
CI Sampler Class Reference

[Cocoa > Graphics & Imaging](#)



2006-12-07



Apple Inc.
© 2006 Apple Computer, 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, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

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

CISampler Class Reference 5

Overview 5

Tasks 5

 Creating a Sampler 5

 Initializing a Sampler 6

 Getting Information About the Sampler Object 6

Class Methods 6

 samplerWithImage: 6

 samplerWithImage:keysAndValues: 6

 samplerWithImage:options: 7

Instance Methods 8

 definition 8

 extent 8

 initWithImage: 8

 initWithImage:keysAndValues: 9

 initWithImage:options: 9

Constants 10

 Sampler Option Keys 10

 Sampler Option Values 10

Document Revision History 13

Index 15

CISampler Class Reference

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

Overview

The `CISampler` class retrieves samples of images for processing by a `CIKernel` object. A `CISampler` object defines a coordinate transform, and modes for interpolation and wrapping. You use `CISampler` objects in conjunction with other Core Image classes, such as `CIFilter`, `CIKernel`, and `CIFilterShape`, to create custom filters.

Tasks

Creating a Sampler

- + [samplerWithImage:](#) (page 6)
Creates and returns a sampler that references an image.
- + [samplerWithImage:keysAndValues:](#) (page 6)
Creates and returns a sampler that references an image using options specified as key-value pairs.
- + [samplerWithImage:options:](#) (page 7)
Creates and returns a sampler that references an image using options specified in a dictionary.

Initializing a Sampler

- [initWithImage:](#) (page 8)
Initializes a sampler with an image object.
- [initWithImage:keysAndValues:](#) (page 9)
Initializes the sampler with an image object using options specified as key-value pairs.
- [initWithImage:options:](#) (page 9)
Initializes the sampler with an image object using options specified in a dictionary.

Getting Information About the Sampler Object

- [definition](#) (page 8)
Gets the domain of definition (DOD) of the sampler.
- [extent](#) (page 8)
Gets the rectangle that specifies the extent of the sampler.

Class Methods

samplerWithImage:

Creates and returns a sampler that references an image.

```
+ (CISampler *)samplerWithImage:(CIImage *)im
```

Parameters

im

The image that you want the sampler to reference.

Return Value

A sampler object that references the image specified by the *im* argument.

Availability

Mac OS X v10.4 and later.

See Also

- + [samplerWithImage:keysAndValues:](#) (page 6)
- + [samplerWithImage:options:](#) (page 7)

Related Sample Code

CIAnnotation

Declared In

CISampler.h

samplerWithImage:keysAndValues:

Creates and returns a sampler that references an image using options specified as key-value pairs.

```
+ (CISampler *)samplerWithImage:(CIImage *)im keysAndValues:key0, ...
```

Parameters*im*

The image that you want the sampler to reference.

key0

A list of key-value pairs that represent options. Each key needs to be followed by that appropriate value. You can supply one or more key-value pairs. Use `nil` to specify the end of the key-value options. See “[Sampler Option Keys](#)” (page 10).

Return Value

A sampler that references the image specified by the *im* argument and uses the specified options.

Availability

Mac OS X v10.4 and later.

See Also

+ [samplerWithImage:](#) (page 6)

+ [samplerWithImage:options:](#) (page 7)

Declared In

CISampler.h

samplerWithImage:options:

Creates and returns a sampler that references an image using options specified in a dictionary.

```
+ (CISampler *)samplerWithImage:(CIImage *)im options:(NSDictionary *)dict
```

Parameters*im*

The image that you want the sampler to reference.

dict

A dictionary that contains options specified as key-value pairs. See “[Sampler Option Keys](#)” (page 10).

Return Value

A sampler that references the image specified by the *im* argument and uses the options specified in the dictionary.

Availability

Mac OS X v10.4 and later.

See Also

+ [samplerWithImage:](#) (page 6)

+ [samplerWithImage:keysAndValues:](#) (page 6)

Declared In

CISampler.h

Instance Methods

definition

Gets the domain of definition (DOD) of the sampler.

```
- (CIFilterShape *)definition
```

Return Value

The filter shape object that contains the DOD.

Discussion

The DOD contains all nontransparent pixels produced by referencing the sampler.

Availability

Mac OS X v10.4 and later.

Related Sample Code

CIAnnotation

Declared In

CISampler.h

extent

Gets the rectangle that specifies the extent of the sampler.

```
- (CGRect)extent
```

Return Value

The rectangle that specifies the area outside which the wrap mode set for the sampler is invoked.

Availability

Mac OS X v10.4 and later.

Related Sample Code

CIAnnotation

Declared In

CISampler.h

initWithImage:

Initializes a sampler with an image object.

```
- (id)initWithImage:(CIImage *)im
```

Parameters

im

The image object to initialize the sampler with.

Availability

Mac OS X v10.4 and later.

See Also

- [initWithImage:keysAndValues:](#) (page 9)
- [initWithImage:options:](#) (page 9)

Declared In

CISampler.h

initWithImage:keysAndValues:

Initializes the sampler with an image object using options specified as key-value pairs.

```
- (id)initWithImage:(CIImage *)im keysAndValues:key0, ...
```

Parameters

im

The image object to initialize the sampler with.

key0

A list of key-value pairs that represent options. Each key needs to be followed by that appropriate value. You can supply one or more key-value pairs. Use `nil` to specify the end of the key-value options. See “[Sampler Option Keys](#)” (page 10).

Availability

Mac OS X v10.4 and later.

See Also

- [initWithImage:](#) (page 8)
- [initWithImage:options:](#) (page 9)

Declared In

CISampler.h

initWithImage:options:

Initializes the sampler with an image object using options specified in a dictionary.

```
- (id)initWithImage:(CIImage *)im options:(NSDictionary *)dict
```

Parameters

im

The image to initialize the sampler with.

dict

A dictionary that contains options specified as key-value pairs. See “[Sampler Option Keys](#)” (page 10).

Availability

Mac OS X v10.4 and later.

See Also

- [initWithImage:](#) (page 8)

- initWithImage:keysAndValues: (page 9)

Declared In
CISampler.h

Constants

Sampler Option Keys

Keys for creating a sampler.

```
extern NSString *kCISamplerAffineMatrix;
extern NSString *kCISamplerWrapMode;
extern NSString *kCISamplerFilterMode
```

Constants

kCISamplerAffineMatrix

The key for an affine matrix. The associated value is an NSArray object ($[a\ b\ c\ d\ tx\ ty]$) that defines the transformation to apply to the sampler.

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

kCISamplerWrapMode

The key for the sampler wrap mode. The wrap mode specifies how Core Image produces pixels that are outside the extent of the sample. Possible values are [kCISamplerWrapBlack](#) (page 11) and [kCISamplerWrapClamp](#) (page 11).

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

kCISamplerFilterMode

The key for the filtering to use when sampling the image. Possible values are [kCISamplerFilterNearest](#) (page 11) and [kCISamplerFilterLinear](#) (page 11).

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

Declared In
CISampler.h

Sampler Option Values

Values for sampler option keys.

```
extern NSString *kCISamplerWrapBlack;  
extern NSString *kCISamplerWrapClamp;  
extern NSString *kCISamplerFilterNearest;  
extern NSString *kCISamplerFilterLinear;
```

Constants

kCISamplerWrapBlack

Pixels are transparent black.

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

kCISamplerWrapClamp

Coordinates are clamped to the extent.

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

kCISamplerFilterNearest

Nearest neighbor sampling.

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

kCISamplerFilterLinear

Bilinear interpolation.

Available in Mac OS X v10.4 and later.

Declared in CISampler.h.

Declared In

CISampler.h

Document Revision History

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

Date	Notes
2006-12-07	Updated formatting.
2006-05-23	First publication of this content as a separate document.
	Added parameter description and updated Class Description.

REVISION HISTORY

Document Revision History

Index

D

definition [instance method 8](#)

E

extent [instance method 8](#)

I

initWithImage: [instance method 8](#)

initWithImage:keysAndValues: [instance method 9](#)

initWithImage:options: [instance method 9](#)

K

kCISamplerAffineMatrix [constant 10](#)

kCISamplerFilterLinear [constant 11](#)

kCISamplerFilterMode [constant 10](#)

kCISamplerFilterNearest [constant 11](#)

kCISamplerWrapBlack [constant 11](#)

kCISamplerWrapClamp [constant 11](#)

kCISamplerWrapMode [constant 10](#)

S

Sampler Option Keys [10](#)

Sampler Option Values [10](#)

samplerWithImage: [class method 6](#)

samplerWithImage:keysAndValues: [class method 6](#)

samplerWithImage:options: [class method 7](#)