PVRShaderEditor

User Manual

Filename	:	PVRShaderEditor.User Manual
Version	:	PowerVR SDK REL_3.1@2284953a External Issue
Issue Date	:	15 Feb 2013
Author	:	Imagination Technologies Limited

Contents

List of Figures

1. Introduction

1.1. Software Overview

1.1.1. PVRShaderEditor

PVRShaderEditor is a shader editor and graphical front-end for the GLSL profiling compilers shader compiler. It currently offers syntax highlighting for GLSL, GLSL ES, HLSL shaders, VGP vertex programs, Microsoft Effects (FX), and PowerVR FX (PFX) files. The Editor is a standalone version of the shader editing functionality that can be found in PVRShaman.

1.1.2. GLSL Profiling Compilers

The GLSL profiling compilers are a series of shader compilers that can be used for gaining profiling information for GLSL and GLSL ES shaders. They provide PVRShaderEditor with profiling output and per line cycle counts to assist in shader development and optimization.

1.2. Document Overview

The purpose of this document is to serve as a complete user manual for PVRShaderEditor. It includes installation instructions, a guide to the functionality of the application, and a complete listing of all interface options and preferences for the GUI.

2. Installation

2.1. From Installer

Download the PowerVR Insider SDK and follow the on screen instructions. Once the package has successfully installed, the application is available in:

```
<InstallDir>\PVRShaderEditor\<PLATFORM>\
```

3. Interface Overview

3.1. Main Interface

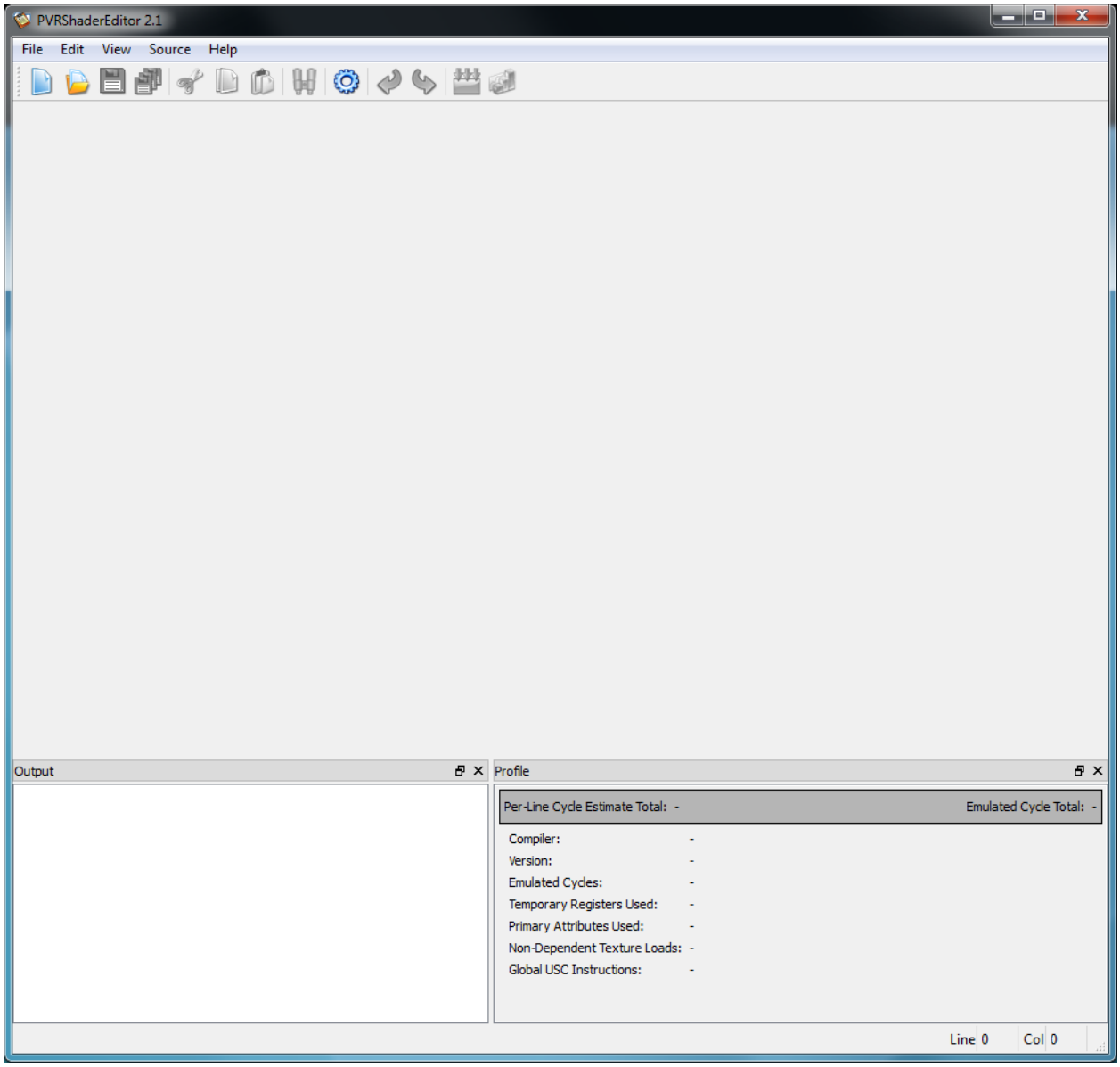

Figure 3-1 Main Interface

3.1.1. Multiple Document Interface

The main interface of PVRShaderEditor consists of three parts, the Effects Editor, the Effects Debug Panel, and the Profile Output Panel. As a tab-based MDI (multiple document interface) application, PVRShaderEditor may have multiple files open simultaneously, displaying a tab for each. In these instances any action performed is performed on the tab that currently has focus.

3.1.2. Effects Editor

```
38  7     if(GeometryCounter == 0)
39        {
40  1       vertex = inVertex + vec3(ripple,ripple2,ripple2);
41        Color = vec4(1.0,0.5,1.0,1);
42        }
43        else
44        {
45        Color = vec4(0.5,0.0,1.0,1);
46  3     vertex = inVertex + vec3(ripple2,ripple,ripple2);
47        }
48
49  5     gl_Position = WorldViewProj * vec4(vertex,1.0);
50  8     Normal = normalize(WorldViewIT * inNormal);
51  3     EyeDir = -vec3(gl_Position);
```

Figure 3-2 Effects Editor

The effects editor is the area used to edit shader code, PFX files, and FX files, as well as plain text files. Open files are arranged as tabs so multiple files can be open at any given time. The editor is syntax highlighted, based on the type of file identified (a setting that can be overridden either selecting the file type from the drop down menu on the bottom bar of the screen.). The effects editor is also the home of the beginning of PVRShaderEditor's profile output.

Cycle Counts

If PVRShaderEditor has the correct compiler settings, and the 'Compile in the background' option is ticked in the preferences, then PVRShaderEditor can display approximate cycle count information for the currently open shader.

Two sets of cycle counts are available:

- per-line cycle counts. In the blue column, next to the line numbers, a line by line breakdown of the cycle cost of the shader will be displayed.
- per-shader cycle counts; these counts are available in the Profile Output Panel and give the overall cost for that shader.

It should be noted that all cycle count values in the effects editor window are approximations intended to assist in identifying areas for optimization. Hardware may be able to reduce these numbers through instruction scheduling etc. that cannot be taken into account by this offline analysis. More accurate information can be found in the Profile Output Panel.

3.1.3. Effects Debug Panel

The Effects Debug Panel contains the compile output of PVRShaderEditor's compiler, either via the 'Compile in the background' option set in Compile Dialog, or through the General Preferences Dialog.

Feedback is broken down per shader with line numbers and error details provided.

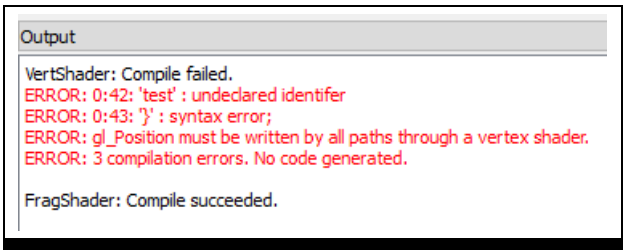

Figure 3-3 Effects Debug Panel

3.1.4. Profile Output Panel

The Profile Output Panel contains more detailed and more accurate profiling information for the shader currently selected from the drop down menu. This information is gained by running the shader through a cycle-accurate emulator.

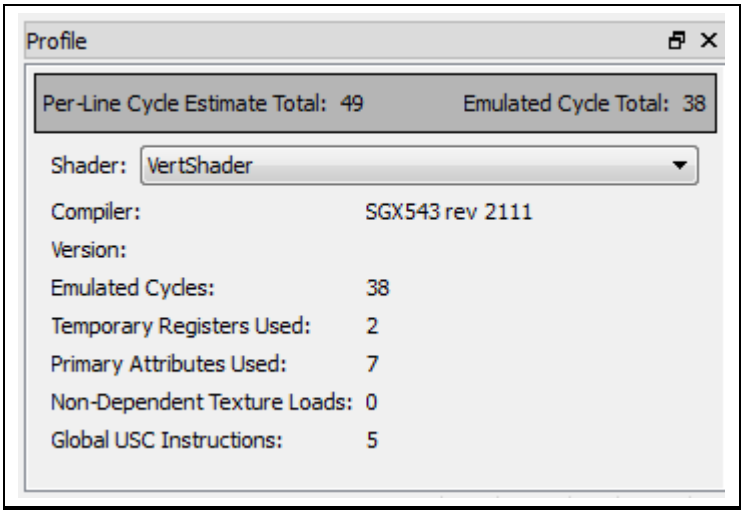

Figure 3-4 Profile Output Panel

Compiler

This value identifies the GPU name targeted by the compiler.

Version

This value shows the compiler version string, where available.

Emulated Cycles Best

This number represents the number of cycles the shader will use when all conditional branches fail and are not processed.

Emulation gives a much more accurate measuring of cycles than the per-line cycle count.

Emulated Cycles Worst

This number represents the number of cycles the shader uses when all conditional branches succeed and are processed.

In many cases the best and worst cycle counts will be identical.

Emulated Cycles

This number represents the number of cycles the shader will us, and is only displayed when there are no conditional blocks present.

Primary Attributes Used

`'Primary Attributes'` are the number of logical input entities. The number of registers consumed by a primary attribute will depend on the number of elements and on its precision.

For indication only; input data storage typically has 1280 registers allocated (varying from platform to platform). These registers are shared by all the shaders being run at any one time. Running out of registers will force reading and writing from external memory which might affect performance.

Temporary Registers

`'Temporary Registers'` are the extra data storage required to process a shader.

As in the case of primary attributes, overflowing the storage allocated might cause degradation in performance. Temporary data storage typically has 384 registers allocated (this number might vary depending on the platform).

Non-Dependent Texture Loads

`'Non-Dependent Texture Loads'` represents the number of texture loads within the selected shader that can be pre-fetched; they are 'independent' of the processing of the shader.

Global USC Instructions

`'Global USC Instructions'` represents the number of instructions that can be extracted and performed once per frame rather than once per use of the shader.

3.2. Menus

3.2.1. File Menu

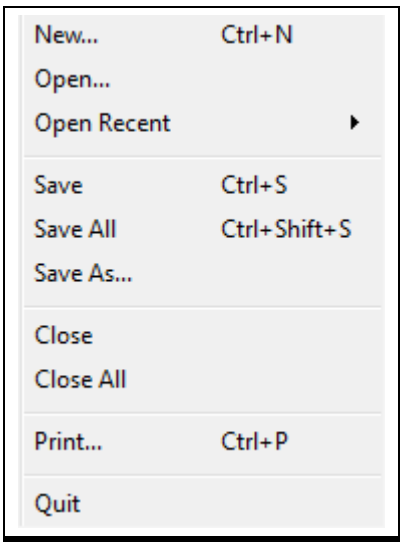

Figure 3-5 File Menu

New

'New…' creates a new text file and opens it in the Effects Editor.

Open

'Open…' opens a file for editing.

Open Recent

'Open Recent' contains a list of the ten most recent files opened, clicking on these files will open them for editing.

Save

'Save' saves the file that currently has focus.

Save All

'Save All' saves all the currently open files.

Save As

'Save As…' saves the file that currently has focus as a new file.

Close

'Close' closes the file that currently has focus.

Close All

'Close All' closes all the currently open files.

Print

'Print…' the file that currently has focus.

Quit

'Quit' closes the application.

3.2.2. Edit Menu

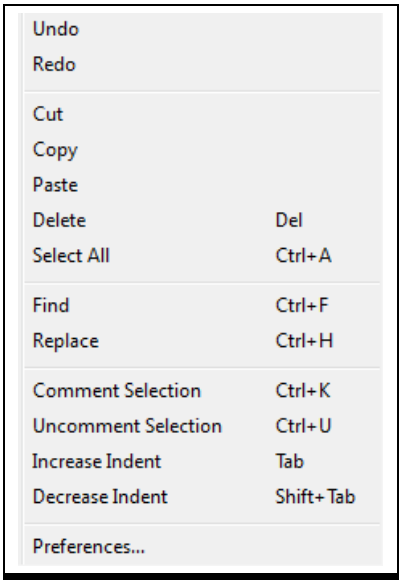

Figure 3-6 Edit Menu

Undo

'Undo' undoes the last performed action.

Redo

'Redo' redoes the last undone action.

Cut

'Cut' cuts the selected text to the clipboard.

Copy

'Copy' copies the selected text to the clipboard.

Paste

'Paste' pastes the contents of the clipboard.

Delete

'Delete' deletes the currently selected text.

Select All

'Select All' selects the entire contents of the file that currently has focus.

Find

'Find' opens the Find Toolbar

Replace

'Replace' opens the Replace Toolbar

Comment Selection

'Comment Selection' comments out the selected text from the file; if a complete line is selected that line will begin with '//'; if part of a line is selected that part will be surrounded in '/* */'.

Uncomment Selection

'Uncomment Selection' removes the commenting from the selected text, either removing a surrounding '/* */' or the '//' at the beginning of the line.

Indent Selection

'Indent Selection' indents the selected text by a single tab.

Outdent Selection

'Outdent Selection' removes an indent from the selected text.

Preferences

'Preferences…' opens the General Preferences Dialog.

3.2.3. View Menu

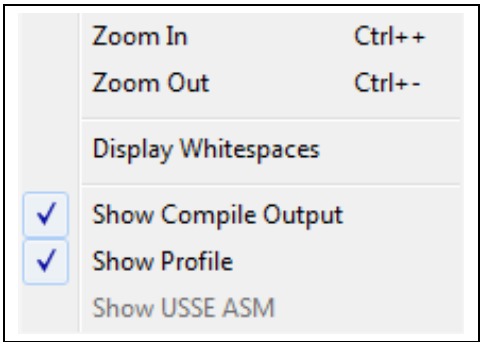

Figure 3-7 File Menu

Zoom In

'Zoom In' enlarges the size of the text in the Effects Editor.

Zoom Out

'Zoom Out' reduces the size of the text in the Effects Editor.

Display Whitespaces

'Display Whitespaces' shows characters in the Effects Editor to illustrate white space characters (i.e. spaces, tabs, etc.).

Show Compile Output

Shows/hides the Effects Debug Panel.

Show Profile

Shows/hides the Profile Output Panel.

Show USSE ASM

Shows/hides the USSE assembly instructions panel. This display of this panel is dependent on the Compiler selected by the user.

Note: Disassembly compilers are available upon signing of an NDA. Contact devtech@imgtec.com for information.

3.2.4. Source Menu

Compile

`'Compile…'` opens the Compile Dialog

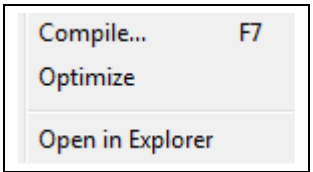

Figure 3-8 Compile Menu

Optimize

`'Optimize'` optimizes the code in a file for better performance.

Open in Explorer/Finder

`'Open in Explorer'` opens the folder which contains the effect file or shader file in Explorer (for Windows) or Finder (for MacOS).

Note: The 'file type' for a file can now be set using the drop down menu found at the bottom of the PVRShaderEditor window.

3.2.5. Help Menu

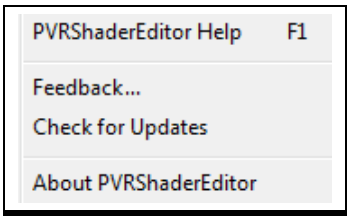

Figure 3-9 Help Menu

PVRShaderEditor Help

`'Help…'` opens this document.

Feedback

`'Feedback…'` opens a panel for giving feedback on the PVRShaderEditor application.

Check for Updates

As of SDK release 3.0 PVRShaderEditor can auto-update. 'Check for Updates' is used to force an update.

About PVRShaderEditor

`'About PVRShaderEditor'` opens a page containing PVRShaderEditor version information, contact details etc.

3.3. Toolbars

3.3.1. Main Toolbar

New File
Opens the new file dialogue, used for creating new effect files or shader files.


Figure 3-10 New File

Open File
Opens an effect file or shader file.

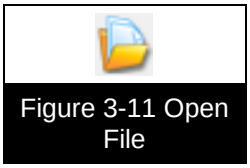
Figure 3-11 Open File

Save File
Saves the file currently open in the editor.

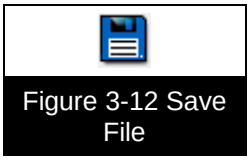
Figure 3-12 Save File

Save All
Saves all files open in the editor.

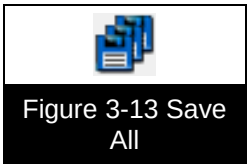
Figure 3-13 Save All

Cut
Cuts the selected text to the clipboard.


Figure 3-14 Cut

Copy
Copies the selected text to the clipboard.

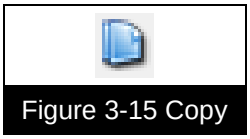
Figure 3-15 Copy

Paste
Pastes text currently in the clipboard.

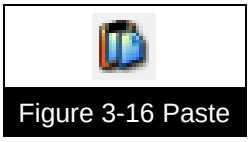
Figure 3-16 Paste

Find
Opens the Find/Replace Toolbar.

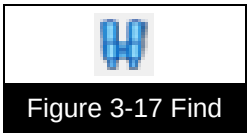
Figure 3-17 Find

Preferences
Opens the General Preferences Dialog.

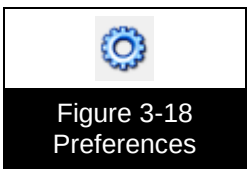
Figure 3-18 Preferences

Undo
Undoes the last action.


Figure 3-19 Undo

Redo
Redoes an action that has been undone.


Figure 3-20 Redo

Compile
Opens the Compile Dialog.


Figure 3-21 Compile

Optimize
Optimizes the effect file or shader file for better performance.


Figure 3-22 Optimize

3.3.2. The Tab Bar

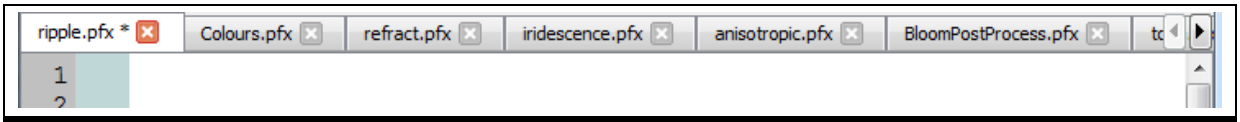

Figure 3-23 The Tab Bar

The tab bar gives quick access to all open shader and profile tabs. The currently shown tab is highlighted. The arrow buttons at the sides of the tab bar allow scrolling left and right if there is not enough space to show all the open tabs at once.

3.3.3. Find/Replace Toolbar

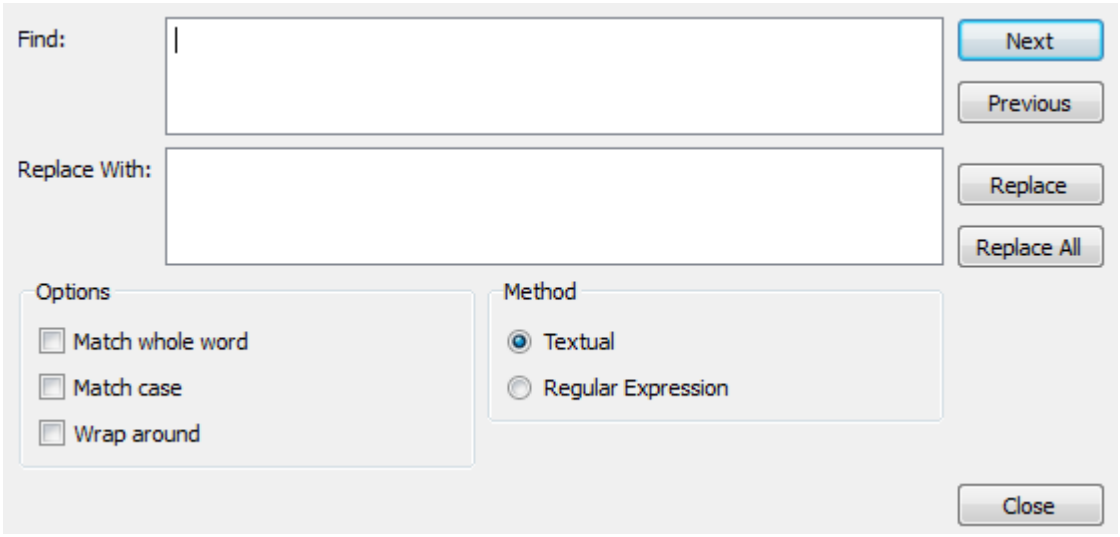

Figure 3-24 Find/Replace Toolbar

The Find and Replace Dialog can be accessed via the `'Edit'` menu, the Effects Editor Toolbar, or by pressing `'Ctrl+F'` (to find) or `'Ctrl+H'` (to replace).

Regular Expressions Syntax

The `'Regular Expression'` method enables the interpretation of regular expressions in the `'Find'` and `'Replace with'` fields. The `'Find'` field is shown in red if the search term is not a complete regular expression.

Information on which regular expressions are supported can be found in Appendix A. Regular Expression Syntax. Back references (`\1` to `\9`) can be used in the `'Replace with'` field as well as the 'Find' field.

3.4. Dialogs

3.4.1. General Preferences Dialog

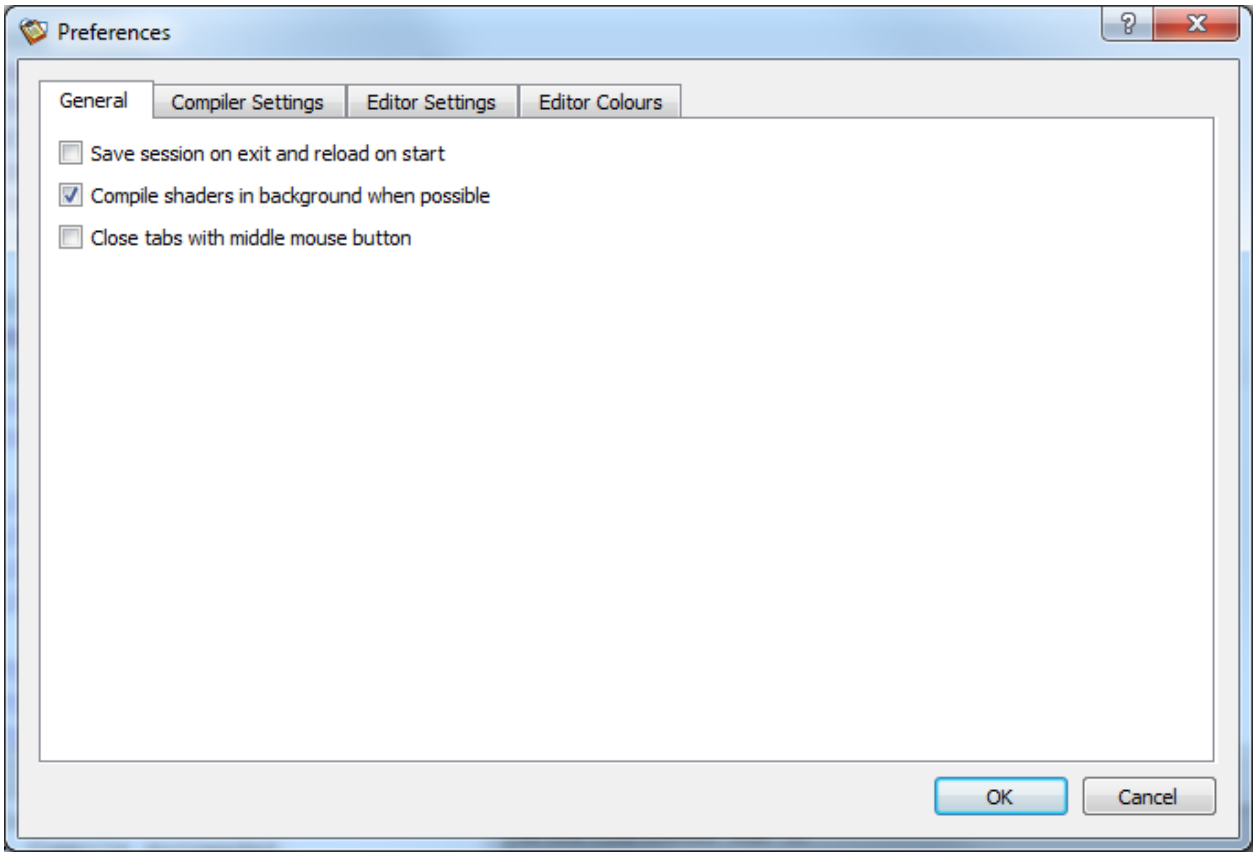

Figure 3-25 General Preferences Dialog

'Save Session on Exit…'

With this option enabled, PVRShaderEditor saves the session on exit and restart. Next time PVRShaderEditor is started the previous session is reloaded.

'Compile Shaders in Background…'

With this option enabled, PVRShaderEditor attempts to compile shaders in the background allowing it to display per-line cycle counts and profiling output as the user types.

'Close Tabs with Middle Mouse Button'

With this option enabled, middle clicking on a tab in the tab bar closes the tab.

3.4.2. Compiler Paths Preferences Dialog

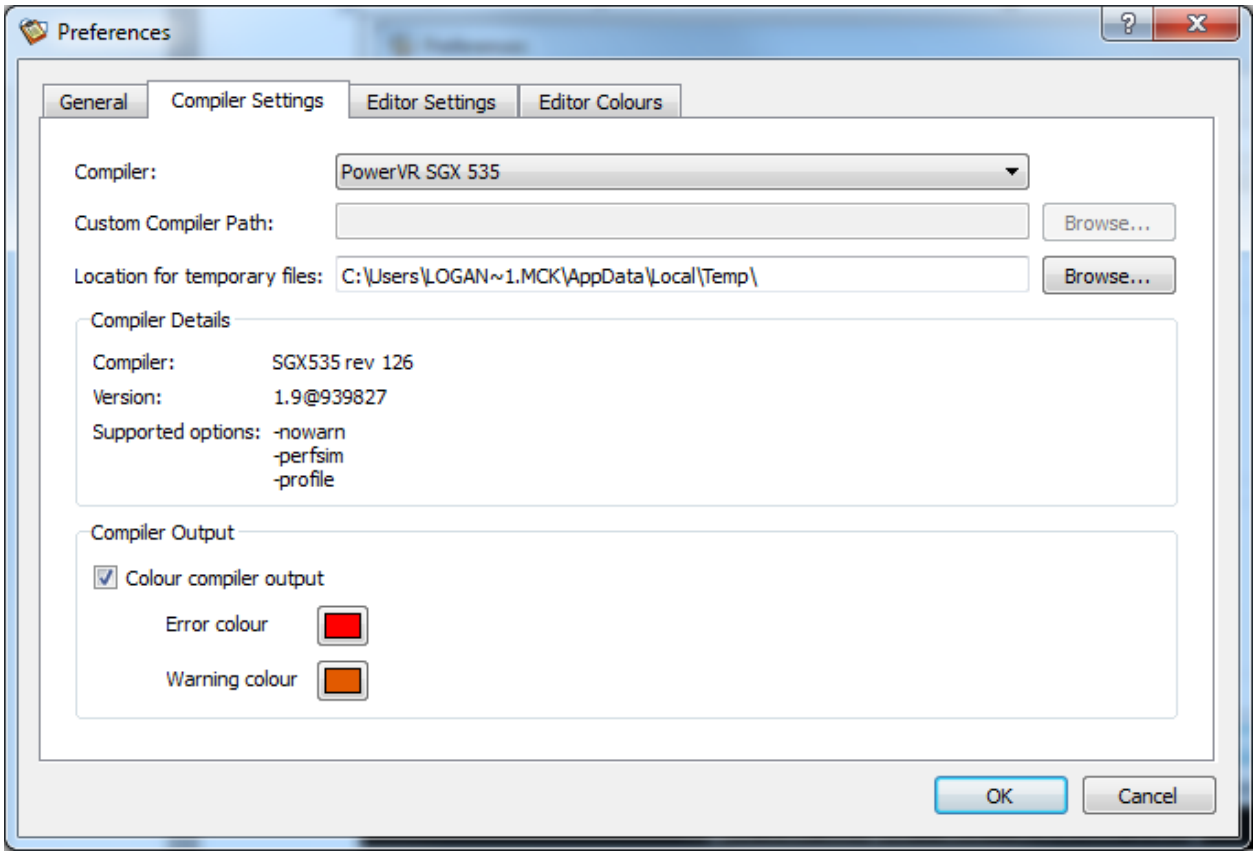

Figure 3-26 Compiler Paths Preferences Dialog

Compiler

'Compiler' shows the compiler currently in use.

Compiler Path

'Custom Compiler Path' shows the path to the compiler currently in use.

Location for Temporary Files

'Location for Temporary Files' specifies the directory where PVRShaderEditor temporarily saves the profiling information.

Colour Compiler Output

With this option ticked the colouration of compiler warnings and errors can be adjusted.

Error Colour

This button opens a colour selection window. The selected colour is used in the Effects Debug Panel when an error occurs.

Warning Colour

This button opens a colour selection window. The selected colour is used in the Effects Debug Panel when a warning occurs.

3.4.3. Editor Settings Preferences Dialog

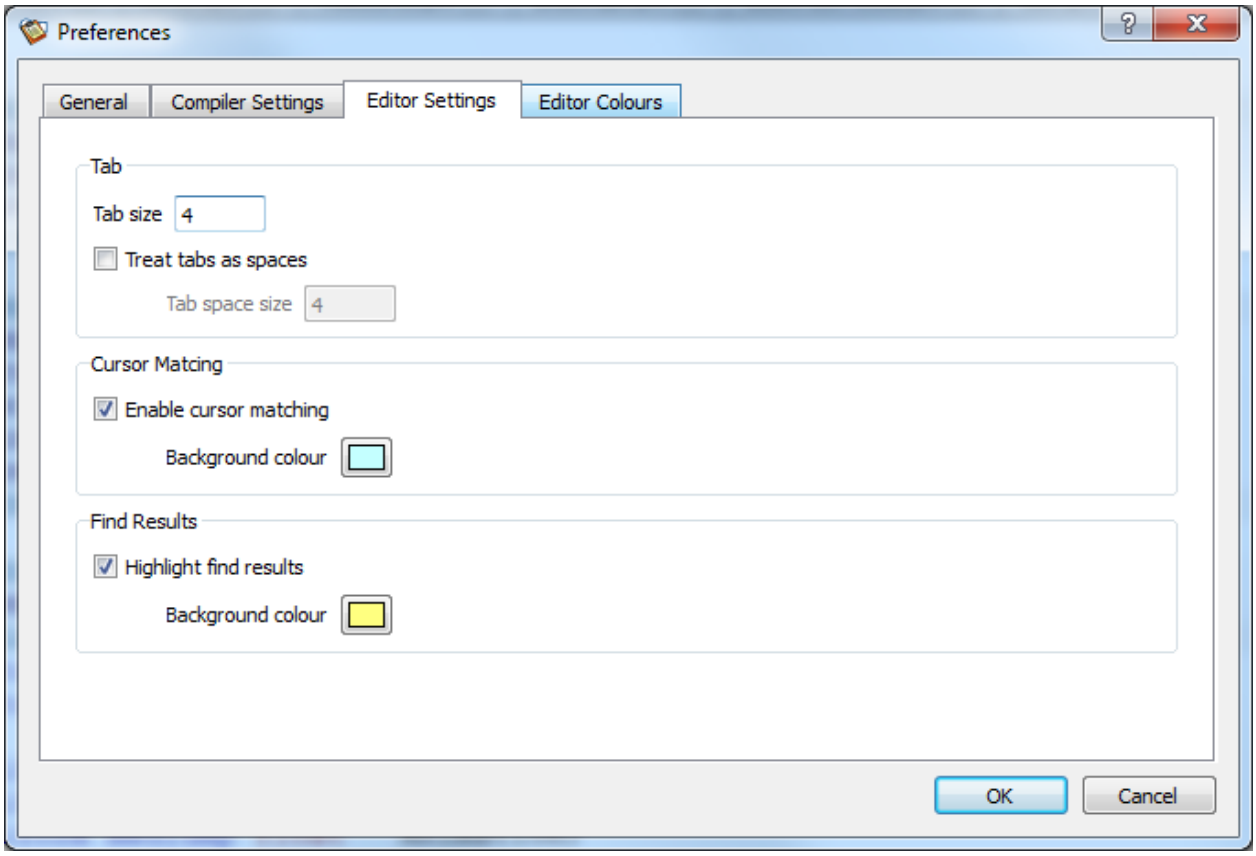

Figure 3-27 Editor Settings Preferences Dialog

Tab

'Treat Tabs as Spaces'

If enabled, PVRShaderEditor treats [tab] as a sequence of spaces.

'Tab Size'

Sets the number of spaces a tab uses if `'Treat Tabs as Spaces'` is ticked.

Cursor Matching

'Enable Cursor Matching'

With this option enabled, all text in the Effects Editor matching the text under the cursor is highlighted.

'Background Colour'

This button opens a colour selection window. The selected colour is used in the Effects Editor for highlighting text highlighted by `'Enable Cursor Matching'`.

Find Results

'Highlight Find Results'

With this option ticked, text that matches the contents of the `'Find'` dialog box is highlighted in the Effects Editor.

'Background Colour'

This button opens a colour selection dialog box. The selected colour is used in the Effects Editor for highlighting text highlighted by `'Highlight Find Results'`.

3.4.4. Editor Colours Preferences Dialog

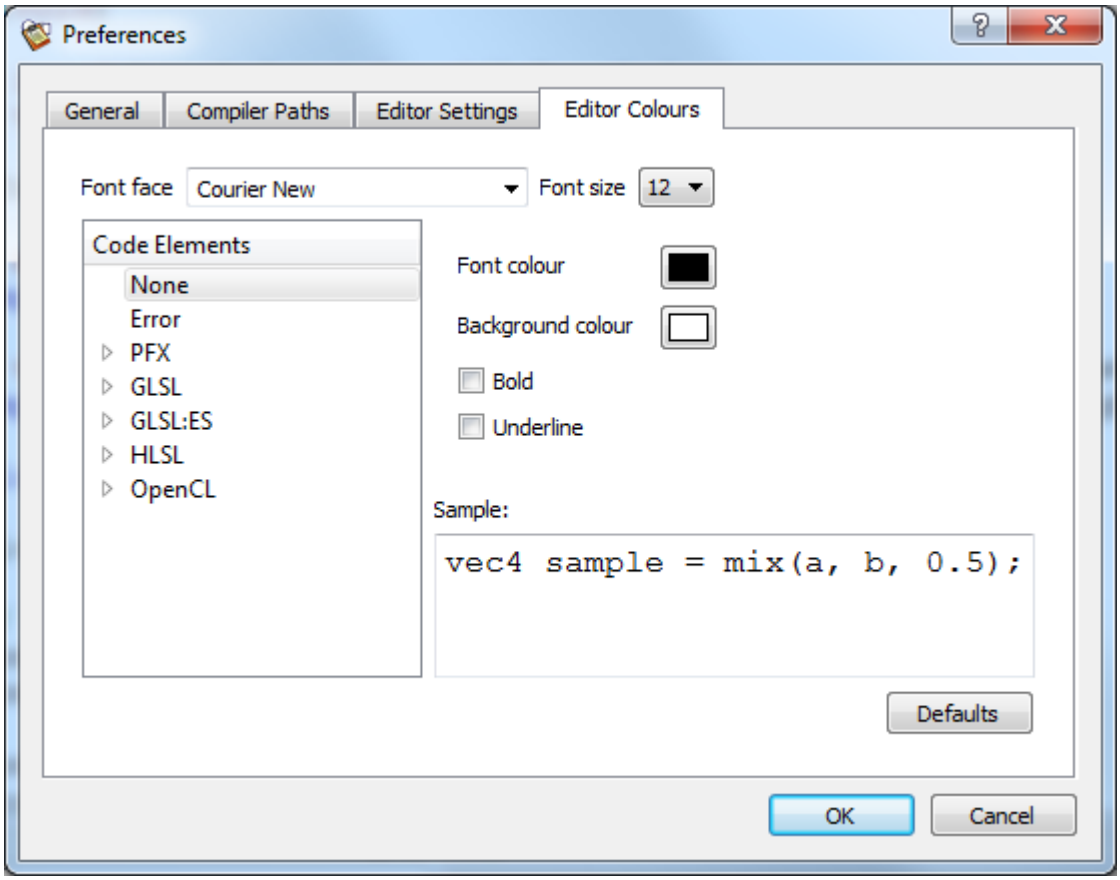

Figure 3-28 Editor Colours Preferences Dialog

This window allows for control of the appearance of the ‘Effects Editor’; everything from background colour to syntax highlighting colours for the various supported languages.

4. Related Materials

Training Courses

- Introducing PFX

Software

- PVRShaman

Documentation

- PVRShaman User Manual
- PFX Language Format Specification

5. Contact Details

For further support contact:

devtech@imgtec.com

PowerVR Developer Technology
Imagination Technologies Limited
Home Park Estate
Kings Langley
Herts, WD4 8LZ
United Kingdom

Tel: +44 (0) 1923 260511
Fax: +44 (0) 1923 277463

Alternatively, you can use the PowerVR Insider forums:

www.imgtec.com/forum

For more information about PowerVR or Imagination Technologies Limited visit our web pages at:

www.imgtec.com

Appendix A. Regular Expression Syntax

Special Constructs	
`(?i X )`	Match sub pattern case insensitive
`(?I X )`	Match sub pattern case sensitive
`(?n X )`	Match sub pattern with newlines
`(?N X )`	Match sub pattern with no newlines
`( X )`	Capturing parentheses (use with back references, see below)
`(?: X )`	Non-capturing parentheses
`(?= X )`	Zero width positive look ahead
`(?! X )`	Zero width negative look ahead
`(?<= X )`	Zero width positive look behind
`(?<! X )`	Zero width negative look behind
`(?> X )`	Atomic grouping (possessive match)
Logical Operators	
`X Y`	X followed by Y
`X | Y`	Either X or Y
Quantifiers	
`X *`	Match 0 or more
`X +`	Match 1 or more
`X ?`	Match 0 or 1
`X {}`	Match 0 or more
`X {n}`	Match n times
`X {,m}`	Match no more than m times
`X {n,}`	Match n or more
`X {n,m}`	Match at least n but no more than m times
These quantifiers are greedy. By following them with '`?`' you can turn them into lazy quantifiers, or follow them by '`+`' for possessive (non-backtracking) quantifiers.	
Boundary Matching	
`^`	Match begin of line [if at begin of pattern]
`$`	Match end of line [if at end of pattern]
`\<`	Begin of word
`\>`	End of word
`\b`	Word boundary
`\B`	Word interior
`\A`	Match only beginning of file
`\Z`	Match only end of file

Character Classes	
`[abc]`	Match a, b, or c
`[^abc]`	Match any but a, b, or c
`[a-zA-Z]`	Match upper- or lower-case a through z
`[]]`	Matches]
`[-]`	Matches -
Predefined Character Classes	
`.`	Match any character
`\d`	Digit [0-9]
`\D`	Non-digit
`\s`	Space
`\S`	Non-space
`\w`	Word character [a-zA-Z_0-9]
`\W`	Non-word character
`\l`	Letter [a-zA-Z]
`\L`	Non-letter
`\h`	Hex digit [0-9a-fA-F]
`\H`	Non-hex digit
`\u`	Single uppercase character
`\U`	Single lowercase character
`\p`	Punctuation (not including '_')
`\P`	Non punctuation
Characters	
`\\`	Back slash character
`\033`	Octal
`\x1b`	Hex
`\t`	Tab
`\n`	Newline
Back References	
`\1` to `\9`	Reference to 1st to 9th capturing group