QCPlugInInputImageSource Protocol Reference

Cocoa > Graphics & Imaging

2007-07-12

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

Simultaneously published in the United States and Canada.

Contents

QCPlugInInputImageSource 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 `QCPlugInInputImageSource` protocol eliminates the need to use explicit image types for the image input ports on your custom patch. Not only does using the protocol avoid restrictions of a specific image type, but it avoids impedance mismatches, and provides better performance by deferring pixel computation until it is needed. When you need to access the pixels in an image, you simply convert the image to a representation (texture or buffer) using one of the methods defined by the `QCPlugInInputImageSource` protocol. Use a texture representation when you want to use input images on the GPU. Use a buffer representation when you want to use input images on the CPU.

Input images are opaque source objects that comply to this protocol. To create an image input port as an Objective-C 2.0 property, declare it as follows:

```
@property(dynamic) id<QCPlugInInputImageSource> inputImage;
```

To create an image input port dynamically. use the type `QCPortTypeImage`:

```
[self addInputPortWithType:QCPortTypeImage
                    forKey:@"inputImage"
            withAttributes:nil];
```

Tasks

Converting an Image to a Representation

- `lockTextureRepresentationWithColorSpace:forBounds:` (page 10)
  Creates an OpenGL texture representation from a subregion of the image source using the provided color space.
- `unlockTextureRepresentation` (page 14)
  Releases the OpenGL texture representation of the image source.
- `lockBufferRepresentationWithPixelFormat:colorSpace:forBounds:` (page 10)
  Creates a memory buffer representation from a subregion of the image source using the provided pixel format and color space.

- `bindTextureRepresentationToCGLContext:textureUnit:normalizeCoordinates:` (page 7)
  Binds the texture to a given texture unit and optionally scales or flips the texture.
- `unbindTextureRepresentationFromCGLContext:textureUnit:` (page 14)
  Unbinds the texture from a texture unit.
- `unlockBufferRepresentation` (page 14)
  Releases the memory buffer representation of the image source.

Getting Color Space Information

- `imageColorSpace` (page 9)
  Returns the color space of the image source.
- `shouldColorMatch` (page 11)
  Returns whether or not the image source should be color matched.

Getting Texture Information

- `texturePixelsWide` (page 13)
  Returns the width of the texture representation.
- `texturePixelsHigh` (page 13)
  Returns the height of the texture representation.
- `textureTarget` (page 13)
  Returns the texture target.
- `textureName` (page 12)
  Returns the texture name.
- `textureColorSpace` (page 11)
  Returns the color space of the texture representation.
- `textureFlipped` (page 11)
  Returns whether or not the contents of the texture are flipped vertically.
- `textureMatrix` (page 12)
  Returns a texture matrix.

Getting Image Buffer Information

- `imageBounds` (page 9)
  Returns the actual bounds of the image source expressed in pixels and aligned to integer boundaries.
- `bufferPixelsWide` (page 9)
  Returns the width of the image buffer representation.
- `bufferPixelsHigh` (page 8)
  Returns the height of the image buffer representation.
- `bufferPixelFormat` (page 8)
  Returns the pixel format of the image buffer representation.
- `bufferColorSpace` (page 8)
  Returns the color space of the image buffer representation.

- `bufferBaseAddress` (page 7)
  Returns the base address of the image buffer.
- `bufferBytesPerRow` (page 8)
  Returns the bytes per row of the buffer representation.

Instance Methods

bindTextureRepresentationToCGLContext:textureUnit:normalizeCoordinates:

Binds the texture to a given texture unit and optionally scales or flips the texture.

```
- (void) bindTextureRepresentationToCGLContext:(CGLContextObj)cgl_ctx
    textureUnit:(GLenum)unit normalizeCoordinates:(BOOL)flag
```

Parameters

cgl_ctx
  The CGL context to render to.)

unit
  The texture unit to bind to (such as, `GL_TEXTURE0`)

flag
  To apply a texture matrix to scale coordinates (from `[0, pixels]` to `[0,1]`) and flip them vertically (if necessary), pass `YES`.

Discussion
When you no longer need the texture, call `unbindTextureRepresentationFromCGLContext:textureUnit:` (page 14).

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

Declared In
`QCPlugIn.h`

bufferBaseAddress

Returns the base address of the image buffer.

```
- (const void*) bufferBaseAddress
```

Return Value
The base address of the buffer.

Discussion
The base address is guaranteed to be aligned on a 16-byte boundary.

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

Declared In
`QCPlugIn.h`

bufferBytesPerRow

Returns the bytes per row of the buffer representation.

```
- (NSUInteger) bufferBytesPerRow
```

Return Value
The number of bytes per row of the buffer.

Discussion
The number of bytes per row is guaranteed to be a multiple of 16.

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

Declared In
`QCPlugIn.h`

bufferColorSpace

Returns the color space of the image buffer representation.

```
- (CGColorSpaceRef) bufferColorSpace
```

Return Value
The color space of the image buffer.

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

Declared In
`QCPlugIn.h`

bufferPixelFormat

Returns the pixel format of the image buffer representation.

```
- (NSString*) bufferPixelFormat
```

Return Value
A string that specifies the pixel format. The supported formats are ARGB8 (8-bit alpha, red, green, blue), BGRA8 (8-bit blue, green, red, and alpha), RGBAf (floating-point, red, green, blue, alpha), I8 (8-bit intensity), and If (floating-point intensity).

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

Declared In
`QCPlugIn.h`

bufferPixelsHigh

Returns the height of the image buffer representation.

```
- (NSUInteger) bufferPixelsHigh
```

Return Value
The height, expressed in pixels.

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

See Also
- bufferPixelsHigh (page 8)

Declared In
QCPlugIn.h

bufferPixelsWide

Returns the width of the image buffer representation.

```
- (NSUInteger) bufferPixelsWide
```

Return Value
The width, expressed in pixels.

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

See Also
- bufferPixelsHigh (page 8)

Declared In
QCPlugIn.h

imageBounds

Returns the actual bounds of the image source expressed in pixels and aligned to integer boundaries.

```
- (NSRect) imageBounds;
```

Return Value
The bounds of the image source.

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

Declared In
QCPlugIn.h

imageColorSpace

Returns the color space of the image source.

```
- (CGColorSpaceRef) imageColorSpace
```

Return Value
The color space of the image source, typically RGB or Gray type.

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

Declared In
`QCPlugIn.h`

lockBufferRepresentationWithPixelFormat:colorSpace:forBounds:

Creates a memory buffer representation from a subregion of the image source using the provided pixel format and color space.

```
- (BOOL) lockBufferRepresentationWithPixelFormat:(NSString*)format
    colorSpace:(CGColorSpaceRef)colorSpace forBounds:(NSRect)bounds
```

Parameters

format
A pixel format that is compatible with the color space.

colorSpace
A Quartz color space that is compatible with the pixel format.

bounds
The bounds of the subregion, expressed as pixels, and aligned to integer boundaries.

Return Value
`YES` if successful; otherwise `NO`.

Discussion
The content of the buffer is read-only. You should not attempt to modify it.

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

See Also
`- unlockBufferRepresentation` (page 14)

Declared In
`QCPlugIn.h`

lockTextureRepresentationWithColorSpace:forBounds:

Creates an OpenGL texture representation from a subregion of the image source using the provided color space.

```
- (BOOL) lockTextureRepresentationWithColorSpace:(CGColorSpaceRef)colorSpace
    forBounds:(NSRect)bounds
```

Parameters

colorSpace
A Quartz color space.

bounds

The bounds of the subregion, expressed in pixels. They must be aligned to integer boundaries.

Return Value
`YES` is successful; `NO` if texture can't be created.

Discussion
Neither the content of the texture nor its states (for example, the wrap mode) must be modified; you can only draw with it. The texture is valid only in the plug-in context.

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

See Also
`- unlockTextureRepresentation` (page 14)

Declared In
`QCPlugIn.h`

shouldColorMatch

Returns whether or not the image source should be color matched.

```
- (BOOL) shouldColorMatch
```

Return Value
`NO` if the source is a mask or gradient; `YES` otherwise.

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

Declared In
`QCPlugIn.h`

textureColorSpace

Returns the color space of the texture representation.

```
- (CGColorSpaceRef) textureColorSpace
```

Return Value
The color space of the texture.

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

Declared In
`QCPlugIn.h`

textureFlipped

Returns whether or not the contents of the texture are flipped vertically.

```
- (BOOL) textureFlipped
```

Return Value
`YES` if the contents of the texture are flipped (upside-down); `NO` otherwise.

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

Declared In
`QCPlugIn.h`

textureMatrix

Returns a texture matrix.

```
- (const GLfloat*) textureMatrix
```

Return Value
A 4x4 texture matrix created by scaling (from [`0`, pixels] to [`0`,`1`]) and vertically flipping the texture coordinates; `NULL` if coordinate transformation is not required.

Discussion
This method is provided as a convenience for 2D textures to take care of two issues:

- Coordinates for rectangular textures are expressed in pixels rather than the normalized units used for power-of-two textures. The coordinates need to be normalized before you can process the texture.
- Texture coordinates are typically flipped by OpenGL for processing on the GPU and need to be flipped to the original coordinates.

You can take care of these two issues simply by loading a the matrix returned by this method onto the OpenGL stack. If you are not sure that your texture needs either of these operations, you can load the matrix on the OpenGL stack anyway, as it acts as an identity matrix if it's not needed.

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

Declared In
`QCPlugIn.h`

textureName

Returns the texture name.

```
- (GLuint) textureName
```

Return Value
The texture name.

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

Declared In
`QCPlugIn.h`

texturePixelsHigh

Returns the height of the texture representation.

```
- (NSUInteger) texturePixelsHigh
```

Return Value
The height of the texture, expressed in pixels.

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

See Also
- texturePixelsWide (page 13)

Declared In
QCPlugIn.h

texturePixelsWide

Returns the width of the texture representation.

```
- (NSUInteger) texturePixelsWide
```

Return Value
The width of the texture, expressed in pixels.

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

See Also
- texturePixelsHigh (page 13)

Declared In
QCPlugIn.h

textureTarget

Returns the texture target.

```
- (GLenum) textureTarget
```

Return Value
The texture target, either GL_TEXTURE_2D or GL_TEXTURE_RECTANGLE_EXT.

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

Declared In
QCPlugIn.h

unbindTextureRepresentationFromCGLContext:textureUnit:

Unbinds the texture from a texture unit.

```
- (void) unbindTextureRepresentationFromCGLContext:(CGLContextObj)cgl_ctx
    textureUnit:(GLenum)unit
```

Parameters

cgl_ctx

A CGL context.)

unit

The texture unit to unbind from (such as, `GL_TEXTURE0`)

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

See Also
`- bindTextureRepresentationToTextureUnit:normalizeCoordinates:` (page 7)

Declared In
`QCPlugIn.h`

unlockBufferRepresentation

Releases the memory buffer representation of the image source.

```
- (void) unlockBufferRepresentation
```

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

See Also
`- lockBufferRepresentationWithPixelFormat:colorSpace:` (page 10)

Declared In
`QCPlugIn.h`

unlockTextureRepresentation

Releases the OpenGL texture representation of the image source.

```
- (void) unlockTextureRepresentation
```

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

See Also
`- lockTextureRepresentationWithTarget:colorSpace:forBounds:` (page 10)

Declared In
`QCPlugIn.h`

Document Revision History

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

Date	Notes
2007-07-12	New document that describes the methods for managing image data that's input to a QCPlugIn object.

Document Revision History

Index