

---

# CIImageAccumulator Class Reference

Graphics & Animation: 2D Drawing



2007-01-05



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, 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

## **CIImageAccumulator Class Reference 5**

---

Overview	5
Tasks	5
Creating an Image Accumulator	5
Initializing an Image Accumulator	5
Setting an Image	6
Obtaining Data From an Image Accumulator	6
Resetting an Accumulator	6
Class Methods	6
imageAccumulatorWithExtent:format:	6
Instance Methods	7
clear	7
extent	7
format	7
image	8
initWithExtent:format:	8
setImage:	8
setImage:dirtyRect:	9

## **Document Revision History 11**

---



# CIImageAccumulator Class Reference

---

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

## Overview

The `CIImageAccumulator` class enables feedback-based image processing for such things as iterative painting operations or fluid dynamics simulations. You use `CIImageAccumulator` objects in conjunction with other Core Image classes, such as `CIFilter`, `CIImage`, `CIVector`, and `CIText`, to take advantage of the built-in Core Image filters when processing images.

## Tasks

### Creating an Image Accumulator

- + `imageAccumulatorWithExtent:format:` (page 6)  
Creates an image accumulator with the specified extent and pixel format.

### Initializing an Image Accumulator

- `initWithExtent:format:` (page 8)  
Initializes an image accumulator with the specified extent and pixel format.

## Setting an Image

- [setImage:](#) (page 8)  
Sets the contents of the image accumulator to the contents of the specified image object.
- [setImage:dirtyRect:](#) (page 9)  
Updates an image accumulator with a subregion of an image object.

## Obtaining Data From an Image Accumulator

- [extent](#) (page 7)  
Returns the extent of the image associated with the image accumulator.
- [format](#) (page 7)  
Returns the pixel format of the image accumulator.
- [image](#) (page 8)  
Returns the current contents of the image accumulator.

## Resetting an Accumulator

- [clear](#) (page 7)  
Resets the accumulator, discarding any pending updates and the current content.

## Class Methods

### imageAccumulatorWithExtent:format:

Creates an image accumulator with the specified extent and pixel format.

```
+ (CImageAccumulator *)imageAccumulatorWithExtent:(CGRect)r format:(CIFormat)f
```

#### Parameters

*r*  
A rectangle that specifies the x-value of the rectangle origin, the y-value of the rectangle origin, and the width and height.

*f*  
The format and size of each pixel. You must supply a pixel format constant, such as `kCIFormatARGB8` (32 bit-per-pixel, fixed-point pixel format) or `kCIFormatRGBAf` (128 bit-per-pixel, floating-point pixel format). See *CImage Class Reference* for more information about pixel format constants.

#### Return Value

The image accumulator object.

#### Availability

Mac OS X v10.4 and later.

#### See Also

- [initWithExtent:format:](#) (page 8)

**Declared In**

CIImageAccumulator.h

## Instance Methods

### clear

Resets the accumulator, discarding any pending updates and the current content.

- (void)clear

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CIImageAccumulator.h

### extent

Returns the extent of the image associated with the image accumulator.

- (CGRect)extent

**Return Value**

The rectangle that specifies the size of the image associated with the image accumulator. This rectangle is the size of the complete region of the working coordinate space, and is a fixed area. It specifies the x-value of the rectangle origin, the y-value of the rectangle origin, and the width and height.

**Availability**

Mac OS X v10.4 and later.

**Declared In**

CIImageAccumulator.h

### format

Returns the pixel format of the image accumulator.

- (CIFormat)format

**Return Value**

The pixel format of the image accumulator.

**Availability**

Mac OS X v10.4 and later.

**Declared In**

CIImageAccumulator.h

## image

Returns the current contents of the image accumulator.

```
- (CIImage *)image
```

### Return Value

The image object that represents the current contents of the image accumulator.

### Availability

Mac OS X v10.4 and later.

### Related Sample Code

CIMicroPaint

### Declared In

CIImageAccumulator.h

## initWithExtent:format:

Initializes an image accumulator with the specified extent and pixel format.

```
- (id)initWithExtent:(CGRect)r format:(CIFormat)f
```

### Parameters

*r*

A rectangle that specifies the x-value of the rectangle origin, the y-value of the rectangle origin, and the width and height.

*f*

The format and size of each pixel. You must supply a pixel format constant, such as `kCIFormatARGB8` (32 bit-per-pixel, fixed-point pixel format) or `kCIFormatRGBAf` (128 bit-per-pixel, floating-point pixel format). See *CImage Class Reference* for more information about pixel format constants.

### Return Value

The initialized image accumulator object.

### Availability

Mac OS X v10.4 and later.

### See Also

[+ imageAccumulatorWithExtent:format:](#) (page 6)

### Related Sample Code

CIAnnotation

CIMicroPaint

### Declared In

CIImageAccumulator.h

## setImage:

Sets the contents of the image accumulator to the contents of the specified image object.



```
- (void)setImage:(CIImage *)im
```

#### Parameters

*im*

The image object whose contents you want to assign to the image accumulator.

#### Availability

Mac OS X v10.4 and later.

#### See Also

- [setImage:dirtyRect:](#) (page 9)

#### Related Sample Code

CIAnnotation

CIMicroPaint

#### Declared In

CIImageAccumulator.h

## setImage:dirtyRect:

Updates an image accumulator with a subregion of an image object.

```
- (void)setImage:(CIImage *)im dirtyRect:(CGRect)r
```

#### Parameters

*im*

The image object whose contents you want to assign to the image accumulator.

*r*

A rectangle that defines the subregion of the image object that's changed since the last time you updated the image accumulator. You must guarantee that the new contents differ from the old only within the region specified by the this argument.

#### Discussion

For additional details on using this method, see “Imaging Dynamical Systems” in *Core Image Programming Guide*.

#### Availability

Mac OS X v10.4 and later.

#### See Also

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

#### Declared In

CIImageAccumulator.h



# Document Revision History

---

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

Date	Notes
2007-01-05	Added information to setImage:dirtyRect:.
2006-05-23	First publication of this content as a separate document.
	Added parameter descriptions and updated Class Description.

## REVISION HISTORY

### Document Revision History