QCPlugInContext Protocol Reference

Cocoa > Graphics & Imaging

2007-05-09

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

QCPlugInContext Protocol Reference

Framework	/System/Library/Frameworks/Quartz.framework/Frameworks/QuartzComposer.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	QCPlugIn.h

Overview

The `QCPlugInContext` protocol defines methods that you use only from within the execution method (`execute:atTime:withArguments:`) of a `QCPlugIn` object.

Tasks

Getting the OpenGL Context

- `CGLContextObj` (page 6)
  Returns the destination CGL context to use for OpenGL rendering from within the execution method.

Logging Messages

- `logMessage:` (page 7)
  Writes a message to the Quartz Composer log.

Getting Execution Context Information

- `userInfo` (page 10)
  Returns a mutable dictionary that contains information that can be shared between all instances of the `QCPlugIn` subclass, running in the same Quartz Composer context.
- `bounds` (page 6)
  Returns the bounds of the rendering context.
- `colorSpace` (page 7)
  Returns the color space used by the rendering context.

Getting an Image Provider

- outputImageProviderFromBufferWithPixelFormat:pixelsWide:pixelsHigh:baseAddress:bytesPerRow:releaseCallback:releaseContext:colorSpace:shouldColorMatch: (page 8)

  Returns an image provider from a single memory buffer.

- outputImageProviderFromTextureWithPixelFormat:pixelsWide:pixelsHigh:name:flipped:releaseCallback:releaseContext:colorSpace:shouldColorMatch: (page 9)

  Returns an image provider from an OpenGL texture.

Instance Methods

bounds

Returns the bounds of the rendering context.

```
- (NSRect) bounds
```

Return Value
The bounds of the rendering context expressed in Quartz Composer units.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

CGLContextObj

Returns the destination CGL context to use for OpenGL rendering from within the execution method.

```
- (CGLContextObj) CGLContextObj
```

Return Value
The destination CGL context.

Discussion
To send commands to the OpenGL context:

- Use CGL macros instead of changing the current OpenGL context.
- Save and restore all OpenGL states except those defines by `GL_CURRENT_BIT` (vertex position, color, texture, and so on)

The following code shows how you'd use the method `CGLContextObj`:

```
// Set up using CGL macros.
#import <OpenGL/CGLMacro.h>

- (BOOL) execute:(id<QCPlugInContext>)context
              atTime:(NSTimeInterval)time
           withArguments:(NSDictionary *)arguments
```

```
{
    // Set the CGL context to a local variable.
    CGLContextObj cgl_ctx = [context CGLContextObj];
    if(cgl_ctx == NULL)
    return NO;

    // Save and set OpenGL states.
    // Put your OpenGL code here.
    // Restore the OpenGL states.
    return YES;
}
```

You can retrieve the corresponding OpenGL pixel format by calling the function `CGLGetPixelFormat`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

colorSpace

Returns the color space used by the rendering context.

```
- (CGColorSpaceRef) colorSpace
```

Return Value
An RGB color space; `NULL` if the custom patch execution mode is not consumer.

Discussion
If the method returns a color space, it must be an RGB color space.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

logMessage:

Writes a message to the Quartz Composer log.

```
- (void) logMessage:(NSString*)format, ...
```

Parameters

format

The string to write to the log. The default location for the log is the standard output.

Discussion
This method is an alternative to using the functions `NSLog` or `printf`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

outputImageProviderFromBufferWithPixelFormat:pixelsWide:pixelsHigh:baseAddress:bytesPerRow:releaseCallback:releaseContext:colorSpace:shouldColorMatch:

Returns an image provider from a single memory buffer.

```
- (id) outputImageProviderFromBufferWithPixelFormat:(NSString*)format
    pixelsWide:(NSUInteger)width pixelsHigh:(NSUInteger)height baseAddress:(const
     void*)baseAddress bytesPerRow:(NSUInteger)rowBytes
    releaseCallback:(QCPlugInBufferReleaseCallback)callback
    releaseContext:(void*)context colorSpace:(CGColorSpaceRef)colorSpace
    shouldColorMatch:(BOOL)colorMatch
```

Parameters

format

The pixel format of the memory buffer. This must be compatible with the color space.

width

The width, in bytes, of the memory buffer.

height

The height, in bytes, of the memory buffer.

baseAddress

The base address of the memory buffer, which must be multiple of 16.

rowBytes

The number of bytes per row of the memory buffer, which must be multiple of 16.

callback

The release callback. Your callback must use this type definition:

```
typedef void (*QCPlugInBufferReleaseCallback)(const void* address, void* context);
```

If you name your callback function `MyQCPlugInBufferReleaseCallback`, you would declare it like this:

```
void MyQCPlugInBufferReleaseCallback (const void address,
              void * context);
```

Quartz Composer invokes your callback when the memory buffer is no longer needed. The callback can be called from any thread at any time

context

The context to pass to the release callback.

colorSpace

The color space of the memory buffer. This must be compatible with the pixel format.

colorMatch

A Boolean that specifies whether Quartz Composer should color match the image. Pass `NO` if the image is a mask or gradient or should not be color matched for some other reason. Otherwise, pass `YES`.

Return Value
An image provider.

Discussion
You must not modify the image until the release callback is invoked.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

outputImageProviderFromTextureWithPixelFormat:pixelsWide:pixelsHigh:name:flipped:releaseCallback:releaseContext:colorSpace:shouldColorMatch:

Returns an image provider from an OpenGL texture.

```
- (id) outputImageProviderFromTextureWithPixelFormat:(NSString*)format
    pixelsWide:(NSUInteger)width pixelsHigh:(NSUInteger)height name:(GLuint)name
    flipped:(BOOL)flipped releaseCallback:(QCPlugInTextureReleaseCallback)callback
     releaseContext:(void*)context colorSpace:(CGColorSpaceRef)colorSpace
    shouldColorMatch:(BOOL)colorMatch;
```

Parameters

format

The pixel format of the texture. This must be compatible with the color space.

width

The width, in bytes, of the texture.

height

The height, in bytes, of the texture.

name

An OpenGL texture of type `GL_TEXTURE_RECTANGLE_EXT` that is valid on the Quartz Composer OpenGL context. Note that textures do not have a retain and release mechanism. This means that your application must make sure that the texture exists for the life cycle of the image provider.

flipped

`YES` to have Quartz Composer flip the contents of the texture vertically.

callback

The release callback. Your callback must use this type definition:

```
typedef void (*QCPlugInTextureReleaseCallback)(CGLContextObj cgl_ctx, GLuint
name, void* context);
```

If you name your callback function `MyQCPlugInTextureReleaseCallback`, you would declare it like this:

```
void MyQCPlugInTextureReleaseCallback (CGLContextObj cgl_ctx,
              GLuint name,
              void* context);
```

Quartz Composer invokes your callback when the memory buffer is no longer needed. The callback can be called from any thread at any time

context

The context to pass to the release callback.

colorSpace

The color space of the texture. This must be compatible with the pixel format.

colorMatch

A Boolean that specifies whether Quartz Composer should color match the texture. Pass `NO` if the texture is a mask or gradient or should not be color matched for some other reason. Otherwise, pass `YES`.

Return Value
An image provider.

Discussion
You must not modify the texture until the release callback is invoked.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

userInfo

Returns a mutable dictionary that contains information that can be shared between all instances of the `QCPlugIn` subclass, running in the same Quartz Composer context.

```
- (NSMutableDictionary*) userInfo
```

Return Value
A mutable dictionary.

Discussion
When you add information to the dictionary, make sure that you use unique keys, such as `com.myCompany.foo`. You can use this dictionary to cache data that you want to share.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`QCPlugIn.h`

Document Revision History

This table describes the changes to *QCPlugInContext Protocol Reference*.

Date	Notes
2007-05-09	New document that describes the protocol used by execution contexts.

Document Revision History

Index