
Khronos API Implementers Guide

Edited by Mark Callow

Copyright © 2008 The Khronos Group Inc.

This document is protected by copyright laws and contains material proprietary to the Khronos Group, Inc. It or any components may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of Khronos Group. The receipt or possession of this document does not convey any rights to reproduce, disclose, or distribute its contents, or to manufacture, use, or sell anything that it may describe, in whole or in part.

Khronos Group grants express permission to any current Promoter, Contributor or Adopter member of Khronos to copy and redistribute UNMODIFIED versions of this document in any fashion, provided that NO CHARGE is made for the document and the latest available update of the document is used whenever possible. Such distributed document may be re-formatted AS LONG AS the contents of the document are not changed in any way. The document may be incorporated into a product that is sold as long as such product includes significant independent work developed by the seller. A link to the current version of this document on the Khronos Group web-site should be included whenever possible with document distributions.

Khronos Group makes no, and expressly disclaims any, representations or warranties, express or implied, regarding this document, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose or non-infringement of any intellectual property. Khronos Group makes no, and expressly disclaims any, warranties, express or implied, regarding the correctness, accuracy, completeness, timeliness, and reliability of the document. Under no circumstances will the Khronos Group, or any of its Promoters, Contributors or Members or their respective partners, officers, directors, employees, agents or representatives be liable for any damages, whether direct, indirect, special or consequential damages for lost revenues, lost profits, or otherwise, arising from or in connection with these materials.

Khronos™ is a trademark of The Khronos Group Inc. OpenGL® is a registered trademark, and OpenGL ES™ is a trademark, of Silicon Graphics, Inc.

Revision History		
Revision 1.01.0	2008-01-11 17:00:45 +0900	msc
	Added brief note about OpenKODE extensions.	
Revision 1.00.0	2007-12-28 12:47:36 +0900	msc
	First Official Release. Added eglx.h & links to Khronos-provided header files; removed draft status.	
Revision 0.92.1	2007-11-29 16:44:09 +0900	msc
	Added links to extension and enumerant rules and reduced repetition even more.	
Revision 0.92.0	2007-11-29 12:30:00 +0900	msc
	Added glex.h & stronger statements about not modifying standard header files; reworked header guidelines to avoid repetition.	
Revision 0.91.0	2006-12-12 17:26:03 -0800	msc
	Public review version: amended copyright notice; changed OpenKODE and GLES utility library names.	
Revision 0.90.0	2006-12-08 19:15:11 -0800	msc
	First docbook version.	

Abstract

Records guidelines to be followed by implementers of Khronos Group *APIs*.

Table of Contents

Introduction	1
General Guidelines	2
Vendor Controlled Platforms (e.g. BREW, SymbianOS)	2
Uncontrolled Platforms (e.g. GNU/Linux, Windows, Windows CE)	2
Acknowledgements	7
Glossary	7

Introduction

This document provides guidelines for *implementers* of OpenGL ES, OpenVG and other *API* standards specified by the Khronos Group. The aim of these hints is to provide commonality between implementations to ease the logistical problems faced by developers using multiple different implementations of an *API*.

One of the primary goals is to allow an application binary to run on top of multiple different OpenGL ES / OpenVG / EGL implementations on the same platform.

Implementers are strongly urged to comply with these guidelines.

General Guidelines

Vendor Controlled Platforms (e.g. BREW, SymbianOS)

Implementers should follow the linkage specifications established by the *platform vendor*.

- Use the header files, (e.g., for OpenGL ES, `gl.h` & `egl.h`) provided by the platform vendor.
- Use the function names specified in those header files.
- Implement all *API* entry points in the same way as in the *vendor*-provided *ABI*. That is, functions should be functions, in-line functions should be in-line functions and macros should be macros.
- Use the platform specified library names.

Vendors of controlled platforms are strongly urged to follow the recommendations given below for [Uncontrolled Platforms](#) when adding a Khronos Group *API* to their platform.

Uncontrolled Platforms (e.g. GNU/Linux, Windows, Windows CE)

When providing implementations for platforms where the vendor does not provide standard linkage specifications, implementers are urged to follow the following recommendations.

Header Files

General

- Implementers are strongly encouraged to use the standard header files (`egl.h`, `gl.h`, `kd.h`, etc.) provided by Khronos for each specification. Links are provided in [Table 1, “Header File Names and Locations”](#). Portable and non-portable definitions are separated into `<api>.h` and `<api>platform.h` files. The intention is that implementers do not need to change the former and they are strongly discouraged from doing so. Consider contributing any header file changes back to Khronos by updating the files in the Khronos Subversion tree so that others may benefit from your expertise.
- If you make your own header files use the names given in [Table 1, “Header File Names and Locations”](#).
- If the platform is Windows, make sure your header files are suitable for use with *MFC*. For example `#define EGL_DEFAULT_DISPLAY GetDC(0)` is broken for MFC. You need to use `::GetDC(0)` because several Microsoft Foundation Classes have their own `GetDC(void)` methods.
- When including one header in another, include the parent directory name. For example when including `eglplatform.h` in `egl.h` use `#include <EGL/eglplatform.h>`. Do not use `#include <eglplatform.h>` because application makefiles will then need 2 different `-I<path>` options to find both include files.

- Functions and enumerants for implementer extensions should be declared and defined in an implementer's own header file. Follow the extension writing and naming rules given in [How to Create Khronos API Extensions](#)¹. Use enumerant values obtained from the Khronos Extension Registry, as explained in [OpenGL Enumerant Allocation Policies](#)².

EGL

- Function declarations and constant definitions for EGL are divided into 2 header files: `egl.h` declares and defines all the core functions and constants; `egl_ext.h` declares and defines functions and constants for all Khronos approved extensions and can be used even if the implementation doesn't provide a particular extension.
- Platform dependent and non-portable items are in `eglplatform.h`, which needs to be modified by the implementer. In particular the `eglNativeDisplayType`, `eglNativeWindowType`, and `eglNativePixmapType` typedefs must be defined as appropriate for the platform (typically, they will be typedef'ed to corresponding types in the native window system). Developer documentation should mention the correspondence so that developers know what parameters to pass to `eglCreateWindowSurface`, `eglCreatePixmapSurface`, and `eglCopyBuffers`. Documentation should also describe the format of the `display_id` parameter to `eglGetDisplay`, since this is a platform-specific identifier.
- Package these and any header files for implementer extensions in the folder EGL.
- Do not include `gl.h` in `egl.h`.

OpenGL ES 1.x

- As with EGL, the function declarations and constant definitions for OpenGL ES 1.x are divided into 2 header files: `gl.h` and `glext.h`.
- Platform dependent and non-portable declarations are in `glplatform.h`. This file may need to be modified by the implementer.
- Package these and any header files for implementer extensions in the folder GLES.
- For compatibility with GLES 1.0 implementations, include in GLES a special `egl.h`³ containing the following:

```
#include <EGL/egl.h>
#include <GLES/gl.h>
```

This is because many early OpenGL ES 1.0 implementations included `gl.h` in `egl.h` so many existing applications only include `egl.h`.

- The name `glu.h` is reserved for future use by the Khronos Group.

OpenGL ES 2.x

- The organization of the header files for OpenGL ES 2.x mirrors that for OpenGL ES 1.x, except that the files are called `gl2.h`, `gl2ext.h` & `gl2platform.h` to keep them distinct from the GLES 1.x header files and enable both versions to coexist on a platform.
- Package these and any header files for implementer extensions in the folder GLES2.

¹ <http://www.khronos.org/registry/doc/rules.html>

² <http://www.khronos.org/registry/doc/enums.html>

³ <http://www.khronos.org/registry/gles/api/1.1/egl.h>

- The name `glu2.h` is reserved for future use by the Khronos Group.

OpenKODE

- Function declarations and constant definitions for OpenKODE are in `kd.h`; platform dependent and non-portable declarations are in `kdplatform.h`.
- As noted earlier, implementers are strongly encouraged to use the Khronos provided header files. Implementers, who are creating their own `kd.h` or need to modify `kdplatform.h`, are encouraged to code them such that they include as few as possible of the platform's include files, and to avoid declaring C and POSIX standard functions. This will ease the creation of portable OpenKODE applications, and help stop non-portable code being added accidentally.
- Each OpenKODE extension is defined in its own header file. Khronos provided header files for each ratified extension are available in the *Extension Headers* subsection of the [OpenKODE registry](#)⁴.
- Package all these OpenKODE header files in the folder `KD`.

OpenVG

- Package the OpenVG header files `openvg.h` and, when provided, `vgu.h` in the folder `VG`.

⁴ <http://www.khronos.org/registry/kode/>

Header File Summary

Table 1. Header File Names and Locations

API	Location	Header Files	How to include	Provider
EGL 1.x	EGL	egl.h ⁵	#include <EGL/egl.h>	Khronos
		eglplatform.h ^{6 a}	Included by egl.h	Khronos, probably modified by Vendor or Implementer
		eglext.h ⁷	#include <EGL/eglext.h>	Khronos
OpenGL ES 1.x	GL ES	gl.h ⁸	#include <GL ES/gl.h>	Khronos
		glplatform.h ^{9 b}	Included by gl.h	Khronos, possibly modified by Vendor or Implementer
		glext.h ¹⁰	#include <GL ES/glext.h>	Khronos
		glu.h	Reserved for future use	
OpenGL ES 2.x	GL ES2	gl2.h ¹¹	#include <GL ES2/gl2.h>	Khronos
		gl2platform.h ¹²	Included by gl2.h	Khronos, possibly modified by Vendor or Implementer
		gl2ext.h ¹³	#include <GL ES2/gl2ext.h>	Khronos
		glu2.h	Reserved for future use	
OpenKODE 1.x	KD	kd.h ¹⁴	#include <KD/kd.h>	Khronos
		kdplatform.h ¹⁵	Included by kd.h	Khronos, possibly modified by Vendor or Implementer
OpenVG 1.x	VG	openvg.h ¹⁶	#include <VG/openvg.h>	Khronos
		vgu.h ^{17 c}	#include <VG/vgu.h>	Khronos

^aMany early EGL implementations used `egltypes.h` instead of the now recommended `eglplatform.h`.

^b`glplatform.h` does not exist in many early implementations of OpenGL ES 1.x. Platform dependent declarations were included directly in `gl.h`.

^cRequired, if the OpenVG utility library is provided.

Notes

- To find the include files, use appropriate compiler options in the makefiles for your sample programs; e.g. `-I` (gcc, linux) or `/I` (Visual C++).

⁵ <http://www.khronos.org/registry/egl/api/egl.h>

⁶ <http://www.khronos.org/registry/egl/api/eglplatform.h>

⁷ <http://www.khronos.org/registry/egl/api/eglext.h>

⁸ <http://www.khronos.org/registry/gles/api/1.1/gl.h>

⁹ <http://www.khronos.org/registry/gles/api/1.1/glplatform.h>

¹⁰ <http://www.khronos.org/registry/gles/api/1.1/glext.h>

¹¹ <http://www.khronos.org/registry/gles/api/2.0/gl2.h>

¹² <http://www.khronos.org/registry/gles/api/2.0/gl2platform.h>

¹³ <http://www.khronos.org/registry/gles/api/2.0/gl2ext.h>

¹⁴ <http://www.khronos.org/registry/kode/api/kd.h>

¹⁵ <http://www.khronos.org/registry/kode/api/kdplatform.h>

¹⁶ http://www.khronos.org/openvg/headers/openvg-1_0_1/openvg.h

¹⁷ http://www.khronos.org/openvg/headers/openvg-1_0_1/vgu.h

- Given the different IDEs & compilers people use, especially on Windows, it is not possible to recommend a system location to place these include files. Where obvious choices exist Khronos recommends implementers take advantage of them.
- In particular, GNU/Linux implementations should follow the *Filesystem Hierarchy Standard*¹⁸ for location of headers and libraries.

Libraries

- It is highly desirable to implement all *API* entry points as function calls. However in the OpenKODE core, macros or in-lines may be used instead of a function call provided the following rules are followed:
 - When calling a function, each argument must be evaluated exactly once (although the order of evaluation is undefined).
 - It must be possible to take the address of function.
- Except in cases where macros are allowed, ensure the *API* function names exported by your lib & dll files match the function names specified by the Khronos standard for the *API* you are implementing.

Packaging

- OpenGL ES, EGL, OpenVG and OpenKODE entry points should be packaged in separate libraries.
- However to provide backward compatibility for existing applications, two OpenGL ES 1.1 libraries should be provided: one with and one without the EGL entry points.

Note: There are extant implementations of the dual OpenGL ES libraries demonstrating this is possible on Symbian, GNU/Linux, Win32 and WinCE.

For OpenGL ES 2.x, only a library without EGL entry points is needed.

Naming

Khronos recommends the library names shown in the following table:

¹⁸ <http://www.pathname.com/fhs/>

Table 2. Recommended Library Names

API/Entry Points	Name
EGL	libEGL.{lib,dll}
OpenGL ES 1.x with EGL (Common Profile)	libGLES_CM.{lib,dll} ^{a b}
OpenGL ES 1.x with EGL (Lite Profile)	libGLES_CL.{lib,dll} ^{a b}
OpenGL ES 1.x without EGL	libGLESv1_C[LM].{lib,dll} ^c
OpenGL ES 2.x without EGL	libGLESv2.{lib,dll}
	libGLUESv1.{lib,dll} ^d
	libGLUESv2.{lib,dll} ^d
OpenKODE	libKD{lib,dll}
OpenVG	libOpenVG{lib,dll}
OpenVG Utilities (when present)	libOpenVGU.{lib,dll}

^aThese names are required for OpenGL ES 1.0 and the libraries must contain the EGL entry points as detailed in Chapter 8, *Packaging*, of the OpenGL ES 1.0 specification.

^bThese names are deprecated for OpenGL ES 1.1 and beyond and should only be used for a library that includes the EGL entry points in order to support legacy applications.

^cThe *OpenGL ES 1.1 specification at revision 1.1.09* was updated to specify these alternate names for GLES libraries that do not contain the EGL entry points.

^dThese names are reserved for future use by the Khronos Group.

Notes

- The extensions `.lib` & `.dll` are allowed to vary by platform. On GNU/Linux for example they should be `.a` and `.so` to match that platform's conventions.

Acknowledgements

Thanks to all the members of the Khronos Group for their input and in particular to the following:

Petri Kero
 Jon Leech
 Robert Palmer
 Jani Vaarala
 Hans-Martin Will

Glossary

Application Binary Interface (ABI)	The low-level interface between a compiled application program and the operating system or its libraries.
Application Programming Interface (API)	The source-code level interface between an application program and the operating system or its libraries.
Implementer	A company or person who implements a Khronos API.
Microsoft Foundation Classes (MFC)	A set of C++ utility classes provided by Microsoft Corporation.

Platform Vendor (Vendor)

A company providing an operating system platform that includes an *ABI* specification for one or more Khronos APIs. E.g., Qualcomm (OpenGL ES on BREW) and Symbian (OpenGL ES on Symbian OS). A Vendor may also be an *Implementer*.