QUALCOMM®

Binary Runtime Environment for Wireless™

# OpenGL® ES BREW Extension User Guide

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA. 92121-1714
U.S.A.



OpenGL® BREW Extension User Guide

80-V8361-1 A

March 18, 2004

# Contents

**OpenGL® ES BREW™ Extension Overview 4**

BREW interface usage 4
BREW interface wrapper usage 4
Revision history 4

**Sample Code 5**

Steps for using IGL 5
Steps for using IEGL 6
Steps for using IGL with gl wrappers 6
Steps for using IEGL with egl wrappers 7

# OpenGL® ES BREW™ Extension Overview

The OpenGL ES BREW Extension allows for the use of the OpenGL ES API by BREW developers. See the Khronos webpage at www.khronos.org for detailed information regarding the OpenGL ES 1.0 API.

## BREW interface usage

To use the OpenGL ES BREW Extension, developers need to include the AEEGL.h header file and instantiate instances of the IGL and IEGL interfaces like any other BREW interface. OpenGL ES and EGL functions can be called by using the IGL and IEGL BREW equivalents. All these functions behave as their OpenGL ES and EGL equivalents but take the BREW interface pointer as their first parameter.

## BREW interface wrapper usage

In addition to using the BREW IGL and IEGL functions directly, there are also wrappers for those functions that allow the user to call the OpenGL ES and EGL functions directly and have them mapped to their BREW interface equivalents, making the IGL and IEGL BREW interface pointers transparent to the user when calling OpenGL ES and EGL functions.

This is accomplished by including the file IGL.h and linking in the GL.c source file provided with the OpenGL ES extension. To avoid one function call overhead, be sure to include IGL.h in all files where you call OpenGL ES or EGL functions. See the sample application for an example of this usage.

## Revision history

The revision history for this document is shown in the following table.

***Revision history***

| Version | Date | Description |
|---|---|---|
| A | Mar 2004 | Initial release |

# Sample Code

See the sample application in the Examples directory for a complete, working example of a BREW application using the OpenGL ES BREW extension.

The following code samples are meant as a general overview of the steps involved when using IGL and IEGL interfaces, as well as when using the wrappers for the IGL and IEGL interfaces when creating the IGL and IEGL interfaces and the cleanup to perform when exiting the application.

See OpenGL ES API Documentation and the Sample Application for sample code for initialization and cleanup of OpenGL ES applications in general.

## Steps for using IGL

### To use IGL

1. Include the AEEGL.h header file.

```
#include “AEEGL.h”
```

2. Create an instance of the IGL interface.

```
if( ISHELL_CreateInstance(pMe->a.m_pIShell, AEECLSID_GL,
(void **)&pMe->m_pIGL)

        != SUCCESS ){

   return FALSE;

}

// You can now use the BREW IGL Interface

// Ex.  IGL_glGetError( pMe->m_pIGL );
```

3. When the application is exiting, release the IGL interface.

```
IGL_Release( pMe->m_pIGL );
```

# Steps for using IEGL

## To use IEGL

1. Include the AEEGL.h file.

```
#include “AEEGL.h”
```

2. Create a BREW interface to IEGL.

```
if( ISHELL_CreateInstance(pMe->a.m_pIShell, AEECLSID_EGL,
(void **)&pMe->m_pIEGL)

        != SUCCESS ){

   return FALSE;

}

// You can now use the IEGL interface

// Ex.  IEGL_eglWaitGL( pMe->m_pIEGL );
```

3. When the application is exiting, release the EGL interface.

```
IEGL_Release( pMe->m_pIEGL );
```

# Steps for using IGL with gl wrappers

## To use IGL with gl wrappers

1. Include the IGL.h wrapper header file.

```
#include “IGL.h”
```

2. Create an instance of the IGL interface and initialize the IGL wrappers.

```
if( ISHELL_CreateInstance(pMe->a.m_pIShell, AEECLSID_GL,
(void **)&pMe->m_pIGL)

        == SUCCESS ){

   IGL_Init( pMe->m_pIGL );

} else {

   return FALSE;

}
```

```
// You can now use the standard OpenGL ES functions

// Ex.  glGetError();
```

3. When the application is exiting, release the IGL interface.

```
// Note that no OpenGL ES function calls can be made after

// releasing the interface

IGL_Release( pMe->m_pIGL );
```

# Steps for using IEGL with egl wrappers

## To use IEGL with egl wrappers

1. Include the IGL.h file.

```
#include “IGL.h”
```

2. Create a BREW interface to IEGL and initialize the IEGL wrappers.

```
if( ISHELL_CreateInstance(pMe->a.m_pIShell, AEECLSID_EGL,
(void **)&pMe->m_pIEGL)

        == SUCCESS ){

   IEGL_Init( pMe->m_pIEGL );

}else{

   return FALSE;

}

// You can now use the standard EGL functions

// Ex.  eglWaitGL();
```

3. When the application is exiting, release the EGL interface.

```
// Note that no EGL API calls can be made after releasing

// the interface, so this should come after all egl-related

// cleanup for the application.

IEGL_Release( pMe->m_pIEGL );
```