NSOpenGLPixelFormat Class Reference

Graphics & Animation: 3D Drawing

2009-08-13

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

Simultaneously published in the United States and Canada.

Contents

NSOpenGLPixelFormat Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Cocoa Drawing Guide
Declared in	NSOpenGL.h
Related sample code	Denoise From A View to A Movie From A View to A Picture GLUT QTCoreVideo201

Overview

To render with OpenGL into an `NSOpenGLContext`, you must specify the context's pixel format. An `NSOpenGLPixelFormat` object specifies the types of buffers and other attributes of the `NSOpenGLContext`. This class is similar to the `AGLPixelFormat` type, which is used in Carbon OpenGL applications.

Every `NSOpenGLPixelFormat` object wraps a low-level, platform-specific Core OpenGL (CGL) pixel format object. Your application can retrieve the CGL pixel format object by calling the `CGLPixelFormatObj` (page 6) method. For more information on the underling CGL pixel format object, see *CGL Reference*.

Tasks

Creating an NSOpenGLPixelFormat Object

`- initWithCGLPixelFormatObj:` (page 9)
Returns an `NSOpenGLPixelFormat`object initialized with using an existing CGL pixel format object.

`- initWithData:` (page 10)
Returns an `NSOpenGLPixelFormat` object initialized with specified pixel format attribute data. (Deprecated. Use `initWithAttributes:` (page 7) instead.)

- `initWithAttributes:` (page 7)
  Returns an `NSOpenGLPixelFormat` object initialized with specified pixel format attributes.

Managing the Pixel Format

- `CGLPixelFormatObj` (page 6)
  Returns the low-level, platform-specific Core OpenGL (CGL) pixel format object represented by the receiver.
- `getValues:forAttribute:forVirtualScreen:` (page 7)
  Gets the value for the specified pixel format attribute.
- `numberOfVirtualScreens` (page 10)
  Returns the number of virtual screens associated with the receiver.

Managing Attributes

- `attributes` (page 6)
  Retrieves the attribute data for the pixel format object. (Deprecated.)
- `setAttributes:` (page 11)
  Sets the attribute data for the pixel format object. (Deprecated.)

Instance Methods

attributes

Retrieves the attribute data for the pixel format object. (Deprecated.)

```
- (NSData *)attributes
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSOpenGL.h`

CGLPixelFormatObj

Returns the low-level, platform-specific Core OpenGL (CGL) pixel format object represented by the receiver.

```
- (void *)CGLPixelFormatObj
```

Return Value
A pointer to the underlying `CGLPixelFormatObj` object.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CIColorTracking
CIRAWFilterSample
Denoise
WebKitCIPlugIn
WhackedTV

Declared In
`NSOpenGL.h`

getValues:forAttribute:forVirtualScreen:

Gets the value for the specified pixel format attribute.

```
- (void)getValues:(GLint *)vals forAttribute:(NSOpenGLPixelFormatAttribute)attrib
    forVirtualScreen:(GLint)screen
```

Parameters

vals
On input, a pointer to a `long` variable. On output, the variable contains the value of the requested attribute.

attrib
The requested attribute. For a list of attribute constants, see the table in "Constants" (page 11).

screen
The screen from which you want to retrieve the attribute. This parameter must be a value between 0 and the number of virtual screens (`numberOfVirtualScreens` (page 10)) minus 1.

Discussion
Because the value for an attribute may be different on each virtual screen, the virtual screen must be specified along with the attribute.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- initWithAttributes:` (page 7)

Related Sample Code
CIVideoDemoGL
GLUT
LiveVideoMixer2
LiveVideoMixer3
NSOpenGL Fullscreen

Declared In
`NSOpenGL.h`

initWithAttributes:

Returns an `NSOpenGLPixelFormat` object initialized with specified pixel format attributes.

```
- (id)initWithAttributes:(const NSOpenGLPixelFormatAttribute *)attribs
```

Parameters

attribs

A 0-terminated array containing Boolean and integer attribute constants. The presence of a Boolean attribute implies a value of `YES` while its absence implies a value of `NO`. Integer constants must be followed by the desired value. For a listing of attribute constants, see the table in "Constants" (page 11).

Return Value

An initialized `NSOpenGLPixelFormat` object whose attributes match the desired attributes as close as possible, or `nil` if an object with the desired attributes could not be initialized.

Discussion

On return, the Boolean attributes of the receiver match the values specified in *attribs*, and the integer attributes are as close to the specified values as can be provided by the system. However, if no matching pixel format exists, the receiver releases itself and `nil` is returned. You may deallocate the receiver following its use in the successful initialization of an `NSOpenGLContext`.

The existence of a Boolean attribute constant in *attribs* implies a `YES` value. The Boolean attribute constants are:

```
NSOpenGLPFAAllRenderers
NSOpenGLPFADoubleBuffer
NSOpenGLPFAStereo
NSOpenGLPFAMinimumPolicy
NSOpenGLPFAMaximumPolicy
NSOpenGLPFAOffScreen
NSOpenGLPFAFullScreen
NSOpenGLPFASingleRenderer
NSOpenGLPFANoRecovery
NSOpenGLPFAAccelerated
NSOpenGLPFAClosestPolicy
NSOpenGLPFARobust
NSOpenGLPFABackingStore
NSOpenGLPFAWindow
NSOpenGLPFAMultiScreen
NSOpenGLPFACompliant
NSOpenGLPFAPixelBuffer
```

The integer constants must be followed by a value. These constants are:

```
NSOpenGLPFAAuxBuffers
NSOpenGLPFAColorSize
NSOpenGLPFAAlphaSize
NSOpenGLPFADepthSize
NSOpenGLPFAStencilSize
NSOpenGLPFAAccumSize
NSOpenGLPFARendererID
NSOpenGLPFAScreenMask
```

This code fragment creates a double-buffered pixel format with a 32-bit depth buffer:

```
NSOpenGLPixelFormatAttribute attrs[] =
{
    NSOpenGLPFADoubleBuffer,
    NSOpenGLPFADepthSize, 32,
    0
};

NSOpenGLPixelFormat* pixFmt = [[NSOpenGLPixelFormat alloc]
initWithAttributes:attrs];

/* Check if initWithAttributes succeeded. */
if(pixFmt == nil) {
    /* initWithAttributes failed. Try to alloc/init with a different  list of
attributes. */
}
```

Availability
Available in Mac OS X v10.0 and later.

See Also
`- getValues:forAttribute:forVirtualScreen:` (page 7)

Related Sample Code
BoingX
From A View to A Movie
From A View to A Picture
NSOpenGL Fullscreen
SurfaceVertexProgram

Declared In
`NSOpenGL.h`

initWithCGLPixelFormatObj:

Returns an `NSOpenGLPixelFormat`object initialized with using an existing CGL pixel format object.

```
- (id)initWithCGLPixelFormatObj:(void *)format
```

Parameters

format
An existing CGL pixel format object.

Return Value
An intialized `NSOpenGLPixelFormat` object that wraps the CGL pixel format object.

Discussion
If your application already has a low-level CGL pixel format object, you can create an `NSOpenGLPixelFormat` object to wrap it by calling this initializer. The `NSOpenGLPixelFormat` object retains the CGL pixel format object by calling the `CGLRetainPixelFormat` function.

Your application should not call `CGLDestroyPixelFormat` to dispose of the CGL pixel format object. Instead, your application should call `CGLReleasePixelFormat` to decrement its reference count.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`NSOpenGL.h`

initWithData:

Returns an `NSOpenGLPixelFormat` object initialized with specified pixel format attribute data. (Deprecated. Use `initWithAttributes:` (page 7) instead.)

```
- (id)initWithData:(NSData *)attribs
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSOpenGL.h`

numberOfVirtualScreens

Returns the number of virtual screens associated with the receiver.

```
- (GLint)numberOfVirtualScreens
```

Return Value
The number of virtual screens.

Discussion
When the attributes are set, OpenGL searches for drivers matching the requested attributes. Each matching driver drives a set of displays. For example, a graphics card in a portable computer might drive the internal screen and an external display. This portable computer would have one virtual screen. A desktop computer might have two different graphics cards, each driving one or more displays. The pairing of an OpenGL driver with its set of associated displays corresponds to one virtual screen. In the above examples, the portable computer would have one virtual screen, while the desktop computer would have two. Another desktop computer with a video card driving two displays at once would have one virtual screen.

For more information on virtual screens, consult *OpenGL Programming Guide for Mac OS X*.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- getValues:forAttribute:forVirtualScreen:` (page 7)

Related Sample Code
CIColorTracking
CIVideoDemoGL
LiveVideoMixer
LiveVideoMixer2
LiveVideoMixer3

Declared In
NSOpenGL.h

setAttributes:

Sets the attribute data for the pixel format object. (Deprecated.)

```
- (void)setAttributes:(NSData *)attribs
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
NSOpenGL.h

Constants

NSOpenGLPixelFormatAttribute

The following attribute names are used by `initWithAttributes:` (page 7) and `getValues:forAttribute:forVirtualScreen:` (page 7):

```
enum {
    NSOpenGLPFAAllRenderers       =   1,
    NSOpenGLPFADoubleBuffer       =   5,
    NSOpenGLPFAStereo             =   6,
    NSOpenGLPFAAuxBuffers         =   7,
    NSOpenGLPFAColorSize          =   8,
    NSOpenGLPFAAlphaSize          =  11,
    NSOpenGLPFADepthSize          =  12,
    NSOpenGLPFAStencilSize        =  13,
    NSOpenGLPFAAccumSize          =  14,
    NSOpenGLPFAMinimumPolicy      =  51,
    NSOpenGLPFAMaximumPolicy      =  52,
    NSOpenGLPFAOffScreen          =  53,
    NSOpenGLPFAFullScreen         =  54,
    NSOpenGLPFASampleBuffers      =  55,
    NSOpenGLPFASamples            =  56,
    NSOpenGLPFAAuxDepthStencil    =  57,
    NSOpenGLPFAColorFloat         =  58,
    NSOpenGLPFAMultisample        =  59,
    NSOpenGLPFASupersample        =  60,
    NSOpenGLPFASampleAlpha        =  61,
    NSOpenGLPFARendererID         =  70,
    NSOpenGLPFASingleRenderer     =  71,
    NSOpenGLPFANoRecovery         =  72,
    NSOpenGLPFAAccelerated        =  73,
    NSOpenGLPFAClosestPolicy      =  74,
    NSOpenGLPFARobust             =  75,
    NSOpenGLPFABackingStore       =  76,
    NSOpenGLPFAMPSafe             =  78,
    NSOpenGLPFAWindow             =  80,
    NSOpenGLPFAMultiScreen        =  81,
    NSOpenGLPFACompliant          =  83,
    NSOpenGLPFAScreenMask         =  84,
    NSOpenGLPFAPixelBuffer        =  90,
    NSOpenGLPFARemotePixelBuffer  =  91,
    NSOpenGLPFAAllowOfflineRenderers = 96,
    NSOpenGLPFAAcceleratedCompute =  97,
    NSOpenGLPFAVirtualScreenCount = 128
};
typedef  uint32_t NSOpenGLPixelFormatAttribute;
```

Constants

`NSOpenGLPFAAllRenderers`

A Boolean attribute. If present, this attribute indicates that the pixel format selection is open to all available renderers, including debug and special-purpose renderers that are not OpenGL compliant.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFADoubleBuffer`

A Boolean attribute. If present, this attribute indicates that only double-buffered pixel formats are considered. Otherwise, only single-buffered pixel formats are considered.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAStereo`

A Boolean attribute. If present, this attribute indicates that only stereo pixel formats are considered. Otherwise, only monoscopic pixel formats are considered.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAuxBuffers`

Value is a nonnegative integer that indicates the desired number of auxiliary buffers. Pixel formats with the smallest number of auxiliary buffers that meets or exceeds the specified number are preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAColorSize`

Value is a nonnegative buffer size specification. A color buffer that most closely matches the specified size is preferred. If unspecified, OpenGL chooses a color size that matches the screen.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAlphaSize`

Value is a nonnegative buffer size specification. An alpha buffer that most closely matches the specified size is preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFADepthSize`

Value is a nonnegative depth buffer size specification. A depth buffer that most closely matches the specified size is preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAStencilSize`

Value is a nonnegative integer that indicates the desired number of stencil bitplanes. The smallest stencil buffer of at least the specified size is preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAccumSize`

Value is a nonnegative buffer size specification. An accumulation buffer that most closely matches the specified size is preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAMinimumPolicy`

A Boolean attribute. If present, this attribute indicates that the pixel format choosing policy is altered for the color, depth, and accumulation buffers such that only buffers of size greater than or equal to the desired size are considered.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAMaximumPolicy`

A Boolean attribute. If present, this attribute indicates that the pixel format choosing policy is altered for the color, depth, and accumulation buffers such that, if a nonzero buffer size is requested, the largest available buffer is preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAOffScreen`

A Boolean attribute. If present, this attribute indicates that only renderers that are capable of rendering to an offscreen memory area and have buffer depth exactly equal to the desired buffer depth are considered. The `NSOpenGLPFAClosestPolicy` attribute is implied.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAFullScreen`

A Boolean attribute. If present, this attribute indicates that only renderers that are capable of rendering to a full-screen drawable are considered. The `NSOpenGLPFASingleRenderer` attribute is implied.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFASampleBuffers`

Value is a nonnegative number indicating the number of multisample buffers.

Available in Mac OS X v10.2 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFASamples`

Value is a nonnegative indicating the number of samples per multisample buffer.

Available in Mac OS X v10.2 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAuxDepthStencil`

Each auxiliary buffer has its own depth stencil.

Available in Mac OS X v10.2 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAColorFloat`

A Boolean attribute. If present, this attribute indicates that only renderers that are capable using buffers storing floating point pixels are considered. This should be accompanied by a `NSOpenGLPFAColorSize` of 64 (for half float pixel components) or 128 (for full float pixel components). Note, not all hardware supports floating point color buffers thus the returned pixel format could be NULL.

Available in Mac OS X v10.4 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAMultisample`

A Boolean attribute. If present and used with `NSOpenGLPFASampleBuffers` and `NSOpenGLPFASamples`, this attribute hints to OpenGL to prefer multi-sampling. Multi-sampling will sample textures at the back buffer dimensions vice the multi-sample buffer dimensions and use that single sample for all fragments with coverage on the back buffer location. This means less total texture samples than with super-sampling (by a factor of the number of samples requested) and will likely be faster though less accurate (texture sample wise) than super-sampling. If the underlying video card does not have enough VRAM to support this feature, this hint does nothing.

The `NSOpenGLPFASampleBuffers` and `NSOpenGLPFASamples` attributes must be configured to request anti-aliasing as follows:

```
NSOpenGLPFAMultisample,
NSOpenGLPFASampleBuffers, (NSOpenGLPixelFormatAttribute)1
NSOpenGLPFASamples, (NSOpenGLPixelFormatAttribute)4,
```

If after adding these options, multisamping still does not work, try removing the `NSOpenGLPFAPixelBuffer` attribute (if present). Some graphics cards may not support this option in specific versions of Mac OS X. If removing the attribute still does not enable multisampling, try adding the `NSOpenGLPFANoRecovery` attribute.

Available in Mac OS X v10.4 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFASupersample`

A Boolean attribute. If present and used with `NSOpenGLPFASampleBuffers` and `NSOpenGLPFASamples`, this attribute hints to OpenGL to prefer super-sampling. Super-sampling will process fragments with a texture sample per fragment and would likely be slower than multi-sampling. If the pixel format is not requesting anti-aliasing, this hint does nothing.

Available in Mac OS X v10.4 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFASampleAlpha`

A Boolean attribute. If present and used with `NSOpenGLPFASampleBuffers` and `NSOpenGLPFASampleBuffers`, this attribute hints to OpenGL to update multi-sample alpha values to ensure the most accurate rendering. If pixel format is not requesting anti-aliasing then this hint does nothing.

Available in Mac OS X v10.4 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFARendererID`

Value is a nonnegative renderer ID number. OpenGL renderers that match the specified ID are preferred. Constants to select specific renderers are provided in the `CGLRenderers.h` header of the OpenGL framework. Of note is `kCGLRendererGenericID` which selects the Apple software renderer. The other constants select renderers for specific hardware vendors.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFASingleRenderer`

A Boolean attribute. If present, this attribute indicates that a single rendering engine is chosen. On systems with multiple screens, this disables OpenGL's ability to drive different monitors through different graphics accelerator cards with a single context. This attribute is not generally useful.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFANoRecovery`

A Boolean attribute. If present, this attribute indicates that OpenGL's failure recovery mechanisms are disabled. Normally, if an accelerated renderer fails due to lack of resources, OpenGL automatically switches to another renderer. This attribute disables these features so that rendering is always performed by the chosen renderer. This attribute is not generally useful.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAccelerated`

A Boolean attribute. If present, this attribute indicates that only hardware-accelerated renderers are considered. If not present, accelerated renderers are still preferred.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAClosestPolicy`

A Boolean attribute. If present, this attribute indicates that the pixel format choosing policy is altered for the color buffer such that the buffer closest to the requested size is preferred, regardless of the actual color buffer depth of the supported graphics device.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFARobust`

A Boolean attribute. If present, this attribute indicates that only renderers that do not have any failure modes associated with a lack of video card resources are considered. This attribute is not generally useful.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFABackingStore`

A Boolean attribute. If present, this attribute indicates that OpenGL only considers renderers that have a back color buffer the full size of the drawable (regardless of window visibility) and that guarantee the back buffer contents to be valid after a call to `NSOpenGLContext` object's `flushBuffer`.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAMPSafe`

A Boolean attribute. If present, this attribute indicates that the renderer is multi-processor safe.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAWindow`

A Boolean attribute. If present, this attribute indicates that only renderers that are capable of rendering to a window are considered. This attribute is implied if neither `NSOpenGLPFAFullScreen` nor `NSOpenGLPFAOffScreen` is specified.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAMultiScreen`

A Boolean attribute. If present, this attribute indicates that only renderers capable of driving multiple screens are considered. This attribute is not generally useful.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFACompliant`
A Boolean attribute. If present, this attribute indicates that pixel format selection is only open to OpenGL-compliant renderers. This attribute is implied unless `NSOpenGLPFAAllRenderers` is specified. This attribute is not useful in the attribute array.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAScreenMask`
Value is a bit mask of supported physical screens. All screens specified in the bit mask are guaranteed to be supported by the pixel format. Screens not specified in the bit mask may still be supported. The bit mask is managed by the CoreGraphics's **DirectDisplay**, available in the `CGDirectDisplay.h` header of the ApplicationServices umbrella framework. A `CGDirectDisplayID` must be converted to an OpenGL display mask using the function `CGDisplayIDToOpenGLDisplayMask`. This attribute is not generally useful.

Available in Mac OS X v10.0 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAPixelBuffer`
A Boolean attribute. If present, this attribute indicates that rendering to a pixel buffer is enabled.

Available in Mac OS X v10.3 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFARemotePixelBuffer`
A Boolean attribute. If present, this attribute indicates that rendering to a pixel buffer on an offline renderer is enabled.

Available in Mac OS X v10.6 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAllowOfflineRenderers`
A Boolean attribute. If present, this attribute indicates that offline renderers may be used.

Available in Mac OS X v10.5 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAAcceleratedCompute`
If present, this attribute indicates that only renderers that can execute OpenCL programs should be used.

Available in Mac OS X v10.6 and later.

Declared in `NSOpenGL.h`.

`NSOpenGLPFAVirtualScreenCount`
The number of virtual screens in this format.

Available in Mac OS X v10.2 and later.

Declared in `NSOpenGL.h`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSOpenGL.h`

Document Revision History

This table describes the changes to *NSOpenGLPixelFormat Class Reference*.

Date	Notes
2009-08-13	Added new symbols for Mac OS X v10.6.
2008-10-15	Updated the the constants section to reflect changes in Mac OS X v10.5.
2007-01-26	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index