

---

# QCPlugInInputImageSource Protocol Reference

[Cocoa > Graphics & Imaging](#)





Apple Inc.  
© 2007 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, Cocoa, Mac, Mac OS, Objective-C, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

## **QCPlugInInputImageSource Protocol Reference 5**

---

Overview 5

Tasks 5

    Converting an Image to a Representation 5

    Getting Color Space Information 6

    Getting Texture Information 6

    Getting Image Buffer Information 6

Instance Methods 7

    bindTextureRepresentationToCGLContext:textureUnit:normalizeCoordinates: 7

    bufferBaseAddress 7

    bufferBytesPerRow 8

    bufferColorSpace 8

    bufferPixelFormat 8

    bufferPixelsHigh 8

    bufferPixelsWide 9

    imageBounds 9

    imageColorSpace 9

    lockBufferRepresentationWithPixelFormat:colorSpace:forBounds: 10

    lockTextureRepresentationWithColorSpace:forBounds: 10

    shouldColorMatch 11

    textureColorSpace 11

    textureFlipped 11

    textureMatrix 12

    textureName 12

    texturePixelsHigh 13

    texturePixelsWide 13

    textureTarget 13

    unbindTextureRepresentationFromCGLContext:textureUnit: 14

    unlockBufferRepresentation 14

    unlockTextureRepresentation 14

## **Document Revision History 15**

---

## **Index 17**

---



# QCPlugInInputImageSource Protocol Reference

---

|                     |   |
|---------------------|---|
| <b>Framework</b>    | /System/Library/Frameworks/Quartz.framework/Frameworks/QuartzComposer.framework |
| <b>Availability</b> | Available in Mac OS X v10.5 and later.  |
| <b>Declared in</b>  | 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.

- [bindValueRepresentationToCGLContext: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.



- (NSInteger) 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.

- (NSInteger) 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. |

## REVISION HISTORY

### Document Revision History



# Index

---

## B

---

`bindTextureRepresentationToCGLContext:textureUnit:normalizeCoordinates:` **protocol instance method** [7](#)  
`bufferBaseAddress` **protocol instance method** [7](#)  
`bufferBytesPerRow` **protocol instance method** [8](#)  
`bufferColorSpace` **protocol instance method** [8](#)  
`bufferPixelFormat` **protocol instance method** [8](#)  
`bufferPixelsHigh` **protocol instance method** [8](#)  
`bufferPixelsWide` **protocol instance method** [9](#)

## I

---

`imageBounds` **protocol instance method** [9](#)  
`imageColorSpace` **protocol instance method** [9](#)

## L

---

`lockBufferRepresentationWithPixelFormat:colorSpace:forBounds:` **protocol instance method** [10](#)  
`lockTextureRepresentationWithColorSpace:forBounds:` **protocol instance method** [10](#)

## S

---

`shouldColorMatch` **protocol instance method** [11](#)

## T

---

`textureColorSpace` **protocol instance method** [11](#)  
`textureFlipped` **protocol instance method** [11](#)  
`textureMatrix` **protocol instance method** [12](#)  
`textureName` **protocol instance method** [12](#)

`texturePixelsHigh` **protocol instance method** [13](#)  
`texturePixelsWide` **protocol instance method** [13](#)  
`textureTarget` **protocol instance method** [13](#)

## U

---

`unbindTextureRepresentationFromCGLContext:textureUnit:` **protocol instance method** [14](#)  
`unlockBufferRepresentation` **protocol instance method** [14](#)  
`unlockTextureRepresentation` **protocol instance method** [14](#)