify a required line number or enter + or - to specify a relative number of lines. For example, +4 moves forward four lines. Alternatively, you can go to the next or previous bookmark of a specified class by selecting the *bookmark* option and choosing one of the following options from a dropdown list:

line	Bookmark inserted on a specified line
search	Bookmark inserted by a search operation
named	Bookmark inserted with a specified name
expand	Bookmark at the start of an expandable code block in outline mode
collapse	Bookmark at the start of an collapsible code block in outline mode
errorLine	Bookmark inserted by an error message
activeBp	Bookmark representing an active breakpoint
disabledBp	Bookmark representing a disabled breakpoint

Note

The *expand*, *collapse*, *activeBp* and *disabledBp* bookmarks are not listed when you generate a bookmark report.

If you choose the *named* class, the dialog box allows you to choose from a dropdown list of named bookmarks in the active file.



Go To a Declaration

You can use the F12 shortcut or choose **Go To Declaration** from the **Search** or popup menu to go to the corresponding declaration when the cursor is over a signal, variable or port.

Go To a Code Block

You can choose to go to the **Start of Block**, **End of Block**, **Next Block**, or **Previous Block** by choosing one of these options from the **Go To Block** cascade of the **Search** menu.

Go To a Matching Object

You can use the  button, Ctrl +] shortcut keys or choose **Go To Matching Object** from the **Search** or popup menu to find a matching object. For example, if you have selected a *module* statement, this command finds the matching *endmodule* statement.

If you are using Verilog, this command finds the matching *end* statement:

begin	<->	end
case	<->	endcase
function	<->	endfunction
module	<->	endmodule
primitive	<->	endprimitive
table	<->	endtable
task	<->	endtask
config	<->	endconfig
generate	<->	endgenerate

If you are using VHDL, this command finds the next *end* statement (which may be nested within other *end* statements) after any of the following statements:

entity	units
architecture	record
configuration	block
for	process
package	generate
procedure	if
function	case
component	loop

Go To a Message

You can use the  button, F4 shortcut key or choose **Go To Next Message** from the **Search** menu to go to the next parser message. For example, you can go to the next error line highlighted after using the **Check Syntax** command.

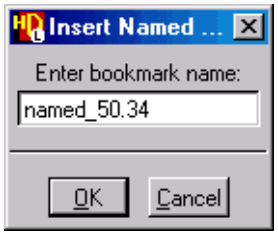
Similarly, you can use the  button, Shift + F4 shortcut keys or choose **Go To Previous Message** from the **Search** menu to go to the previous message.

Using Bookmarks

You can access commands for using bookmarks from the Bookmarks toolbar or from the **Bookmarks** cascade of the **Search** menu.

Insert Bookmarks

You can insert a named bookmark at the current cursor position by using the F7 shortcut or by choosing **Insert Named Bookmark** from the menu. A dialog box is displayed prompting you to enter a unique name for each bookmark on entry.



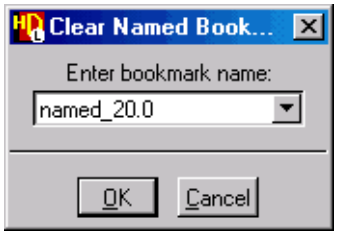
Note

Note that a default name is derived automatically by adding the line and column to the default string *named_*.

This will insert a line bookmark (indicated by an icon) on the current line in the file.

```
44    SIGNAL enable_write    : std_logic;
45    SIGNAL enable_xmit_clk : std_logic;
46    SIGNAL sample          : std_logic;
```

You can clear a named bookmark by using the Ctrl+F7 shortcut or by choosing **Clear Named Bookmark** from the menu. A dialog box is displayed prompting you to select which bookmark you want to delete.



You can insert a line bookmark by selecting a line and then using the button, the Alt+F2 shortcut keys or choosing **Toggle Line Bookmark** from the **Search** or popup menu.

This will insert a line bookmark (indicated by an icon) on the current line in the file.

```
41    SIGNAL clr_int_en      : std_logic;
42    SIGNAL div_data        : std_logic_vector(7 DOWNTO 0);
43    SIGNAL enable_rcv_clk  : std_logic;
```

You can move the cursor to the next line bookmark by using the button, the F2 shortcut key or by selecting **Next Bookmark** from the menu.

You can move the cursor to the previous line bookmark by using the button, the Shift+F2 shortcut keys or by selecting **Previous Bookmark** from the menu.

If bookmarks are enabled in your code browser preferences, the named and line bookmarks are listed below a *Bookmarks* folder in the code browser.

You can also move between bookmarks by selecting the **Line Bookmark** or **Named Bookmark** option as described in "Scrolling Controls" on page 48.

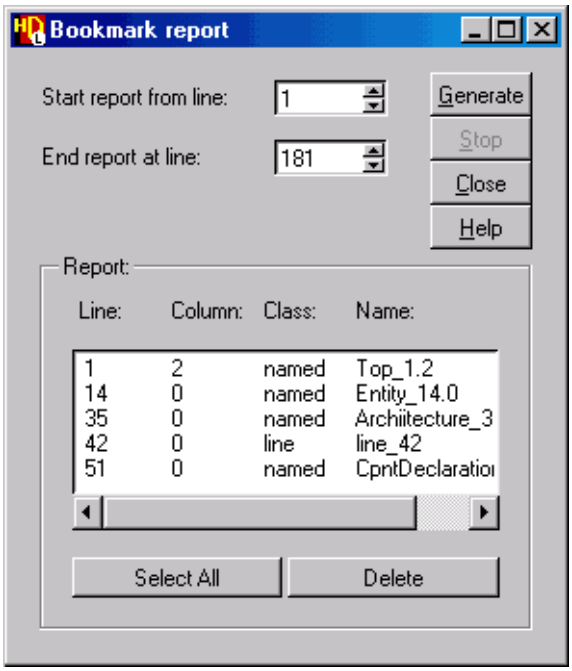
You can clear all of the current bookmarks by using the button, the Ctrl+Shift+F2 shortcut keys or by choosing **Clear All Bookmarks** from the menu.

Generate Bookmark Report

You can view all of the bookmarks which have been entered in a file by generating a bookmark report showing the name and line number of each user-defined bookmark (line, named or search).

Use the button, the Alt+F2 shortcut keys or choose **Report** from the **Bookmark** cascade of the **Search** menu to display the Bookmark Report dialog box.

Optionally specify a start and end line number for the report, then press the Generate button to display a list indicating the line number, column, class and bookmark name.



You can go to the location of any bookmark shown in the report by double-clicking on its entry in the dialog box.

Note

The bookmark report also lists any *errorLine* (indicated by an icon), *warningLine* (indicated by an icon) or *noteLine* (indicated by an icon) bookmarks.

View Operations

A number of commands for view operations are available from the View toolbar or **View** menu.

Showing Line Numbers

You can use the button or choose **Show Line Numbers** from the **View** menu to toggle the display of line numbers.

You can also hide the line numbers by choosing **Hide Line Numbers** from the popup menu when the cursor is in the line numbers column.

Using the Report Pane

You can enable a report pane (which is used to display error messages when syntax checking is performed) by setting the **Enable Report Pane** option in the **View** menu or from the popup window in the report pane.

You can hide or show the report pane by setting the **Show Report Pane** option in the **View** menu or from the popup window in the report pane.

You can save the report as a plain text file by choosing **Save Report As** or print the report by choosing **Print Report** from the popup menu in the report pane.

Highlighting Syntax

You can use the button or choose **Highlight Syntax** from the **View** menu to toggle the highlighting of language keywords for the language of the current file.

Refer to "Setting Preferences" on page 57 for information about setting the highlighting for each of the supported languages.

Showing Whitespace

You can use the ¶ button or choose **Show Whitespace** from the **View** menu to display the space or tab characters underlying whitespace.

Using Language Templates

The language templates help you to quickly enter complex Verilog or VHDL language constructs from a template. They are not intended to replace thorough knowledge of the language but allow you to quickly enter syntactically correct and automatically formatted sections of code.

You can use the  button or choose **Language Templates** from the **View** menu to toggle the display of the language templates window.

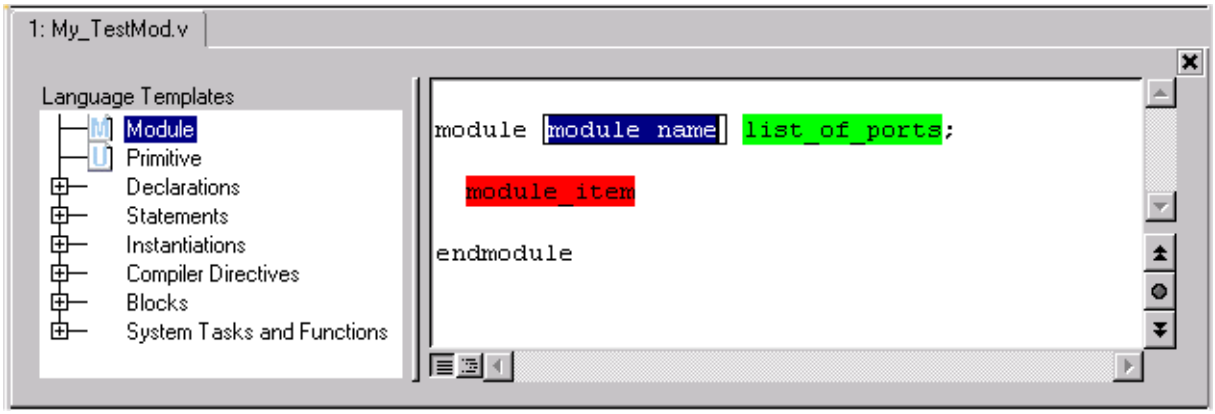
Note

If the language Template window is not displayed, there may be a brief delay while the templates are loaded.

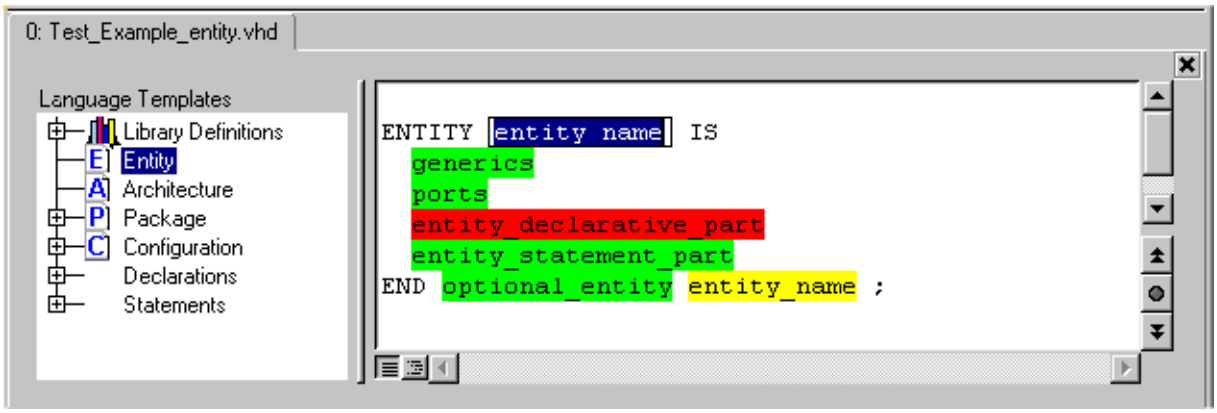
Separate language templates are available for Verilog ‘95, VHDL (which supports VHDL ‘87, VHDL ‘93, VHDL 2002 or VHDL 2008), SystemVerilog, and Verilog (which supports Verilog 2005). The appropriate language is automatically selected for an existing HDL file, or all templates are available in an unsaved file with unknown language.

The language templates are displayed as a hierarchical tree which can be expanded or collapsed by clicking on the ⊞ or ⊟ icons.

You can copy any template construct into the body of the active text file by double-clicking on the required construct in the language templates window. The first editable field is automatically highlighted ready for editing. For example, the following picture shows a module statement added from the Verilog ‘95 template:



The following picture shows an ENTITY statement added from the VHDL '93 template:



Fields requiring user entry (such as an object name) are highlighted in yellow.

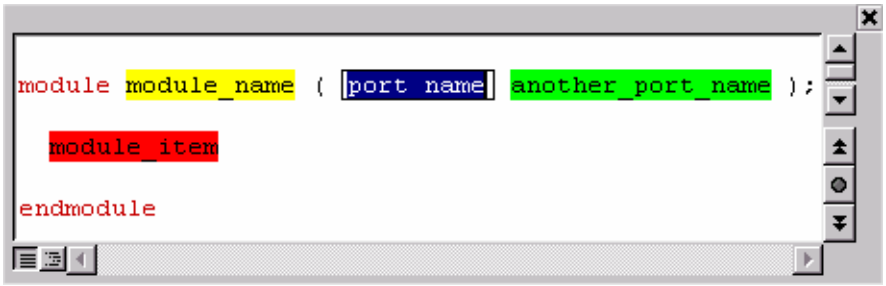
The new name is entered when you press the Return key and the cursor moves to the next field. Note that the name may be used more than once in the same language construct. For example, entity name is the first and also the last editable field in the ENTITY construct. Both of these entries are automatically updated when you edit the first field.

Tip: You can also move between template fields by pressing the key or by choosing **Next Field** or **Previous Field** from the **Templates** cascade of the **Edit** menu.

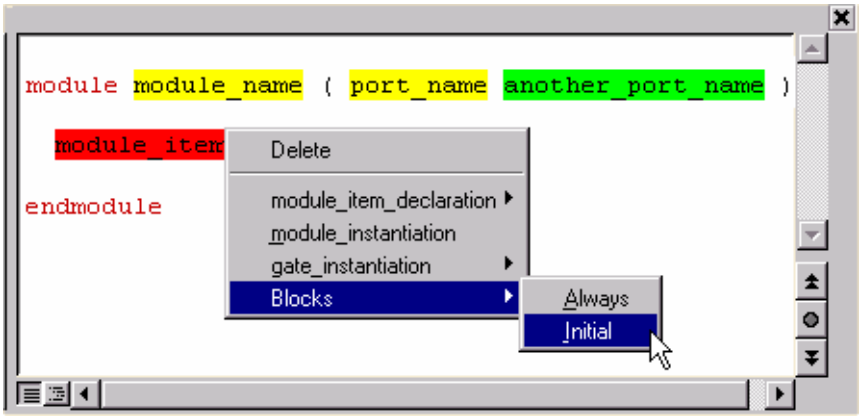
Fields highlighted in green or red can be expanded by double-clicking with the Left mouse button, using the spacebar, choosing **Expand** from the popup menu or by choosing **Expand Field** from the **Templates** cascade of the **Edit** menu.

Some fields display a popup menu which allows you to insert additional template statements. For example, you can choose from a list of types for the VHDL *type* or Verilog *net_type* fields.

Fields highlighted in green are replaced when they are expanded. For example, the VHDL *ports* or Verilog *list_of_ansi_ports* field is expanded to display a new field which allows you to enter a port name. Multiple port names can be entered by expanding the *another_port_name* field. The template is completed by deleting the last unwanted entry field.



Fields highlighted in red support multiple lines. Expanding these fields inserts a new template line before the highlighted field. For example, you can insert a module item declaration, module instantiation, gate instantiation or block statements before the *module_item* field.



You can also enter a language construct by partially typing a keyword and using the **Complete Keyword** command described in "Completing Keywords, Symbols and Signals" on page 19.

For example, if a Verilog language template is active, you can use the A± button or Ctrl + spacebar shortcut keys to complete the word *mod* by choosing *module* from the popup menu to insert the actual keyword or *'module'* to insert a module template.

Displaying Toolbars

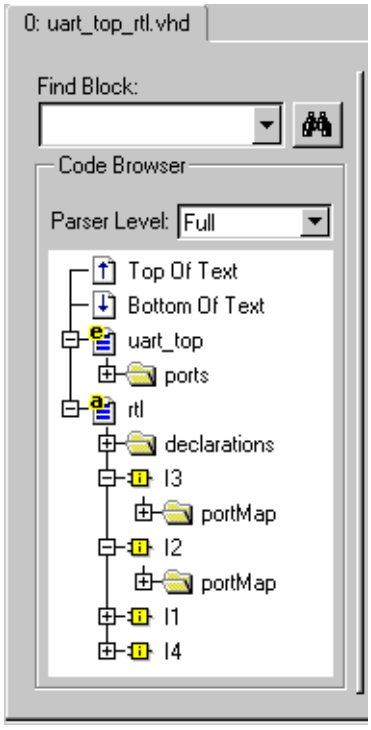
You can display or hide toolbars using the **Toolbars** cascade of the **View** menu.

Refer to "DesignPad Toolbars and Controls" on page 43 for more information about the available toolbars.

Using the Code Browser

The code browser can be displayed or hidden by setting the **Code Browser** option in the **View** menu or by choosing **Hide Code Browser** from the popup menu in the code browser.

The code browser represents the code structure in the active file as a hierarchical tree of recognized code blocks.



Note

Refer to "Code Browser Content" on page 77 for information about the code blocks recognized for each language.

Bi-directional cross-referencing is supported between the code browser and the active file. If you select an object in the code browser, the corresponding line or block of code is highlighted. Similarly, the corresponding object is highlighted in the code browser when the cursor is in the active file. The hierarchical tree is automatically expanded and the windows scrolled when necessary.

You can collapse or expand the tree by clicking on the ⊞ or ⊟ icons or by pressing the right mouse button and choosing **Collapse**, **Collapse All**, **Expand** or **Expand All** from the popup menu.

You can choose to refresh the browser by choosing **Refresh** from the popup menu.

You can find a block of code by entering the name of the code block in the Find Block entry box and using the button. For example, you can enter a signal name (or part of a signal name such as *_clk*) to find the next signal or port declaration block which contains the specified name.

Note that recently entered strings can be accessed from a dropdown list.

You can set the HDL parser level to *Full*, *Fast* or *None*: Full parsing locates all ports and declarations; Fast parsing displays instantiations only; None is the only option available for non-HDL files and locates the top or bottom of the text file and any named or line bookmarks only.

Document Operations

You can perform document commands from the Document Tools and Macros toolbars or from the **Document** menu.

Setting the Language

The language is automatically set when you open a recognized file type or create a new file using the design content creation wizard.

The supported languages normally include: Verilog ‘95, Verilog 2005, SystemVerilog, VHDL ‘87, VHDL ‘93, VHDL 2002, VHDL 2008, Cxx (C or C++), Macro, PSL, Tcl, and XML. If no language is recognized, the default language is set to *none*. Refer to “Document Types” on page 60 for information about adding support for additional languages.

You can set the language for an unrecognized file type or an untitled file by choosing one of the supported languages from the **Language** cascade of the **Document** menu.

Reporting Statistics

You can report statistics about the number of commented and uncommented lines in the active file by using Ctrl+F1 shortcut keys or choosing **Report Statistics** from the **Document** menu

The document statistics are reported on the status bar. For example:

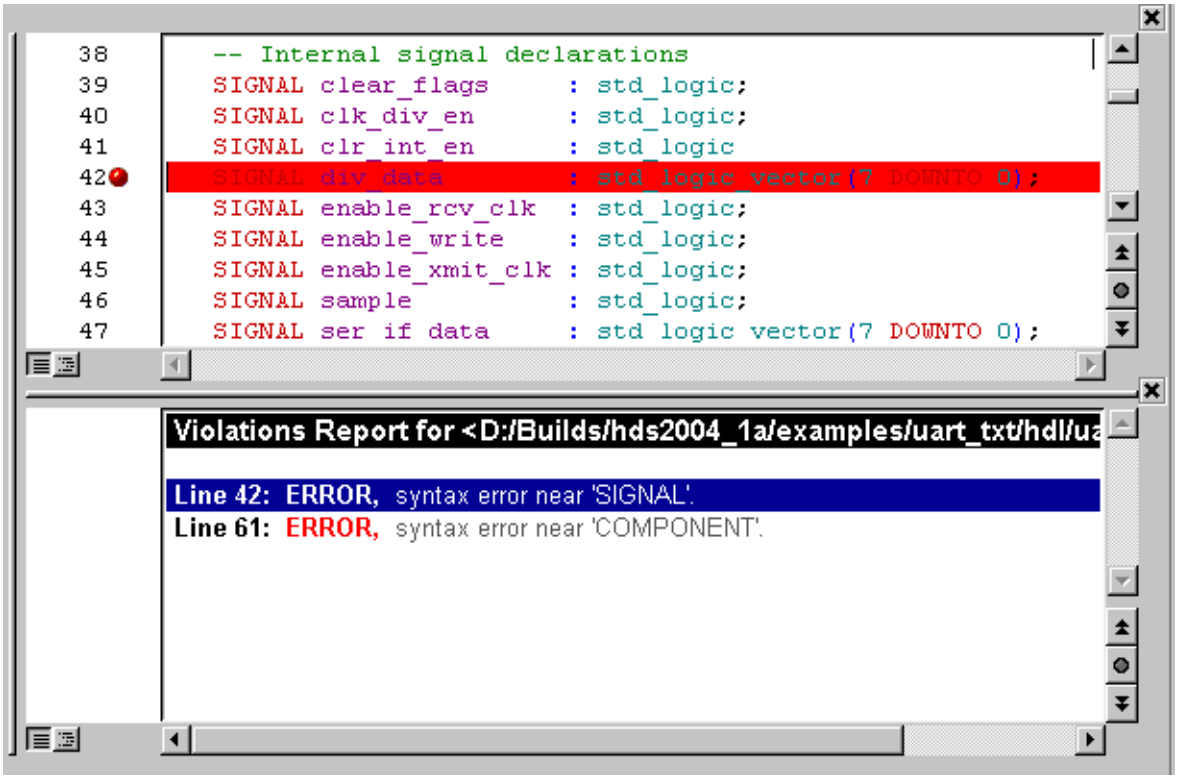
```
Commented Lines: 2; Uncommented Lines: 40
```

Checking Syntax

You can check that the syntax used in the active document is correct for the currently selected language by choosing **Check Syntax** from the **Document** menu.

When an error is detected, the line is highlighted and an *errorLine* bookmark icon is inserted next to the line number for the line of code containing the error.

Error messages are displayed in the status bar and a full list of messages is displayed as a violations report in a report pane below the main HDL text window:



The report pane can be enabled, hidden printed or saved as described in "Using the Report Pane" on page 28.

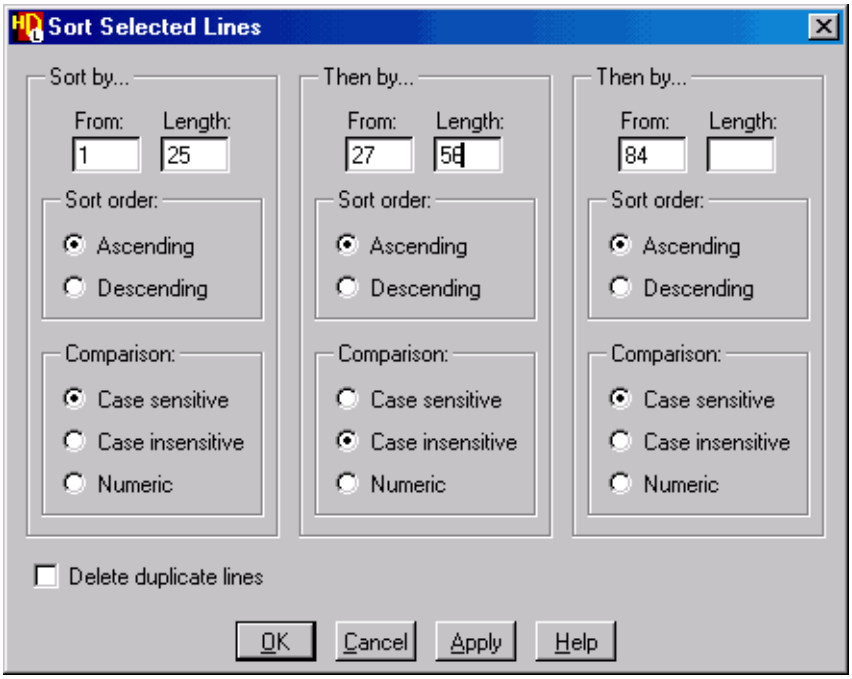
Line numbers can be enabled or disabled as described in "Showing Line Numbers" on page 28.

You can navigate to the next or previous syntax error and its corresponding message in the report pane by using the go to commands described in "Go To a Message" on page 25.

You can cross-reference from a message in the report pane to the error in the code by clicking on the message or cross-reference to the message by clicking on an error line.

Sorting Text

You can sort the selected lines by using the button or choosing **Sort** from the **Document** menu to display the Sort Selected Lines dialog box:



If nothing is selected, the current buffer contents are sorted.

You can sort the selected text in ascending or descending order using options to sort by case sensitive, case insensitive or numeric characters.

You can optionally perform lower level sorts and set an option to automatically delete any duplicated lines.

Using Outline Mode

Outline mode can be used when you do not want to view all of the contents of a long text file. When outline mode is set, each multi-line code block in the file is collapsed and represented by hierarchical ⊞ icons.

You can switch between normal and outline view mode by using the buttons at the bottom of the DesignPad window or by setting **Outline Mode** in the **Document** menu.

You can expand all code blocks by choosing **Expand All** from the **Document** menu or collapse all code blocks by choosing **Collapse All**. You can also expand individual code blocks by clicking on the ⊟ icons.

For example, a combined VHDL entity and architecture file is collapsed in outline mode to show only the entity and architecture declaration lines. You can expand the entity or architecture and then expand lower level code blocks such as the port declarations.

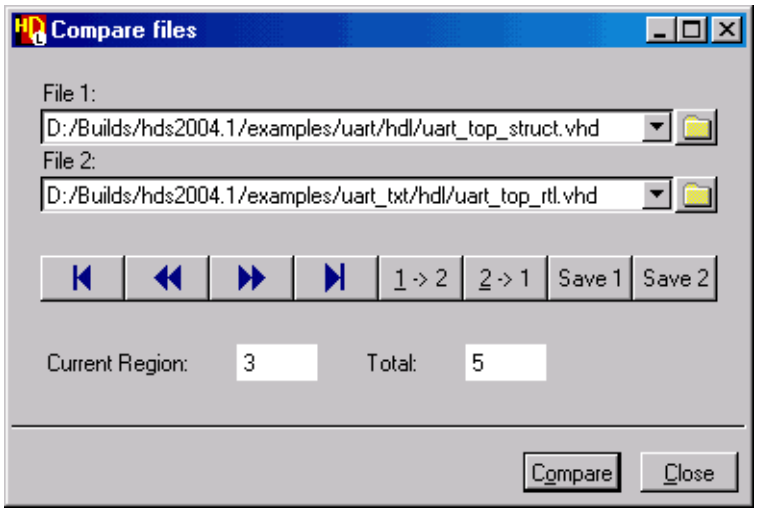
Using Column Select Mode

When **Column Select Mode** is set in the **Document** menu you can select a column of text by holding down the Left mouse button and dragging the cursor vertically down.

When this mode is unset, this action selects all text in the rows under the cursor.

Comparing Text Files

You can compare text files by using the button or choosing **Compare Two Files** from the **Document** menu to display the Compare Files dialog box:



The active file is preselected as *File 1* in the dialog box. You can use the buttons to browse for an alternative file and for the file *File 2* you want to compare.

The files are compared when you press the Compare button.

If there are any differences between the two selected files a split window is displayed in DesignPad highlighting the regions which are different in each of the files. If there are no differences, a message is issued stating that there are no differences between the two files.

Regions of text which contain differences are highlighted in green with the lines which are different shown in yellow.

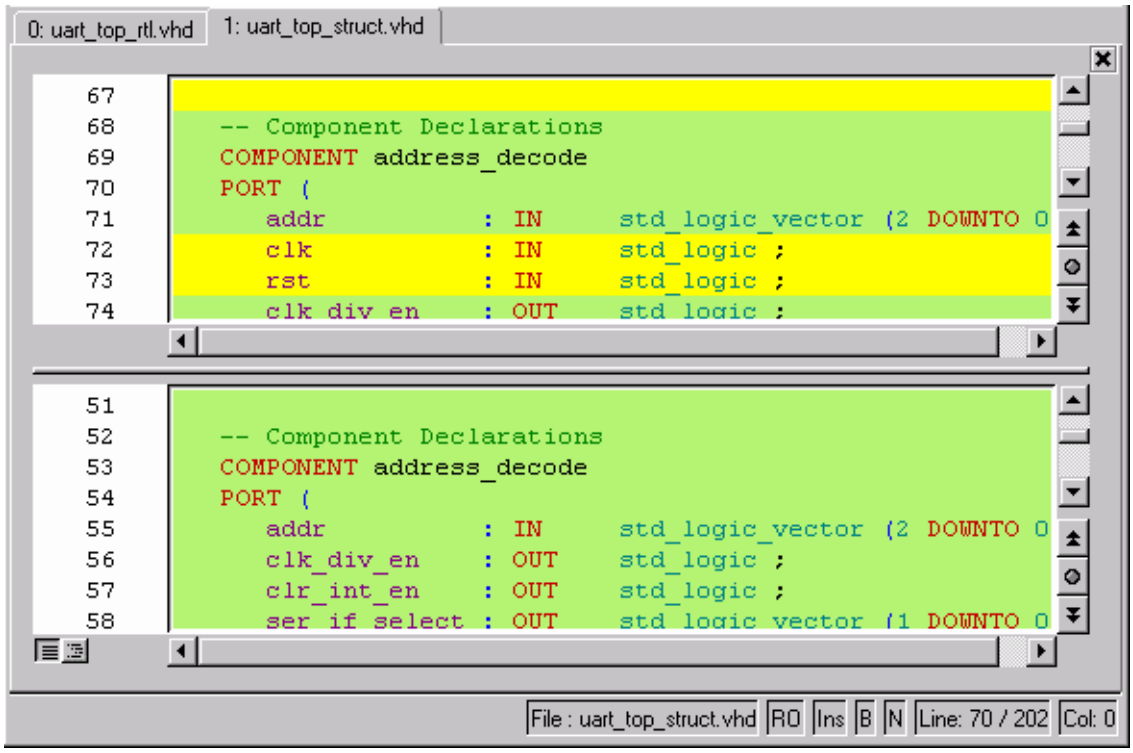
The *Total* field shows the number of regions in the text file where differences are present. The *Current Region* field indicates the current position in the file.

Note

A code block which contains several errors may be shown as one region.

You can navigate backwards and forwards through the regions with differences by using the |◀, ◀◀, ▶|, and ▶▶ buttons. The current region field changes as you move backwards and forwards indicating your position in the file.

The following example shows differences between the VHDL generated from the top level view in the *UART* example design and the corresponding source VHDL for the *UART_TXT* example:



You can merge the differences from the current region of *File 1* in the top window into the corresponding region of *File 2* in the bottom window by using the 1 -> 2 button or merge differences from *File 2* into *File 1* by using the 2 -> 1 button.

Each file can be saved individually using the Save 1 or Save 2 buttons.

Using Macros

Macros can be used to record a sequence of key strokes or commands which perform an operation which you can then repeat by playing back the macro.

Recording a Macro

You can define a macro by using the ● button, the Ctrl+Shift+R shortcut keys or by setting **Record Macro** in the **Macros** cascade of the **Document** menu.

All the subsequent key strokes are recorded until you repeat this command to stop the recording.

Note

You can include a command in a macro by including its keyboard shortcut in the key stroke sequence.

The Save As dialog box for your operating system is displayed when you stop the recording for you to save the keystroke sequence in a new macro file (with the default file extension *.dpm*) at any specified writable location.

This location is saved in your preferences so that the macro files can be read the next time that you invoke DesignPad.

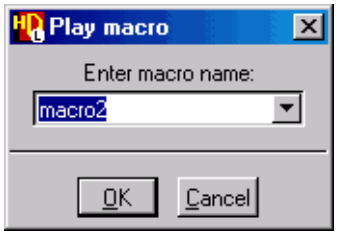
The macro files contain a separate line for each key press and release action that was performed while the macro was recorded.

Pausing a Macro

You can temporarily pause macro recording by using the ⏸ button or by setting **Pause Macro** in the menu. This can be useful if you want to perform keystrokes outside the macro. The macro recording is resumed when you unset this command.

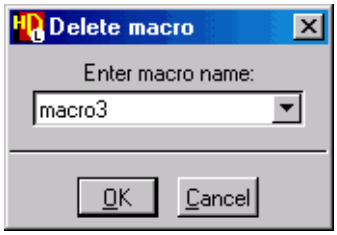
Replaying a Macro

You can repeat the key sequences defined in a stored macro by using the ▶ button, the Ctrl+R shortcut keys or by choosing **Playback Macro** from the menu to choose from a list of available macros in the Play Macro dialog box:



Deleting a Macro

You can delete any existing named macro by choosing **Delete Macro** from the **Macros** cascade of the **Document** menu and selecting from a dropdown list of macro names in the Delete Macro dialog box.

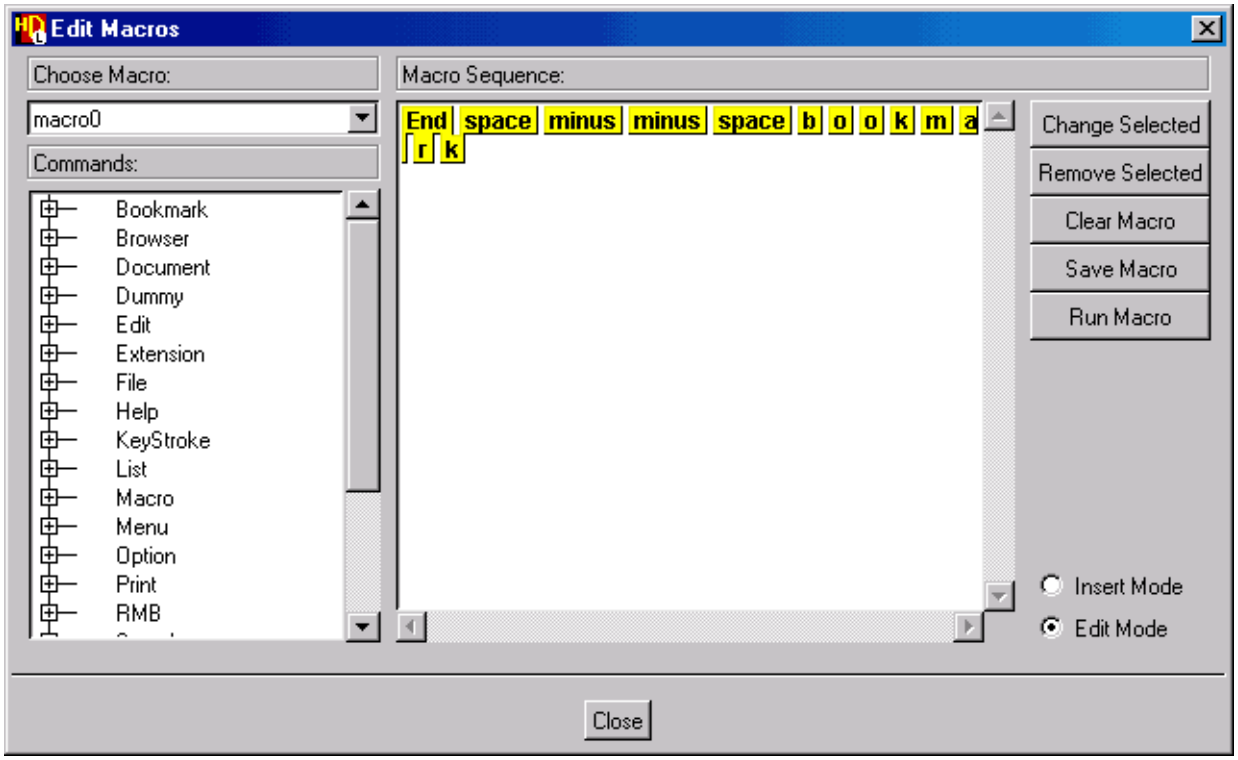


Loading a Macro

You can browse your window system for a predefined macro file by choosing **Load Macro** from the menu.

Editing a Macro

You can edit an existing macro either by directly adding or removing keystrokes in the macro file or by choosing **Edit Macro** from the menu to display the Edit Macros dialog box. The following example shows the dialog box used to edit a simple macro which inserts the comment string -- *bookmark* at the end of the current text line:



The dialog box provides access to all the existing DesignPad commands assigned to toolbars and menus. You can use any of these commands in your macro by expanding the category and using the Left mouse button to drag the required command into the macro sequence list.

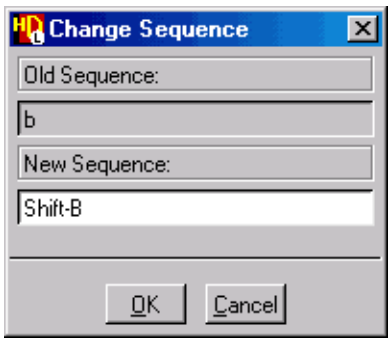
For example, expand the *Bookmark* category and drag the *LineToggle* command into the key sequence shown above to create a macro which inserts a line bookmark and adds an end-of line comment to the current text line.

Note

If your macro contains any commands which cannot be undone, an undo operation will remove any undoable commands and edit strokes but leave non-undoable commands.

You can insert key strokes when **Insert Mode** is set or edit the existing key strokes when **Edit Mode** is set.

In edit mode, you can use the Remove Selected button to remove a key stroke or command and use the Change Selected button when a key stroke is selected to display the Change Sequence dialog box:



You can change the key sequence by typing the required new key stroke into the dialog box.

You can clear all key strokes and commands used in the current macro by using the Clear Macro button or save the macro by using the Save Macro button.

You can test the current macro by using the Run Macro button. This button can be useful to test the changes to a macro before they have been saved.

Refer to "Customizing DesignPad" on page 49 for information about the available commands.

You can also open a macro file as a text view in DesignPad. When opened in this way, the Macro language type is plug-in is enabled and syntax highlighting is available to identify the macro elements. Refer to "Macro" on page 74 for information about the syntax highlighting colors.

Window Controls

You can open a new DesignPad window by using the  button or by choosing **New Window** from the **Window** menu.

You can open several windows and use them simultaneously as separate text editors. When a new window is opened a unique window number is shown before the filename in the title bar. (The window number 0 is used when there is a single window.)

Note

You can set preferences to open new files as a separate top level window or as a tab within the current window by selecting the *Window* category in the Preferences dialog box.

When you have more than one window open, you can activate the next window by using the  button, F6 shortcut key or choosing **Activate Next Window** from the **Window** menu.

Alternatively, you can activate the previous window by using the  button, Shift+F6 shortcut keys or choosing **Activate Previous Window** from the **Window** menu.

You can split the current window by using the  button or choosing **Split** from the **Window** menu. This feature can be useful for working on long text files where you need access to the top and bottom of the file. To remove the split window, choose **Remove Split** from the **Window** menu.

You can also make currently open windows active by choosing from a list of open windows in the **Window** menu or by using the Ctrl+n shortcut keys (where n is the window number). For example, Ctrl+2 opens window number 2.

Note

The window number is shown before the current file pathname in the title bar.

You can close the active window by using the  button, Ctrl+F4 shortcut keys or choosing **Close Window** from the **Window** menu. You are prompted to save if there are any edits to the file in the active window since it was last edited.

You can close all open windows by using the  button or choosing **Close All** from the **Window** menu.

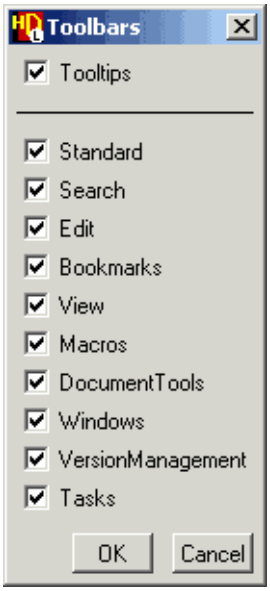
You can also close all windows and exit from DesignPad by choosing **Exit** from the **File** menu.

Appendix A
DesignPad Toolbars and Controls

DesignPad has ten individual toolbars, a menu bar, a status bar and browser controls.

If not already displayed, the toolbars can be individually enabled or disabled from the **Toolbars** cascade of the **View** menu.

Alternatively, you can choose **Settings** from the **Toolbars** cascade of the **View** menu to display the Toolbars dialog box.



The dialog box allows you to enable or disable any of the toolbars. It also allows you to control whether tooltips are displayed when you move the mouse cursor over a toolbar button.

Note

Note that the Version Management toolbar is only available when DesignPad is integrated with HDS and version management is enabled.
You can enable version management by choosing **Version Management** from the **Options** menu in the HDS design manager as described in the "Version Management" chapter of the *HDL Designer Series User Manual*.

The toolbars can be customized by using the **Toolbars** tab of the **Customize** dialog box which can be accessed from the **Options** menu as described in "Customizing DesignPad" on page 49.

Standard Toolbar

The following commands are available from the Standard toolbar:

Table A-1. Standard Toolbar Commands

Button	Description
	Create new Design File (Design Content Creation Wizard)
	Open an existing file
	Go to previous browser position
	Go to next browser position
	Open up
	Open down
	Save active file
	Save all files
	Print active file
	Document and Visualize
	Cut to clipboard
	Copy to clipboard
	Append to clipboard
	Paste from clipboard
	Paste column from clipboard
	Undo
	Redo

Search Toolbar

The following commands are available from the Search toolbar:

Table A-2. Search Toolbar Commands

Button	Description
	Display the Find dialog box
	Find next occurrence of search string specified in Find dialog box
	Find previous occurrence of search string specified in Find dialog box
	Find next occurrence matching the current word
	Find previous occurrence matching the current word

Table A-2. Search Toolbar Commands

Button	Description
	Find a specified search string in files
	Go to matching text
	Go to next message
	Go to previous message
	Clear highlights in text

Edit Toolbar

The following commands are available from the Edit toolbar:

Table A-3. Edit Toolbar Commands

Button	Description
	Complete keyword
	Comment selected text or current line
	Uncomment current comment
	Increase indent for selected text or current line
	Decrease indent for selected text or current line
	Automatic indent for selected text or all text
	Make uppercase selected text or current word
	Make lowercase selected text or current word

Bookmarks Toolbar

The following commands are available from the Bookmarks toolbar:

Table A-4. Bookmarks Toolbar Commands

Button	Description
	Toggle line bookmark
	Go to next bookmark
	Go to previous bookmark
	Clear all bookmarks
	Generate bookmark report

View Toolbar

The following commands are available from the View toolbar:

Table A-5. View Toolbar Commands

Button	Description
	Show or hide line numbers
	Show or hide highlight syntax
	Show or hide whitespace
	Show or hide language templates

Macros Toolbar

The following commands are available from the Macros toolbar:

Table A-6. Macros Toolbar Commands

Button	Description
	Toggle record macro
	Pause record macro
	Playback macro

Document Tools Toolbar

The following commands are available from the Document Tools toolbar:

Table A-7. Document Tools Toolbar Commands

Button	Description
	Run syntax checker
	Sort buffer or selection
	Convert to graphics
	Visualize code
	Trace to graphics
	Compare two files

Windows Toolbar

The following commands are available from the Windows toolbar:

Table A-8. Windows Toolbar Commands

Button	Description
	Open new window
	Close current window
	Close all windows
	Activate next window
	Activate previous window
	Split current window
	Component Browser

Version Management Toolbar

The following commands are available from the Version Management toolbar:

Table A-9. Version Management Toolbar Commands

Button	Description
	Check in the selected objects
	Check out writable copies of the selected objects
	Check out (get) read-only copies of the selected objects
	Change the lock for the selected objects
	Tag the selected objects with a symbolic label
	Synchronize the workspace with objects in the repository
	Report version management status information
	Report the revision control history
	Compare the selected HDL file with a version in the repository

The Version Management toolbar is only available when DesignPad is integrated with HDS and a version management interface is enabled in your HDS preferences. The buttons shown above are available when you are using the GNU Revision Control System (RCS). Additional buttons may be available if a different version management interface is selected.

Refer to the *HDL Designer Series User Manual* for information about the version management features.

Tasks Toolbar

The following commands are available from the Tasks toolbar:

Table A-10. Tasks Toolbar Commands

Button	Description
	Run the LeonardoSpectrum synthesis flow
	Run the ModelSim simulation flow
	Run the Precision Synthesis flow
	Generate HDL from graphical views
	Run the QuestaSim simulation flow

You can display a pulldown palette on the  ,  ,   and  buttons to operate on a single design level, hierarchically through components (indicated by ) or hierarchically through components from the design root (indicated by ). When you select one of these options, it becomes the default operation for the button.

Note that the Tasks toolbar is only available when DesignPad is integrated with HDS. The task buttons shown above are normally available by default. However, the tasks available from the toolbar and menu can be changed using the HDS task manager and are automatically loaded when DesignPad is invoked.

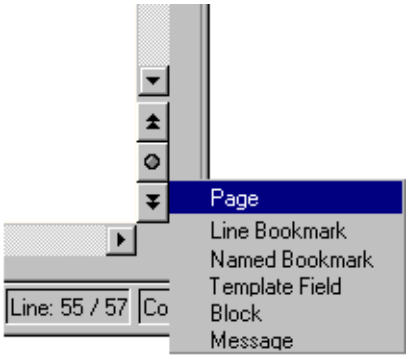
Refer to the *HDL Designer Series User Manual* for information about tasks.

Scrolling Controls

You can use the window scroll bars to move up or down in the current window.

You can also use the  or  buttons. These buttons scroll up or down to the next page by default but you can choose alternative scrolling modes by pressing the  mouse button over the  button on the scroll bar.

A popup menu allows you to set scrolling by page, line bookmark, named bookmark, template field, code block or message:



Menu Bar

The following pulldown menus are provided in DesignPad when it is invoked from within a HDL Designer Series product:

A short message describing the associated command is displayed in the status bar when the cursor is moved over any pulldown menu item or toolbar button.

You can also enable or disable the display of tooltips by choosing **Tooltips** from the **Toolbars** cascade of the **View** menu. When tooltips are enabled, the command name is shown in a tooltip window under the button when the cursor is moved over a toolbar.

Many commands are also available in a context-sensitive popup menu which is displayed when you press and release the right mouse button.

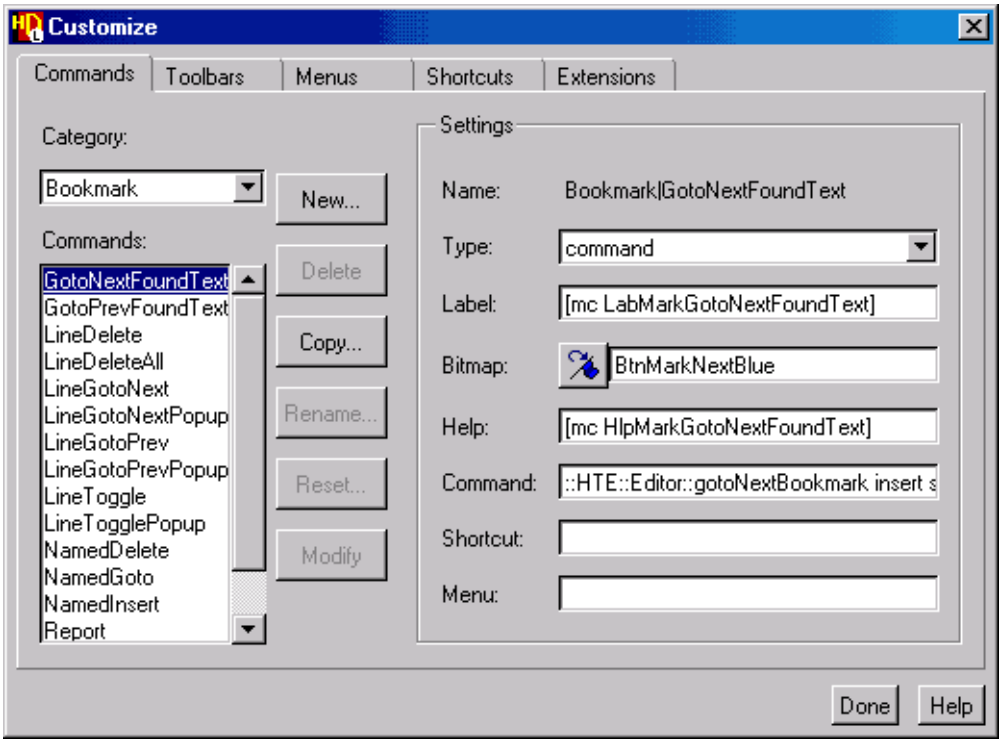
Customizing DesignPad

DesignPad supports customizable commands, toolbars, menus, shortcut keys and a facility to reference your own Tcl scripts defining extension functions.

You can invoke and customize each of these features by choosing **Customize** from the **Options** menu to display the Customize dialog box.

Customizing Commands

The following picture shows the Customize dialog box when it is first invoked. The **Commands** tab appears by default showing the default settings for the first command selected in the commands list.



The category and name of a built-in command cannot be changed although you can use the New... button to create a new command with unspecified settings or the Copy... button to create a new command with the same initial settings as the selected command.

New commands are created with a default category *Custom* and name *Cmd1* but can be renamed in a dialog box which is displayed when they are created. You can also use the Rename... and Delete buttons to rename or delete a custom command. (You cannot delete or rename a built-in command).

You can change any of the following command settings:

- **Type** is normally set to *command* but can be changed to *popup*, *checkbutton* or *radiobutton*. *popup* is used to specify a submenu. *checkbutton* for a command in a menu or toolbar which can be set on or off. *radiobutton* for a command in a menu which selects one only of several choices.
- **Label** is the text displayed when the command is used in a menu.

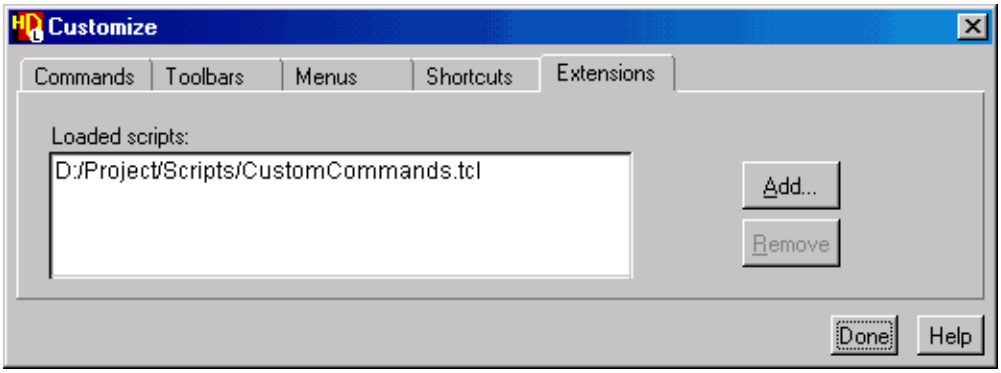
- **Bitmap** is the name of the icon file used when the command is used in a toolbar. You can browse for an alternative icon file (in GIF format) by clicking on the icon shown in the dialog box.
- **Help** is the text shown in the status bar for a menu command or the tooltip text for a toolbar command.
- **Command** identifies the namespace and definition of the command in the Tcl code.
- **Shortcut** is the shortcut text shown when the command is used in a menu.
- **Menu** identifies the submenu name for the *popup* command type.

The Label and Help fields for the built-in commands are references to the actual text strings which are specified in a *.msg* file. Alternatively, the text can be entered directly into the dialog box. Note that you can include the ampersand character before a letter which you want to use as a mnemonic shortcut in a menu option.

The new settings are applied to the selected command when you click the Modify button.

Note that you can use the Reset... button to reset all settings for the selected built-in command to its default settings.

You can reference custom commands which define extensions to the standard features by loading a user-written Tcl script in the **Extensions** tab.



You can use the Add... button to browse for a Tcl script containing custom command definitions and add the selected script to the loaded scripts list. Tcl scripts should have one of the extensions *.tcl*, *.tk*, *.itcl* or *.itk*.

You can use the Remove button to unload a custom Tcl script.

The Tcl script must contain a valid command namespace and definition. The following simple example shows the built-in *AcceptFileDrops* command which allows DesignPad to accept file drops from the Windows explorer:

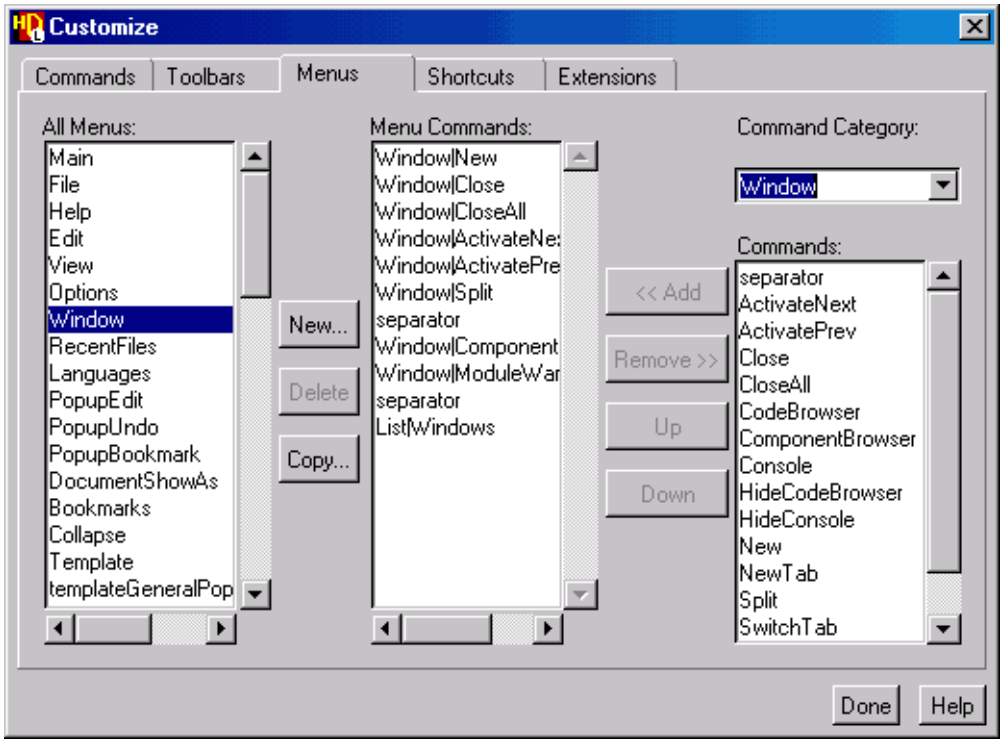
```
namespace eval ::HteAcceptFileDrops {
   ::HTE::Editor::DND::addMIMEType text/uri-list {
```

```
        HTE::API::weakOpenFile [lindex $data 0]
    } 50
}
```

Customizing Menus and Toolbars

Any of the commands defined in the **Commands** tab can be added to custom menus or toolbars using the **Menus** and **Toolbars** tabs.

For example, the following picture shows the **Menus** tab being used to edit the Window menu:



You can use the New... button to create a new empty menu or the Copy... button to create a new menu with the same initial content as the selected menu.

New menus are created with a default name *Menu1* but can be renamed in a dialog box which is displayed when they are created. You can also use the Delete button to delete a custom menu. (You cannot delete a built-in menu).

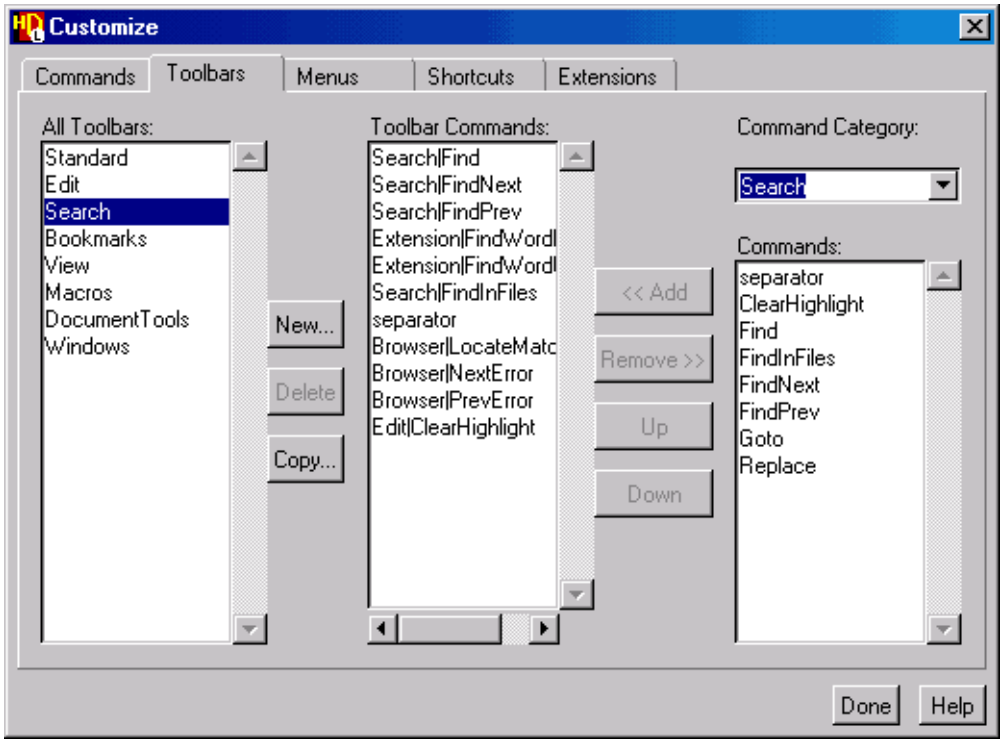
You can add or remove commands in the menus by using the << Add or Remove >> buttons. You can add a special separator character by choosing *separator* from the *Others* category.

You can move a command up or down the menu list by using the Up or Down buttons. Up corresponds to left in the main menu bar; down corresponds to right in the main menu bar.

Note

Note that the shortcut text and brief description shown in the status bar are defined in the **Commands** tab. The shortcut key sequence is defined in the **Shortcuts** tab.

You can customize toolbars in the same way using the **Toolbars** tab. For example, the following picture shows the **Toolbars** tab being used to edit the Search toolbar:



You can use the New... button to create a new empty toolbar or the Copy... button to create a new toolbar with the same initial content as the selected toolbar.

New toolbars are created with a default name *Toolbar1* but can be renamed in a dialog box which is displayed when they are created. You can also use the Delete button to delete a custom toolbar. (You cannot delete a built-in toolbar).

You can add or remove commands in the toolbars by using the << Add or Remove >> buttons. You can add a special separator character by choosing *separator* from the *Others* category.

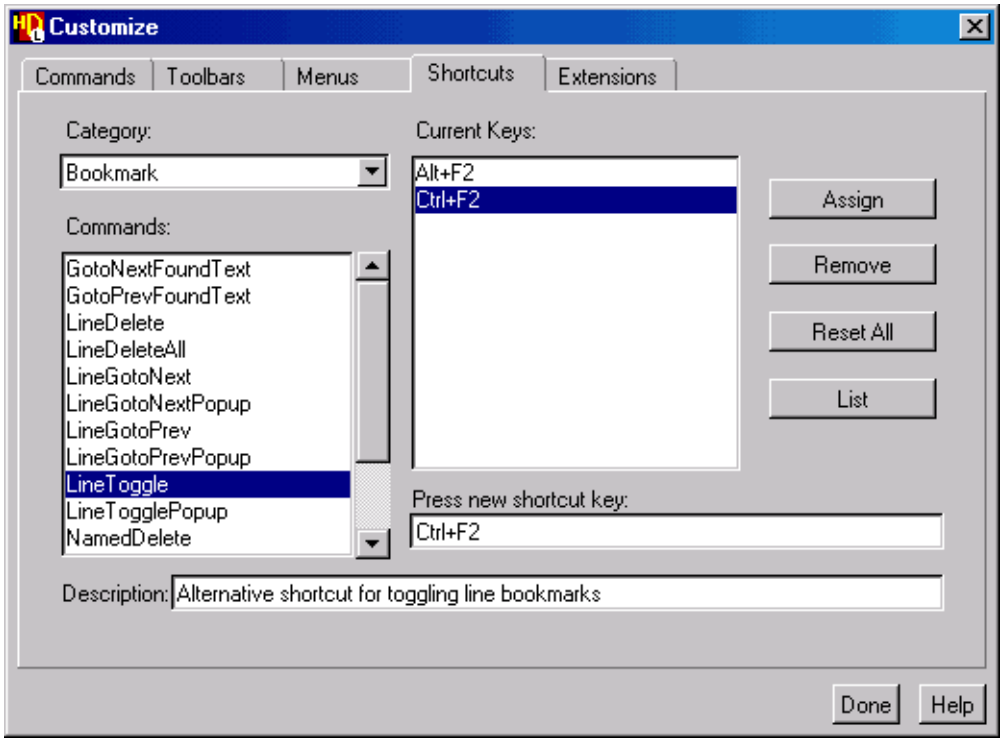
You can move the selected command up or down the toolbar list by using the Up or Down buttons. Up corresponds to left in the horizontal toolbar; down corresponds to right in the horizontal toolbar.

Note

Note that the tooltips and icons used for toolbar buttons are defined in the **Commands** tab.

Customizing Shortcuts

You can use the **Shortcuts** tab to modify or add shortcut key strokes to any of the commands defined in the **Commands** tab.



The dialog box displays the current key shortcut or key shortcuts for the selected command.

You can add a new shortcut key by placing the mouse cursor in the *Press new shortcut key* field and pressing the required key or sequence of keys. You can optionally enter a description for the new shortcut.

The shortcut is added to the list of current key shortcuts for the selected command when you click the Assign button.

You can clear all the custom setting fields by using the Reset All button.

You can display a list of all currently defined shortcuts in a new text editor window by using the List button.

Note

Note that the shortcut can be shown next to the corresponding menu command by specifying the required text string in the *Shortcut* field on the **Commands** tab.

Keystroke Modes

You can configure DesignPad to operate in *vi* or *emacs* emulation mode instead of the normal DesignPad keystroke mode. When either of these modes is enabled, you can use most of the standard keyboard commands recognized by the *vi* or *emacs* editor.

You can choose the keystroke mode by choosing **Vi**, **Emacs** or **DesignPad** from the **Keystroke Mode** cascade of the **Options** menu.

If you use a *vi* or *emacs* keystroke which requires arguments (for example, /*string* to search for the specified string), the arguments can be entered in an entry box which is provided above the status bar at the bottom of the editor window.

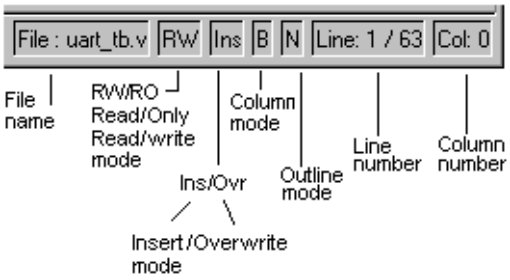
Note

A list of standard DesignPad *Keyboard Shortcuts* can be viewed by choosing **Shortcuts** from the **Help** menu. A list of *vi Emulation Mode Keystrokes* and *Emacs Emulation Mode Keystrokes* can be accessed from a link at the bottom of the standard keystroke list.

Status Bar

The status bar provides information about the current status of the associated file which appears in the main window. It typically displays information about the file and the current line and column numbers.

The following example shows each section of the status bar:



Appendix B
Setting Preferences

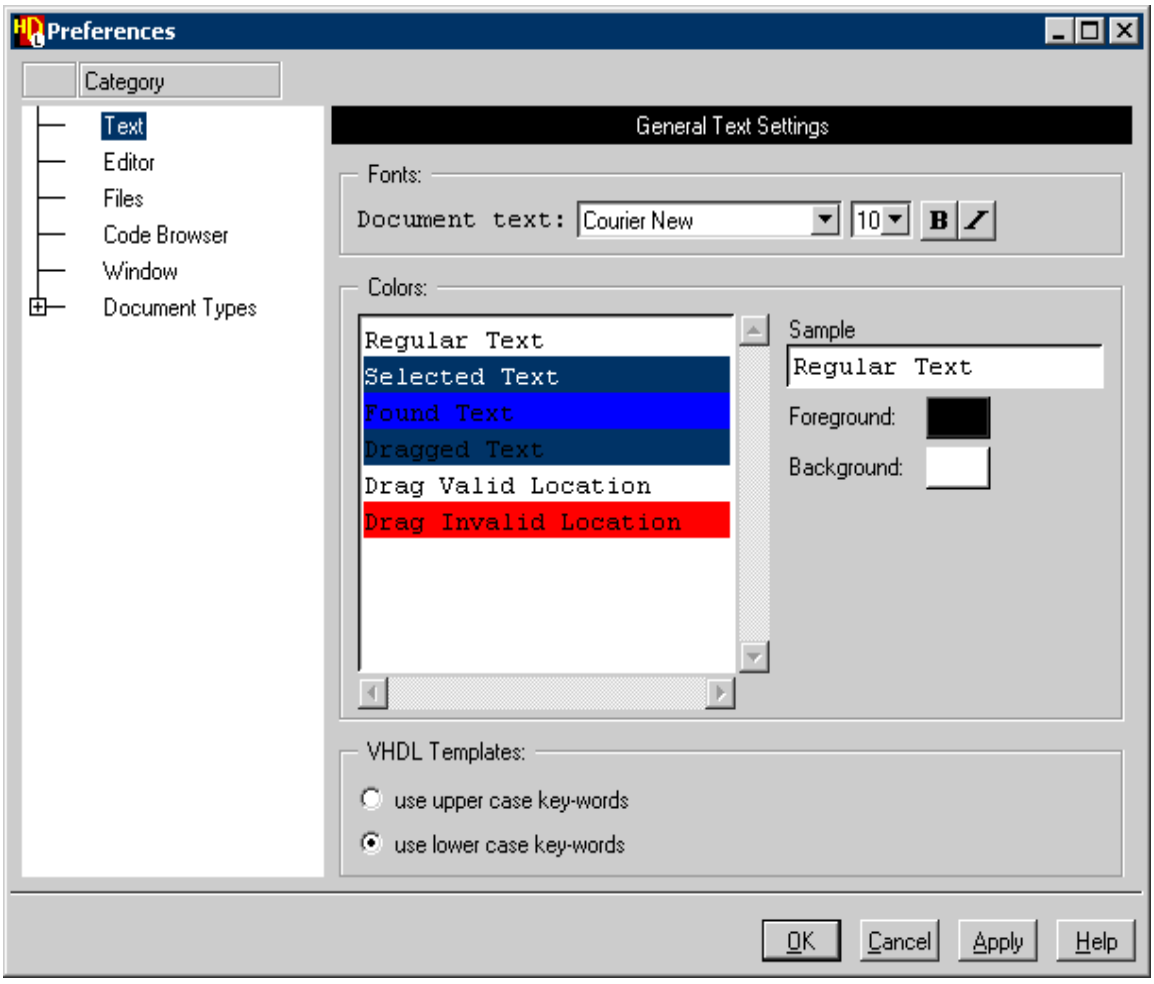
You can set preferences for DesignPad by choosing **Preferences** from the **Options** menu to display the Preferences dialog box. The dialog box displays the preference categories as a hierarchical tree. You can select any of the categories to display the corresponding edit page.

Note

You can access information about the options on each page by using the Help button on the dialog box.

General Text Settings

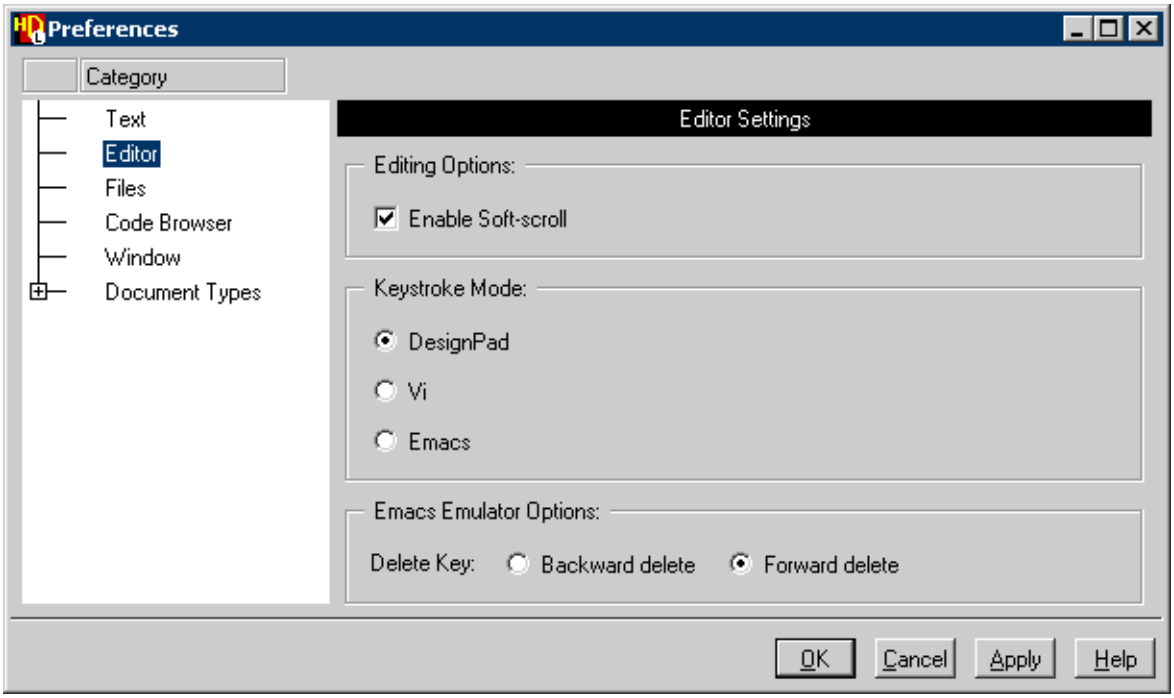
The General Text Settings page allows you to set default fonts and colors:



You can also choose whether upper or lower case is used for language keywords.

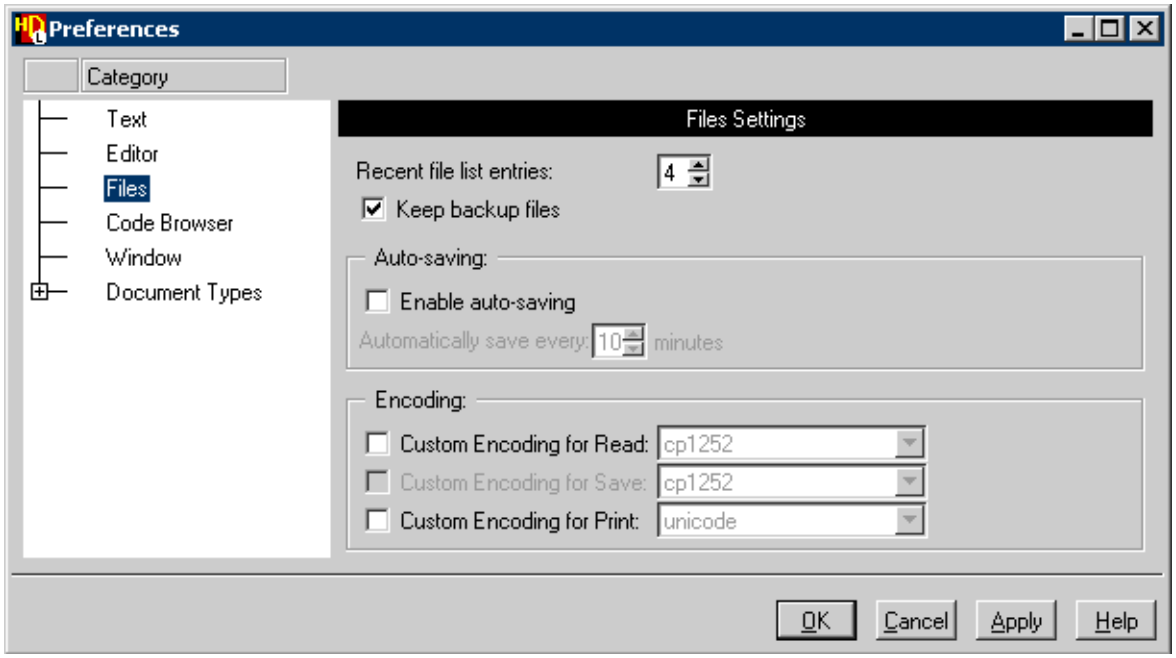
Editor Settings

The Editor page provides options to set the keystroke mode according to the required operating mode. You can choose the to use the DesignPad, Vi or the Emacs operating mode.You can further specify the Emacs Emulator delete key.You can also choose to enable soft-scroll.



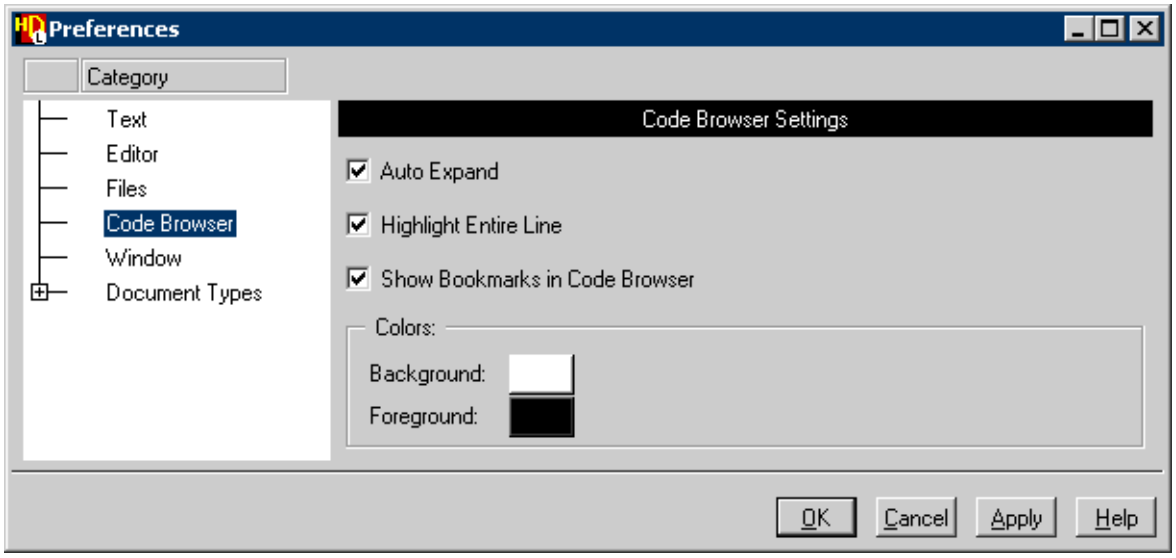
File Settings

The Files page provides options to control the number of entries in the recent files list, whether to keep backup files and enable automatic saves. You can also set separate ISO8859 character transmission encodings for read, save and print:



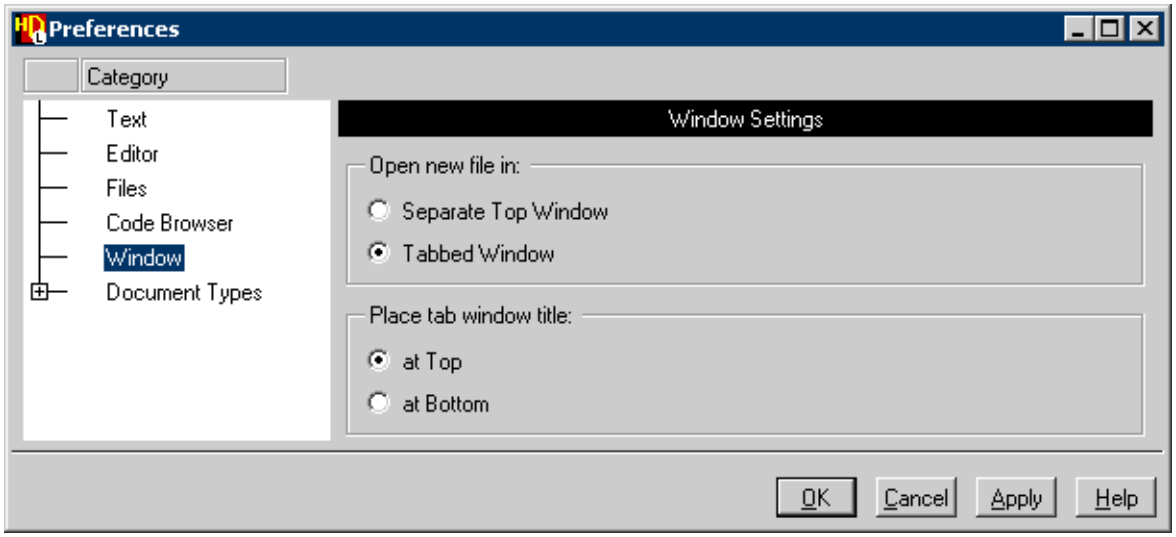
Code Browser Settings

The Code Browser Settings page allows you to enable auto expansion and choose whether to highlight the whole line. You can also choose whether to show named and line bookmarks and set the code browser background and foreground color:



Window Settings

The Window Settings page provides options to control whether windows are opened in a new or separate window and whether the window title is placed at the top or bottom:



Document Types

The Document Types page allows you to add or remove language plug-ins and enable language parsing.

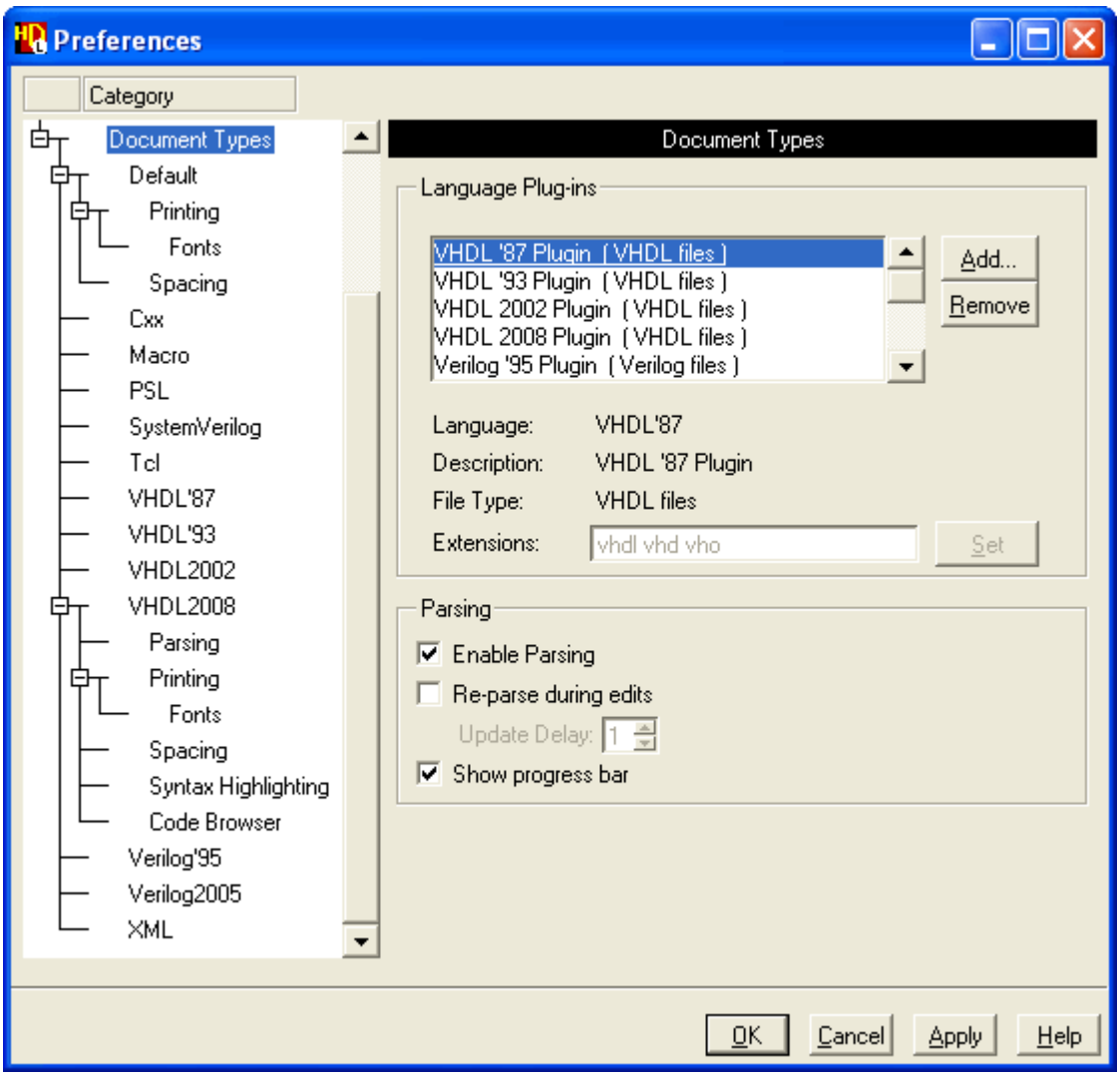
Language plug-ins are automatically loaded for VHDL ‘87, VHDL ‘93, VHDL 2002, VHDL 2008, Verilog ‘95, Verilog 2005, SystemVerilog, PSL, Cxx (C or C++), Tcl, XML and DesignPad Macro document types.

You can use the Add... button to display the Add Plug-ins dialog box and choose an alternative plug-in. For example, a SPICE plug-in for use with analog circuit simulators is available in the *new_plugins* directory.

You can also edit a list of file extensions that are recognized for each language. The following extensions are set by default:

VHDL (‘87, ‘93, 2002 or 2008)	vhdl, vhd, vho
Verilog (‘95 or 2005)	vlg, sv, v, vo, verilog, svh
System Verilog	vlg, sv, v, vo, verilog, svh
PSL	psl
Cxx (C or C++)	c, h, cpp, hpp
Tcl	tcl, tk, itcl, itk
XML	xml, htt, htl
Macro	dpm
SPICE	cir, ckt, cmd, lib, model, sp, spi

If language parsing is enabled, the parser is automatically selected for the language which corresponds to the file extension used by the active file. You can also choose whether to re-parse during edits and optionally show a parser progress bar.



The *Document Types* category can be expanded to access a hierarchy of pages which can be used to set additional preferences for each document type.

You can expand the *Default* document type to display pages which allow you to set default printing, printer font (for Windows only), spacing and indent preferences. These options are used when no specific preference has been set for the active language.

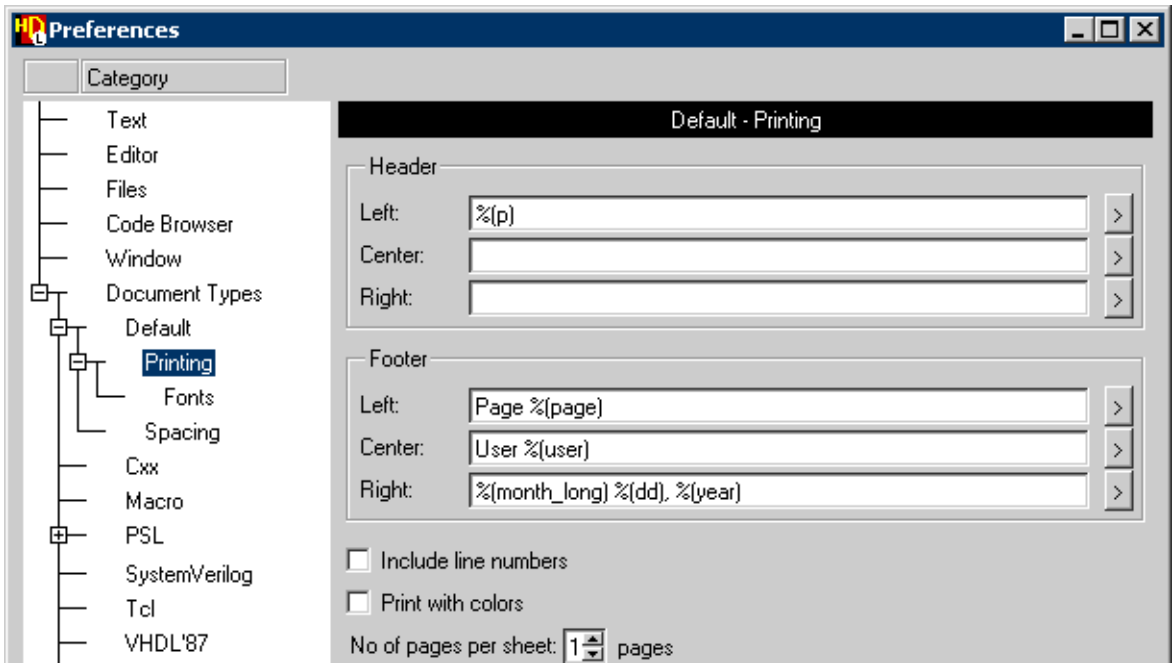
You can expand each language document type to set specific printing, printer font, spacing and indent preferences for each language. When specified, these preferences override the default document type preferences.

The language document types also have additional preference pages for parsing, syntax highlighting and code browser options.

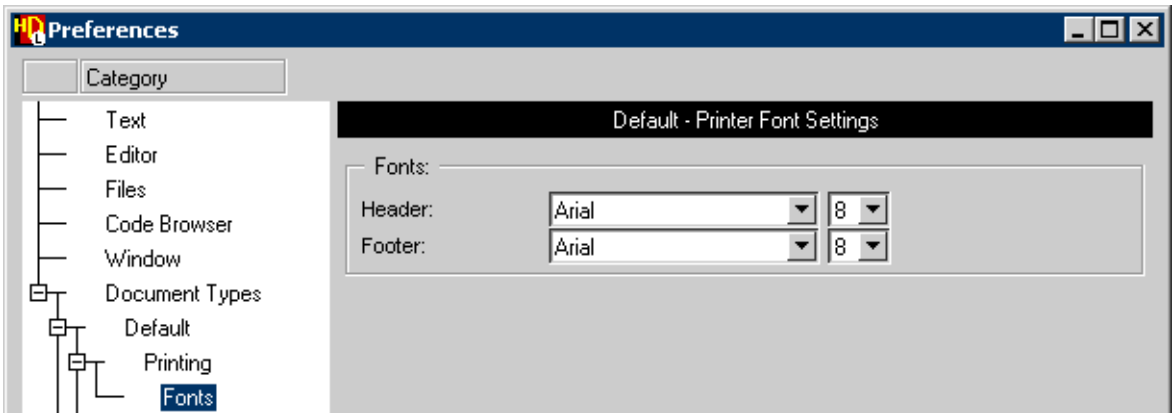
Printing and Font Settings

You can expand the *Default* or language document types to display pages which allow you to setup printer headers and footers.

You can choose whether to include line numbers and (if a color printer is available) whether to reproduce highlight colors. You can also specify the number of pages to print on each sheet of paper.

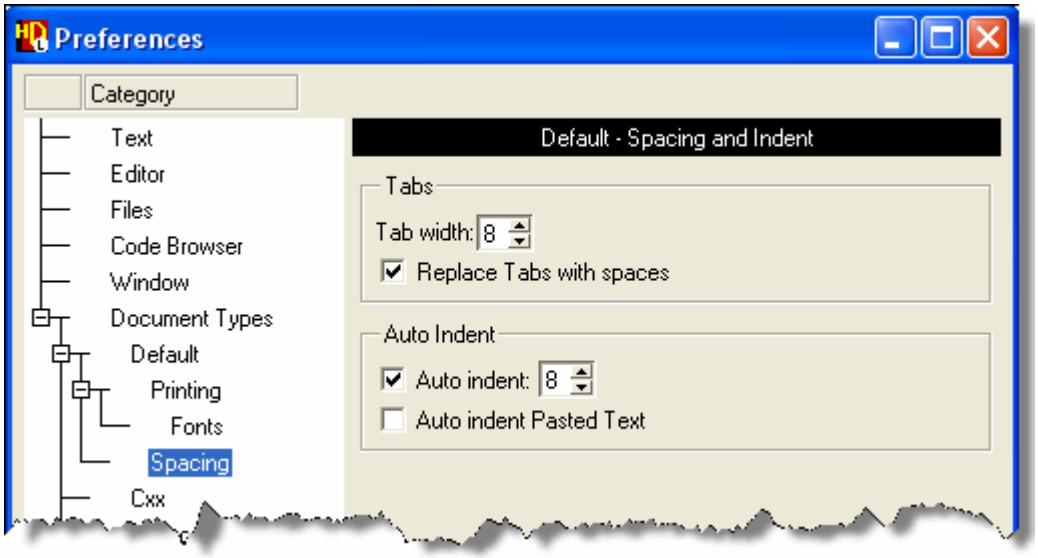


On Windows workstations, a separate page is available which allows you to set a separate printer font and font size for headers and footers.



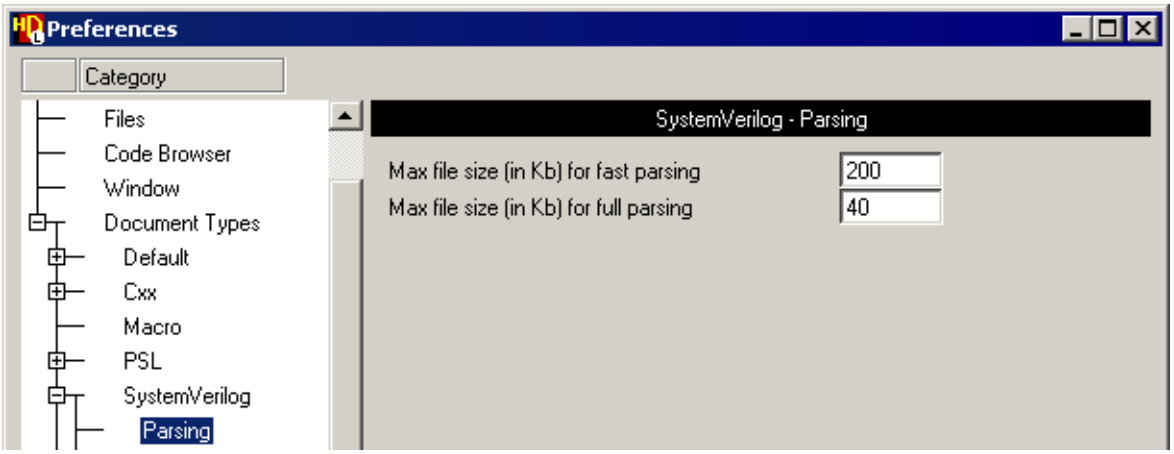
Spacing and Indent

The Spacing and Indent page allows you to set default and language options for tab width and automatic indent. The language pages also have an option to automatically indent pasted text.



Parsing

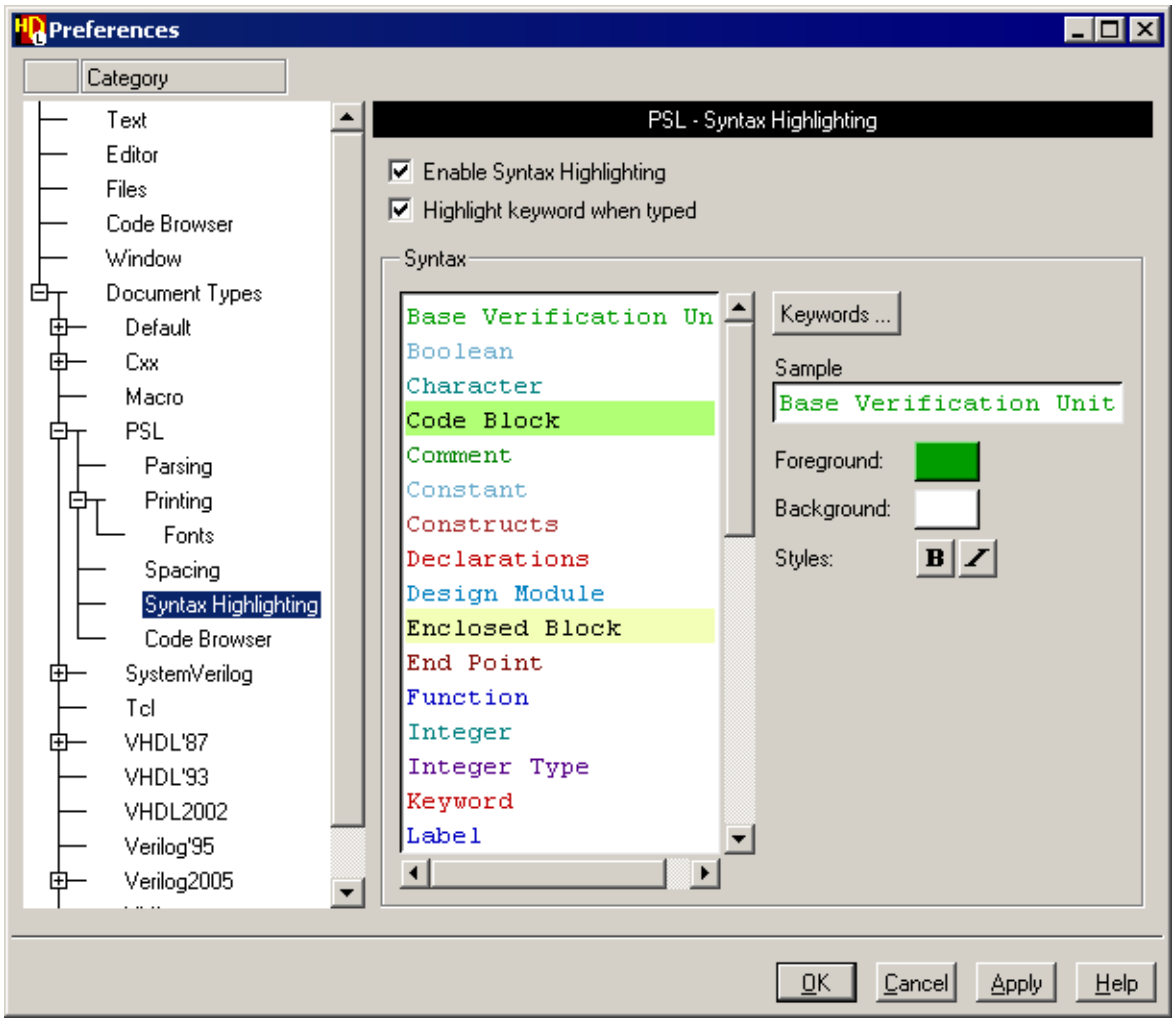
The VHDL, Verilog and the SystemVerilog language types have an additional Parser page which allows you to set the maximum file sizes for fast and full parsing:



Syntax Highlighting

You can set syntax highlighting preferences separately for each language.

For example the following picture shows the syntax highlighting page for PSL:



You can setup syntax highlighting separately for each document type. You can also use the Keywords ... button to set the highlight colors used for each individual keyword.

C or C++

The following table lists the default colors for the tags defined when you are using the C or C++ language:

Table B-1. C and C++ Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: 'x'	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black

Table B-1. C and C++ Syntax Highlighting

Tag	Description	Background	Foreground
Code Block Bound	The braces }{ bounding a code block.	white	purple
Comment	A comment line (preceded by // or text enclosed between /* and */).	white	green
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Keyword	All pre-defined C or C++ keywords.	white	red
Line Continuation	A line continuation operator \.	white	light red
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Operator	Any C or C++ operator.	white	blue
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
Preprocessor Directives	Any preprocessor directives.	white	navy blue
String	A string enclosed in double quotes. For example: ”mystring”	white	turquoise
URL	A unified resource locator anywhere in the text.	white	blue italic

PSL

The following table lists the default colors for the tags defined when you are using the Accellera Property Specification Language (PSL):

Table B-2. PSL Syntax Highlighting

Tag	Description	Background	Foreground
Base Verification Unit	The name of an inherited verification unit.	white	light green

Table B-2. PSL Syntax Highlighting

Tag	Description	Background	Foreground
Boolean	Name of a boolean	white	sea blue
Character	A character enclosed in single quotes. For example: ‘x’	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Comment	A comment line (preceded by // or text enclosed between /* and */ in Verilog; preceded by -- in VHDL).	white	green
Constant	A constant name.	white	sea blue
Constructs	Preprocessor constructs (%if, %for, %end etc.)	white	brown
Declarations	Named declaration keywords.	white	red
Design Module	HDL design module under verification.	white	light blue
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
End Point	Name of a user-defined end point.	white	brown
Function	A boolean level built-in function.	white	dark blue
Integer	An integer not inside a string	white	turquoise
Integer Type	The name of an integer	white	dark purple
Keyword	All pre-defined keywords.	white	red
Label	Name of a label	white	dark blue
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Operator	Any operator (such as |=>).	white	blue
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
Parentheses	Parentheses, braces and square brackets.	white	blue

Table B-2. PSL Syntax Highlighting

Tag	Description	Background	Foreground
Preprocessor Directives	Any preprocessor directives (#define, #include etc.).	white	dark blue
Property	The name of a user-defined property.	white	light blue
Sequence	The name of a user-defined sequence.	white	purple
String	A string enclosed in double quotes. For example: ”`mystring`”	white	turquoise
Struct Type	The name of a structure.	white	dark red
System Verilog Keyword	Supported System Verilog keywords.	white	red
URL	A unified resource locator anywhere in the text.	white	blue italic
Verification Directive	All verification layer directives (assert, assume, restrict etc.).	white	blue
Verification Unit	Name of a user-defined verification unit, mode or property.	white	purple
Verilog Keywords	Supported Verilog keywords.	white	red
VHDL Keywords	Supported VHDL keywords.	white	red

Tcl

The following table lists the default colors for the tags defined when you are using the tool command language (Tcl):

Table B-3. Tcl Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: ‘`x`’	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Code block Bound	The brackets `][` or braces `}{` bounding a code block.	white	purple
Comment	A comment line (preceded by #).	white	green

Table B-3. Tcl Syntax Highlighting

Tag	Description	Background	Foreground
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Instruction	Command line enclosed between brackets }{.	white	violet
Keyword	All pre-defined C or C++ keywords.	white	red
Line Continuation	A line continuation operator \.	white	light red
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Operator	Any Tcl operator.	white	blue
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
String	A string enclosed in double quotes. For example: ”`mystring`”	white	turquoise
URL	A unified resource locator anywhere in the text.	white	blue italic
Variable	Any variable name.	white	saddle brown

VHDL’87, VHDL’93, VHDL 2002 or VHDL 2008

The following table lists the default colors for the tags defined when you are using the VHDL’87, VHDL’93, VHDL 2002 or VHDL 2008 language:

Table B-4. VHDL’87, VHDL’93, VHDL 2008 and VHDL 2008 Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: ‘`x`’	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Comment	A comment line (preceded by --)	white	green

Table B-4. VHDL'87, VHDL'93, VHDL 2008 and VHDL 2008 Syntax Highlighting

Tag	Description	Background	Foreground
Constant	A constant name defined by the CONSTANT keyword. For example: `CONSTANT MyConst`	white	purple
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Enum Literal	The identifiers or literals of an enumeration type. For example: `type states is (StateA,StateB)`	white	purple
Generic	Name of a generic constant. For example: `GENERIC (Tpd: time)`	white	dark blue
Identifier - Assn List	An identifier in an association list. For example: `clk => clk,`	white	maroon

Identifier - Record	An Identifier in a record. For example:
	`TYPE switch_info IS RECORD`
	`status : BINARY;`
	`IDnumber : INTEGER;`
	`END RECORD`
		white	violet

Integer	Any integer not inside a string.	white	turquoise
Keyword	All pre-defined VHDL keywords.	white	red
Label	Any label.	white	dark blue
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Mode	The port mode (in, out, inout, buffer or linkage).	white	dark blue
Operator	Any operator (colon, assignment, comparison, slash etc.).	white	blue
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
Parentheses	Parentheses.	white	blue
Port	The name of a port.	white	purple

Table B-4. VHDL'87, VHDL'93, VHDL 2008 and VHDL 2008 Syntax Highlighting

Tag	Description	Background	Foreground
Signal	The name of a signal.	white	purple
String	A string enclosed in double quotes. For example: "`mystring`"	white	turquoise
Type Name	Type names. For example: `std_logic_vector`	white	turquoise
URL	A unified resource locator anywhere in the text.	white	blue italic
Variable	A variable name defined by the VARIABLE keyword. For example: `Variable MyVar`	white	purple

The following additional PSL syntax highlighting tags are also supported when you are using VHDL '87, VHDL '93, VHDL 2002 or VHDL 2008:

PSL Base Verification
PSL Boolean
PSL Constant
PSL Constructs
PSL Declarations
PSL Design Module
PSL Preprocessor Directives
PSL Property
PSL Sequence
PSL End Point
PSL Function
PSL Integer Type
PSL Parentheses
PSL Struct Type
PSL System Verilog Keywords
PSL Verification Directive
PSL Verification Unit
PSL Verilog Keywords
PSL VHDL Keyword

Verilog'95 or Verilog2005

The following table lists the default colors for the tags defined when you are using the Verilog'95 or Verilog2005 language:

Table B-5. Verilog'95 and Verilog2005 Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: '`x`'	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Comment	A comment line (preceded by // or text enclosed between /* and */)	white	green
Compiler Directives	A conditional compilation directive	white	blue
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black

Table B-5. Verilog'95 and Verilog2005 Syntax Highlighting

Tag	Description	Background	Foreground
Function Name	The name of a function. For example: `function increment`	white	deep sky blue
Gate	The name of a gate. For example: `pullup gate_name(output)`	white	light red
Keyword	All pre-defined VHDL keywords.	white	red
Macro Name	The name of a macro	white	navy blue
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Module Instance Name	The name of an instance of a module.	white	navy blue
Module Name	The name of a module (defined using the *module* keyword). For example: `module Tester`	white	deep sky blue
Net	The name of a net. For example: `wire [7:0] data_out;`	white	purple
Operator	Any operator (colon, assignment, comparison, slash etc.).	white	blue
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
Parameter	An identifier in a *trireg* statement. For example: `trireg small storeit`	white	purple
Port	The name of a port. For example: `input [7:0] data_out`	white	purple

Reg	The name of a variable. For example:
	`integer int_name`
	`reg [25:0] clkdiv`
		white	purple

String	A string enclosed in double quotes. For example: "mystring"	white	turquoise

Table B-5. Verilog'95 and Verilog2005 Syntax Highlighting

Tag	Description	Background	Foreground
Task Name	A task name (defined using the *task* keyword). For example: `task uart_write;`	white	deep sky blue
URL	A unified resource locator anywhere in the text.	white	blue italic

The following additional PSL syntax highlighting tags are also supported when you are using Verilog '95 or Verilog 2005:

PSL Base Verification
PSL Boolean
PSL Constant
PSL Constructs
PSL Declarations
PSL Design Module
PSL Preprocessor Directives
PSL Property
PSL Sequence
PSL End Point
PSL Function
PSL Integer
PSL Integer Type
PSL Label
PSL Parenthesis
PSL Struct Type
PSL System Verilog Keywords
PSL Verification Directive
PSL Verification Unit
PSL Verilog Keywords
PSL VHDL Keyword

XML

The following table lists the default colors for the tags defined when you are using the eXtensible Markup Language (XML):

Table B-6. XML Syntax Highlighting

Tag	Description	Background	Foreground
Code Block	Highlights the code chosen in the code browser.	light green	black
Comment	A comment line (enclosed between <!-- and -->).	white	green
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Keyword	All pre-defined XML keywords.	white	red
Markup	A markup delimiter.	white	navy blue
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black

Table B-6. XML Syntax Highlighting

Tag	Description	Background	Foreground
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
URL	A unified resource locator anywhere in the text.	white	blue italic

SPICE

The following table lists the default colors for the tags defined when you are using the example plug-in which supports analog circuit simulation using SPICE (Simulation Program with Integrated Circuit Emphasis):

Table B-7. SPICE Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: ‘x’	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Commands	A SPICE command	white	dark blue
Comment	A comment line (enclosed between #COM and #endcom).	white	green
Component	Name of a component	white	light red
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Keyword	All pre-defined SPICE keywords.	white	red
LHS Parameter	left hand side parameter in an assignment statement	white	purple
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic

Table B-7. SPICE Syntax Highlighting

Tag	Description	Background	Foreground
Number	A number	white	turquoise
Operator	Any operator (colon, assignment, comparison, slash etc.)	white	blue
Output Variable	An output variable	white	violet
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
Preprocessor Directive	A preprocessor directive	white	navy blue
String	A string within double quotes	white	turquoise
URL	A unified resource locator anywhere in the text.	white	blue italic

Macro

The following table lists the default colors for the tags defined when you are using the Macro plug-in which supports editing DesignPad macros:

Table B-8. Macro Syntax Highlighting

Tag	Description	Background	Foreground
Character	A character enclosed in single quotes. For example: ‘x’	white	turquoise
Code Block	Highlights the code chosen in the code browser.	light green	black
Comment	A comment line (enclosed between #COM and #endcom).	white	green
Enclosed Block	Any text enclosed between (),{} or [].	pale yellow	black
Keyword	All pre-defined SPICE keywords.	white	red
Menu name	DesignPad menu names.	white	purple
Message - Error	Error messages.	red	black
Message - Note	Note messages.	green	black
Message - Warning	Warning messages.	blue	black

Table B-8. Macro Syntax Highlighting

Tag	Description	Background	Foreground
Misspelled Word	Reserved for misspelled words when spelling checker is available.	white	black bold italic
Modifiers	Modifier key	white	red
Operator	Any operator (colon, assignment, comparison, slash etc.)	white	blue
Output Variable	An output variable	white	violet
Overlapping Messages	The overlapping part when error or warning messages overlap.	yellow	black
String	A string within double quotes	white	turquoise
URL	A unified resource locator anywhere in the text.	white	blue italic

SystemVerilog

The following table lists the default colors for the tags defined when you are using the **SystemVerilog** language:

Table B-9. SystemVerilog Syntax Highlighting

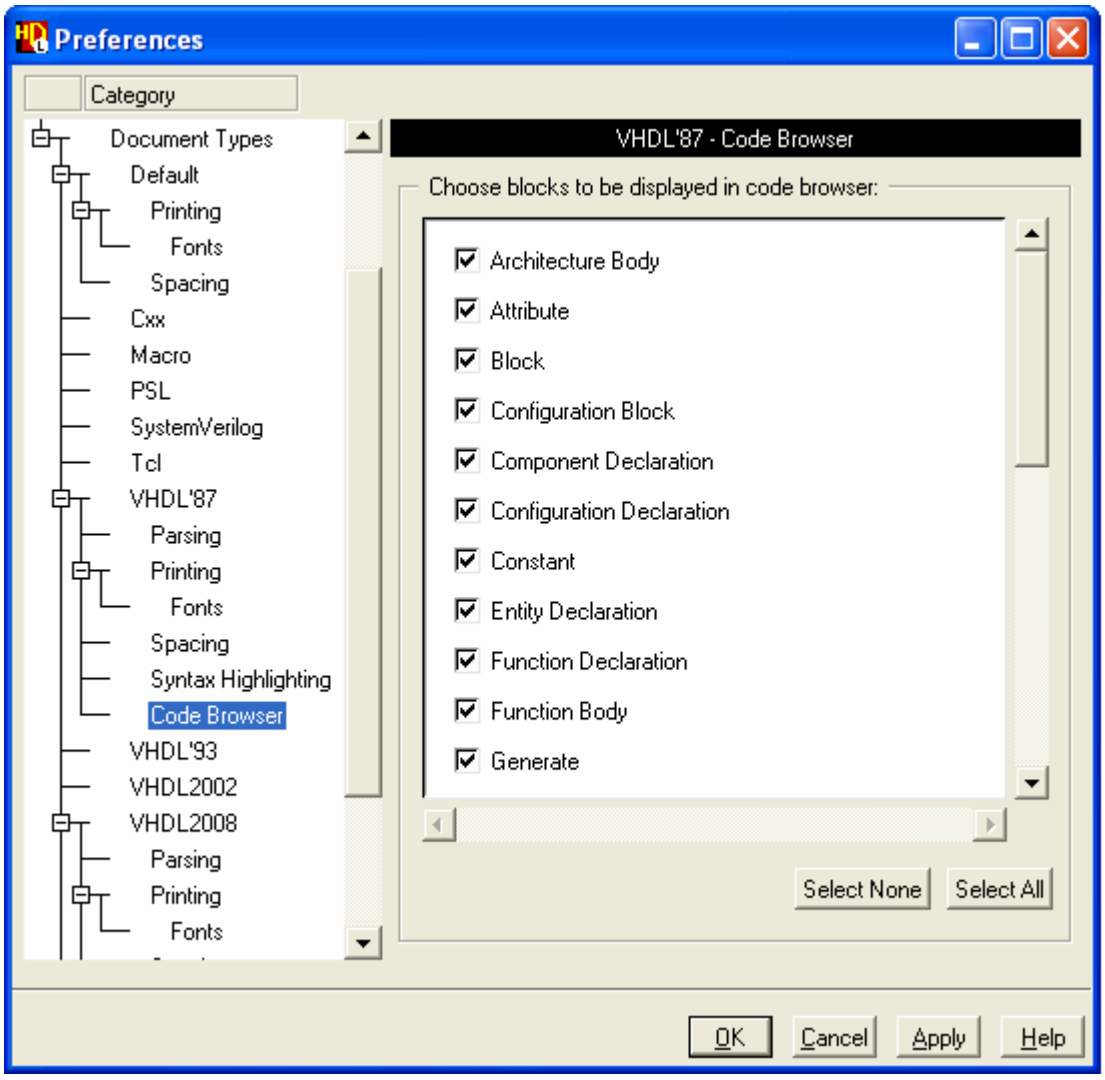
Tag	Description	Background	Foreground
Argument	Argument for a sequence or property.	white	sea blue
Character	Character within single quotes.	white	cyan
Clocking Identifier	Identifier of a clocking statement.	white	light brown
Comment	Commented line (preceded by //) or a commented paragraph (enclosed between "/*" and "*/").	white	green
Compiler Directives	Compiler directive (eg 'define).	white	navy blue
Integer	Integer not inside a string.	white	magenta
Enumerated Type declaration	Enumerated type declaration	white	magenta
Function Name	Name of a Function.	white	light blue
Gate	Name of a Gate.	white	red

Table B-9. SystemVerilog Syntax Highlighting

Tag	Description	Background	Foreground
Interface Name	Name of an interface.	white	orange
Keyword	Verilog 2001 pre-defined keywords.	white	maroon
Macro Name	Name of a Macro.	white	dark blue
Module Instance Name	Name of an Instance of a Module.	white	dark blue
Module Name	Name of a Module.	white	purple
System Verilog Keyword	System Verilog additional keywords.	white	blue
Net	Name of a Net.	white	dark magenta
Operator	Any operator (colon, assignment, comparison, slash,...etc).	white	blue
Package Name	Name of a package.	white	dark magenta
Parameter	Identifier in a "trireg" statement.	white	dark magenta
Port	Name of a Port.	white	dark magenta
Program Identifier	Name of a program.	white	cyan
Property Name	Name of a property.	white	light blue
Sequence Name	Name of a sequence.	white	dark pink
String	String within double quotes.	white	grey
Struct or Union	Name of a struct or union.	white	light blue
Task Name	Name of a Task.	white	dark magenta
Variable Declaration	Name of a Variable.	white	sea blue

Code Browser Content

You can set code browser content preferences separately for the Cxx (C or C+), Tcl, VHDL (‘87, ‘93, 2002 or 2008), Verilog (‘95 or 2005), SystemVerilog, PSL and SPICE document types.



The following icons are used for code blocks in the C or C++ document type:

Table B-10. C and C++ Code Block Icons

Icon	Code Block	Icon	Code Block
	Class		Method
	Function		Namespace

The following icons are used for code blocks in the PSL document type:

Table B-11. PSL Code Block Icons

Icon	Code Block	Icon	Code Block
	Default clock		Verification directive
	End point		Verification mode
	Property		Verification property
	Sequence		Verification unit

The following icons are used for code blocks in the Tcl document type:

Table B-12. Tcl Code Block Icons

Icon	Code Block	Icon	Code Block
	Procedure		Namespace

The following icons are used for code blocks in the VHDL ‘87, VHDL’93, or VHDL 2002 document types:

Table B-13. VHDL’87, VHDL’93 and VHDL 2002 Code Block Icons

Icon	Code Block	Icon	Code Block
	Architecture body		Package body
	Attribute		Package header
	Block statement		Buffer port
	Component declaration		Input port
	Configuration block		Inout port
	Configuration declaration		Port map
	Constant		Output port
	Entity declaration		Procedure declaration
	Function declaration		Procedure body
	Function body		Process
	Generate frame		Signal
	Generic declaration		Subtype
	Generic map		Type
	Instance		Type Body

The following icons are used for code blocks in the Verilog'95 document type:

Table B-14. Verilog'95 Code Block Icons

Icon	Code Block	Icon	Code Block
	Always		Parameter map
	Function		Input port
	Gate instance		Inout port
	Initial		Port map
	Instance		Output port
	External instance		Wire
	Module		Task
	Parameter declaration		

The following icons are used for code blocks in the Verilog2005 document type:

Table B-15. Verilog2005 Code Block Icons

Icon	Code Block	Icon	Code Block
	Always		Module
	Function		Parameter
	Gate instance		Parameter map
	Generate statement		Input port
	Generate variable		Inout port
	Initial		Port map
	Instance		Output port
	External instance		Wire
	Local parameter		Task

The following icons are used for code blocks in the SPICE document type:

Table B-16. SPICE Code Block Icons

Icon	Code Block	Icon	Code Block
	Sub-circuit		Sub-circuit parameter
	Sub-circuit node		

Preference File

Preference settings are saved in a *dp_user* directory at the same location as other HDL Designer Series resource files.

These files are normally saved in the *Profiles* or *Documents and Settings* directory on Windows workstations or in the home directory on UNIX and Linux workstations.

The preference file (*dp_user_prefs*) is written out in a versioned sub-directory when you exit from the application from which DesignPad was invoked.

For example, the following sub-directory is created for version 2.43 preferences when DesignPad is integrated with HDS:

Note

The directory containing the HDS resource files is named *HDL Designer Series* when it is created in the *Profiles* or *Documents and Settings* directory on Windows workstations or *hdl_designer_series* or when it is created elsewhere.

You can use the HDS_USER_HOME environment variable or **-user_home** command line switch to specify an alternative location for the HDL Designer Series user resource files directory. The DesignPad preferences are always written to a *dp_user* directory at the same relative location shown above (provided that you have write permissions to this location).

Refer to the “Resource Files” section in the *HDL Designer Series User Manual* for more information about reading and saving preference files.

Index

— R —

— S —

— T —

End-User License Agreement

The latest version of the End-User License Agreement is available on-line at:
www.mentor.com/eula

IMPORTANT INFORMATION

USE OF ALL SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF SOFTWARE INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.

END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section 2) and hardware (collectively "Products") between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties' entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

1. **ORDERS, FEES AND PAYMENT.**

   1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer's appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement ("Order(s)"), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not these documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order will not be effective unless agreed in writing by an authorized representative of Customer and Mentor Graphics.

   1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice(s). Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer's sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer's behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.

   1.3. All Products are delivered FCA factory (Incoterms 2000), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics' delivery of Software by electronic means is subject to Customer's provision of both a primary and an alternate e-mail address.

2. **GRANT OF LICENSE.** The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation and design data ("Software") are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection 5.2); (b) for Customer's internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee's residence, an airport or hotel, provided that such employee's primary place of employment is the site where the Software is authorized for use. Mentor Graphics' standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer requests any change or enhancement to Software, whether in the course of

receiving support or consulting services, evaluating Software, performing beta testing or otherwise, any inventions, product improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics' sole discretion) will be the exclusive property of Mentor Graphics.

3. **ESC SOFTWARE.** If Customer purchases a license to use development or prototyping tools of Mentor Graphics' Embedded Software Channel ("ESC"), Mentor Graphics grants to Customer a nontransferable, nonexclusive license to reproduce and distribute executable files created using ESC compilers, including the ESC run-time libraries distributed with ESC C and C++ compiler Software that are linked into a composite program as an integral part of Customer's compiled computer program, provided that Customer distributes these files only in conjunction with Customer's compiled computer program. Mentor Graphics does NOT grant Customer any right to duplicate, incorporate or embed copies of Mentor Graphics' real-time operating systems or other embedded software products into Customer's products or applications without first signing or otherwise agreeing to a separate agreement with Mentor Graphics for such purpose.

4. **BETA CODE.**

4.1. Portions or all of certain Software may contain code for experimental testing and evaluation ("Beta Code"), which may not be used without Mentor Graphics' explicit authorization. Upon Mentor Graphics' authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. This grant and Customer's use of the Beta Code shall not be construed as marketing or offering to sell a license to the Beta Code, which Mentor Graphics may choose not to release commercially in any form.

4.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer's use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer's evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.

4.3. Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer's feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection 4.3 shall survive termination of this Agreement.

5. **RESTRICTIONS ON USE.**

5.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer's employees and on-site contractors, excluding Mentor Graphics competitors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use it except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer learns or becomes aware of such unauthorized disclosure or use. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse-assemble, reverse-compile, reverse-engineer or in any way derive any source code from Software. Log files, data files, rule files and script files generated by or for the Software (collectively "Files"), including without limitation files containing Standard Verification Rule Format ("SVRF") and Tcl Verification Format ("TVF") which are Mentor Graphics' proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Software or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Software, or disclose to any third party the results of, or information pertaining to, any benchmark.

5.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized use. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer's employees or contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.

5.3. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense or otherwise transfer the Products, whether by operation of law or otherwise ("Attempted Transfer"), without Mentor Graphics' prior written consent and payment of Mentor Graphics' then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics' prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics' option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms

of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer's permitted successors in interest and assigns.

5.4. The provisions of this Section 5 shall survive the termination of this Agreement.

6. **SUPPORT SERVICES.** To the extent Customer purchases support services, Mentor Graphics will provide Customer updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics' then current End-User Support Terms located at http://supportnet.mentor.com/about/legal/.

7. **AUTOMATIC CHECK FOR UPDATES; PRIVACY.** Technological measures in Software may communicate with servers of Mentor Graphics or its contractors for the purpose of checking for and notifying the user of updates and to ensure that the Software in use is licensed in compliance with this Agreement. Mentor Graphics will not collect any personally identifiable data in this process and will not disclose any data collected to any third party without the prior written consent of Customer, except to Mentor Graphics' outside attorneys or as may be required by a court of competent jurisdiction.

8. **LIMITED WARRANTY.**

8.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Products will meet Customer's requirements or that operation of Products will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates or (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification or improper installation. MENTOR GRAPHICS' ENTIRE LIABILITY AND CUSTOMER'S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY, PROVIDED CUSTOMER HAS OTHERWISE COMPLIED WITH THIS AGREEMENT. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."

8.2. THE WARRANTIES SET FORTH IN THIS SECTION 8 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

9. **LIMITATION OF LIABILITY.** EXCEPT WHERE THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 9 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

10. **HAZARDOUS APPLICATIONS.** CUSTOMER ACKNOWLEDGES IT IS SOLELY RESPONSIBLE FOR TESTING ITS PRODUCTS USED IN APPLICATIONS WHERE THE FAILURE OR INACCURACY OF ITS PRODUCTS MIGHT RESULT IN DEATH OR PERSONAL INJURY ("HAZARDOUS APPLICATIONS"). NEITHER MENTOR GRAPHICS NOR ITS LICENSORS SHALL BE LIABLE FOR ANY DAMAGES RESULTING FROM OR IN CONNECTION WITH THE USE OF MENTOR GRAPHICS PRODUCTS IN OR FOR HAZARDOUS APPLICATIONS. THE PROVISIONS OF THIS SECTION 10 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

11. **INDEMNIFICATION.** CUSTOMER AGREES TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE OR LIABILITY, INCLUDING ATTORNEYS' FEES, ARISING OUT OF OR IN CONNECTION WITH THE USE OF PRODUCTS AS DESCRIBED IN SECTION 10. THE PROVISIONS OF THIS SECTION 11 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

12. **INFRINGEMENT.**

12.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will pay costs and damages finally awarded against Customer that are attributable to the action. Customer understands and agrees that as conditions to Mentor Graphics' obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance

to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.

12.2. If a claim is made under Subsection 12.1 Mentor Graphics may, at its option and expense, (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.

12.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics' licensors who do not provide such indemnification to Mentor Graphics' customers; or (h) infringement by Customer that is deemed willful. In the case of (h), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.

12.4. THIS SECTION 12 IS SUBJECT TO SECTION 9 ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS FOR DEFENSE, SETTLEMENT AND DAMAGES, AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.

13. **TERMINATION AND EFFECT OF TERMINATION.** If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term.

13.1. Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer's obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.

13.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer's possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.

14. **EXPORT.** The Products provided hereunder are subject to regulation by local laws and United States government agencies, which prohibit export or diversion of certain products and information about the products to certain countries and certain persons. Customer agrees that it will not export Products in any manner without first obtaining all necessary approval from appropriate local and United States government agencies.

15. **U.S. GOVERNMENT LICENSE RIGHTS.** Software was developed entirely at private expense. All Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to US FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in this Agreement, except for provisions which are contrary to applicable mandatory federal laws.

16. **THIRD PARTY BENEFICIARY.** Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.

17. **REVIEW OF LICENSE USAGE.** Customer will monitor the access to and use of Software. With prior written notice and during Customer's normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer's software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer's compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FLEXlm or FLEXnet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics' request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section 17 shall survive the termination of this Agreement.

18. **CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION.** The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the United States. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, USA, if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply. Notwithstanding the foregoing, all disputes in Asia arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International

Arbitration Centre ("SIAC") to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. This section shall not restrict Mentor Graphics' right to bring an action against Customer in the jurisdiction where Customer's place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.

19. **SEVERABILITY.** If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.

20. **MISCELLANEOUS.** This Agreement contains the parties' entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements, including but not limited to any purchase order terms and conditions. Some Software may contain code distributed under a third party license agreement that may provide additional rights to Customer. Please see the applicable Software documentation for details. This Agreement may only be modified in writing by authorized representatives of the parties. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.

Rev. 100615, Part No. 246066