PFX Language Format

Specification

Filename	:	PFX Language Format.Specification
Version	:	PowerVR SDK REL_17.1@4658063a External Issue
Issue Date	:	07 Apr 2017
Author	:	Imagination Technologies Limited

Contents

1. Introduction 3
1.1. Document Overview 3
1.2. PFX Overview 3
2. Blocks 4
2.1. Header 4
2.2. Texture 4
2.3. Target 6
2.4. Vertex Shader and Fragment Shader 7
2.5. Effect 8
2.6. Deprecated Blocks 9
2.6.1. Textures 9
3. Contact Details 10

List of Tables

Table 1. Keywords (header block) 4
Table 2. Keywords (texture block) 4
Table 3. Values (texture block) 5
Table 4. Keywords (target block) 6
Table 5. Values (target block) 7
Table 6. Keywords (vertex shader and fragment shader blocks) 7
Table 7. Sub-blocks (vertex shader and fragment shader blocks) 7
Table 8. Keywords (effect block) 8
Table 9. Sub-blocks (effect block) 9
Table 10. Values (textures block) 9

1. Introduction

1.1. Document Overview

The purpose of this document is to act as a specification for the PowerVR Effects (PFX) format (PFX specification version 2.0.2).

1.2. PFX Overview

The PFX format is a small, simple, easy to use effects format consisting of several blocks that describe how a given graphics effect is put together (see Section 2). As a minimum, a correctly formatted PFX consists of:

- One `EFFECT` block.
- One `VERTEXSHADER` block.
- One `FRAGMENTSHADER` block.

It is also possible for PFXs to contain the following:

- One `TARGET` block.
- Zero or more `TEXTURE` blocks.

By default PFXs are stored in PFX files. It is possible for multiple PFXs to exist within a single PFX file, each described by a separate effect block. In this instance multiple PFXs may share blocks. Finally, it is possible for a PFX to reference a `TARGET` block as an input as if it were a `TEXTURE` block, enabling the simple creation of complex post-processing effects. For this to function correctly the `TARGET` block render should be completed prior to being read as an input. If the `TARGET` block render has not been completed prior to being read as an input, the behaviour will vary based on the render target implementation of the platform.

2. Blocks

2.1. Header

The header block contains metadata used for description and labelling purposes (see Table 1 for keywords).

Table 1. Keywords (header block)

Keyword	Description
VERSION	The version of this PFX. The format follows this convention: MAJOR.MINOR.BUILD.REVISION
DESCRIPTION	A plain-text description of what this PFX file contains and the desired effect.
COPYRIGHT	Copyright descriptor of author(s).

2.2. Texture

A TEXTURE block describes a surface that can either be the contents of a texture file or the contents of a framebuffer/render texture (see Table 2 and Table 3 for keywords and values, respectively).

Table 2. Keywords (texture block)

Keyword	Description
NAME	A text identifier for this texture.
PATH	The filename of the texture. If spaces are included, enclose the path in quotation marks, e.g., "base map.pvr".
MINIFICATION	Minification texture filter flags. Valid values: • NEAREST • LINEAR
MAGNIFICATION	Magnification texture filter flags. Valid values: • NEAREST • LINEAR
MIPMAP	MIP-map texture filter flags. Valid values: • NEAREST • LINEAR • NONE
VIEW	Specifying the VIEW keyword modifies the functionality of the defined texture to be a render texture of the current scene. Valid values: • PFX_CURRENTVIEW • POD camera name in optional quotation mark, e.g., "Camera01".
CAMERA	An alias for VIEW.

Keyword	Description
RESOLUTION	Describes the resolution of the texture. This will be ignored if PATH is specified, where the resolution will be that of the loaded texture.
WRAP_*x*	Where *x* is a valid axis (S, T, or R). Specifies the texture wrapping in the defined axis. Valid values: • REPEAT • CLAMP
SURFACETYPE	Describes the surface/pixel type of the texture. This will be ignored if PATH is specified, where the surface type will be that of the loaded texture. Valid values: • RGBA8888 • RGBA4444 • RGB888 • RGB565
FILTER	[Deprecated] Allows the specification of texture filter flags in short-hand. Valid values are hyphen separated filter flags in the order of minification, magnification, and MIP-map, e.g., LINEAR-LINEAR-LINEAR enables trilinear texture filtering, or LINEAR-LINEAR-NONE enables bilinear.
WRAP	[Deprecated] Allows the specification of texture wrapping flags in short-hand. Valid values are hyphen separated wrapping flags in the order of axis S, T, and R., e.g., REPEAT-CLAMP-CLAMP.

Table 3. Values (texture block)

Value	Associated Keyword	Description
PFX_CURRENTVIEW	VIEW CAMERA	Indicates that the render texture should be derived from the current view of the scene and not from a specific POD file camera.
LINEAR	MINIFICATION MAGNIFICATION MIP-MAP	Linearly interpolates between sampled texels.
NEAREST		Chooses nearest texel based on Manhattan distance.
NONE	MIP-MAP	Disable MIP-mapping.
CLAMP	WRAP_S WRAP_T WRAP_R	Clamps to texture border.
REPEAT		Repeats at texture border.
RGBA8888	SURFACETYPE	32bit RGBA texture format.
RGBA4444		16bit RGBA texture format.
RGB888		24bit RGB texture format.
RGB565		16bit RGB texture format.

Value	Associated Keyword	Description
INTENSITY8		8bit intensity texture format.

2.3. Target

A TARGET block specifies a surface that an EFFECT block can render to. A PFX may read from a TARGET block as if it were a TEXTURE block as long as the render to that block has been completed prior to the read (see Table 4 and Table 5 for keywords and values, respectively).

Table 4. Keywords (target block)

Keyword	Description
NAME	A text identifier for this target.
MINIFICATION	Minification texture filter flags. Valid values: • NEAREST • LINEAR
MAGNIFICATION	Magnification texture filter flags. Valid values: • NEAREST • LINEAR
MIPMAP	MIP-map texture filter flags. Valid values: • NEAREST • LINEAR • NONE
RESOLUTION	Describes the resolution of the texture. This will be ignored if PATH is specified, where the resultant resolution will be that of the loaded texture.
WRAP_*x*	Where *x* is a valid axis (S, T, or R). Specifies the texture wrapping in the defined axis. Valid values: • REPEAT • CLAMP
SURFACETYPE	Describes the surface/pixel type of the texture. This will be ignored if PATH is specified, where the resultant surface type will be that of the loaded texture. Valid values: • RGBA8888 • RGBA4444 • RGB888 • RGB565

Table 5. Values (target block)

Value	Associated Keyword	Description
PFX_CURRENTVIEW	VIEW CAMERA	Indicates that the render texture should be derived from the current view of the scene and not from a specific POD file camera.
LINEAR	MINIFICATION MAGNIFICATION MIP-MAP	Linear interpolates between sampled texels.
NEAREST		Chooses nearest texel based on Manhattan distance.
NONE	MIP-MAP	Disable MIP-mapping.
CLAMP	WRAP_S WRAP_T WRAP_R	Clamps to texture border.
REPEAT		Repeats at texture border.
RGBA8888	SURFACETYPE	32bit RGBA texture format.
RGBA4444		16bit RGBA texture format.
RGB888		24bit RGB texture format.
RGB565		16bit RGB texture format.
INTENSITY8		8bit intensity texture format.

2.4. Vertex Shader and Fragment Shader

VERTEXSHADER and FRAGMENTSHADER blocks are the location in which the GLSL code for an effect is located. The code can either be embedded within the PFX file itself or be located elsewhere and referenced with a filename using the FILE keyword (see Table 6 and Table 7 for keywords and sub-blocks, respectively).

Table 6. Keywords (vertex shader and fragment shader blocks)

Keyword	Description
NAME	A unique identifier for this shader.
FILE	A file name of a text file containing valid GLSL code.

Table 7. Sub-blocks (vertex shader and fragment shader blocks)

Block	Description
GLSL_CODE	A block containing plain-text GLSL code.

2.5. Effect

The EFFECT block is the primary block used in describing a PFX. It references other blocks which can contain textures, targets and shaders as well as containing a number of application-specific 'semantics' which can be used by an application to identify the meaning of a given attribute (see Table 8 and Table 9 for keywords and sub-blocks, respectively).

Table 8. Keywords (effect block)

Keyword	Description
NAME	A text identifier for this effect.
ATTRIBUTE	Specifies GLSL attribute variable. Format: `ATTRIBUTE varName SEMANTIC` • varName references a variable as specified in the shader blocks. • SEMANTIC references an application specific semantic.
UNIFORM	Specifies GLSL uniform variable. Format: `UNIFORM varName SEMANTIC` • varName references a variable as specified in the shader blocks. • SEMANTIC references an application-specific semantic.
TEXTURE	Specifies a texture name which will be bound to the given unit. Format: `TEXTURE UNIT TextureName` • UNIT specifies an integer-based texture unit to bind to. • TextureName references a TEXTURE block of a given name.
TARGET	Specifies a target which this effect will write to, instead of the default frame buffer. Two types of targets exist, colour targets and depth targets. Only colour target support is required to be compliant with this specification. Format: `TARGET BUFFERTYPE<UNIT> TargetName` • BUFFERTYPE can be of type COLOR or, optionally, DEPTH. • UNIT is an integer based value defining the buffer unit. Only 0 is required to be compliant with this specification. • TargetName references a TARGET block of a given name.
VERTEXSHADER	References a VERTEXSHADER block of a given name.
FRAGMENTSHADER	References a FRAGMENTSHADER block of a given name.

Table 9. Sub-blocks (effect block)

Block	Description
ANNOTATION	A block containing plain-text which will be copied as-is into a text buffer, readable by the application.

2.6. Deprecated Blocks

The following blocks are now deprecated and will be removed in a future version of the specification and, therefore, their use is discouraged.

2.6.1. Textures

The TEXTURES block has been replaced with multiple TEXTURE blocks, which allows the specification of individual textures in a more verbose manner. Each line of the TEXTURES block takes the form identified next (see Table 10 for values).

```
FILE    TextureName    FileName.pvr    FILTERFLAGS WRAPFLAGS
```

Table 10. Values (textures block)

Value	Description
TextureName	Specifies a text identifier.
FileName.pvr	Specifies the filename of the texture.
FILTERFLAGS	Specifies a hyphen separated list of texture filter flags (see Section 2.2).
WRAPFLAGS	Specifies a hyphen separated list of texture wrapping flags (see Section 2.2).

3. Contact Details

For further support, visit our forum:
http://forum.imgtec.com

Or file a ticket in our support system:
https://pvrsupport.imgtec.com

To learn more about our PowerVR Graphics SDK and Insider programme, please visit:
http://www.powervrinsider.com

For general enquiries, please visit our website:
http://imgtec.com/corporate/contactus.asp