

---

# CIFilterGenerator Class Reference

[Graphics & Imaging](#) > Quartz



2006-12-05



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, 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

---

## **CIFilterGenerator Class Reference 5**

---

Overview	5
Tasks	5
Creating Filter Generator Objects	5
Initializing a Filter Generator Object	6
Connecting and Disconnecting Objects	6
Managing Exported Keys	6
Setting and Getting Class Attributes	6
Archiving a Filter Generator Object	6
Registering a Filter Chain	7
Creating a Filter from a Filter Chain	7
Class Methods	7
filterGenerator	7
filterGeneratorWithContentsOfURL:	7
Instance Methods	8
classAttributes	8
connectObject:withKey:toObject:withKey:	8
disconnectObject:withKey:toObject:withKey:	9
exportedKeys	9
exportKey:fromObject:withName:	10
filter	10
initWithContentsOfURL:	11
registerFilterName:	11
removeExportedKey:	11
setAttributes:forExportedKey:	12
setClassAttributes:	12
writeToURL:atomically:	13
Constants	13
Exported Keys	13

---

## **Document Revision History 15**

---

## **Index 17**

---



# CIFilterGenerator Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying NSObject (NSObject)
<b>Framework</b>	Library/Frameworks/QuartzCore.framework
<b>Availability</b>	Mac OS X v10.5 and later
<b>Declared in</b>	CIFilterGenerator.h
<b>Companion guides</b>	Core Image Programming Guide Core Image Filter Reference

## Overview

The `CIFilterGenerator` class provides methods for creating a `CIFilter` object by chaining together existing `CIFilter` objects to create complex effects. (A **filter chain** refers to the `CIFilter` objects that are connected in the `CIFilterGenerator` object.) The complex effect can be encapsulated as a `CIFilterGenerator` object and saved as a file so that it can be used again. The **filter generator file** contains an archived instance of all the `CIFilter` objects that are chained together.

Any filter generator files that you copy to `/Library/Graphics/Image Units/` are loaded when any of the loading methods provided by the `CIPugin` class are invoked. A `CIFilterGenerator` object is registered by its filename or, if present, by a class attribute that you supply in its description.

You can create a `CIFilterGenerator` object programmatically, using the methods provided by the `CIFilterGenerator` class, or by using the editor view provided by Core Image (see *CIFilter Image Kit Additions*).

## Tasks

### Creating Filter Generator Objects

+ [filterGenerator](#) (page 7)

Creates and returns an empty filter generator object.

- + [filterGeneratorWithContentsOfURL:](#) (page 7)  
Creates and returns a filter generator object and initializes it with the contents of a filter generator file.

## Initializing a Filter Generator Object

- [initWithContentsOfURL:](#) (page 11)  
Initializes a filter generator object with the contents of a filter generator file.

## Connecting and Disconnecting Objects

- [connectObject:withKey:toObject:withKey:](#) (page 8)  
Adds an object to the filter chain.
- [disconnectObject:withKey:toObject:withKey:](#) (page 9)  
Removes the connection between two objects in the filter chain.

## Managing Exported Keys

- [exportedKeys](#) (page 9)  
Returns an array of the exported keys.
- [exportKey:fromObject:withName:](#) (page 10)  
Exports an input or output key of an object in the filter chain.
- [removeExportedKey:](#) (page 11)  
Removes a key that was previously exported.
- [setAttributes:forExportedKey:](#) (page 12)  
Sets a dictionary of attributes for an exported key.

## Setting and Getting Class Attributes

- [classAttributes](#) (page 8)  
Retrieves the class attributes associated with a filter.
- [setClassAttributes:](#) (page 12)  
Sets the class attributes for a filter.

## Archiving a Filter Generator Object

- [writeToURL:atomically:](#) (page 13)  
Archives a filter generator object to a filter generator file.

## Registering a Filter Chain

- [registerFilterName:](#) (page 11)  
Registers the name associated with a filter chain.

## Creating a Filter from a Filter Chain

- [filter](#) (page 10)  
Creates a filter object based on the filter chain.

## Class Methods

### filterGenerator

Creates and returns an empty filter generator object.

```
+ (CIFilterGenerator *)filterGenerator
```

#### Return Value

A `CIFilterGenerator` object.

#### Discussion

You use the returned object to connect two or more `CIFilter` objects and input images. It is also valid to have only one `CIFilter` object in a filter generator.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

+ [filterGeneratorWithContentsOfURL:](#) (page 7)

#### Declared In

`CIFilterGenerator.h`

### filterGeneratorWithContentsOfURL:

Creates and returns a filter generator object and initializes it with the contents of a filter generator file.

```
+ (CIFilterGenerator *)filterGeneratorWithContentsOfURL:(NSURL *)aURL
```

#### Parameters

*aURL*

The location of a filter generator file.

#### Return Value

A `CIFilterGenerator` object; returns `nil` if the file can't be read.

#### Availability

Available in Mac OS X v10.5 and later.

**See Also**

+ [filterGenerator](#) (page 7)

**Declared In**

CIFilterGenerator.h

## Instance Methods

### classAttributes

Retrieves the class attributes associated with a filter.

```
- (NSDictionary *)classAttributes
```

**Return Value**

An `NSDictionary` object that contains the class attributes for a filter, or `nil` if attributes are not set for the filter.

**Discussion**

For more information about class attributes for a filter, see *Core Image Programming Guide* and the filter attributes key constants defined in *CIFilter Class Reference*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setClassAttributes:](#) (page 12)

**Declared In**

CIFilterGenerator.h

### connectObject:withKey:toObject:withKey:

Adds an object to the filter chain.

```
- (void)connectObject:(id)sourceObject withKey:(NSString *)sourceKey
      toObject:(id)targetObject withKey:(NSString *)targetKey
```

**Parameters**

*sourceObject*

A `CIFilter` object, a `CIImage` object, or a the path (an `NSString` or `NSURL` object) to an image.

*sourceKey*

The key that specifies the source object. For example, if the source is the output image of a filter, pass the `outputImage` key. Pass `nil` if the source object is used directly.

*targetObject*

The object that to link the source object to.

*targetKey*

The key that specifies the target for the source. For example, if you are connecting the source to the input image of a `CIFilter` object, you would pass the `inputImage` key.



**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [disconnectObject:withKey:toObject:withKey:](#) (page 9)

**Declared In**

CIFilterGenerator.h

**disconnectObject:withKey:toObject:withKey:**

Removes the connection between two objects in the filter chain.

```
- (void)disconnectObject:(id)sourceObject withKey:(NSString *)key
    toObject:(id)targetObject withKey:(NSString *)targetKey
```

**Parameters**

*sourceObject*

A **CIFilter** object, a **CIImage** object, or a the path (an **NSString** or **NSURL** object) to an image.

*sourceKey*

The key that specifies the source object. Pass `nil` if the source object is used directly.

*targetObject*

The object that you want to disconnect the source object from.

*targetKey*

The key that specifies the target that the source object is currently connected to.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [connectObject:withKey:toObject:withKey:](#) (page 8)

**Declared In**

CIFilterGenerator.h

**exportedKeys**

Returns an array of the exported keys.

```
- (NSDictionary *)exportedKeys
```

**Return Value**

An array of dictionaries that describe the exported key and target object. See [kCIFilterGeneratorExportedKey](#) (page 14), [kCIFilterGeneratorExportedKeyTargetObject](#) (page 14), and [kCIFilterGeneratorExportedKey](#) (page 14) for keys used in the dictionary.

**Discussion**

This method returns the keys that you exported using the [exportKey:fromObject:withName:](#) (page 10) method or that were exported before being written to the file from which you read the filter chain.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [exportKey:fromObject:withName:](#) (page 10)

**Declared In**

CIFilterGenerator.h

**exportKey:fromObject:withName:**

Exports an input or output key of an object in the filter chain.

```
- (void)exportKey:(NSString *)key fromObject:(id)targetObject withName:(NSString *)exportedKeyName
```

**Parameters**

*key*

The key to export from the target object (for example, `inputImage`).

*targetObject*

The object associated with the key (for example, the filter).

*exportedKeyName*

A unique name to use for the exported key. Pass `nil` to use the original key name.

**Discussion**

When you create a `CIFilter` object from a `CIFilterGenerator` object, you might want the filter client to be able to set some of the parameters associated with the filter chain. You can make a parameter settable by exporting the key associated with the parameter. If the exported key represents an input parameter of the filter, the key is exported as an input key. If the key represents an output parameter, it is exported as an output key.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [exportedKeys](#) (page 9)
- [setAttributes:forExportedKey:](#) (page 12)
- [removeExportedKey:](#) (page 11)

**Declared In**

CIFilterGenerator.h

**filter**

Creates a filter object based on the filter chain.

```
- (CIFilter *)filter
```

**Return Value**

A `CIFilter` object.

**Discussion**

The topology of the filter chain is immutable, meaning that any changes you make to the filter chain are not reflected in the filter. The returned filter has the input and output keys that are exported.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CIFilterGenerator.h

**initWithContentsOfURL:**

Initializes a filter generator object with the contents of a filter generator file.

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

**Parameters**

*aURL*

The location of a filter generator file.

**Return Value**

The initialized `CIFilterGenerator` object. Returns `nil` if the file can't be read.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

+ [filterGenerator](#) (page 7)

+ [filterGeneratorWithContentsOfURL:](#) (page 7)

**Declared In**

CIFilterGenerator.h

**registerFilterName:**

Registers the name associated with a filter chain.

```
- (void)registerFilterName:(NSString *)name
```

**Parameters**

*name*

A unique name for the filter chain you want to register.

**Discussion**

This method allows you to register the filter chain as a named filter in the Core Image filter repository. You can then create a `CIFilter` object from it using the `filterWithName:` method of the `CIFilter` class.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CIFilterGenerator.h

**removeExportedKey:**

Removes a key that was previously exported.

```
- (void)removeExportedKey:(NSString *)exportedKeyName
```

**Parameters**

*exportedKeyName*

The name of the key you want to remove.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [exportKey:fromObject:withName:](#) (page 10)

**Declared In**

CIFilterGenerator.h

**setAttributes:forExportedKey:**

Sets a dictionary of attributes for an exported key.

```
- (void)setAttributes:(NSDictionary *)attributes forExportedKey:(NSString *)key
```

**Parameters**

*attributes*

A dictionary that describes the attributes associated with the specified key.

*key*

The exported key whose attributes you want to set.

**Discussion**

By default, the exported key inherits the attributes from its original key and target object. You can use this method to change one or more of the existing attributes for the key, such as the default value or maximum value. For more information on attributes, see *CIFilter Class Reference* and *Core Image Programming Guide*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [exportedKeys](#) (page 9)

- [exportKey:fromObject:withName:](#) (page 10)

**Declared In**

CIFilterGenerator.h

**setClassAttributes:**

Sets the class attributes for a filter.

```
- (void)setClassAttributes:(NSDictionary *)attributes
```

**Parameters**

*attributes*

An `NSDictionary` object that contains the class attributes for a filter. For information on the required attributes, see *CIFilter Class Reference* and *Core Image Programming Guide*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [classAttributes](#) (page 8)

**Declared In**

CIFilterGenerator.h

**writeToURL:atomically:**

Archives a filter generator object to a filter generator file.

- (BOOL)writeToURL:(NSURL \*)*aURL* atomically:(BOOL)*flag*

**Parameters**

*aURL*

A location for the file generator file.

*flag*

Pass `true` to specify that Core Image should create an interim file to avoid overwriting an existing file.

**Return Value**

Returns `true` if the the object is successfully archived to the file.

**Discussion**

Use this method to save your filter chain to a file for later use.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CIFilterGenerator.h

## Constants

**Exported Keys**

Keys for the exported parameters of a filter generator object.

```
extern NSString *const kCIFilterGeneratorExportedKey;  
extern NSString *const kCIFilterGeneratorExportedKeyTargetObject;  
extern NSString *const kCIFilterGeneratorExportedKeyName;
```

### Constants

`kCIFilterGeneratorExportedKeyName`

The key (`CIFilterGeneratorExportedKeyName`) for the name used to export the `CIFilterGenerator` object. The associated value is a string that specifies a unique name for the filter generator object.

Available in Mac OS X v10.5 and later.

Declared in `CIFilterGenerator.h`.

`kCIFilterGeneratorExportedKey`

The key (`CIFilterGeneratorExportedKey`) for the exported parameter. The associated value is the key name of the parameter you are exporting, such as `inputRadius`.

Available in Mac OS X v10.5 and later.

Declared in `CIFilterGenerator.h`.

`kCIFilterGeneratorExportedKeyTargetObject`

The target object (`CIFilterGeneratorExportedKeyTargetObject`) for the exported key. The associated value is the name of the object, such as `CIMotionBlur`.

Available in Mac OS X v10.5 and later.

Declared in `CIFilterGenerator.h`.

### Declared In

`CIFilterGenerator.h`

# Document Revision History

---

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

Date	Notes
2006-12-05	New document that describes the class used to chain together filter objects.

**REVISION HISTORY**

Document Revision History



# Index

---

## C

---

classAttributes **instance method** [8](#)  
connectObject:withKey:toObject:withKey:  
    **instance method** [8](#)

## D

---

disconnectObject:withKey:toObject:withKey:  
    **instance method** [9](#)

## E

---

Exported Keys [13](#)  
exportedKeys **instance method** [9](#)  
exportKey:fromObject:withName: **instance method**  
    [10](#)

## F

---

filter **instance method** [10](#)  
filterGenerator **class method** [7](#)  
filterGeneratorWithContentsOfURL: **class method**  
    [7](#)

## I

---

initWithContentsOfURL: **instance method** [11](#)

## K

---

kCIFilterGeneratorExportedKey **constant** [14](#)  
kCIFilterGeneratorExportedKeyName **constant** [14](#)

kCIFilterGeneratorExportedKeyTargetObject  
    **constant** [14](#)

## R

---

registerFilterName: **instance method** [11](#)  
removeExportedKey: **instance method** [11](#)

## S

---

setAttributes:forExportedKey: **instance method**  
    [12](#)  
setClassAttributes: **instance method** [12](#)

## W

---

writeToURL:atomically: **instance method** [13](#)