CIFilter Image Kit Additions

Graphics & Imaging > Quartz

2006-12-06

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

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

CIFilter Image Kit Additions

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Companion guide	Core Image Programming Guide
Declared in	IKFilterUI.h

Overview

This Image Kit addition to the `CIFilter` class, introduced in Mac OS X v10.5, consists of one method and a set of constants that generate a view with input parameter controls for a Core Image filter. Using this method, it is easier for applications to present a user interface for a filter than it was in Mac OS X v10.4. Then, applications could create a filter user interface only by analyzing the keys and key attributes of a filter and then writing the code to implement the user interface.

You use the `viewForUIConfiguration:excludedKeys:` method to request a view from Core Image. The view is a subclass of the `NSView` class so that you can insert it easily into any other view as a subview or into an `NSWindow` object as a content view. Core Image automatically generates the view for you unless you implement the `IKFilterCustomUIProvider` protocol, in which case calling `viewForUIConfiguration:excludedKeys:` causes Core Image to provide your custom view.

Tasks

Creating a View for a Filter

`– viewForUIConfiguration:excludedKeys:` (page 6)
Returns a filter view for the filter.

Instance Methods

viewForUIConfiguration:excludedKeys:

Returns a filter view for the filter.

```
-(IKFilterUIView*)viewForUIConfiguration:(NSDictionary*)inUIConfiguration
    excludedKeys:(NSArray*)inKeys;
```

Parameters

inUIConfiguration

A dictionary that contains values for the IKUISizeFlavor and kCIUIParameterSet keys. See "User Interface Options" (page 7) for the constants that you can provide as values for IKUISizeFlavor. For kCIUIParameterSet you can provide one of the following values: kCIUISetBasic, kCIUISetIntermediate, kCIUISetAdvanced, or kCIUISetDevelopment. When you request a user interface for a parameter set, all keys for that set and below are included. For example, the advanced set consists of all parameters in the basic, intermediate and advanced sets. The development set should contain parameters that are either experimental or for debugging purposes. You should use them only during the development of filters and client applications, and not in a shipping product.

inKeys

An array of the input keys for which you do *not* want to provide a user interface. Pass nil if you want all input keys to be represented in the user interface.

Return Value

An IKFilterUIView object. You should retain the view as long as you need it, but make sure to release it when you no longer need it as the view is retaining the filter.

Discussion

Calling this method to receive a view for a filter causes the CIFilter class to invoke the provideViewForUIConfiguration:excludedKeys: method. If you override provideViewForUIConfiguration:excludedKeys: the user interface is created by your filter subclass . Otherwise, Core Image automatically generates the user interface based on the filter keys and attributes.

The algorithm used to lay out the controls for a filter operates in a manner similar to the Core Image Fun House application (/Developer/Applications/Graphics Tools/). Applications can retrieve a view whose control sizes complement the size of user interface elements already used in the application. It is also possible to choose which filter input parameters appear in the view. Consumer applications, for example, may want to show a small, basic set of input parameters whereas professional applications may want to provide access to all input parameters.

The controls in the view use bindings to set the values of the filter. See *Cocoa Bindings Programming Topics* if you are unfamiliar with bindings.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IKFilterUI.h

Constants

User Interface Options

Keys or values for the size of the input parameter controls for a filter view.

```
NSString *IKUISizeFlavor;
NSString *IKUISizeMini;
NSString *IKUISizeSmall;
NSString *IKUISizeRegular;
NSString *IKUImaxSize;
NSString *IKUIFlavorAllowFallback;
```

Constants

`IKUISizeFlavor`

A key for the size of the controls in a filter view. The associated value can be `IKUISizeMini`, `IKUISizeSmall`, or `IKUISizeRegular`.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

`IKUISizeMini`

Controls whose size is mini, as defined by Interface Builder 2.5.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

`IKUISizeSmall`

Controls whose size is small, as defined by Interface Builder 2.5.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

`IKUISizeRegular`

Controls whose size is regular or normal, as defined by Interface Builder 2.5.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

`IKUImaxSize`

Controls whose dimensions are the maximum allowable for the filter view. A width or height of `0` indicates that that dimension of the view is not restricted. If the size requested is too small, the filter is expected to return a view as small as possible. It is up to the client to verify that the returned view fits into the context.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

`IKUIFlavorAllowFallback`

Substitute controls of another size. The associated value is a Boolean value. If the filter cannot provide a view for the requested size and a fallback is allowed, the filter can use controls of a different size.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterUI.h`.

Declared In

`IKFilterUI.h`

Document Revision History

This table describes the changes to *CIFilter Image Kit Additions*.

Date	Notes
2006-12-06	New document that describes the category that extends Core Image filters for use with Image Kit.

Index