Filewrap

User Manual

Filename	:	Filewrap.User Manual.1.6f.External.doc
Version	:	1.6f External Issue (Package: POWERVR SDK 2.09.29.0656)
Issue Date	:	15 Sep 2011
Author	:	Gordon MacLachlan

Contents

List of Figures

1. Introduction

1.1. Software Overview

Filewrap is a tool used to convert a resource file of an application into a source/header file that can be integrated into that application. This allows executable and resources to be combined in to a single file for distribution or obfuscation purposes. It can also be used to create a memory file system for instances where no file system is present.

1.2. Document Overview

The purpose of this document is to serve as a complete user manual for Filewrap. It includes installation instructions, a guide to the functionality of the application, and a complete listing of all command-line options.

2. Installation

Filewrap is a standalone executable that is available as part of the PowerVR Insider SDK which can be downloaded from the PowerVR Insider website.

2.1.1. From Installer

Download one of the PowerVR Insider SDKs and run the installer following the on screen instructions. Once the package has successfully installed, the application will be available in the `SDK folder`.

2.1.2. From GZIP

Download the PowerVR Insider SDK. Unzip the .tar.gz file, and then untar the .tar file. From the ensuing folder, browse to:

```
<SDK_ROOT>\Utilities\Filewrap\<PLATFORM>\
```

This folder will contain the Filewrap executable.

3. Usage Instructions

3.1. Wrapping a File

Wrapping a file is performed as follows:

```
Filewrap –o OutFile.cpp inputFile1 inputFile2 inputFile3
```

This example shows three input files being wrapped. Multiple input files can be wrapped and output to a single output file safely as loading is based on the original file name not the output file name; as these original file names serve as the index in the memory file system it is important that they are unique.

3.2. Command-Line Options

-o[a] outFile

Outputs [by appending] to outFile.

e.g –oa OutFile.cpp InputFile.txt will wrap InputFile.txt and append it to OutFile.cpp.

-b

Do not append 0 byte (for binary data only).

-s

String literal mode

-h

Create header file (doesn't register file in the memory file system).

-be

Use big endian mode (default is little endian).

3.3. Compile Time Wrapping

The wrapping of files can be automated at compile time using makefiles or custom build steps (in Visual Studio for example). An example of this is given below.

3.3.1. Example: Visual Studio

- Add the file you wish to be wrapped to your project.
- Select the file in the solution explorer.
- Right click the file and select properties.
- Click on Custom Built Tool.
- Set 'Command Line' to:
  `Filewrap.exe –o "$(InputDir)Outputfile.cpp" "$(InputPath)"`
- Set 'Outputs' to:
  `$(InputDir)Outputfile.cpp`

This process can also be automated by editing the Visual Studio solution file with a script thereby ensuring that it does not need to be done for each individual file that is to be wrapped.

3.4. Loading a File

All file read access in the PVRTools library uses the `CPVRTResourceFile` class. When loading a file, `CPVRTResourceFile` first looks in the read path set by the application before checking for the file by name in the memory file system. This order of precedence allows files that have been wrapped and linked into the executable to be overridden.

An example of reading a file using the `CPVRTResourceFile` class can be seen below:

```
	// Get and set the read path for content files
	CPVRTResourceFile::SetReadPath((char*)PVRShellGet(prefReadPath));

	// Get and set the load/release functions for loading external files.
	// In the majority of cases the PVRShell will return NULL function pointers
	// implying that nothing special is required to load external files.
	CPVRTResourceFile::SetLoadReleaseFunctions(  PVRShellGet(prefLoadFileFunc),
	                    PVRShellGet(prefReleaseFileFunc) );

	// Set the error string, and the file to be opened
	CPVRTString* const pReturnError = "";
	const char* const pszSrcFile = "fileName.txt"

	CPVRTResourceFile TextFile(pszSrcFile);
	if (!TextFile.IsOpen())
	{
		*pReturnError += CPVRTString("Failed to open file ") + pszSrcFile + "\n";
		return PVR_FAIL;
	}

	const char* pTextData = (const char*) TextFile.DataPtr();
```

It is important to note that when reading a file from the memory file system that only the original (pre-wrapping) file name is used, it is thus very important that these filenames be unique even if multiple are stored within a single wrapped file. Finally, in order to access the `CPVRTResourceFile` class and for it to function correctly the application must be linked to the PVRTools library for the target API.

4. Contact Details

For further support contact:

devtech@imgtec.com

PowerVR Developer Technology
Imagination Technologies Ltd.
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 Ltd. visit our web pages at:

www.imgtec.com