
UIImage Class Reference

[Cocoa](#) > [Graphics & Imaging](#)



2007-10-31



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

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Carbon, Cocoa, Mac, Mac OS, 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

UIImage Class Reference 5

Overview	5
Tasks	6
Creating an Image	6
Creating an Image by Modifying an Existing Image	6
Initializing an Image	7
Getting Image Information	7
Class Methods	8
emptyImage	8
imageWithBitmapData:bytesPerRow:size:format:colorSpace:	8
imageWithCGImage:	9
imageWithCGImage:options:	9
imageWithCGLayer:	10
imageWithCGLayer:options:	10
imageWithColor:	10
imageWithContentsOfURL:	11
imageWithContentsOfURL:options:	11
imageWithCVImageBuffer:	12
imageWithCVImageBuffer:options:	13
imageWithData:	13
imageWithData:options:	14
imageWithImageProvider:size:format:colorSpace:options:	14
imageWithTexture:size:flipped:colorSpace:	15
Instance Methods	16
definition	16
extent	16
imageByApplyingTransform:	17
imageByCroppingToRect:	17
initWithBitmapData:bytesPerRow:size:format:colorSpace:	17
initWithCGImage:	18
initWithCGImage:options:	19
initWithCGLayer:	19
initWithCGLayer:options:	20
initWithColor:	20
initWithContentsOfURL:	20
initWithContentsOfURL:options:	21
initWithCVImageBuffer:	21
initWithCVImageBuffer:options:	22
initWithData:	22
initWithData:options:	23
initWithImageProvider:size:format:colorSpace:options:	23

initWithTexture:size:flipped:colorSpace: 24
Constants 25
Pixel Formats 25
Color Space Key 26

Document Revision History 27

Index 29

CIImage Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	Library/Frameworks/QuartzCore.framework
Availability	Mac OS X v10.4 and later
Companion guide	Core Image Programming Guide
Declared in	CIImage.h
Related sample code	CarbonCocoaCoreImageTab CIAnnotation CITransitionSelectorSample2 Reducer WebKitCIPlugIn

Overview

The `CIImage` class represents an image. Core Image images are immutable. You use `CIImage` objects in conjunction with other Core Image classes, such as `CIFilter`, `CIContext`, `CIVector`, and `CIColor`, to take advantage of the built-in Core Image filters when processing images. You can create `CIImage` objects with data supplied from a variety of sources, including Quartz 2D images, Core Video image buffers (`CVImageBufferRef`), URL-based objects, and `NSData` objects.

Although a `CIImage` object has image data associated with it, it is not an image. You can think of a `CIImage` object as an image “recipe.” A `CIImage` object has all the information necessary to produce an image, but Core Image doesn’t actually render an image until it is told to do so. This “lazy evaluation” method allows Core Image to operate as efficiently as possible.

Core Image defines methods for creating and initializing images. Additional methods that support drawing and initializing an image with an `NSBitmapImageRep` object are defined in *CIImage Additions Reference*.

Tasks

Creating an Image

- + [emptyImage](#) (page 8)
Creates and returns an empty image object.
- + [imageWithColor:](#) (page 10)
Creates and returns an image of infinite extent that is initialized the specified color.
- + [imageWithBitmapData:bytesPerRow:size:format:colorSpace:](#) (page 8)
Creates and returns an image object from bitmap data.
- + [imageWithCGImage:](#) (page 9)
Creates and returns an image object from a Quartz 2D image.
- + [imageWithCGImage:options:](#) (page 9)
Creates and returns an image object from a Quartz 2D image using the specified color space.
- + [imageWithCGLayer:](#) (page 10)
Creates and returns an image object from the contents supplied by a `CGLayer` object.
- + [imageWithCGLayer:options:](#) (page 10)
Creates and returns an image object from the contents supplied by a `CGLayer` object, using the specified options.
- + [imageWithContentsOfURL:](#) (page 11)
Creates and returns an image object from the contents of a file.
- + [imageWithContentsOfURL:options:](#) (page 11)
Creates and returns an image object from the contents of a file, using the specified options.
- + [imageWithCVImageBuffer:](#) (page 12)
Creates and returns an image object from the contents of `CVImageBuffer` object.
- + [imageWithCVImageBuffer:options:](#) (page 13)
Creates and returns an image object from the contents of `CVImageBuffer` object, using the specified options.
- + [imageWithData:](#) (page 13)
Creates and returns an image object initialized with the supplied image data.
- + [imageWithData:options:](#) (page 14)
Creates and returns an image object initialized with the supplied image data, using the specified options.
- + [imageWithImageProvider:size:format:colorSpace:options:](#) (page 14)
Creates and returns an image object initialized with data provided by an image provider.
- + [imageWithTexture:size:flipped:colorSpace:](#) (page 15)
Creates and returns an image object initialized with data supplied by an OpenGL texture.

Creating an Image by Modifying an Existing Image

- [imageByApplyingTransform:](#) (page 17)
Returns a new image that represents the original image after applying an affine transform.

- [imageByCroppingToRect:](#) (page 17)
Returns a new image that represents the original image after cropping to a rectangle.

Initializing an Image

- [initWithColor:](#) (page 20)
Initializes an image with the specified color.
- [initWithBitmapData:bytesPerRow:size:format:colorSpace:](#) (page 17)
Initializes an image object with bitmap data.
- [initWithCGImage:](#) (page 18)
Initializes an image object with a Quartz 2D image.
- [initWithCGImage:options:](#) (page 19)
Initializes an image object with a Quartz 2D image, using the specified options.
- [initWithCGLayer:](#) (page 19)
Initializes an image object from the contents supplied by a CGLayer object.
- [initWithCGLayer:options:](#) (page 20)
Initializes an image object from the contents supplied by a CGLayer object, using the specified options.
- [initWithContentsOfURL:](#) (page 20)
Initializes an image object from the contents of a file.
- [initWithContentsOfURL:options:](#) (page 21)
Initializes an image object from the contents of a file, using the specified options.
- [initWithCVImageBuffer:](#) (page 21)
Initializes an image object from the contents of CVImageBuffer object.
- [initWithCVImageBuffer:options:](#) (page 22)
Initializes an image object from the contents of CVImageBuffer object, using the specified options.
- [initWithData:](#) (page 22)
Initializes an image object with the supplied image data.
- [initWithData:options:](#) (page 23)
Initializes an image object with the supplied image data, using the specified options.
- [initWithImageProvider:size:format:colorSpace:options:](#) (page 23)
Initializes an image object with data provided by an image provider, using the specified options.
- [initWithTexture:size:flipped:colorSpace:](#) (page 24)
Initializes an image object with data supplied by an OpenGL texture.

Getting Image Information

- [definition](#) (page 16)
Returns a filter shape object that represents the domain of definition of the image.
- [extent](#) (page 16)
Returns a rectangle that specifies the extent of the image.

Class Methods

emptyImage

Creates and returns an empty image object.

```
+ (CImage *)emptyImage
```

Return Value

An image object.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CImage.h

initWithBitmapData:bytesPerRow:size:format:colorSpace:

Creates and returns an image object from bitmap data.

```
+ (CImage *)initWithBitmapData:(NSData *)d bytesPerRow:(size_t)bpr
    size:(CGSize)size format:(CIFormat)f colorSpace:(CGColorSpaceRef)cs
```

Parameters

d

The bitmap data for the image. This data must be premultiplied.

bpr

The number of bytes per row.

size

The dimensions of the image.

f

The format and size of each pixel. You must supply a pixel format constant. See “[Pixel Formats](#)” (page 25).

cs

The color space that the image is defined in. If this value is `nil`, the image is not color matched. Pass `nil` for images that don't contain color data (such as elevation maps, normal vector maps, and sampled function tables).

Return Value

An image object.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithBitmapData:bytesPerRow:size:format:colorSpace:](#) (page 17)

Declared In

CImage.h

initWithCGImage:

Creates and returns an image object from a Quartz 2D image.

```
+ (CImage *)initWithCGImage:(CGImageRef) image
```

Parameters

image

A Quartz 2D image (CGImageRef) object. For more information, see *Quartz 2D Programming Guide* and *CImage Reference*.

Return Value

An image object initialized with the contents of the Quartz 2D image.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithCGImage:options:](#) (page 9)

- [initWithCGImage:](#) (page 18)

Related Sample Code

CVideoDemoGL

Declared In

CImage.h

initWithCGImage:options:

Creates and returns an image object from a Quartz 2D image using the specified color space.

```
+ (CImage *)initWithCGImage:(CGImageRef) image options:(NSDictionary *)d
```

Parameters

image

A Quartz 2D image (CGImageRef) object. For more information, see *Quartz 2D Programming Guide* and *CImage Reference*.

d

A dictionary that contains a color space key ([kCIColorSpace](#) (page 26)) whose value is a CGColorSpace object. (See CGColorSpaceRef.)

Return Value

An image object initialized with the contents of the Quartz 2D image and the specified color space.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithCGImage:](#) (page 9)

- [initWithCGImage:options:](#) (page 19)

Declared In

CImage.h

imageWithCGLayer:

Creates and returns an image object from the contents supplied by a `CGLayer` object.

```
+ (CImage *)imageWithCGLayer:(CGLayerRef)layer
```

Parameters

layer

A `CGLayer` object. For more information see *Quartz 2D Programming Guide* and *CGLayer Reference*.

Return Value

An image object initialized with the contents of the layer object.

Availability

Mac OS X v10.4 and later.

See Also

+ [imageWithCGLayer:options:](#) (page 10)

- [initWithCGLayer:](#) (page 19)

Declared In

`CImage.h`

imageWithCGLayer:options:

Creates and returns an image object from the contents supplied by a `CGLayer` object, using the specified options.

```
+ (CImage *)imageWithCGLayer:(CGLayerRef)layer options:(NSDictionary *)d
```

Parameters

layer

A `CGLayer` object. For more information see *Quartz 2D Programming Guide* and *CGLayer Reference*.

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See [“Pixel Formats”](#) (page 25).

Return Value

An image object initialized with the contents of the layer object and set up with the specified options.

Availability

Mac OS X v10.4 and later.

See Also

+ [imageWithCGLayer:](#) (page 10)

- [initWithCGLayer:options:](#) (page 20)

Declared In

`CImage.h`

imageWithColor:

Creates and returns an image of infinite extent that is initialized the specified color.

```
+ (CIImage *)initWithColor:(CIColor *)color
```

Parameters

color

A color object.

Return Value

The image object initialized with the color represented by the `CIColor` object.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [initWithColor:](#) (page 20)

Declared In

CIImage.h

initWithContentsOfURL:

Creates and returns an image object from the contents of a file.

```
+ (CIImage *)initWithContentsOfURL:(NSURL *)url
```

Parameters

url

The location of the file.

Return Value

An image object initialized with the contents of the file.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithContentsOfURL:options:](#) (page 11)

- [initWithContentsOfURL:](#) (page 20)

Related Sample Code

CIAnnotation

CITransitionSelectorSample2

Declared In

CIImage.h

initWithContentsOfURL:options:

Creates and returns an image object from the contents of a file, using the specified options.

```
+ (CIImage *)initWithContentsOfURL:(NSURL *)url options:(NSDictionary *)d
```

Parameters

url

The location of the file.

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).

Return Value

An image object initialized with the contents of the file and set up with the specified options.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithContentsOfURL:](#) (page 11)

- [initWithContentsOfURL:options:](#) (page 21)

Declared In

CIImage.h

initWithCVMImageBuffer:

Creates and returns an image object from the contents of CVMImageBuffer object.

+ (CIImage *)initWithCVMImageBuffer:(CVMImageBufferRef) *imageBuffer*

Parameters

imageBuffer

A CVMImageBuffer object. For more information, see *Core Video Programming Guide* and *Core Video Reference*.

Return Value

An image object initialized with the contents of the image buffer object.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithCVMImageBuffer:options:](#) (page 13)

- [initWithCVMImageBuffer:](#) (page 21)

Related Sample Code

CIVideoDemoGL

QTCarbonCoreImage101

QTCoreImage101

WhackedTV

Declared In

CIImage.h

initWithCVImageBuffer:options:

Creates and returns an image object from the contents of `CVImageBuffer` object, using the specified options.

```
+ (UIImage *)initWithCVImageBuffer:(CVImageBufferRef)imageBuffer
    options:(NSDictionary *)dict
```

Parameters

imageBuffer

A `CVImageBuffer` object. For more information, see *Core Video Programming Guide* and *Core Video Reference*.

dict

A dictionary that contains options for creating an image object. You can supply such options as a color space. (The pixel format is supplied by the `CVImageBuffer` object.)

Return Value

An image object initialized with the contents of the image buffer object and set up with the specified options.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithCVImageBuffer:](#) (page 12)

- [initWithCVImageBuffer:options:](#) (page 22)

Declared In

`UIImage.h`

initWithData:

Creates and returns an image object initialized with the supplied image data.

```
+ (UIImage *)initWithData:(NSData *)data
```

Parameters

data

The data object that holds the contents of an image file (such as TIFF, GIF, JPG, or whatever else the system supports). The image data must be premultiplied.

Return Value

An image object initialized with the supplied data, or `nil` if the method cannot create an image representation from the contents of the supplied data object.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithData:options:](#) (page 14)

- [initWithData:](#) (page 22)

Related Sample Code

[LayerBackedOpenGLView](#)

[WebKitCIPlugin](#)

Declared In

CIImage.h

initWithData:options:

Creates and returns an image object initialized with the supplied image data, using the specified options.

```
+ (CIImage *)initWithData:(NSData *)data options:(NSDictionary *)d
```

Parameters*data*

A pointer to the image data. The data must be premultiplied

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).

Return Value

An image object initialized with the supplied data and set up with the specified options.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithData:](#) (page 13)

- [initWithData:options:](#) (page 23)

Declared In

CIImage.h

initWithImageProvider:size:format:colorSpace:options:

Creates and returns an image object initialized with data provided by an image provider.

```
+ (CIImage *)initWithImageProvider:(id)p size:(size_t)width :(size_t)height
    format(CIFormat)f colorSpace:(CGColorSpaceRef)cs options:(NSDictionary *)dict
```

Parameters*p*

A data provider that implements the `CIImageProvider` informal protocol. Core Image retains this data until the image is deallocated.

width

The width of the image.

height

The height of the image.

f

A pixel format constant. See “[Pixel Formats](#)” (page 25).

cs

The color space that the image is defined in. If the this value is `nil`, the image is not color matched. Pass `nil` for images that don’t contain color data (such as elevation maps, normal vector maps, and sampled function tables).

dict

A dictionary that specifies image-creation options, which can be `kCImageProviderTileSize` or `kCImageProviderUserInfo`. See *CImageProvider Protocol Reference* for more information on these options.

Return Value

An image object initialized with the data from the data provider. Core Image does not populate the image object until the object needs the data.

Availability

Mac OS X v10.4 and later.

Declared In

`CImageProvider.h`

See Also

- [initWithImageProvider:size::format:colorSpace:options:](#) (page 23)

initWithTexture:size:flipped:colorSpace:

Creates and returns an image object initialized with data supplied by an OpenGL texture.

```
+ (CImage *)initWithTexture:(unsigned int)name size:(CGSize)size flipped:(BOOL)flag
  colorSpace:(CGColorSpaceRef)cs
```

Parameters

name

An OpenGL texture. Because `CImage` objects are immutable, the texture must remain unchanged for the life of the image object. See the discussion for more information.

size

The dimensions of the texture.

flag

YES to have Core Image flip the contents of the texture vertically.

cs

The color space that the image is defined in. If the `colorSpace` value is `nil`, the image is not color matched. Pass `nil` for images that don't contain color data (such as elevation maps, normal vector maps, and sampled function tables).

Return Value

An image object initialized with the texture data.

Discussion

When using a texture to create a `CImage` object, the texture must be valid in the Core Image context (`CImageContext`) that you draw the `CImage` object into. This means that one of the following must be true:

- The texture must be created using the `CGLContext` object that the `CImageContext` is based on.
- The context that the texture was created in must be shared with the `CGLContext` that the `CImageContext` is based on.

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. When you no longer need the image, you can delete the texture.

Core Image ignores the texture filtering and wrap modes (`GL_TEXTURE_FILTER` and `GL_TEXTURE_WRAP`) that you set through OpenGL. The filter and wrap modes are overridden by what the `CISampler` object specifies when you apply a filter to the `CImage` object.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithTexture:size:flipped:colorSpace:](#) (page 24)

Declared In

`CImage.h`

Instance Methods

definition

Returns a filter shape object that represents the domain of definition of the image.

- (`CIFilterShape *`)`definition`

Return Value

A filter shape object.

Availability

Mac OS X v10.4 and later.

See Also

- [extent](#) (page 16)

Declared In

`CImage.h`

extent

Returns a rectangle that specifies the extent of the image.

- (`CGRect`)`extent`

Return Value

A rectangle that specifies the extent of the image in working space coordinates.

Availability

Mac OS X v10.4 and later.

See Also

- [definition](#) (page 16)

Related Sample Code

`CIVideoDemoGL`

`QTCarbonCoreImage101`

QTCoreImage101

Reducer

UnsharpMask

Declared In

CIImage.h

imageByApplyingTransform:

Returns a new image that represents the original image after applying an affine transform.

```
- (CIImage *)imageByApplyingTransform:(CGAffineTransform)matrix
```

Parameters

matrix

An affine transform.

Return Value

The transformed image object.

Availability

Mac OS X v10.4 and later.

See Also

- [imageByCroppingToRect:](#) (page 17)

Declared In

CIImage.h

imageByCroppingToRect:

Returns a new image that represents the original image after cropping to a rectangle.

```
- (CIImage *)imageByCroppingToRect:(CGRect)r
```

Return Value

An image object cropped to the specified rectangle.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [imageByApplyingTransform:](#) (page 17)

Declared In

CIImage.h

initWithBitmapData:bytesPerRow:size:format:colorSpace:

Initializes an image object with bitmap data.

```
- (id)initWithBitmapData:(NSData *)d bytesPerRow:(size_t)bpr size:(CGSize)size
  format:(CIFormat)f colorSpace:(CGColorSpaceRef)c
```

Parameters*d*

The bitmap data to use for the image. The data you supply must be premultiplied.

bpr

The number of bytes per row.

size

The size of the image data.

f

A pixel format constant. See [“Pixel Formats”](#) (page 25).

c

The color space that the image is defined in and must be a Quartz 2D color space ([CGColorSpaceRef](#)). Pass `nil` for images that don’t contain color data (such as elevation maps, normal vector maps, and sampled function tables).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

+ [initWithBitmapData:bytesPerRow:size:format:colorSpace:](#) (page 8)

Declared In

`CIImage.h`

initWithCGImage:

Initializes an image object with a Quartz 2D image.

```
- (id)initWithCGImage:(CGImageRef)image
```

Parameters*image*

A Quartz 2D image ([CGImageRef](#)) object. For more information, see [Quartz 2D Programming Guide](#) and [CGImage Reference](#).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCGImage:options:](#) (page 19)

+ [initWithCGImage:](#) (page 9)

Declared In

`CIImage.h`

initWithCGImage:options:

Initializes an image object with a Quartz 2D image, using the specified options.

```
- (id)initWithCGImage:(CGImageRef)image options:(NSDictionary *)d
```

Parameters

image

A Quartz 2D image (CGImageRef) object. For more information, see *Quartz 2D Programming Guide* and *CImage Reference*.

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCGImage:](#) (page 18)

+ [imageWithCGImage:options:](#) (page 9)

Declared In

CImage.h

initWithCGLayer:

Initializes an image object from the contents supplied by a CGLayer object.

```
- (id)initWithCGLayer:(CGLayerRef)layer
```

Parameters

layer

A CGLayer object. For more information see *Quartz 2D Programming Guide* and *CGLayer Reference*.

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCGLayer:options:](#) (page 20)

+ [imageWithCGLayer:](#) (page 10)

Related Sample Code

CIAnnotation

QTCarbonCoreImage101

Declared In

CImage.h

initWithCGLayer:options:

Initializes an image object from the contents supplied by a CGLayer object, using the specified options.

```
- (id)initWithCGLayer:(CGLayerRef)layer options:(NSDictionary *)d
```

Parameters

layer

A CGLayer object. For more information see *Quartz 2D Programming Guide* and *CGLayer Reference*.

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCGLayer:](#) (page 19)
- + [initWithCGLayer:options:](#) (page 10)

Declared In

CIImage.h

initWithColor:

Initializes an image with the specified color.

```
- (id)initWithColor:(CIColor *)color
```

Parameters

color

A color object.

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Available in Mac OS X v10.5 and later.

See Also

- + [initWithColor:](#) (page 10)

Declared In

CIImage.h

initWithContentsOfURL:

Initializes an image object from the contents of a file.

```
- (id)initWithContentsOfURL:(NSURL *)url
```

Parameters*url*

The location of the file.

Return ValueThe initialized image object or `nil` if the object could not be initialized.**Availability**

Mac OS X v10.4 and later.

See Also- [initWithContentsOfURL:options:](#) (page 21)+ [initWithContentsOfURL:](#) (page 11)**Declared In**

CImage.h

initWithContentsOfURL:options:

Initializes an image object from the contents of a file, using the specified options.

- (id)initWithContentsOfURL:(NSURL *)*url* options:(NSDictionary *)*d***Parameters***url*

The location of the file.

*d*A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).**Return Value**The initialized image object or `nil` if the object could not be initialized.**Availability**

Mac OS X v10.4 and later.

See Also- [initWithContentsOfURL:](#) (page 20)+ [initWithContentsOfURL:options:](#) (page 11)**Declared In**

CImage.h

initWithCVMImageBuffer:

Initializes an image object from the contents of CVMImageBuffer object.

- (id)initWithCVMImageBuffer:(CVMImageBufferRef) *imageBuffer*

Parameters*imageBuffer*

A `CVImageBuffer` object. For more information, see *Core Video Programming Guide* and *Core Video Reference*.

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCVImageBuffer:options:](#) (page 22)

+ [imageWithCVImageBuffer:](#) (page 12)

Related Sample Code

`VideoViewer`

Declared In

`CIImage.h`

initWithCVImageBuffer:options:

Initializes an image object from the contents of `CVImageBuffer` object, using the specified options.

```
- (id)initWithCVImageBuffer:(CVImageBufferRef)imageBuffer options:(NSDictionary *)dict
```

Parameters*imageBuffer*

A `CVImageBuffer` object. For more information, see *Core Video Programming Guide* and *Core Video Reference*.

dict

A dictionary that contains options for creating an image object. You can supply such options as a color space. (The pixel format is supplied by the `CVImageBuffer` object.)

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithCVImageBuffer:](#) (page 21)

+ [imageWithCVImageBuffer:options:](#) (page 13)

Declared In

`CIImage.h`

initWithData:

Initializes an image object with the supplied image data.

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

Parameters

data

The image data. The data you supply must be premultiplied.

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithData:options:](#) (page 23)

+ [imageWithData:](#) (page 13)

Declared In

CImage.h

initWithData:options:

Initializes an image object with the supplied image data, using the specified options.

```
- (id)initWithData:(NSData *)data options:(NSDictionary *)d
```

Parameters

data

The image data. The data you supply must be premultiplied.

d

A dictionary that contains options for creating an image object. You can supply such options as a pixel format and a color space. See “[Pixel Formats](#)” (page 25).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithData:](#) (page 22)

+ [imageWithData:options:](#) (page 14)

Declared In

CImage.h

initWithImageProvider:size:format:colorSpace:options:

Initializes an image object with data provided by an image provider, using the specified options.

```
- (id)initWithImageProvider:(id)p size:(size_t)width:(size_t)height
    format:(CIFormat)f colorSpace:(CGColorSpaceRef)cs options:(NSDictionary *)dict
```

Parameters*p*

A data provider that implements the `CIColorProvider` informal protocol. Core Image retains this data until the image is deallocated.

width

The width of the image data.

height

The height of the image data.

f

A pixel format constant. See “[Pixel Formats](#)” (page 25).

cs

The color space of the image. If this value is `nil`, the image is not color matched. Pass `nil` for images that don't contain color data (such as elevation maps, normal vector maps, and sampled function tables).

dict

A dictionary that specifies image-creation options, which can be `kCIColorProviderTileSize` or `kCIColorProviderUserInfo`. See *CIColorProvider Protocol Reference* for more information on these options.

Return Value

The initialized image object or `nil` if the object could not be initialized.

Discussion

Core Image does not populate the image until it actually needs the data.

Availability

Mac OS X v10.4 and later.

Declared In

`CIColorProvider.h`

See Also

+ [initWithImageProvider:size::format:colorSpace:options:](#) (page 14)

initWithTexture:size:flipped:colorSpace:

Initializes an image object with data supplied by an OpenGL texture.

```
- (id)initWithTexture:(unsigned int)name size:(CGSize)size flipped:(BOOL)flag
  colorSpace:(CGColorSpaceRef)cs
```

Parameters*name*

An OpenGL texture. Because `CIColor` objects are immutable, the texture must remain unchanged for the life of the image object. See the discussion for more information.

size

The dimensions of the texture.

flag

YES to have Core Image flip the contents of the texture vertically.

`CS`

The color space that the image is defined in. This must be a Quartz color space (`CGColorSpaceRef`). If the `colorSpace` value is `nil`, the image is not color matched. Pass `nil` for images that don't contain color data (such as elevation maps, normal vector maps, and sampled function tables).

Return Value

The initialized image object or `nil` if the object could not be initialized.

Discussion

When using a texture to create a `CImage` object, the texture must be valid in the Core Image context (`CImageContext`) that you draw the `CImage` object into. This means that one of the following must be true:

- The texture must be created using the `CGLContext` object that the `CImageContext` is based on.
- The context that the texture was created in must be shared with the `CGLContext` that the `CImageContext` is based on.

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. When you no longer need the image, you can delete the texture.

Core Image ignores the texture filtering and wrap modes (`GL_TEXTURE_FILTER` and `GL_TEXTURE_WRAP`) that you set through OpenGL. The filter and wrap modes are overridden by what the `CISampler` object specifies when you apply a filter to the `CImage` object.

Availability

Mac OS X v10.4 and later.

See Also

+ [imageWithTexture:size:flipped:colorSpace:](#) (page 15)

Declared In

`CImage.h`

Constants

Pixel Formats

Image data pixel formats.

```
extern CFormat kCFormatARGB8;
extern CFormat kCFormatRGBA16;
extern CFormat kCFormatRGBAf;
```

Constants

`CFormat`

The data type for a pixel format.

`kCFormatARGB8`

A 32 bit-per-pixel, fixed-point pixel format.

`kCFormatRGBA16`

A 64 bit-per-pixel, fixed-point pixel format.

`kCIFormatRGBAf`

A 128 bit-per-pixel, floating-point pixel format.

Declared In

`CIImage.h`

Color Space Key

A key for the color space of an image.

```
extern NSString *kCIImageColorSpace;
```

Constants

`kCIImageColorSpace`

The key for a color space. The value you supply for this dictionary key must be a `CGColorSpaceRef` data type. For more information on this data type see *CGColorSpace Reference*. Typically you use this option when you need to load an elevation, mask, normal vector, or RAW sensor data directly from a file without color correcting it. This constant specifies to override Core Image, which, by default, assumes that data is in `GenericRGB`.

Declared In

`CIImage.h`

Document Revision History

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

Date	Notes
2007-10-31	Updated for Mac OS X v10.5.
	Added information about texture filtering and wrap modes to initWithTexture:size:flipped:colorSpace: (page 15) and initWithTexture:size:flipped:colorSpace: (page 24).
	Revised information on options for initWithCGImage:options: (page 9).
2006-05-23	First publication of this content as a separate document.
	Added parameter descriptions and updated Class Description.
	Revised discussion for initWithTexture:size:flipped:colorSpace: (page 15) and initWithTexture:size:flipped:colorSpace: (page 24).
	Added the constant kCIColorFormatRGBA16 (page 25).

REVISION HISTORY

Document Revision History

Index

C

CIFFormat **constant** 25
Color Space Key 26

D

definition **instance method** 16

E

emptyImage **class method** 8
extent **instance method** 16

I

imageByApplyingTransform: **instance method** 17
imageByCroppingToRect: **instance method** 17
imageWithBitmapData:bytesPerRow:size:format:
colorSpace: **class method** 8
imageWithCGImage: **class method** 9
imageWithCGImage:options: **class method** 9
imageWithCGLayer: **class method** 10
imageWithCGLayer:options: **class method** 10
imageWithColor: **class method** 10
imageWithContentsOfURL: **class method** 11
imageWithContentsOfURL:options: **class method**
11
imageWithCVImageBuffer: **class method** 12
imageWithCVImageBuffer:options: **class method**
13
imageWithData: **class method** 13
imageWithData:options: **class method** 14
imageWithImageProvider:size:format:colorSpace:
options: **class method** 14

imageWithTexture:size:flipped:colorSpace:
class method 15
initWithBitmapData:bytesPerRow:size:format:
colorSpace: **instance method** 17
initWithCGImage: **instance method** 18
initWithCGImage:options: **instance method** 19
initWithCGLayer: **instance method** 19
initWithCGLayer:options: **instance method** 20
initWithColor: **instance method** 20
initWithContentsOfURL: **instance method** 20
initWithContentsOfURL:options: **instance method**
21
initWithCVImageBuffer: **instance method** 21
initWithCVImageBuffer:options: **instance method**
22
initWithData: **instance method** 22
initWithData:options: **instance method** 23
initWithImageProvider:size:format:colorSpace:
options: **instance method** 23
initWithTexture:size:flipped:colorSpace:
instance method 24

K

kCIFFormatARGB8 **constant** 25
kCIFFormatRGBA16 **constant** 25
kCIFFormatRGBAf **constant** 26
kCIImageColorSpace **constant** 26

P

Pixel Formats 25