Image Kit Reference Collection

Graphics & Animation: 2D Drawing

2010-03-24

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

Image Kit Reference Collection

Framework	System/Library/Frameworks/Quartz.framework
Header file directories	Quartz.framework/ImageKit.framework
Declared in	IKCameraDeviceView.h IKDeviceBrowserView.h IKFilterBrowserPanel.h IKFilterBrowserView.h IKFilterUI.h IKFilterUIView.h IKImageBrowserCell.h IKImageBrowserView.h IKImageEditPanel.h IKImageView.h IKPictureTaker.h IKSaveOptions.h IKScannerDeviceView.h IKSlideshow.h

Introduction

The Image Kit framework is an image handling framework that's based on Quartz, Core Image, OpenGL, and Core Animation. It defines a set of Objective-C classes that provide user interface support for:

- Finding, browsing, viewing, and editing images. The Image Kit framework builds on the `NSView` class to provide a view that is optimized for image operations such as rotating, zooming, scrolling, and dragging. Client applications can support powerful in-place editing with a minimum effort.
- Browsing, previewing, and setting input parameters for Core Image filters. The Image Kit framework extends the Core Image API by defining additions to the `CIFilter` class that make it easy for applications to present a user interface for a Core Image filter.
- Extending the Save panel with options for saving images
- Showing slideshows

Image Kit Reference Collection

Classes

PART I

Classes

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.
Declared in	IIKFilterUI.h
Companion guide	Core Image Programming Guide

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 12)
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 13) 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:` (page 145) 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.

Related Sample Code

CIFilterGeneratorTest

CIRAWFilterSample

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`

IKCameraDeviceView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKCameraDeviceView.h
Related sample code	ImageKitDemo

Overview

The `IKCameraDeviceView` class displays the contents of the selected camera.

Tasks

Getting and Setting the Camera Device

`cameraDevice` (page 17) *property*
The current camera device.

View Display Mode

`iconSize` (page 21) *property*
Specifies the icon size.

`mode` (page 21) *property*
Specifies the display mode of the camera device view.

`hasDisplayModeIcon` (page 20) *property*
Returns whether the device view is being displayed in icon mode.

`hasDisplayModeTable` (page 20) *property*
Returns whether the device view is being displayed in table mode.

Selecting the File Transfer Mode

`transferMode` (page 22) *property*
Determines how the contents are saved by the delegate.

Configuring Download Interface and Downloading Files

`canDownloadSelectedItems` (page 18) *property*
Returns whether the selected items can be downloaded (read-only)

`downloadsDirectory` (page 20) *property*
Specifies the directory where files are downloaded

- `downloadSelectedItems:` (page 23)
Deletes the selected items from the camera.

- `downloadAllItems:` (page 22)
Downloads all the items.

`downloadSelectedControlLabel` (page 20) *property*
Allows the "Download Selected" control to be renamed.

`downloadAllControlLabel` (page 19) *property*
Allows the "Download All" control to be renamed.

`displaysDownloadsDirectoryControl` (page 19) *property*
Specifies whether the downloads directory control should be displayed.

Getting and Setting the Post Processing Application

`displaysPostProcessApplicationControl` (page 19) *property*
Displays whether the post process application control should be displayed.

`postProcessApplication` (page 21) *property*
The URL of the application used to post process the image.

Deleting Selected Items

`canDeleteSelectedItems` (page 17) *property*
Returns whether the selected items can be deleted. (read-only)

- `deleteSelectedItems:` (page 22)
Deletes the currently selected items.

Selection Management

- `selectIndexes:byExtendingSelection:` (page 24)
Invoked to select the specified files, extending the selection if specified.

- `selectedIndexes` (page 24)
The selected indexes of the camera files.

Getting and Setting the Delegate

`delegate` (page 18) *property*
The camera device view delegate.

Selected Item Rotation

`canRotateSelectedItemsLeft` (page 18) *property*
Returns whether the selected items can be rotated left. (read-only)

`canRotateSelectedItemsRight` (page 18) *property*
Returns whether the selected items can be rotated right. (read-only)

`- rotateLeft:` (page 23)
Rotates the selected image to the left.

`- rotateRight:` (page 24)
Rotates the selected image to the right.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

cameraDevice

The current camera device.

```
@property(assign) ICCameraDevice *cameraDevice
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

canDeleteSelectedItems

Returns whether the selected items can be deleted. (read-only)

```
@property(readonly) BOOL canDeleteSelectedItems
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

canDownloadSelectedItems

Returns whether the selected items can be downloaded (read-only)

```
@property(readonly) BOOL canDownloadSelectedItems
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

canRotateSelectedItemsLeft

Returns whether the selected items can be rotated left. (read-only)

```
@property(readonly) BOOL canRotateSelectedItemsLeft
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property canRotateSelectedItemsRight` (page 18)

Declared In
`IKCameraDeviceView.h`

canRotateSelectedItemsRight

Returns whether the selected items can be rotated right. (read-only)

```
@property(readonly) BOOL canRotateSelectedItemsRight
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property canRotateSelectedItemsLeft` (page 18)

Declared In
`IKCameraDeviceView.h`

delegate

The camera device view delegate.

```
@property(assign) id<IKCameraDeviceViewDelegate> delegate
```

Discussion
The delegate must conform to the `IKCameraDeviceViewDelegate` protocol.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKCameraDeviceView.h

displaysDownloadsDirectoryControl

Specifies whether the downloads directory control should be displayed.

```
@property BOOL displaysDownloadsDirectoryControl
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property downloadsDirectory` (page 20)

Declared In
IKCameraDeviceView.h

displaysPostProcessApplicationControl

Displays whether the post process application control should be displayed.

```
@property BOOL displaysPostProcessApplicationControl
```

Discussion
The post process application is only relevant when the `transferMode` (page 22) is set to `IKCameraDeviceViewTransferModeFileBased` (page 25).

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property postProcessApplication` (page 21)

Declared In
IKCameraDeviceView.h

downloadAllControlLabel

Allows the “Download All” control to be renamed.

```
@property(copy) NSString *downloadAllControlLabel
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKCameraDeviceView.h

downloadsDirectory

Specifies the directory where files are downloaded

```
@property(retain) NSURL *downloadsDirectory
```

Availability
Available in Mac OS X v10.6 and later.

See Also
- `downloadSelectedItems:` (page 23)
- `downloadAllItems:` (page 22)

Declared In
`IKCameraDeviceView.h`

downloadSelectedControlLabel

Allows the “Download Selected” control to be renamed.

```
@property(copy) NSString *downloadSelectedControlLabel
```

Availability
Available in Mac OS X v10.6 and later.

See Also
- `downloadSelectedItems:` (page 23)

Declared In
`IKCameraDeviceView.h`

hasDisplayModeIcon

Returns whether the device view is being displayed in icon mode.

```
@property BOOL hasDisplayModeIcon
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property hasDisplayModeTable` (page 20)

Declared In
`IKCameraDeviceView.h`

hasDisplayModeTable

Returns whether the device view is being displayed in table mode.

```
@property BOOL hasDisplayModeTable
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property hasDisplayModeIcon` (page 20)

Declared In
`IKCameraDeviceView.h`

iconSize

Specifies the icon size.

```
@property NSUInteger iconSize
```

Discussion
The icon is square, so only a single dimension is required.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

mode

Specifies the display mode of the camera device view.

```
@property IKCameraDeviceViewDisplayMode mode
```

Discussion
The possible values are listed in "Camera View Display Modes" (page 25)

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

postProcessApplication

The URL of the application used to post process the image.

```
@property(retain) NSURL *postProcessApplication
```

Discussion
The post process application is only relevant when the `transferMode` (page 22) is set to `IKCameraDeviceViewTransferModeFileBased` (page 25).

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property displaysPostProcessApplicationControl` (page 19)

Related Sample Code
ImageKitDemo

Declared In
`IKCameraDeviceView.h`

transferMode

Determines how the contents are saved by the delegate.

```
@property IKCameraDeviceViewTransferMode transferMode
```

Discussion
The possible values are described in "Camera File Transfer Modes" (page 25).

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

Instance Methods

deleteSelectedItems:

Deletes the currently selected items.

```
- (void)deleteSelectedItems:(id)sender
```

Parameters

sender
The object that sent the message.

Discussion
This method is can be connected to a user interface item in Interface Builder.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKCameraDeviceView.h`

downloadAllItems:

Downloads all the items.

```
- (void)downloadAllItems:(id)sender
```

Parameters

sender

The object that sent the message.

Discussion

This method is can be connected to a user interface item in Interface Builder.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

downloadSelectedItems:

Deletes the selected items from the camera.

```
- (void)downloadSelectedItems:(id)sender
```

Parameters

sender

The object that sent the message.

Discussion

This method is can be connected to a user interface item in Interface Builder.

Availability

Available in Mac OS X v10.6 and later.

See Also

- downloadAllItems: (page 22)
- @property downloadsDirectory (page 20)

Declared In

IKCameraDeviceView.h

rotateLeft:

Rotates the selected image to the left.

```
- (void)rotateLeft:(id)sender
```

Parameters

sender

The object that sent the message.

Discussion

This method is can be connected to a user interface item in Interface Builder.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

rotateRight:

Rotates the selected image to the right.

```
- (void)rotateRight:(id)sender
```

Parameters

sender

The object that sent the message.

Discussion

This method is can be connected to a user interface item in Interface Builder.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

selectedIndexes

The selected indexes of the camera files.

```
- (NSIndexSet *)selectedIndexes
```

Return Value

The indexes of the selected files.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

selectIndexes:byExtendingSelection:

Invoked to select the specified files, extending the selection if specified.

```
- (void)selectIndexes:(NSIndexSet *)indexes byExtendingSelection:(BOOL)extend
```

Parameters

indexes

The indexes of the files to select.

extend

YES if the selection should be extended, otherwise NO.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

Constants

Camera View Display Modes

These constants specify the display mode used by the camera view. These constants are used by `mode` (page 21).

```
enum
{
    IKCameraDeviceViewDisplayModeTable = 0,
    IKCameraDeviceViewDisplayModeIcon
};
typedef NSInteger IKCameraDeviceViewDisplayMode;
```

Constants

`IKCameraDeviceViewDisplayModeTable`

Display the devices in as a table.

Available in Mac OS X v10.6 and later.

Declared in `IKCameraDeviceView.h`.

`IKCameraDeviceViewDisplayModeIcon`

Display the devices as icons.

Available in Mac OS X v10.6 and later.

Declared in `IKCameraDeviceView.h`.

Camera File Transfer Modes

These constants specify the transfer mode used by the camera view. These constants are used by `mode` (page 21).

```
enum
{
    IKCameraDeviceViewTransferModeFileBased = 0,
    IKCameraDeviceViewTransferModeMemoryBased
};
typedef NSInteger IKCameraDeviceViewTransferMode;
```

Constants

`IKCameraDeviceViewTransferModeFileBased`

Transferred files will be saved to disk by the delegate.

Available in Mac OS X v10.6 and later.

Declared in `IKCameraDeviceView.h`.

`IKCameraDeviceViewTransferModeMemoryBased`

Transferred files will be supplied to the delegate as an `NSData` object.

Available in Mac OS X v10.6 and later.

Declared in `IKCameraDeviceView.h`.

IKDeviceBrowserView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKDeviceBrowserView.h
Related sample code	ImageKitDemo

Overview

The `IKDeviceBrowserView` allows you to select a camera or scanner from a list of the available devices.

Delegation

The `IKDeviceBrowserView` delegate must conform to the `IKDeviceBrowserViewDelegate` protocol. The delegate provides methods to inform you of selection changes in the browser as well as errors encountered when creating the browser list.

Tasks

Getting the Selected Device

`selectedDevice` (page 30) *property*
Returns the selected device. (read-only)

Specifying the Device Types to Display

`displaysLocalCameras` (page 28) *property*
Specifies whether local cameras are displayed by the browser.

`displaysNetworkCameras` (page 29) *property*
Specifies whether network cameras are displayed by the browser.

`displaysLocalScanners` (page 29) *property*
Specifies whether local scanners are displayed by the browser.

`displaysNetworkScanners` (page 29) *property*
Specifies whether network scanners are displayed by the browser.

Specifying the Display Mode

`mode` (page 30) *property*
Specifies the browser display mode.

Getting and Setting the Delegate

`delegate` (page 28) *property*
Specifies the delegate object.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

delegate

Specifies the delegate object.

```
@property(assign) id<IKDeviceBrowserViewDelegate> delegate
```

Discussion
The delegate object must conform to the `IKDeviceBrowserViewDelegate` protocol.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKDeviceBrowserView.h`

displaysLocalCameras

Specifies whether local cameras are displayed by the browser.

```
@property BOOL displaysLocalCameras
```

Availability
Available in Mac OS X v10.6 and later.

See Also
- `@property displaysNetworkCameras` (page 29)
- `@property displaysLocalScanners` (page 29)
- `@property displaysNetworkScanners` (page 29)

Declared In
`IKDeviceBrowserView.h`

displaysLocalScanners

Specifies whether local scanners are displayed by the browser.

```
@property BOOL displaysLocalScanners
```

Availability
Available in Mac OS X v10.6 and later.

See Also
- `@property displaysNetworkScanners` (page 29)
- `@property displaysLocalCameras` (page 28)
- `@property displaysNetworkCameras` (page 29)

Declared In
`IKDeviceBrowserView.h`

displaysNetworkCameras

Specifies whether network cameras are displayed by the browser.

```
@property BOOL displaysNetworkCameras
```

Availability
Available in Mac OS X v10.6 and later.

See Also
- `@property displaysLocalCameras` (page 28)
- `@property displaysLocalScanners` (page 29)
- `@property displaysNetworkScanners` (page 29)

Declared In
`IKDeviceBrowserView.h`

displaysNetworkScanners

Specifies whether network scanners are displayed by the browser.

```
@property BOOL displaysNetworkScanners
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property displaysLocalScanners` (page 29)
`@property displaysLocalCameras` (page 28)
`@property displaysNetworkCameras` (page 29)

Declared In
`IKDeviceBrowserView.h`

mode

Specifies the browser display mode.

```
@property IKDeviceBrowserViewDisplayMode mode
```

Discussion
The supported display modes are defined in "Device Browser Display Modes" (page 30)

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKDeviceBrowserView.h`

selectedDevice

Returns the selected device. (read-only)

```
@property(readonly) ICDevice *selectedDevice
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`deviceBrowserView:selectionDidChange:` (page 144) (IKDeviceBrowserViewDelegate)

Declared In
`IKDeviceBrowserView.h`

Constants

Device Browser Display Modes

These constants specify the display mode of the device browser.

```
enum
{
    IKDeviceBrowserViewDisplayModeTable,
    IKDeviceBrowserViewDisplayModeOutline,
    IKDeviceBrowserViewDisplayModeIcon
};
typedef NSInteger IKDeviceBrowserViewDisplayMode;
```

Constants

`IKDeviceBrowserViewDisplayModeTable`

The devices are displayed in a table.

Available in Mac OS X v10.6 and later.

Declared in `IKDeviceBrowserView.h`.

`IKDeviceBrowserViewDisplayModeOutline`

The devices are displayed in an outline.

Available in Mac OS X v10.6 and later.

Declared in `IKDeviceBrowserView.h`.

`IKDeviceBrowserViewDisplayModeIcon`

The devices are displayed as icons.

Available in Mac OS X v10.6 and later.

Declared in `IKDeviceBrowserView.h`.

IKFilterBrowserPanel Class Reference

Inherits from	NSPanel : NSWindow : NSResponder : NSObject
Conforms to	NSUserInterfaceValidations (NSWindow) NSAnimatablePropertyContainer (NSWindow) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKFilterBrowserPanel.h
Related sample code	CIRAWFilterSample

Overview

The `IKFilterBrowserPanel` class provides a user interface that allows users to browse Core Image filters (`CIFilter`), to preview a filter, and to get additional information about the filter, such as its description.

An `IKFilterBrowserPanel` object can be displayed as:

- a separate panel, that is, a utility window that floats on top of document windows
- a modal dialog
- a sheet, that is, a dialog that is attached to its parent window and must be dismissed by the user
- a view that an application can insert into a custom user interface

An `IKFilterBrowserPanel` object can be configured through a style mask to use either the default or brushed metal look for windows. The size and number of visible controls are specified through an options dictionary. An `IKFilterBrowserPanel` object communicates selection changes through notifications.

The `IKFilterBrowserPanel` class allows the user to create filter collections that are stored with the `filterCollections` key in the `com.apple.CoreImageKit.plist` property list located in `~/Library/Preferences/`.

Tasks

Getting a Filter Name

`- filterName` (page 37)
Returns the name of the filter that is currently selected in the filter browser.

Displaying and Running the Panel

`- filterBrowserViewWithOptions:` (page 37)
Returns a view that contains a filter browser.

`- beginWithOptions:modelessDelegate:didEndSelector:contextInfo:` (page 36)
Displays the filter browser in a new utility window, unless the filter browser is already open.

`- beginSheetWithOptions:modalForWindow:modalDelegate:didEndSelector:contextInfo:` (page 35)
Displays the filter browser in a sheet—that is, a dialog that is attached to its parent window and must be dismissed by the user.

`- runModalWithOptions:` (page 38)
Displays the filter browser in a modal dialog that must be dismissed by the user but that is not attached to a window.

`- finish:` (page 37)
Closes a filter browser view.

Creating a Filter Browser Panel

`+ filterBrowserPanelWithStyleMask:` (page 34)
Creates a shared instance of the `IKFilterBrowserPanel` class.

Class Methods

filterBrowserPanelWithStyleMask:

Creates a shared instance of the `IKFilterBrowserPanel` class.

```
+ (id)filterBrowserPanelWithStyleMask:(unsigned int)styleMask;
```

Parameters

styleMask
A mask that specifies whether to use the default or brushed metal look for the window. You can select or deselect the `NSTexturedBackgroundWindowMask` style bit.

Return Value
The shared instance.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
CIRAWFilterSample

Declared In
`IKFilterBrowserPanel.h`

Instance Methods

beginSheetWithOptions:modalForWindow:modalDelegate:didEndSelector:contextInfo:

Displays the filter browser in a sheet—that is, a dialog that is attached to its parent window and must be dismissed by the user.

```
- (void)beginSheetWithOptions:(NSDictionary*)inOptions modalForWindow:(NSWindow
    *)docWindow modalDelegate:(id)modalDelegate didEndSelector:(SEL)didEndSelector
     contextInfo:(void *)contextInfo;
```

Parameters

inOptions
: A dictionary of options that describe the configuration to use for the filter browser user interface. For the possible keys you can supply see "Filter Browser Option Keys" (page 38) and the constant `IKUISizeFlavor`.

modalForWindow
: The parent window for the dialog.

modalDelegate
: The object that will invoke the selector `didEndSelector` when the filter browser session terminates.

didEndSelector
: The selector to invoke when the filter browser session terminates.

contextInfo
: Any data that must be passed as an argument to the delegate through `didEndSelector` after the filter browser session terminates.

Discussion
When the filter browser session ends, `didEndSelector` is invoked on the modeless delegate, passing `contextInfo` as an argument. The selector `didEndSelector` must have the following signature:

```
- (void)openPanelDidEnd:(NSOpenPanel *)panel returnCode:(int)returnCode
contextInfo:(void *)contextInfo
```

The `returnCode` value passed to the selector is set to `NSOKButton` if the user validates, or to `NSCancelButton` if the user cancels.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- beginWithOptions:modelessDelegate:didEndSelector:contextInfo:` (page 36)

- runModalWithOptions (page 38)

Declared In
IKFilterBrowserPanel.h

beginWithOptions:modelessDelegate:didEndSelector:contextInfo:

Displays the filter browser in a new utility window, unless the filter browser is already open.

```
- (void)beginWithOptions:(NSDictionary*)inOptions
    modelessDelegate:(id)modelessDelegate didEndSelector:(SEL)didEndSelector
    contextInfo:(void *)contextInfo;
```

Parameters

inOptions

A dictionary of options that describe the configuration to use for the filter browser user interface. For the possible keys you can supply see "Filter Browser Option Keys" (page 38) and the constant IKUISizeFlavor.

modelessDelegate

The object that will invoke the selector didEndSelector when the filter browser session terminates.

didEndSelector

The selector to invoke when the filter browser session terminates.

contextInfo

Any data that must be passed as an argument to the delegate through didEndSelector after the filter browser session terminates.

Discussion
When the filter browser session ends, didEndSelector is invoked on the modeless delegate, passing contextInfo as an argument. The selector didEndSelector must have the following signature:

```
- (void)openPanelDidEnd:(NSOpenPanel *)panel returnCode:(int)returnCode
contextInfo:(void *)contextInfo
```

The returnCode value passed to the selector is set to NSOKButton if the user validates, or to NSCancelButton if the user cancels.

Availability
Available in Mac OS X v10.5 and later.

See Also
- beginSheetWithOptions:modalForWindow:modalDelegate:didEndSelector:contextInfo: (page 35)
- runModalWithOptions (page 38)

Related Sample Code
CIRAWFilterSample

Declared In
IKFilterBrowserPanel.h

filterBrowserViewWithOptions:

Returns a view that contains a filter browser.

```
- (IKFilterBrowserView*)filterBrowserViewWithOptions:(NSDictionary*)inOptions;
```

Parameters

inOptions

A dictionary of options that describe the configuration to use for the filter browser user interface. For the possible keys you can supply see "Filter Browser Option Keys" (page 38) and the constant IKUISizeFlavor.

Return Value
A filter browser view that is configured as specified.

Discussion
Use this method to add a view that contains the filter browser to your custom user interface. To dismiss the filter browser view, invoke the finish (page 37) method.

Availability
Available in Mac OS X v10.5 and later.

Declared In
IKFilterBrowserPanel.h

filterName

Returns the name of the filter that is currently selected in the filter browser.

```
- (NSString*)filterName;
```

Return Value
The name of the currently selected filter.

Discussion
Use this method in response to the notifications IKFilterBrowserFilterSelectedNotification (page 40) or IKFilterBrowserFilterDoubleClickNotification (page 40), or after the user makes a choice in a dialog.

Availability
Available in Mac OS X v10.5 and later.

Declared In
IKFilterBrowserPanel.h

finish:

Closes a filter browser view.

```
- (void)finish:(id)sender;
```

Parameters

sender

The object that invokes the action, such as an OK or Cancel button.

Discussion
Invoke this action when you want to dismiss the filter browser.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- filterBrowserViewWithOptions` (page 37)

Declared In
`IKFilterBrowserPanel.h`

runModalWithOptions:

Displays the filter browser in a modal dialog that must be dismissed by the user but that is not attached to a window.

```
- (int)runModalWithOptions:(NSDictionary*)inOptions;
```

Parameters

inOptions

A dictionary of options that describe the configuration to use for the filter browser user interface. For the possible keys you can supply see "`Filter Browser Option Keys`" (page 38) and the constant `IKUISizeFlavor`.

Return Value
Either `NSOKButton` if the user validates, or `NSCancelButton` if the user cancels.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- beginSheetWithOptions:modalForWindow:modalDelegate:didEndSelector:contextInfo:` (page 35)
`- beginWithOptions:modelessDelegate:didEndSelector:contextInfo:` (page 36)

Declared In
`IKFilterBrowserPanel.h`

Constants

Filter Browser Option Keys

Keys for filter browser options.

```
NSString *const IKFilterBrowserDefaultInputImage;
NSString *const IKFilterBrowserExcludeCategories;
NSString *const IKFilterBrowserExcludeFilters;
NSString *const IKFilterBrowserShowCategories;
NSString *const IKFilterBrowserShowPreview;
```

Constants

`IKFilterBrowserDefaultInputImage`

The key for the default input image. The associated value is the `CIImage` object to use as the default input image for the filter preview. Setting the image to `nil` causes Image Kit to use the image supplied by the framework. You can also set the input image and other parameters during the notification IKFilterBrowserWillPreviewFilterNotification (page 39).

Available in Mac OS X v10.5 and later.

Declared in `IKFilterBrowserPanel.h`.

`IKFilterBrowserExcludeCategories`

The key for excluding filter categories. The associated value is an `NSArray` object that lists the categories that you do *not* want to display in the filter browser.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterBrowserPanel.h`.

`IKFilterBrowserExcludeFilters`

The key for excluding filters. The associated value is an `NSArray` object that lists the filters that you do *not* want to display in the filter browser.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterBrowserPanel.h`.

`IKFilterBrowserShowCategories`

The key for showing categories. The associated value is a `BOOL` value that determines if the filter browser should show the category list.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterBrowserPanel.h`.

`IKFilterBrowserShowPreview`

The associated value is a `BOOL` value that determines if the filter browser should provide a preview.

Available in Mac OS X v10.5 and later.

Declared in `IKFilterBrowserPanel.h`.

Declared In

`IKFilterBrowserPanel.h`

Notifications

IKFilterBrowserWillPreviewFilterNotification

Posted before showing a filter preview, allowing an application to set the parameters of a filter.

The selected filter is sent as the object in the notification.

Availability

Available in Mac OS X v10.5 and later.

Declared In
`IKFilterBrowserPanel.h`

IKFilterBrowserFilterSelectedNotification

Posted when the user clicks a filter name in the filter browser.

The name of the selected filter is sent as the object in the notification.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterBrowserPanel.h`

IKFilterBrowserFilterDoubleClickNotification

Posted when the user double-clicks a filter in the filter browser.

The name of the selected filter is send as the object in the notification.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterBrowserPanel.h`

IKFilterBrowserView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKFilterBrowserView.h

Overview

The `IKFilterBrowserView` class is used as a container for the elements of an `IKFilterBrowserPanel` object.

Tasks

Setting the Preview State

`- setPreviewState:` (page 42)
Sets the preview state.

Getting the Filter Name

`- filterName` (page 41)
Returns the name of the filter that is currently selected in the filter browser.

Instance Methods

filterName

Returns the name of the filter that is currently selected in the filter browser.

```
- (NSString*)filterName;
```

Return Value
The name of the currently selected filter.

Discussion
Use this method in response to the notifications `IKFilterBrowserFilterSelectedNotification` (page 40) or `IKFilterBrowserFilterDoubleClickNotification` (page 40), or after the user makes a choice in a dialog.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterBrowserView.h`

setPreviewState:

Sets the preview state.

```
- (void)setPreviewState:(BOOL)inState;
```

Parameters
inState
A state (`YES` or `NO`) that represents whether a preview is visible.

Discussion
Use this method to show and hide the preview programmatically.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterBrowserView.h`

IKFilterUIView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKFilterUIView.h
Related sample code	CIFilterGeneratorTest CIRAWFilterSample

Overview

The `IKFilterUIView` class provides a view that contains input parameter controls for a Core Image filter (`CIFilter`). You need to use this class when providing a user interface for a custom filter. The class creates a view that has an object controller for the given filter. It also retains the filter.

Tasks

Creating and Initializing a Filter UI View

`+ viewWithFrame:filter:` (page 44)
Creates a view that contains controls for the input parameters of a filter.

`- initWithFrame:filter:` (page 44)
Initializes a view that contains controls for the input parameters of a filter.

Getting Data from the Filter View

`- filter` (page 44)
Returns the Core Image filter associated with the view.

`- objectController` (page 45)
Returns the object controller for the bindings between the filter and its view.

Class Methods

viewWithFrame:filter:

Creates a view that contains controls for the input parameters of a filter.

```
+ (id)viewWithFrame:(NSRect)frameRect filter:(CIFilter *)inFilter
```

Parameters

frameRect

The rectangle that defines the area of the view.

inFilter

A Core Image filter. The view retains the filter.

Return Value

An `IKFilterUIView` object that contains controls for the input parameters of the provided filter.

Availability

Available in Mac OS X v10.5 and later.

See Also

`- initWithFrame:filter:` (page 44)

Declared In

`IKFilterUIView.h`

Instance Methods

filter

Returns the Core Image filter associated with the view.

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

Return Value

The Core Image filter associated with the view.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`IKFilterUIView.h`

initWithFrame:filter:

Initializes a view that contains controls for the input parameters of a filter.

```
- (id)initWithFrame:(NSRect)frameRect filter:(CIFilter *)inFilter
```

Parameters

frameRect

The rectangle that defines the area of the view.

inFilter

A Core Image filter. The view retains the filter.

Return Value
The `IKFilterUIView` object initialized with controls for the input parameters of the provided filter.

Availability
Available in Mac OS X v10.5 and later.

See Also
`+ viewWithFrame:filter:` (page 44)

Declared In
`IKFilterUIView.h`

objectController

Returns the object controller for the bindings between the filter and its view.

```
- (NSObjectController *)objectController
```

Return Value
The object controller for the bindings between the filter and its view.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterUIView.h`

IKImageBrowserCell Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKImageBrowserCell.h
Related sample code	ImageBrowserViewAppearance ImageKitDemo

Overview

The `IKImageBrowserCell` class is used to display a cell conforming to the `IKImageBrowserItem Protocol` protocol in an `IKImageBrowserView`.

Tasks

Cell Component Frames

- `frame` (page 49)
  Returns the receiver's frame rectangle, which defines its position in its `IKImageBrowserView`.
- `imageFrame` (page 51)
  Returns the receiver's image frame rectangle, which defines the position of the thumbnail in its `IKImageBrowserView`.
- `subtitleFrame` (page 53)
  Returns the receiver's subtitle frame rectangle.
- `titleFrame` (page 54)
  Returns the receiver's title frame rectangle.
- `imageContainerFrame` (page 50)
  Returns the receiver's image container frame rectangle, which defines the position of the container of the thumbnail.

Represented Item

- `indexOfRepresentedItem` (page 51)
  Returns the index of the receiver's represented object in the datasource.
- `representedItem` (page 53)
  Returns the receiver's represented object.

Selection Handling

- `isSelected` (page 51)
  Returns whether the cell is selected.
- `selectionFrame` (page 53)
  Returns the receiver's selection frame rectangle, which defines the position of the selection rectangle in its `IKImageBrowserView`.

Cell Content Display

- `imageAlignment` (page 49)
  Returns the position of the cell's image in the frame.
- `opacity` (page 52)
  Returns the opacity of the receiver.

Getting The Cell State

- `cellState` (page 49)
  Returns the current cell state of the receiver.

Core Animation Integration

- `layerForType:` (page 52)
  Returns a layer for the specified position.

Getting The Parent Browser View

- `imageBrowserView` (page 50)
  Returns the view the receiver uses to display the cell.

Instance Methods

cellState

Returns the current cell state of the receiver.

```
- (IKImageBrowserCellState)cellState
```

Return Value
The current state of the cell. See "IKImageBrowserCellState" (page 54) for possible values.

Discussion
The `IKImageBrowserView` creates thumbnails asynchronously. This method returns the current state.

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserCell.h`

frame

Returns the receiver's frame rectangle, which defines its position in its `IKImageBrowserView`.

```
- (NSRect)frame
```

Return Value
The coordinates of the frame, in the `IKImageBrowserView` coordinate space.

Discussion
Subclasses should not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

imageAlignment

Returns the position of the cell's image in the frame.

```
- (NSImageAlignment)imageAlignment
```

Return Value
The alignment of the image. See `NSImageAlignment` for possible values.

Discussion
Subclasses can override this method to customize the image alignment.

The image frame will be computed automatically from the image container frame by taking in account the image alignment and the image aspect ratio.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

imageBrowserView

Returns the view the receiver uses to display the cell.

```
- (IKImageBrowserView *)imageBrowserView
```

Return Value
The browser view containing the cell.

Discussion
Subclasses should not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

imageContainerFrame

Returns the receiver's image container frame rectangle, which defines the position of the container of the thumbnail.

```
- (NSRect)imageContainerFrame
```

Return Value
The coordinates of image container frame, in the `IKImageBrowserView` coordinate space.

Discussion
The image frame is computed automatically from the image container frame by taking in account the image alignment and the image aspect ratio.

Subclasses can override this method to customize the position of the thumbnail container.

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserCell.h`

imageFrame

Returns the receiver's image frame rectangle, which defines the position of the thumbnail in its `IKImageBrowserView`.

```
- (NSRect)imageFrame
```

Return Value
The coordinates of the frame, in the `IKImageBrowserView` coordinate space.

Discussion
It is the developer's responsibility to compute the `imageFrame` such that it lies entirely within the cell's `frame` (page 49) rectangle.

Subclasses can override this method to customize the position of the thumbnail.

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserCell.h`

indexOfRepresentedItem

Returns the index of the receiver's represented object in the datasource.

```
- (NSUInteger)indexOfRepresentedItem
```

Return Value
The index of the represented object in the datasource.

Discussion
Subclasses should not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

isSelected

Returns whether the cell is selected.

```
- (BOOL)isSelected
```

Return Value
`YES` if the cell is selected, otherwise `NO`.

Discussion
Subclasses should not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

layerForType:

Returns a layer for the specified position.

```
- (CALayer *)layerForType:(NSString *)type
```

Parameters

type
A string representing the layer location. See "Cell Layer Positions" (page 55) for possible values.

Return Value
The `CALayer` to display in the specified position.

Discussion
Subclasses can override this method to add a Core Animation layer to the cell

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

opacity

Returns the opacity of the receiver.

```
- (CGFloat)opacity
```

Return Value
The cell's opacity.

Discussion
Possible values are between 0.0 (transparent) and 1.0 (opaque).

Subclasses can override this method to customize the opacity of the cell.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

representedItem

Returns the receiver's represented object.

```
- (id)representedItem
```

Return Value
The item represented by the cell.

Discussion
Subclasses should not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

selectionFrame

Returns the receiver's selection frame rectangle, which defines the position of the selection rectangle in its `IKImageBrowserView`.

```
- (NSRect)selectionFrame
```

Return Value
The cells selection frame, in the `IKImageBrowserView` coordinate space.

Discussion
Subclasses can override this method to customize the position of the selection frame.

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserCell.h`

subtitleFrame

Returns the receiver's subtitle frame rectangle.

```
- (NSRect)subtitleFrame
```

Return Value
The coordinates of the subtitle frame, in the `IKImageBrowserView` coordinate space.

Discussion
It is the developer's responsibility to compute the `subtitleFrame` such that it lies entirely within the cell's `frame` (page 49) rectangle.

Subclasses can override this method to customize the position of the subtitle.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserCell.h`

titleFrame

Returns the receiver's title frame rectangle.

```
- (NSRect)titleFrame
```

Return Value
The coordinates of the title frame, in the `IKImageBrowserView` coordinate space.

Discussion
It is the developer's responsibility to compute the `titleFrame` such that it lies entirely within the cell's `frame` (page 49) rectangle.

Subclasses can override this method to customize the position of the title.

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserCell.h`

Constants

IKImageBrowserCellState

The possible states for the browser cell. These values are used by the `cellState` (page 49) method.

```
typedef enum{
    IKImageStateNoImage=0,
    IKImageStateInvalid,
    IKImageStateReady,
} IKImageBrowserCellState;
```

Constants
`IKImageStateNoImage`
Returned until a thumbnail has been created from the represented object.
Available in Mac OS X v10.6 and later.
Declared in `IKImageBrowserCell.h`.

`IKImageStateInvalid`
The thumbnail is invalid. For example, an unsupported image is provided.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

`IKImageStateReady`
The receiver's represented object has been set and the cell is ready to display.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

Cell Layer Positions

Optional positioning of additional layers displayed with the cell. Used by the `layerForType:` (page 52) method.

```
NSString *const IKImageBrowserCellBackgroundLayer;
NSString *const IKImageBrowserCellForegroundLayer;
NSString *const IKImageBrowserCellSelectionLayer;
NSString *const IKImageBrowserCellPlaceHolderLayer;
```

Constants

`IKImageBrowserCellBackgroundLayer`
Layer displayed in the background.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

`IKImageBrowserCellForegroundLayer`
Layer displayed in the foreground.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

`IKImageBrowserCellSelectionLayer`
Layer displayed as the selection.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

`IKImageBrowserCellPlaceHolderLayer`
Layer displayed as a placeholder when an image is not yet available.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserCell.h`.

IKImageBrowserView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKImageBrowserView.h
Related sample code	AutomatorHandsOn DesktopImage ImageBrowser ImageBrowserViewAppearance ImageKitDemo

Overview

The `IKImageBrowserView` class is a view for displaying and browsing a large amount of images and movies efficiently.

You must set a datasource for the view and implement, at a minimum, the `numberOfItemsInImageBrowser:` (page 151) and `imageBrowser:itemAtIndex:` (page 148) described in *IKImageBrowserDataSource Protocol Reference*. The items must conform to the `IKImageBrowserItem Protocol` protocol.

The class's delegate object must conform to `IKImageBrowserDelegate` protocol. It receives notification of changes in selection, as well as mouse events in the cells.

Tasks

Initializing and Setting Up an Image Browser View

`- initWithFrame:` (page 69)
Initializes a newly allocated image browser view with the provided frame rectangle.

Updating the Display of the Content

- `reloadData` (page 72)
  Marks the receiver as needing its data reloaded.

Getting and Setting the Delegate

- `setDelegate:` (page 78)
  Sets the delegate of the receiver.
- `delegate` (page 67)
  Returns the delegate of the receiver.

Getting and Setting the Data Source

- `setDataSource:` (page 78)
  Sets the data source of the receiver.
- `dataSource` (page 66)
  Returns the data source of the receiver.

Setting the Appearance

- `setCellsStyleMask:` (page 77)
  Defines the appearance style of the cells.
- `cellsStyleMask` (page 65)
  Returns the appearance style mask for the cell.
- `setConstrainsToOriginalSize:` (page 77)
  Sets whether the receiver constrains the cell's image to its original size.
- `constrainsToOriginalSize` (page 66)
  Returns whether the receiver constrains the cell's image to its original size.
- `setIntercellSpacing:` (page 80)
  Sets the spacing between cells in the view.
- `intercellSpacing` (page 70)
  Returns the spacing between cells in the view.

Creating a Custom Cell for an Item

- `newCellForRepresentedItem:` (page 71)
  Returns the cell to use for the specified item.

Zooming and Resizing

- `setZoomValue:` (page 81)
  Sets the zoom value.
- `zoomValue` (page 82)
  Returns the current zoom value.
- `setContentResizingMask:` (page 77)
  Determines how the receiver resizes its content when zooming.
- `contentResizingMask` (page 66)
  Returns the receiver's content resizing mask, which determines how its content is resized while zooming.

Scrolling

- `scrollIndexToVisible:` (page 73)
  Scrolls the receiver to the item at the specified index.

Setting and Getting Cell Size

- `setCellSize:` (page 76)
  Sets the cell size.
- `cellSize` (page 64)
  Returns the cell size.

Getting Item Information

- `indexOfItemAtPoint:` (page 69)
  Returns the index of the item at the specified location.
- `itemFrameAtIndex:` (page 70)
  Returns the frame rectangle for the item located at the specified index.
- `visibleItemIndexes` (page 81)
  Returns the indexes of the view's currently visible items.
- `cellForItemAtIndex:` (page 64)
  Returns the browser cell for the item at the specified index.

Reordering and Groups Items

- `selectionIndexes` (page 73)
  Returns the indexes of the selected cells.
- `setSelectionIndexes:byExtendingSelection:` (page 80)
  Selects cells at the specified indexes.
- `setAllowsMultipleSelection:` (page 74)
  Controls whether the user can select more than one cell at a time.

- `allowsMultipleSelection` (page 62)
  Returns whether multiple selections are allowed.
- `setAllowsEmptySelection:` (page 74)
  Controls whether an empty selection is allowed.
- `allowsEmptySelection` (page 62)
  Returns whether an empty selection is allowed.
- `setAllowsReordering:` (page 75)
  Controls whether the user can reorder items.
- `allowsReordering` (page 62)
  Returns whether the user can reorder items.
- `setAnimates:` (page 75)
  Controls whether the receiver animates reordering and changes of the data source.
- `animates` (page 63)
  Returns whether the receiver animates reordering and changes of the data source.
- `expandGroupAtIndex:` (page 68)
  Expands a group at the specified index.
- `collapseGroupAtIndex:` (page 65)
  Collapses a group at the specified index.
- `isGroupExpandedAtIndex:` (page 70)
  Returns whether the group at the provided index is expanded.

Supporting Drag and Drop

- `setDraggingDestinationDelegate:` (page 79)
  Sets the dragging destination delegate of the receiver.
- `draggingDestinationDelegate` (page 67)
  Returns the dragging destination delegate of the receiver.
- `setDropIndex:dropOperation:` (page 79)
  Allows the class to retarget the drop action.
- `indexAtLocationOfDroppedItem` (page 69)
  Returns the index of the cell where the drop operation occurred.
- `setAllowsDroppingOnItems:` (page 74)
  Specifies whether the user can drop on items.
- `allowsDroppingOnItems` (page 61)
  Returns whether the user can drop on items.
- `dropOperation` (page 67)
  Returns the current drop operation.

Core Animation Layer Integration

- `setForegroundLayer:` (page 79)
  The Core Animation layer used as the foreground overlay.

- `foregroundLayer` (page 68)
  Returns the foreground Core Animation layer
- `setBackgroundLayer:` (page 75)
  The Core Animation layer used as the view's background.
- `backgroundLayer` (page 63)
  Returns the foreground Core Animation layer

QuickLook Support

- `setCanControlQuickLookPanel:` (page 76)
  Specifies whether the view can automatically take control of the QuickLook panel.
- `canControlQuickLookPanel` (page 63)
  Returns whether the view can automatically take control of the QuickLook panel.

Getting Columns and Rows Information

- `numberOfColumns` (page 71)
  Returns the current number of columns.
- `numberOfRows` (page 71)
  Returns the current number of rows.
- `rectOfColumn:` (page 72)
  Returns the rectangle containing the specified column.
- `columnIndexesInRect:` (page 65)
  Returns the column indexes in the specified rectangle.
- `rectOfRow:` (page 72)
  Returns the rectangle containing the specified row.
- `rowIndexesInRect:` (page 73)
  Returns the row indexes in the specified rectangle.

Instance Methods

allowsDroppingOnItems

Returns whether the user can drop on items.

```
- (BOOL)allowsDroppingOnItems
```

Return Value
`YES` if the user is able to drop on items, otherwise `NO`.

Discussion
The default is `NO`.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

allowsEmptySelection

Returns whether an empty selection is allowed.

```
- (BOOL)allowsEmptySelection
```

Return Value
`YES` if the receiver allows an empty selection; `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setAllowsEmptySelection:` (page 74)

Declared In
`IKImageBrowserView.h`

allowsMultipleSelection

Returns whether multiple selections are allowed.

```
- (BOOL)allowsMultipleSelection
```

Return Value
`YES` if the receiver allows the user to select more than one cell at a time; `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setAllowsEmptySelection:` (page 74)

Declared In
`IKImageBrowserView.h`

allowsReordering

Returns whether the user can reorder items.

```
- (BOOL)allowsReordering
```

Return Value
`YES` if the user can reorder items; `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setAllowsReordering:` (page 75)

Declared In
`IKImageBrowserView.h`

animates

Returns whether the receiver animates reordering and changes of the data source.

```
- (BOOL)animates
```

Return Value
`YES` if the receiver animates reordering and changes of the data source; `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setAnimates:` (page 75)

Declared In
`IKImageBrowserView.h`

backgroundLayer

Returns the foreground Core Animation layer

```
- (CALayer *)backgroundLayer
```

Return Value
A `CALayer` instance.

Availability
Available in Mac OS X v10.6 and later.

See Also
`- setBackgroundLayer:` (page 75)

Declared In
`IKImageBrowserView.h`

canControlQuickLookPanel

Returns whether the view can automatically take control of the QuickLook panel.

```
- (BOOL)canControlQuickLookPanel
```

Return Value
YES, if the view can display the Quick Look panel, otherwise NO.

Availability
Available in Mac OS X v10.6 and later.

See Also
- setCanControlQuickLookPanel: (page 76)

Declared In
IKImageBrowserView.h

cellForItemAtIndex:

Returns the browser cell for the item at the specified index.

```
- (IKImageBrowserCell *)cellForItemAtIndex:(NSUInteger)index
```

Parameters
index
The index.

Return Value
The browser cell at the specified index.

Discussion
Subclasses must not override this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

cellSize

Returns the cell size.

```
- (NSSize)cellSize
```

Return Value
The current size for the cells in the image browser view.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setCellSize: (page 76)

Declared In
IKImageBrowserView.h

cellsStyleMask

Returns the appearance style mask for the cell.

```
- (NSUInteger)cellsStyleMask
```

Return Value
The appearance style mask for the cell.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `setCellsStyleMask:` (page 77)

Declared In
`IKImageBrowserView.h`

collapseGroupAtIndex:

Collapses a group at the specified index.

```
- (void)collapseGroupAtIndex:(NSUInteger)index
```

Parameters
index
The index of the group you want to collapse.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `expandGroupAtIndex:` (page 68)
- `isGroupExpandedAtIndex:` (page 70)

Declared In
`IKImageBrowserView.h`

columnIndexesInRect:

Returns the column indexes in the specified rectangle.

```
- (NSIndexSet *)columnIndexesInRect:(NSRect)rect
```

Parameters
rect
The rectangle in the view's coordinate system.

Return Value
An index set containing the cell indexes.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

constrainsToOriginalSize

Returns whether the receiver constrains the cell's image to its original size.

```
- (BOOL)constrainsToOriginalSize
```

Return Value
NO if the image is not constrained; otherwise YES.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setConstrainsToOriginalSize: (page 77)

Declared In
IKImageBrowserView.h

contentResizingMask

Returns the receiver's content resizing mask, which determines how its content is resized while zooming.

```
- (NSUInteger)contentResizingMask
```

Return Value
The content resizing mask.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setContentResizingMask: (page 77)

Declared In
IKImageBrowserView.h

dataSource

Returns the data source of the receiver.

```
- (id)dataSource
```

Return Value
The data source (IKImageBrowserDataSource). The data source is not retained by the receiver.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setDataSource: (page 78)

Declared In
IKImageBrowserView.h

delegate

Returns the delegate of the receiver.

```
- (id)delegate
```

Return Value
The delegate.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setDelegate: (page 78)

Declared In
IKImageBrowserView.h

draggingDestinationDelegate

Returns the dragging destination delegate of the receiver.

```
- (id)draggingDestinationDelegate
```

Return Value
The receiver's dragging destination delegate.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setDraggingDestinationDelegate: (page 79)

Declared In
IKImageBrowserView.h

dropOperation

Returns the current drop operation.

```
- (IKImageBrowserDropOperation)dropOperation
```

Return Value
IKImageBrowserDropOn (page 86) if the drop occurs on an item, otherwise IKImageBrowserDropBefore (page 86).

Discussion
The returned value is valid when a drop occurred and until next drop.

For example, given a browser with `N` cells , a cell of `N-1` and operation of `IKImageBrowserDropOn` (page 86) would specify a drop on the last cell. To specify a drop after the last cell, one would use an index of `N` and `IKImageBrowserDropBefore` (page 86) for the operation.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

expandGroupAtIndex:

Expands a group at the specified index.

```
- (void)expandGroupAtIndex:(NSUInteger)index
```

Parameters
index
The index of the group you want to expand.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `collapseGroupAtIndex:` (page 65)
- `isGroupExpandedAtIndex:` (page 70)

Declared In
`IKImageBrowserView.h`

foregroundLayer

Returns the foreground Core Animation layer

```
- (CALayer *)foregroundLayer
```

Return Value
A `CALayer` instance.

Availability
Available in Mac OS X v10.6 and later.

See Also
- `setForegroundLayer:` (page 79)

Declared In
`IKImageBrowserView.h`

indexAtLocationOfDroppedItem

Returns the index of the cell where the drop operation occurred.

```
- (NSUInteger)indexAtLocationOfDroppedItem
```

Return Value
The index of the cell where the drop operation occurred.

Discussion
The returned index is valid until the next drop occurs.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

indexOfItemAtPoint:

Returns the index of the item at the specified location.

```
- (NSInteger)indexOfItemAtPoint:(NSPoint)point
```

Parameters
point
The location of the item.

Return Value
The index of the item or `NSNotFound` if no item at this location.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

initWithFrame:

Initializes a newly allocated image browser view with the provided frame rectangle.

```
- (id)initWithFrame:(NSRect)frame
```

Parameters
frame
The rectangle for the image browser.

Return Value
The initialized object.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

intercellSpacing

Returns the spacing between cells in the view.

```
- (NSSize)intercellSpacing
```

Return Value
The vertical and horizontal spacing between cells.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

isGroupExpandedAtIndex:

Returns whether the group at the provided index is expanded.

```
- (BOOL)isGroupExpandedAtIndex:(NSUInteger)index
```

Parameters
index
The index you want to check.

Return Value
`YES` if the group is expanded; `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `expandGroupAtIndex:` (page 68)
- `collapseGroupAtIndex:` (page 65)

Declared In
`IKImageBrowserView.h`

itemFrameAtIndex:

Returns the frame rectangle for the item located at the specified index.

```
- (NSRect)itemFrameAtIndex:(NSInteger)index
```

Parameters
index
The index of the item whose frame rectangle you want to obtain.

Return Value
The frame rectangle of the item.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

newCellForRepresentedItem:

Returns the cell to use for the specified item.

```
- (IKImageBrowserCell *)newCellForRepresentedItem:(id)anItem
```

Parameters

anItem
The item that the returned cell will represent.

Return Value
A new cell. The cell should not be return as autoreleased.

Discussion
Subclasses can override this method to customize the appearance of the cell that will represent *anItem*.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

numberOfColumns

Returns the current number of columns.

```
- (NSUInteger)numberOfColumns
```

Return Value
The number of columns.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

numberOfRows

Returns the current number of rows.

```
- (NSUInteger)numberOfRows
```

Return Value
The number of rows.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

rectOfColumn:

Returns the rectangle containing the specified column.

```
- (NSRect)rectOfColumn:(NSUInteger)columnIndex
```

Parameters

columnIndex
The column index.

Return Value
A rectangle containing the column. Specified in the view's coordinate system.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

rectOfRow:

Returns the rectangle containing the specified row.

```
- (NSRect)rectOfRow:(NSUInteger)rowIndex
```

Parameters

rowIndex
The row index.

Return Value
A rectangle containing the column. Specified in the view's coordinate system.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

reloadData

Marks the receiver as needing its data reloaded.

```
- (void)reloadData
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
IKImageBrowserView.h

rowIndexesInRect:

Returns the row indexes in the specified rectangle.

```
- (NSIndexSet *)rowIndexesInRect:(NSRect)rect
```

Parameters

rect

A rectangle in the view's coordinate system.

Return Value
An index set containing the item indexes.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

scrollIndexToVisible:

Scrolls the receiver to the item at the specified index.

```
- (void)scrollIndexToVisible:(NSInteger)index
```

Parameters

index

The index of the item to scroll to.

Availability
Available in Mac OS X v10.5 and later.

Declared In
IKImageBrowserView.h

selectionIndexes

Returns the indexes of the selected cells.

```
- (NSIndexSet *)selectionIndexes
```

Return Value
The indexes of the selected cells.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setSelectionIndexes:byExtendingSelection: (page 80)

Declared In
IKImageBrowserView.h

setAllowsDroppingOnItems:

Specifies whether the user can drop on items.

```
- (void)setAllowsDroppingOnItems:(BOOL)flag
```

Parameters

flag

YES if the user is able to drop on items, otherwise NO.

Discussion
The default is NO.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

setAllowsEmptySelection:

Controls whether an empty selection is allowed.

```
- (void)setAllowsEmptySelection:(BOOL)flag
```

Parameters

flag

A BOOL value that specifies whether to allow an empty selection.

Availability
Available in Mac OS X v10.5 and later.

See Also
- allowsEmptySelection (page 62)

Declared In
IKImageBrowserView.h

setAllowsMultipleSelection:

Controls whether the user can select more than one cell at a time.

```
- (void)setAllowsMultipleSelection:(BOOL)flag
```

Parameters

flag

A BOOL value that specifies whether to allow multiple selections.

Availability
Available in Mac OS X v10.5 and later.

See Also
- allowsMultipleSelection (page 62)

Declared In
IKImageBrowserView.h

setAllowsReordering:

Controls whether the user can reorder items.

```
- (void)setAllowsReordering:(BOOL)flag
```

Parameters

flag

A BOOL value that specifies whether the user can reorder items.

Availability
Available in Mac OS X v10.5 and later.

See Also
- allowsReordering (page 62)

Declared In
IKImageBrowserView.h

setAnimates:

Controls whether the receiver animates reordering and changes of the data source.

```
- (void)setAnimates:(BOOL)flag
```

Parameters

flag

A BOOL value that specifies whether the receiver animates reordering and changes of the data source.

Availability
Available in Mac OS X v10.5 and later.

See Also
- animates (page 63)

Declared In
IKImageBrowserView.h

setBackgroundLayer:

The Core Animation layer used as the view's background.

```
- (void)setBackgroundLayer:(CALayer *)aLayer
```

Parameters

aLayer

A CALayer instance.

Discussion
The background layer can have sublayers. Additionally, the layers can also contain animations.

The layer is optional.

Availability
Available in Mac OS X v10.6 and later.

See Also
- `backgroundLayer` (page 63)

Declared In
`IKImageBrowserView.h`

setCanControlQuickLookPanel:

Specifies whether the view can automatically take control of the QuickLook panel.

```
- (void)setCanControlQuickLookPanel:(BOOL)flag
```

Parameters

flag

`YES`, if the view can display the QuickLook panel, otherwise `NO`.

Discussion
When the browser view displays the QuickLook panel it sets itself as the QuickLook datasource. If the browser cells returned by the datasource return items that are URLs or paths, then the QuickLook panel will display the image at that location. Otherwise, the browser cell must implement the `QLPreviewItem` protocol and return the requested URL for the custom cell.

Availability
Available in Mac OS X v10.6 and later.

See Also
- `canControlQuickLookPanel` (page 63)

Declared In
`IKImageBrowserView.h`

setCellSize:

Sets the cell size.

```
- (void)setCellSize:(NSSize)size
```

Parameters

size

The size to set.

Discussion
You must use `setCellSize` or `setZoomValue:` (page 81), but not both. Setting the zoom value changes the cell size, and vice versa.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `cellSize` (page 64)
- `setZoomValue:` (page 81)

Declared In
`IKImageBrowserView.h`

setCellsStyleMask:

Defines the appearance style of the cells.

```
- (void)setCellsStyleMask:(NSUInteger)mask
```

Parameters

mask

An integer bit mask. A mask can be specified by combining any of the options described in "`Cell Appearance Style Masks`" (page 82) using the C bitwise `OR` operator.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `cellsStyleMask` (page 65)

Declared In
`IKImageBrowserView.h`

setConstrainsToOriginalSize:

Sets whether the receiver constrains the cell's image to its original size.

```
- (void)setConstrainsToOriginalSize:(BOOL)flag
```

Parameters

flag

A flag that specifies whether to constrain the image. The default value is `NO`.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `constrainsToOriginalSize` (page 66)

Declared In
`IKImageBrowserView.h`

setContentResizingMask:

Determines how the receiver resizes its content when zooming.

```
- (void)setContentResizingMask:(NSUInteger)mask
```

Parameters

mask

A resizing mask. You specify a mask by combining any of the following options using the C bitwise OR operator: `NSViewWidthSizable`, `NSViewHeightSizable`. Other values are ignored.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `contentResizingMask` (page 66)

Declared In
`IKImageBrowserView.h`

setDataSource:

Sets the data source of the receiver.

```
- (void)setDataSource:(id)source
```

Parameters

source

A data source (`IKImageBrowserDataSource`).

Availability
Available in Mac OS X v10.5 and later.

See Also
- `dataSource` (page 66)

Declared In
`IKImageBrowserView.h`

setDelegate:

Sets the delegate of the receiver.

```
- (void)setDelegate:(id)aDelegate
```

Parameters

aDelegate

The delegate must implement the `IKImageBrowserDelegate` informal protocol.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `delegate` (page 67)

Declared In
`IKImageBrowserView.h`

setDraggingDestinationDelegate:

Sets the dragging destination delegate of the receiver.

```
- (void)setDraggingDestinationDelegate:(id)delegate
```

Parameters

delegate

The delegate (`NSDraggingDestination`) to set.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- draggingDestinationDelegate` (page 67)

Declared In
`IKImageBrowserView.h`

setDropIndex:dropOperation:

Allows the class to retarget the drop action.

```
- (void)setDropIndex:(NSInteger)index
    dropOperation:(IKImageBrowserDropOperation)operation
```

Parameters

index

The requested drop index.

operation

The requested drop operation. The possible values are described in `IKImageBrowserDropOperation` (page 85).

Discussion
For example, To specify a drop on the second item, one would specify index as `1`, and operation as `IKImageBrowserDropOn` (page 86). To specify a drop after the last item, one would specify index as the number of items and operation as `IKImageBrowserDropBefore` (page 86).

Passing a value of `-1` for *index*, and `IKImageBrowserDropOn` (page 86) as the operation causes the entire browser view to be highlighted rather than a specific item. This is useful if the data displayed by the receiver does not allow the user to drop items at a specific item location

.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageBrowserView.h`

setForegroundLayer:

The Core Animation layer used as the foreground overlay.

```
- (void)setForegroundLayer:(CALayer *)aLayer
```

Parameters

aLayer

A `CALayer` instance.

Discussion

The foreground overlay layer can have sublayers. Additionally, the layers can also contain animations.

The foreground layer is an overlay that is applied to the view. It can be used to provide information such as loading progress or for pure cosmetic purposes, such as dark gradients on top and bottom of the browser view.

This layer is optional.

Availability

Available in Mac OS X v10.6 and later.

See Also

`- foregroundLayer` (page 68)

Declared In

`IKImageBrowserView.h`

setIntercellSpacing:

Sets the spacing between cells in the view.

```
- (void)setIntercellSpacing:(NSSize)aSize
```

Parameters

aSize

The vertical and horizontal spacing between cells.

Discussion

By default, both values are `10.0` in the receiver's coordinate system.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`IKImageBrowserView.h`

setSelectionIndexes:byExtendingSelection:

Selects cells at the specified indexes.

```
- (void)setSelectionIndexes:(NSIndexSet *)indexes
    byExtendingSelection:(BOOL)extendSelection
```

Parameters

indexes

The indexes of the cells you want to select.

extendSelection

A BOOL value that specifies whether to extend the current selection. Pass YES to extends the selection; NO replaces the current selection.

Availability
Available in Mac OS X v10.5 and later.

See Also
- selectionIndexes (page 73)

Related Sample Code
ImageKitDemo

Declared In
IKImageBrowserView.h

setZoomValue:

Sets the zoom value.

```
- (void)setZoomValue:(float)aValue
```

Parameters

aValue

The zoom value. This value should be greater or equal to zero and less or equal than one. A zoom value of zero corresponds to the minimum size (40x40 pixels). A zoom value of one means images fits the browser bounds. Other values are interpolated.

Discussion
You must use setZoomValue or setCellSize: (page 76), but not both. Setting the zoom value changes the cell size, and vice versa.

Availability
Available in Mac OS X v10.5 and later.

See Also
- zoomValue (page 82)
- setCellSize: (page 76)

Declared In
IKImageBrowserView.h

visibleItemIndexes

Returns the indexes of the view's currently visible items.

```
- (NSIndexSet *)visibleItemIndexes
```

Return Value
A set containing the indexes.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageBrowserView.h

zoomValue

Returns the current zoom value.

```
- (float)zoomValue
```

Return Value
The zoom value.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setZoomValue: (page 81)

Declared In
IKImageBrowserView.h

Constants

Cell Appearance Style Masks

Masks for the appearance style bit field.

```
enum{
    IKCellsStyleNone              =0,
    IKCellsStyleShadowed          =1,
    IKCellsStyleOutlined          =2,
    IKCellsStyleTitled            =4,
    IKCellsStyleSubtitled         =8
};
```

Constants

IKCellsStyleNone
No style.
Available in Mac OS X v10.5 and later.
Declared in IKImageBrowserView.h.

IKCellsStyleShadowed
Cells use shadows.
Available in Mac OS X v10.5 and later.
Declared in IKImageBrowserView.h.

IKCellsStyleOutlined
Cells are outlined.
Available in Mac OS X v10.5 and later.
Declared in IKImageBrowserView.h.

`IKCellsStyleTitled`
Cells display a title.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKCellsStyleSubtitled`
Cells display a subtitle.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

Declared In
`IKImageBrowserView.h`

Group Style Attributes

Attributes for the group style. Used by the

```
enum{
    IKGroupBezelStyle,
    IKGroupDisclosureStyle,
};
```

Constants
`IKGroupBezelStyle`
A bezel style.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKGroupDisclosureStyle`
A disclosure triangle.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

Discussion
These constants affect the appearance of a group.

Declared In
`IKImageBrowserView.h`

View Options Keys

Keys for image browser view options. You set and retrieve values for these keys by sending the view `setValue:forKey` and `valueForKey:` messages.

```
NSString * const IKImageBrowserBackgroundColorKey;
NSString * const IKImageBrowserSelectionColorKey;
NSString * const IKImageBrowserCellsOutlineColorKey;
NSString * const IKImageBrowserCellsTitleAttributesKey;
NSString * const IKImageBrowserCellsHighlightedTitleAttributesKey;
NSString * const IKImageBrowserCellsSubtitleAttributesKey;
```

Constants

`IKImageBrowserBackgroundColorKey`

A key for the background color of the image browser view. The associated value is an `NSColor` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserSelectionColorKey`

A key for the color that indicates a selection. The associated value is an `NSColor` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserCellsOutlineColorKey`

A key for the outline color for an item in the image browser view. The associated value is an `NSColor` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserCellsTitleAttributesKey`

A key for title attribute of an item in the image browser view. The associated value is an `NSDictionary` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserCellsHighlightedTitleAttributesKey`

A key for the highlighted title attribute for an item in the image browser view. The associated value is an `NSDictionary` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserCellsSubtitleAttributesKey`

A key for a subtitle attribute for an item in the image browser view. The associated value is an `NSDictionary` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

Discussion

Declared In

`IKImageBrowserView.h`

Group Keys

Keys for group attributes.

```
NSString * const IKImageBrowserGroupRangeKey;
NSString * const IKImageBrowserGroupBackgroundColorKey;
NSString * const IKImageBrowserGroupTitleKey;
NSString * const IKImageBrowserGroupStyleKey;
NSString * const IKImageBrowserGroupHeaderLayer;
NSString * const IKImageBrowserGroupFooterLayer;
```

Constants

`IKImageBrowserGroupRangeKey`

A key for the range of a group. The associated value is an `NSValue` object. This is required if the view uses grouping

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserGroupBackgroundColorKey`

A key for the background color of a group. The associated value is an `NSColor` object. This color is used only for the bezel style.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserGroupTitleKey`

A key for the title of a group. The associated value is an `NSString` object. This string is used for the disclosure style only.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserGroupStyleKey`

A key for the style of a group. The associated value is one of the constants defined in "Group Style Attributes" (page 83).

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserGroupHeaderLayer`

A key for the header layer of the group. The associated value is a `CALayer`.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserGroupFooterLayer`

A key for the header layer of the group. The associated value is a `CALayer`.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserView.h`.

Declared In

`IKImageBrowserView.h`

IKImageBrowserDropOperation

These constants specify the locations for dropping items onto the browser view. Used by the method `setDropIndex:dropOperation:` (page 79).

```
typedef enum
{
    IKImageBrowserDropOn=0,
    IKImageBrowserDropBefore=1,
}IKImageBrowserDropOperation;
```

Constants

`IKImageBrowserDropOn`

Drop the item on the cell.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserDropBefore`

Drop the item before the cell.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserView.h`.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`IKImageBrowserView.h`

IKImageEditPanel Class Reference

Inherits from	NSPanel : NSWindow : NSResponder : NSObject
Conforms to	NSUserInterfaceValidations (NSWindow) NSAnimatablePropertyContainer (NSWindow) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKImageEditPanel.h

Overview

The `IKImageEditPanel` class provides a panel, that is, a utility window that floats on top of document windows, optimized for image editing.

Tasks

Creating an Image Editing Panel

`+ sharedImageEditPanel` (page 88)
Creates a shared instance of an image editing panel.

Getting the User Adjustments and Effects

`filterArray` (page 88) *property*
Returns the current array of user adjustments to effects. (read-only)

Getting, Setting, and Reloading Data

`dataSource` (page 88) *property*
Specifies the edit panel's dataSource.

`- reloadData` (page 89)
Reloads the data from the data associated with an image editing panel.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

dataSource

Specifies the edit panel's dataSource.

```
@property(assign) id<IKImageEditPanelDataSource> dataSource
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageEditPanel.h`

filterArray

Returns the current array of user adjustments to effects. (read-only)

```
@property(readonly) NSArray *filterArray
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKImageEditPanel.h`

Class Methods

sharedImageEditPanel

Creates a shared instance of an image editing panel.

```
+ (IKImageEditPanel *)sharedImageEditPanel
```

Return Value
An `IKImageEditPanel` object.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageEditPanel.h`

Instance Methods

reloadData

Reloads the data from the data associated with an image editing panel.

```
- (void)reloadData
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageEditPanel.h`

IKImageView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKImageView.h
Related sample code	ImageKitDemo

Overview

The `IKImageView` class provides an efficient way to display images in a view while at the same time supporting a number of image editing operations such as rotating, zooming, and cropping. It supports drag and drop, so that the user can drag an image to the view. If possible, image rendering uses hardware acceleration to achieve optimal performance. The `IKImageView` class is implemented as a subclass of `NSView`. Similar to `NSImageView`, the `IKImageView` class is used to display a single image.

You can provide an images for the view in any of these formats:

- File reference (`NSURL`, `CFURLRef`, or a path)
- `CGImageSourceRef`
- Data (`NSData` or `CFDataRef`)
- Image (`CGImageRef` or `CIImage`)

Providing a file reference is the preferred way to set the the image for a view because in addition to the actual image data, `IKImageView` also handles the image metadata embedded in the file. The image view automatically fetches the metadata from a file reference, whereas for the other sources (except for a `CGImageSourceRef` source), it cannot. For images set from other sources, you need to set the metadata separately.

`IKImageView` supports multi-frame images (TIFF, GIF, and so forth) and animated images.

Tasks

Getting and Setting Image View Characteristics

`delegate` (page 95) *property*
Specifies the delegate object of the receiver.

`zoomFactor` (page 97) *property*
Specifies the zoom factor for the image view.

`rotationAngle` (page 96) *property*
Specifies the rotation angle for the image view.

`currentToolMode` (page 95) *property*
Specifies the current tool mode for the image view.

`autoresizes` (page 94) *property*
Specifies the automatic resizing state for the image view.

`hasHorizontalScroller` (page 96) *property*
Specifies the horizontal scroll bar state for the image view.

`hasVerticalScroller` (page 96) *property*
Specifies the vertical scroll bar state for the image view.

`autohidesScrollers` (page 94) *property*
Specifies the automatic-hiding scroll bar state for the image view.

`supportsDragAndDrop` (page 97) *property*
Specifies the drag-and-drop support state for the image view.

`editable` (page 95) *property*
Specifies the editable state for the image view.

`doubleClickOpensImageEditPanel` (page 95) *property*
Specifies the image-opening state of the editing pane in the image view.

`imageCorrection` (page 96) *property*
Specifies a Core Image filter for image correction.

`backgroundColor` (page 94) *property*
Specifies the background color for the image view.

- `imageSize` (page 101)
Returns the size of the image in the image view.

- `imageProperties` (page 100)
Returns the metadata for the image in the view.

Getting and Setting Images

- `image` (page 100)
Returns the image associated with the view, after any image corrections.

- `setImage:imageProperties:` (page 103)
Sets the image to display in an image view.

- `setImageWithURL:` (page 103)
  Initializes an image view with the image specified by a URL.

Manipulating the Image in a View

- `setRotationAngle:centerPoint:` (page 104)
  Sets the rotation angle at the provided origin.
- `setImageZoomFactor:centerPoint:` (page 104)
  Sets the zoom factor at the provided origin.
- `zoomImageToFit:` (page 105)
  Zooms the image so that it fits in the image view.
- `zoomImageToActualSize:` (page 105)
  Zooms the image so that it is displayed using its true size.
- `zoomImageToRect:` (page 106)
  Zooms the image so that it fits in the specified rectangle.
- `zoomIn:` (page 106)
  Zooms the image in.
- `zoomOut:` (page 106)
  Zooms the image out.
- `crop:` (page 99)
  Crops the image using the current selection.
- `flipImageHorizontal:` (page 99)
  Flips an image along the horizontal axis.
- `flipImageVertical:` (page 100)
  Flips an image along the vertical axis.
- `rotateImageLeft:` (page 101)
  Rotates the image left (counter-clockwise).
- `rotateImageRight:` (page 102)
  Rotates the image right (clockwise).

Working With Core Animation

- `setOverlay:forType:` (page 104)
  Sets an overlay type for a Core Animation layer.
- `overlayForType:` (page 101)
  Returns the Core Animation layer associated with a layer type.

Scrolling

- `scrollToPoint:` (page 102)
  Scrolls the view to the specified point.
- `scrollToRect:` (page 102)
  Scrolls the view so that it includes the provided rectangular area.

Converting Points and Rectangles

- `convertViewPointToImagePoint:` (page 98)
  Converts an image view coordinate to an image coordinate.
- `convertViewRectToImageRect:` (page 98)
  Converts an image view rectangle to an image rectangle.
- `convertImagePointToViewPoint:` (page 97)
  Converts an image coordinate to an image view coordinate.
- `convertImageRectToViewRect:` (page 98)
  Converts an image rectangle to an image view rectangle.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

autohidesScrollers

Specifies the automatic-hiding scroll bar state for the image view.

```
@property BOOL autohidesScrollers
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

autoresizes

Specifies the automatic resizing state for the image view.

```
@property BOOL autoresizes
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

backgroundColor

Specifies the background color for the image view.

```
@property(assign) NSColor *backgroundColor
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

currentToolMode

Specifies the current tool mode for the image view.

```
@property(copy) NSString *currentToolMode
```

Discussion
See "Tool Modes" (page 107) for possible values.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

delegate

Specifies the delegate object of the receiver.

```
@property(assign) id delegate
```

Discussion
An `IKImageView` object's delegate is inserted in the responder chain after the image view itself and is informed of various actions by the image view through delegation messages.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

doubleClickOpensImageEditPanel

Specifies the image-opening state of the editing pane in the image view.

```
@property BOOL doubleClickOpensImageEditPanel
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

editable

Specifies the editable state for the image view.

```
@property BOOL editable
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

hasHorizontalScroller

Specifies the horizontal scroll bar state for the image view.

```
@property BOOL hasHorizontalScroller
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

hasVerticalScroller

Specifies the vertical scroll bar state for the image view.

```
@property BOOL hasVerticalScroller
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

imageCorrection

Specifies a Core Image filter for image correction.

```
@property(assign) CIFilter *imageCorrection
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

rotationAngle

Specifies the rotation angle for the image view.

```
@property CGFloat rotationAngle
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

supportsDragAndDrop

Specifies the drag-and-drop support state for the image view.

```
@property BOOL supportsDragAndDrop
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

zoomFactor

Specifies the zoom factor for the image view.

```
@property CGFloat zoomFactor
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

Instance Methods

convertImagePointToViewPoint:

Converts an image coordinate to an image view coordinate.

```
- (NSPoint)convertImagePointToViewPoint:(NSPoint)imagePoint
```

Parameters
imagePoint
A point specified in coordinates relative to the image.

Return Value
A point specified in coordinates relative to the image view.

Availability
Available in Mac OS X v10.5 and later.

See Also
- convertViewPointToImagePoint: (page 98)

Declared In
IKImageView.h

convertImageRectToViewRect:

Converts an image rectangle to an image view rectangle.

```
- (NSRect)convertImageRectToViewRect:(NSRect)imageRect
```

Parameters
imageRect
An rectangle specified in coordinates relative to the image.

Return Value
An rectangle specified in coordinates relative to the image view.

Availability
Available in Mac OS X v10.5 and later.

See Also
- convertViewRectToImageRect: (page 98)

Declared In
IKImageView.h

convertViewPointToImagePoint:

Converts an image view coordinate to an image coordinate.

```
- (NSPoint)convertViewPointToImagePoint:(NSPoint)viewPoint
```

Parameters
viewPoint
A point specified in coordinates relative to the image view.

Return Value
The point specified in coordinates relative to the image.

Availability
Available in Mac OS X v10.5 and later.

See Also
- convertImagePointToViewPoint: (page 97)

Declared In
IKImageView.h

convertViewRectToImageRect:

Converts an image view rectangle to an image rectangle.

```
- (NSRect)convertViewRectToImageRect:(NSRect)viewRect
```

Parameters

viewRect

An rectangle specified in coordinates relative to the image view.

Return Value

The rectangle specified in coordinates relative to the image.

Availability

Available in Mac OS X v10.5 and later.

See Also

- convertImageRectToViewRect: (page 98)

Declared In

IKImageView.h

crop:

Crops the image using the current selection.

```
- (void)crop:(id)sender
```

Parameters

sender

Typically the object that invoked this method.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKImageView.h

flipImageHorizontal:

Flips an image along the horizontal axis.

```
- (void)flipImageHorizontal:(id)sender
```

Parameters

sender

The object initiating the action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- flipImageVertical: (page 100)

Declared In

IKImageView.h

flipImageVertical:

Flips an image along the vertical axis.

```
- (void)flipImageVertical:(id)sender
```

Parameters

sender

The object initiating the action.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `flipImageHorizontal:` (page 99)

Declared In
`IKImageView.h`

image

Returns the image associated with the view, after any image corrections.

```
- (CGImageRef)image
```

Return Value
The image.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `setImage:imageProperties:` (page 103)
- `setImageWithURL:` (page 103)

Declared In
`IKImageView.h`

imageProperties

Returns the metadata for the image in the view.

```
- (NSDictionary *)imageProperties
```

Return Value
A dictionary of metadata that specifies the image properties.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

imageSize

Returns the size of the image in the image view.

```
- (NSSize)imageSize
```

Return Value
The size of the image.

Discussion
The image size changes whenever an image is rotates or cropped.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageView.h`

overlayForType:

Returns the Core Animation layer associated with a layer type.

```
- (CALayer *)overlayForType:(NSString *)layerType
```

Parameters
layerType
A layer type. See "Overlay Types" (page 108).

Return Value
The Core Animation layer.

Availability
Available in Mac OS X v10.5 and later.

See Also
- setOverlay:forType: (page 104)

Declared In
`IKImageView.h`

rotateImageLeft:

Rotates the image left (counter-clockwise).

```
- (void)rotateImageLeft:(id)sender
```

Parameters
sender
Typically the object that invoked this method.

Availability
Available in Mac OS X v10.6 and later.

See Also
- rotateImageRight: (page 102)

Declared In
IKImageView.h

rotateImageRight:

Rotates the image right (clockwise).

```
- (void)rotateImageRight:(id)sender
```

Parameters

sender
Typically the object that invoked this method.

Availability
Available in Mac OS X v10.6 and later.

See Also
- rotateImageLeft: (page 101)

Declared In
IKImageView.h

scrollToPoint:

Scrolls the view to the specified point.

```
- (void)scrollToPoint:(NSPoint)point
```

Parameters

point
The point to scroll to.

Availability
Available in Mac OS X v10.5 and later.

See Also
- scrollToRect: (page 102)

Declared In
IKImageView.h

scrollToRect:

Scrolls the view so that it includes the provided rectangular area.

```
- (void)scrollToRect:(NSRect)rect
```

Parameters

rect
The rectangular area to include in the view.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `scrollToPoint:` (page 102)

Declared In
`IKImageView.h`

setImage:imageProperties:

Sets the image to display in an image view.

```
- (void)setImage:(CGImageRef)image imageProperties:(NSDictionary *)metaData
```

Parameters

image
The image to set.

metaData
A dictionary that contains metadata that describes the image.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `image` (page 100)
- `imageProperties` (page 100)
- `setImageWithURL:` (page 103)

Declared In
`IKImageView.h`

setImageWithURL:

Initializes an image view with the image specified by a URL.

```
- (void)setImageWithURL:(NSURL *)url
```

Parameters

url
The URL that specifies the location of the image.

Discussion
This method is the preferred initializer for RAW images. If you use this method for a TIFF file that contains multiple images, only the first image is displayed.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `setImage:imageProperties:` (page 103)

Declared In
`IKImageView.h`

setImageZoomFactor:centerPoint:

Sets the zoom factor at the provided origin.

```
- (void)setImageZoomFactor:(CGFloat)zoomFactor centerPoint:(NSPoint)centerPoint
```

Parameters

zoomFactor

The zoom factor to apply to the image.

centerPoint

The point that specifies the origin of the zoom factor.

Availability
Available in Mac OS X v10.5 and later.

See Also
`@property zoomFactor` (page 97)

Declared In
`IKImageView.h`

setOverlay:forType:

Sets an overlay type for a Core Animation layer.

```
- (void)setOverlay:(CALayer *)layer forType:(NSString *)layerType
```

Parameters

layer

A Core Animation layer object.

layerType

A layer type. See "Overlay Types" (page 108).

Availability
Available in Mac OS X v10.5 and later.

See Also
`- overlayForType:` (page 101)

Declared In
`IKImageView.h`

setRotationAngle:centerPoint:

Sets the rotation angle at the provided origin.

```
- (void)setRotationAngle:(CGFloat)rotationAngle centerPoint:(NSPoint)centerPoint
```

Parameters

rotationAngle

The rotation angle to apply to the image.

centerPoint
The point that specifies the origin of the rotation angle.

Availability
Available in Mac OS X v10.5 and later.

See Also
`@property rotationAngle` (page 96)

Declared In
`IKImageView.h`

zoomImageToActualSize:

Zooms the image so that it is displayed using its true size.

```
- (void)zoomImageToActualSize:(id)sender
```

Parameters
sender
The object initiating the action.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- zoomImageToFit:` (page 105)
`- zoomImageToRect:` (page 106)

Declared In
`IKImageView.h`

zoomImageToFit:

Zooms the image so that it fits in the image view.

```
- (void)zoomImageToFit:(id)sender
```

Parameters
sender
The object initiating the action.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- zoomImageToActualSize:` (page 105)
`- zoomImageToRect:` (page 106)

Declared In
`IKImageView.h`

zoomImageToRect:

Zooms the image so that it fits in the specified rectangle.

```
- (void)zoomImageToRect:(NSRect)rect
```

Parameters

rect

The rectangle to fit the image in.

Availability
Available in Mac OS X v10.5 and later.

See Also
- zoomImageToFit: (page 105)
- zoomImageToActualSize: (page 105)

Declared In
IKImageView.h

zoomIn:

Zooms the image in.

```
- (void)zoomIn:(id)sender
```

Parameters

sender

Typically the object that invoked this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageView.h

zoomOut:

Zooms the image out.

```
- (void)zoomOut:(id)sender
```

Parameters

sender

Typically the object that invoked this method.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageView.h

Constants

Tool Modes

Image Kit tools modes referenced by the `currentToolMode` (page 95) property.

```
NSString *const IKToolModeNone;
NSString *const IKToolModeMove;
NSString *const IKToolModeSelect;
NSString *const IKToolModeSelectRect;
NSString *const IKToolModeSelectEllipse;
NSString *const IKToolModeSelectLasso;
NSString *const IKToolModeCrop;
NSString *const IKToolModeRotate;
NSString *const IKToolModeAnnotate;
```

Constants

`IKToolModeNone`
No tool is set.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKToolModeMove`
The move tool.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKToolModeSelect`
The selection tool.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKToolModeSelectRect`
Same as `IKToolModeSelect`.

Available in Mac OS X v10.6 and later.

Declared in `IKImageView.h`.

`IKToolModeSelectEllipse`
The selection ellipse.

Available in Mac OS X v10.6 and later.

Declared in `IKImageView.h`.

`IKToolModeSelectLasso`
The selection lasso.

Available in Mac OS X v10.6 and later.

Declared in `IKImageView.h`.

`IKToolModeCrop`
The crop tool.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKToolModeRotate`
The rotation tool.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKToolModeAnnotate`
The annotation tool.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

Declared In
`IKImageView.h`

Overlay Types

A layer level.

```
NSString *const IKOverlayTypeBackground;
NSString *const IKOverlayTypeImage;
```

Constants
`IKOverlayTypeBackground`
A background.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

`IKOverlayTypeImage`
An image.

Available in Mac OS X v10.5 and later.

Declared in `IKImageView.h`.

Declared In
`IKImageView.h`

IKPictureTaker Class Reference

Inherits from	NSPanel : NSWindow : NSResponder : NSObject
Conforms to	NSUserInterfaceValidations (NSWindow) NSAnimatablePropertyContainer (NSWindow) NSCoding (NSResponder) NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKPictureTaker.h
Related sample code	ImageKitDemo

Overview

The `IKPictureTaker` class represents a panel that allows users to choose images by browsing the file system. The picture taker panel provides an Open Recent menu, supports image cropping, and supports taking snapshots from an iSight or other digital camera.

Tasks

Creating And Displaying The Picture Taker

`+ pictureTaker` (page 110)
Returns a shared `IKPictureTaker` instance, creating it if necessary.

`- beginPictureTakerSheetForWindow:withDelegate:didEndSelector:contextInfo:` (page 110)
Opens a picture taker as a sheet whose parent is the specified window.

`- beginPictureTakerWithDelegate:didEndSelector:contextInfo:` (page 111)
Opens a picture taker pane.

`- popUpRecentsMenuForView:withDelegate:didEndSelector:contextInfo:` (page 113)
Displays the Open Recent popup menu associated with the picture taker.

`- runModal` (page 113)
Opens a modal picture taker dialog.

Getting and Setting Images

- `setInputImage:` (page 114)
  Set the image input for the picture taker.
- `inputImage` (page 112)
  Returns the input image associated with the picture taker.
- `outputImage` (page 112)
  Returns the edited image.

Getting and Setting Mirroring

- `setMirroring:` (page 114)
  Controls whether the receiver enables video mirroring during snapshots.
- `mirroring` (page 112)
  Returns whether video mirroring is enabled during snapshots.

Class Methods

pictureTaker

Returns a shared `IKPictureTaker` instance, creating it if necessary.

```
+ (IKPictureTaker *)pictureTaker
```

Return Value
An `IKPictureTaker` object.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
ImageKitDemo

Declared In
`IKPictureTaker.h`

Instance Methods

beginPictureTakerSheetForWindow:withDelegate:didEndSelector:contextInfo:

Opens a picture taker as a sheet whose parent is the specified window.

```
- (void)beginPictureTakerSheetForWindow:(NSWindow *)aWindow withDelegate:(id)delegate
    didEndSelector:(SEL)didEndSelector contextInfo:(void *)contextInfo
```

Parameters

aWindow

The parent window of the picture taker sheet.

delegate

The object that will invoke the selector `didEndSelector` when the picture taker session terminates.

didEndSelector

The selector to invoke when the picture taker session terminates.

contextInfo

Any data that must be passed as an argument to the delegate through `didEndSelector` after the picture taker session terminates.

Discussion

The `didEndSelector` method should have the following signature:

```
- (void)pictureTakerDidEnd:(IKPictureTaker *)sheet returnCode:(NSInteger)returnCode
contextInfo:(void *)contextInfo;
```

The `returnCode` value is set to `NSOKButton` if the user validates, or to `NSCancelButton` if the user cancels.

Availability

Available in Mac OS X v10.5 and later.

See Also

`- beginPictureTakerWithDelegate:didEndSelector:contextInfo:` (page 111)

Related Sample Code

ImageKitDemo

Declared In

`IKPictureTaker.h`

beginPictureTakerWithDelegate:didEndSelector:contextInfo:

Opens a picture taker pane.

```
- (void)beginPictureTakerWithDelegate:(id)delegate didEndSelector:(SEL)didEndSelector
    contextInfo:(void *)contextInfo
```

Parameters

delegate

The object that will invoke the selector `didEndSelector` when the picture taker session terminates.

didEndSelector

The selector to invoke when the picture taker session terminates.

contextInfo

Any data that must be passed as an argument to the delegate through `didEndSelector` after the picture taker session terminates.

Discussion

The `didEndSelector` method should have the following signature:

```
- (void)pictureTakerDidEnd:(IKPictureTaker *)sheet returnCode:(NSInteger)returnCode
contextInfo:(void *)contextInfo;
```

The `returnCode` value is set to `NSOKButton` if the user validates, or to `NSCancelButton` if the user cancels.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- beginPictureTakerSheetForWindow:withDelegate:didEndSelector:contextInfo:` (page 110)

Declared In
`IKPictureTaker.h`

inputImage

Returns the input image associated with the picture taker.

```
- (NSImage *)inputImage
```

Return Value
The input image.

Discussion
The input image is never modified by the picture taker.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setInputImage:` (page 114)

Declared In
`IKPictureTaker.h`

mirroring

Returns whether video mirroring is enabled during snapshots.

```
- (BOOL)mirroring
```

Return Value
Returns YES if video mirroring is enabled, NO otherwise.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKPictureTaker.h`

outputImage

Returns the edited image.

```
- (NSImage *)outputImage
```

Return Value
The edited image.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
ImageKitDemo
ImagePicker

Declared In
`IKPictureTaker.h`

popUpRecentsMenuForView:withDelegate:didEndSelector:contextInfo:

Displays the Open Recent popup menu associated with the picture taker.

```
- (void)popUpRecentsMenuForView:(NSView *)aView withDelegate:(id)delegate
    didEndSelector:(SEL)didEndSelector contextInfo:(void *)contextInfo
```

Parameters

delegate
The object that will invoke the selector `didEndSelector` when the picture taker session terminates.

didEndSelector
The selector to invoke when the picture taker session terminates.

contextInfo
Any data that must be passed as an argument to the delegate through `didEndSelector` after the picture taker session terminates.

Discussion
The `didEndSelector` method should have the following signature:

```
- (void)pictureTakerDidEnd:(IKPictureTaker *)sheet returnCode:(NSInteger)returnCode
contextInfo:(void *)contextInfo;
```

The `returnCode` value is set to `NSOKButton` if the user validates, or to `NSCancelButton` if the user cancels.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKPictureTaker.h`

runModal

Opens a modal picture taker dialog.

```
- (NSInteger)runModal
```

Return Value
Returns `NSOKButton` if the user edits or chooses an image; `NSCancelButton` if the user cancels or does not change the default image.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKPictureTaker.h`

setInputImage:

Set the image input for the picture taker.

```
- (void)setInputImage:(NSImage *)image
```

Parameters

image

An `NSImage` object.

Discussion
The input image is never modified by the picture taker.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- inputImage` (page 112)

Related Sample Code
ImagePicker

Declared In
`IKPictureTaker.h`

setMirroring:

Controls whether the receiver enables video mirroring during snapshots.

```
- (void)setMirroring:(BOOL)b
```

Parameters

b

The default setting is `YES`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKPictureTaker.h`

Constants

Picture Taker Keys

Keys for customizing the picture taker appearance and behavior. These values are set by sending the picture taker instance `setValue:forKey`.

```
NSString *const IKPictureTakerAllowsVideoCaptureKey;
NSString *const IKPictureTakerAllowsFileChoosingKey;
NSString *const IKPictureTakerShowRecentPictureKey;
NSString *const IKPictureTakerUpdateRecentPictureKey;
NSString *const IKPictureTakerAllowsEditingKey;
NSString *const IKPictureTakerShowEffectsKey;
NSString *const IKPictureTakerInformationalTextKey;
NSString *const IKPictureTakerImageTransformsKey;
NSString *const IKPictureTakerOutputImageMaxSizeKey;
NSString *const IKPictureTakerCropAreaSizeKey;
NSString *const IKPictureTakerShowAddressBookPictureKey;
NSString *const IKPictureTakerShowEmptyPictureKey;
NSString *const IKPictureTakerRemainOpenAfterValidateKey;
```

Constants

`IKPictureTakerAllowsVideoCaptureKey`

A key for allowing video capture. The associated value is an `NSNumber` value (`BOOL`) whose default value is `YES`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerAllowsFileChoosingKey`

A key for allowing the user to choose a file. The associated value is an `NSNumber` object that contains a `BOOL` value whose default value is `YES`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerUpdateRecentPictureKey`

A key for allowing a recent picture to be updated. The associated value is an `NSNumber` object that contains a `BOOL` value whose default value is `YES`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerAllowsEditingKey`

A key for allowing image editing. The associated value is an `NSNumber` object that contains a `BOOL` value whose default value is `YES`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerShowEffectsKey`

A key for showing effects. The associated value is an `NSNumber` object that contains a `BOOL` value whose default value is `NO`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerInformationalTextKey`
A key for informational text. The associated value is an `NSString` or `NSAttributedString` object whose default value is `"Drag Image Here"`.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerImageTransformsKey`
A n image transformation key. The associated value is an `NSDictionary` object that can be serialized.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerOutputImageMaxSizeKey`
A key for the maximum size of the output image. The associated value is an `NSValue` object (`NSSize`).

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerCropAreaSizeKey`
A key for the cropping area size. The associated value is an `NSValue` object (`NSSize`).

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerShowAddressBookPictureKey`
A key for showing the address book picture. The associated value is a Boolean value packages as an `NSNumber` object. The default value is `NO`. If set to YES, the picture taker automatically adds the address book image for the Me user at the end of the Recent Pictures pop-up menu.

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerShowEmptyPictureKey`
A key for showing an empty picture. The associated value is an `NSImage` object. The default value is `nil`. If set to an image, the picture taker automatically shows an image at the end of the Recent Pictures pop-up menu. that means "no picture."

Available in Mac OS X v10.5 and later.

Declared in `IKPictureTaker.h`.

`IKPictureTakerRemainOpenAfterValidateKey`
A key that determines if the picture taker should remain open after the user selects done. This allows the application to programmatically dismiss the panel. The associated value is an `NSNumber` object that contains a `BOOL` value whose default value is `NO`

Available in Mac OS X v10.6 and later.

Declared in `IKPictureTaker.h`.

IKSaveOptions Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKSaveOptions.h
Related sample code	CIRAWFilterSample DispatchFractal ImageKitDemo

Overview

The `IKSaveOptions` class initializes, adds, and manages user interface options for saving image data.

Tasks

Creating A Save Options Accessory View

`- initWithImageProperties:imageUTType:` (page 120)
Initializes a save options accessory pane for the provided image properties and uniform type identifier.

`- addSaveOptionsAccessoryViewToSavePanel:` (page 119)
Adds `IKSaveOptions` accessory view to a `NSSavePanel`.

Retrieving User Responses

`imageProperties` (page 118) *property*
Returns a dictionary of updated image properties that reflects the user's selection. (read-only)

`imageUTType` (page 119) *property*
Returns the uniform type identifier that reflects the user's selection. (read-only)

`userSelection` (page 119) *property*
Returns a dictionary that contains the save options selected by the user. (read-only)

Getting and Setting the Delegate

`delegate` (page 118) *property*
Specifies the delegate object.

File Type Filtering

`- saveOptions:shouldShowUTType:` (page 120) *delegate method*
Called to determine if the specified uniform type identifier should be shown in the save panel.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

delegate

Specifies the delegate object.

```
@property(assign) id delegate
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`- saveOptions:shouldShowUTType:` (page 120)

Declared In
`IKSaveOptions.h`

imageProperties

Returns a dictionary of updated image properties that reflects the user's selection. (read-only)

```
@property(readonly) NSDictionary *imageProperties
```

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
DispatchFractal

Declared In
`IKSaveOptions.h`

imageUTType

Returns the uniform type identifier that reflects the user's selection. (read-only)

```
@property(readonly) NSString *imageUTType
```

Availability
Available in Mac OS X v10.6 and later.

Related Sample Code
DispatchFractal

Declared In
`IKSaveOptions.h`

userSelection

Returns a dictionary that contains the save options selected by the user. (read-only)

```
@property(readonly) NSDictionary *userSelection
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKSaveOptions.h`

Instance Methods

addSaveOptionsAccessoryViewToSavePanel:

Adds `IKSaveOptions` accessory view to a `NSSavePanel`.

```
- (void)addSaveOptionsAccessoryViewToSavePanel:(NSSavePanel *)savePanel
```

Parameters

savePanel
The save panel to add the `IKSaveOptions` to.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
DispatchFractal

Declared In
`IKSaveOptions.h`

initWithImageProperties:imageUTType:

Initializes a save options accessory pane for the provided image properties and uniform type identifier.

```
- (id)initWithImageProperties:(NSDictionary *)imageProperties
    imageUTType:(NSString *)imageUTType
```

Parameters

imageProperties

A dictionary of image properties.

imageUTType

A string that specifies a uniform type identifier, such as `JPEG`. See *Uniform Type Identifiers Overview*.

Return Value
The initialized object.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
CIRAWFilterSample
DispatchFractal
ImageKitDemo

Declared In
`IKSaveOptions.h`

Delegate Methods

saveOptions:shouldShowUTType:

Called to determine if the specified uniform type identifier should be shown in the save panel.

```
- (BOOL)saveOptions:(IKSaveOptions *)saveOptions
    shouldShowUTType:(NSString *)utType
```

Parameters

saveOptions

The `IKSaveOptions` instance that called the delegate.

utType

The uniform type identifier to test.

Return Value
`YES` if the specified type should be shown in the save options, otherwise `NO`.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKSaveOptions.h`

IKScannerDeviceView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKScannerDeviceView.h
Related sample code	ImageKitDemo

Overview

The `IKScannerDeviceView` class displays a view that allows scanning. It can be customized by specifying the display mode. The delegate receives the scanned data and must implement the `IKScannerDeviceViewDelegate` protocol.

Tasks

Setting the Scanner Device

`scannerDevice` (page 126) *property*
The device used for scanning

Display Mode of the Device View

`mode` (page 124) *property*
The display mode used by the device view.

`hasDisplayModeAdvanced` (page 124) *property*
Returns whether the scanner view is using the advanced display mode.

`hasDisplayModeSimple` (page 124) *property*
Returns whether the scanner view is using the simple display mode.

Configuring Downloading

`displaysDownloadsDirectoryControl` (page 123) *property*
Determines whether the downloads directory control is displayed.

`downloadsDirectory` (page 124) *property*
The directory where scans are saved.

`transferMode` (page 126) *property*
Determines how the scanned content is provided to the delegate.

`documentName` (page 123) *property*
Returns the document name.

Specifying a Post Processing Application

`displaysPostProcessApplicationControl` (page 123) *property*
Specifies whether the post processing application control is displayed.

`postProcessApplication` (page 125) *property*
The URL of the application to use for post processing of the scan.

Getting and Setting the Delegate

`delegate` (page 122) *property*
The scanner device delegate

Customizing Button Labels

`overviewControlLabel` (page 125) *property*
Allows customization of the “Overview” label.

`scanControlLabel` (page 125) *property*
Allows customization of the “Scan” label.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

delegate

The scanner device delegate

```
@property(assign) id<IKScannerDeviceViewDelegate> delegate
```

Discussion
The delegate is sent notifications of errors as well as the completed scan content.

The delegate must conform to the `IKScannerDeviceViewDelegate` protocol.

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

displaysDownloadsDirectoryControl

Determines whether the downloads directory control is displayed.

```
@property BOOL displaysDownloadsDirectoryControl
```

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property downloadsDirectory` (page 124)

Declared In
`IKScannerDeviceView.h`

displaysPostProcessApplicationControl

Specifies whether the post processing application control is displayed.

```
@property BOOL displaysPostProcessApplicationControl
```

Discussion
The post processing application is only relevant when the transfer mode is `IKScannerDeviceViewTransferModeFileBased` (page 127).

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property postProcessApplication` (page 125)

Declared In
`IKScannerDeviceView.h`

documentName

Returns the document name.

```
@property(copy) NSString *documentName
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKScannerDeviceView.h

downloadsDirectory

The directory where scans are saved.

```
@property(retain) NSURL *downloadsDirectory
```

Availability
Available in Mac OS X v10.6 and later.

See Also
@property displaysDownloadsDirectoryControl (page 123)

Declared In
IKScannerDeviceView.h

hasDisplayModeAdvanced

Returns whether the scanner view is using the advanced display mode.

```
@property BOOL hasDisplayModeAdvanced
```

Availability
Available in Mac OS X v10.6 and later.

See Also
@property hasDisplayModeSimple (page 124)

Declared In
IKScannerDeviceView.h

hasDisplayModeSimple

Returns whether the scanner view is using the simple display mode.

```
@property BOOL hasDisplayModeSimple
```

Availability
Available in Mac OS X v10.6 and later.

See Also
@property hasDisplayModeAdvanced (page 124)

Declared In
IKScannerDeviceView.h

mode

The display mode used by the device view.

```
@property IKScannerDeviceViewDisplayMode mode
```

Discussion
The possible constants are described in "Scanner View Display Mode" (page 127).

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

overviewControlLabel

Allows customization of the "Overview" label.

```
@property(copy) NSString *overviewControlLabel
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

postProcessApplication

The URL of the application to use for post processing of the scan.

```
@property(retain) NSURL *postProcessApplication
```

Discussion
The post processing application is only relevant when the transfer mode is `IKScannerDeviceViewTransferModeFileBased` (page 127).

Availability
Available in Mac OS X v10.6 and later.

See Also
`@property displaysPostProcessApplicationControl` (page 123)

Related Sample Code
ImageKitDemo

Declared In
`IKScannerDeviceView.h`

scanControlLabel

Allows customization of the "Scan" label.

```
@property(copy) NSString *scanControlLabel
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

scannerDevice

The device used for scanning

```
@property(assign) ICScannerDevice *scannerDevice
```

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

transferMode

Determines how the scanned content is provided to the delegate.

```
@property IKScannerDeviceViewTransferMode transferMode
```

Discussion
The supported constants are defined in "Scanner Transfer Modes" (page 126).

Availability
Available in Mac OS X v10.6 and later.

Declared In
`IKScannerDeviceView.h`

Constants

Scanner Transfer Modes

These constants determine how the scanner data is returned to the delegate. They are used by the `transferMode` (page 126) property.

```
enum {
    IKScannerDeviceViewTransferModeFileBased = 0,
    IKScannerDeviceViewTransferModeMemoryBased
```

```
};
typedef NSInteger IKScannerDeviceViewTransferMode;
```

Constants

`IKScannerDeviceViewTransferModeFileBased`
The scanned content will be saved to the specified download directory.
Available in Mac OS X v10.6 and later.
Declared in `IKScannerDeviceView.h`.

`IKScannerDeviceViewTransferModeMemoryBased`
The scanned data is returned to the delegate as a `NSData` object.
Available in Mac OS X v10.6 and later.
Declared in `IKScannerDeviceView.h`.

Scanner View Display Modes

These constants specify the display mode the scanner view will use. They are used by the `mode` (page 124) property.

```
enum {
    IKScannerDeviceViewDisplayModeSimple,
    IKScannerDeviceViewDisplayModeAdvanced
};
typedef NSInteger IKScannerDeviceViewDisplayMode;
```

Constants

`IKScannerDeviceViewDisplayModeSimple`
The view will display in simple mode.
Available in Mac OS X v10.6 and later.
Declared in `IKScannerDeviceView.h`.

`IKScannerDeviceViewDisplayModeAdvanced`
The view will display in advanced mode.
Available in Mac OS X v10.6 and later.
Declared in `IKScannerDeviceView.h`.

IKSlideshow Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKSlideshow.h
Related sample code	ImageKitDemo

Overview

The `IKSlideshow` class encapsulates a data source and options for a slideshow.

Tasks

Creating a Shared Instance of a Slideshow

`+ sharedSlideshow` (page 131)
Returns a shared instance of a slideshow.

Running and Stopping a Slideshow

`- runSlideshowWithDataSource:inMode:options:` (page 133)
Runs a slideshow that contains the specified kind of items, provided from a data source.

`- stopSlideshow:` (page 133)
Stops a slideshow.

`autoPlayDelay` (page 130) *property*
Controls the interval of time before a slideshow starts to play automatically.

Getting Slideshow Data

- `indexOfCurrentSlideshowItem` (page 132)
  Returns the index of the current slideshow item.

Reloading Data

- `reloadData` (page 132)
  Reloads the data for a slideshow.
- `reloadSlideshowItemAtIndex:` (page 132)
  Reloads the data for a slideshow, starting at the specified index.

Exporting Slideshow Items

+ `canExportToApplication:` (page 130)
  Finds out whether the slideshow can export its contents to an application.
+ `exportSlideshowItem:toApplication:` (page 131)
  Exports a slideshow item to the application that has the provided bundle identifier.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

autoPlayDelay

Controls the interval of time before a slideshow starts to play automatically.

```
@property NSTimeInterval autoPlayDelay
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKSlideshow.h`

Class Methods

canExportToApplication:

Finds out whether the slideshow can export its contents to an application.

```
+ (BOOL)canExportToApplication:(NSString *)applicationBundleIdentifier
```

Parameters

applicationBundleIdentifier

The bundle identifier of the application that you want to export the slideshow to. See "Bundle Identifiers" (page 134).

Return Value

YES if the slideshow can be exported to the specified application; NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

+ exportSlideshowItem:toApplication: (page 131)

Declared In

IKSlideshow.h

exportSlideshowItem:toApplication:

Exports a slideshow item to the application that has the provided bundle identifier.

```
+ (void)exportSlideshowItem:(id)item toApplication:(NSString
    *)applicationBundleIdentifier
```

Parameters

item

The item to export

applicationBundleIdentifier

The bundle identifier of the application that you want to export the item to.

Availability

Available in Mac OS X v10.5 and later.

See Also

+ canExportToApplication: (page 130)

Declared In

IKSlideshow.h

sharedSlideshow

Returns a shared instance of a slideshow.

```
+ (IKSlideshow *)sharedSlideshow
```

Return Value

A slideshow object.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

ImageKitDemo

Declared In
IKSlideshow.h

Instance Methods

indexOfCurrentSlideshowItem

Returns the index of the current slideshow item.

```
- (NSUInteger)indexOfCurrentSlideshowItem
```

Return Value
The index of the current item in the slideshow.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
ImageKitDemo

Declared In
IKSlideshow.h

reloadData

Reloads the data for a slideshow.

```
- (void)reloadData
```

Availability
Available in Mac OS X v10.5 and later.

See Also
- reloadSlideshowItemAtIndex: (page 132)

Declared In
IKSlideshow.h

reloadSlideshowItemAtIndex:

Reloads the data for a slideshow, starting at the specified index.

```
- (void)reloadSlideshowItemAtIndex:(NSUInteger)index
```

Parameters
index
The index that species where to reload the slideshow data.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `reloadData` (page 132)

Declared In
`IKSlideshow.h`

runSlideshowWithDataSource:inMode:options:

Runs a slideshow that contains the specified kind of items, provided from a data source.

```
- (void)runSlideshowWithDataSource:(id < IKSlideshowDataSource >)dataSource
    inMode:(NSString *)slideshowMode options:(NSDictionary *)slideshowOptions
```

Parameters

dataSource
The data source to use for the slideshow.

slideshowMode
A constant that indicate what kind of items are in the slideshow—`IKSlideshowModeImages`, `IKSlideshowModePDF`, or `IKSlideshowModeQuickLook`. See "Slideshow Modes" (page 134).

slideshowOptions
A dictionary of slideshow options. See "Slideshow Option Keys" (page 135).

Availability
Available in Mac OS X v10.5 and later.

See Also
- `stopSlideshow:` (page 133)

Related Sample Code
ImageKitDemo

Declared In
`IKSlideshow.h`

stopSlideshow:

Stops a slideshow.

```
- (void)stopSlideshow:(id)sender
```

Parameters

sender
The object sending the message to stop the slideshow.

Discussion
This method is invoked when the user clicks a button or issues a stop command.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `runSlideshowWithDataSource:inMode:options:` (page 133)

Declared In
`IKSlideshow.h`

Constants

Bundle Identifiers

Identifiers for exporting slideshow items to an application.

```
NSString *const IK_iPhotoBundleIdentifier;
NSString *const IK_ApertureBundleIdentifier;
NSString *const IK_MailBundleIdentifier;
```

Constants

`IK_iPhotoBundleIdentifier`
The iPhoto application—`com.apple.iPhoto`.
Available in Mac OS X v10.5 and later.
Declared in `IKSlideshow.h`.

`IK_ApertureBundleIdentifier`
The Aperature application—`com.apple.Aperture`.
Available in Mac OS X v10.6 and later.
Declared in `IKSlideshow.h`.

`IK_MailBundleIdentifier`
The Mail application—`com.apple.mail`.
Available in Mac OS X v10.6 and later.
Declared in `IKSlideshow.h`.

Slideshow Modes

The kind of items in the slideshow.

```
NSString *const IKSlideshowModeImages;
NSString *const IKSlideshowModePDF;
NSString *const IKSlideshowModeOther;
```

Constants

`IKSlideshowModeImages`
All items in the slideshow are images.
Available in Mac OS X v10.5 and later.
Declared in `IKSlideshow.h`.

`IKSlideshowModePDF`
All items in the slideshow are PDF documents.
Available in Mac OS X v10.5 and later.
Declared in `IKSlideshow.h`.

`IKSlideshowModeOther`
There are a mixture of items in the slideshow (image, PDF, text, HTML, and so on).

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

Slideshow Option Keys

Keys for slideshow options.

```
NSString *const IKSlideshowWrapAround;
NSString *const IKSlideshowStartPaused;
NSString *const IKSlideshowStartIndex;
NSString *const IKSlideshowPDFDisplayBox;
NSString *const IKSlideshowPDFDisplayMode;
NSString *const IKSlideshowPDFDisplaysAsBook;
NSString *const IKSlideshowScreen;
NSString *const IKSlideshowAudioFile;
```

Constants

`IKSlideshowWrapAround`
A key for starting the slideshow over after the last slide shows. The associated value is a `Boolean` data type.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowStartPaused`
A key for starting in a paused state. The associated value is a `Boolean` data type.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowStartIndex`
A key for the slideshow item index. The associated value is an index.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowPDFDisplayBox`
A key for the PDF display box. The associated value is a type of display box, such as `kPDFDisplayBoxMediaBox` or `kPDFDisplayBoxMediaBox`. See *PDFPage Class Reference* for more information.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowPDFDisplayMode`
A key for the PDF display mode. The associated value is a PDF display mode constant, such as `kPDFDisplaySinglePage` or `kPDFDisplayTwoUp`. See *PDFView Class Reference* for more information.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowPDFDisplaysAsBook`
A key for displaying the slideshow as a book. The associated value is a `Boolean` data type.

Available in Mac OS X v10.5 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowScreen`

A key specifying the screen on which the slideshow is displayed. The associated value is an `NSScreen` object. By default `mainScreen` is used.

Available in Mac OS X v10.6 and later.

Declared in `IKSlideshow.h`.

`IKSlideshowAudioFile`

A key specifying the audio file played during the slideshow. The associated value is an `NSURL` object.

Available in Mac OS X v10.6 and later.

Declared in `IKSlideshow.h`.

Protocols

IKCameraDeviceViewDelegate Protocol Reference

Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKCameraDeviceView.h

Overview

The `IKCameraDeviceViewDelegate` protocol is adopted by the delegate of the `IKCameraDeviceView` class. It allows downloading of camera content, handling selection changes, and handling errors.

Tasks

Downloading Camera Content

`- cameraDeviceView:didDownloadFile:location:fileData:error:` (page 139)
Invoked for each file that is downloaded from the camera device.

Detecting Selection Changes

`- cameraDeviceViewSelectionDidChange:` (page 141)
Invoked when the selection changed.

Managing Errors

`- cameraDeviceView:didEncounterError:` (page 140)
Invoked when the camera encounters an error.

Instance Methods

cameraDeviceView:didDownloadFile:location:fileData:error:

Invoked for each file that is downloaded from the camera device.

```
- (void)cameraDeviceView:(IKCameraDeviceView *)cameraDeviceView
    didDownloadFile:(ICCameraFile *)file location:(NSURL *)url fileData:(NSData
    *)data error:(NSError *)error
```

Parameters

cameraDeviceView

The camera device view that sent the message.

file

The camera file.

url

The URL to save the data.

data

The data.

error

Any errors encountered during downloading.

Discussion

If the `IKCameraDeviceView` `transferMode` (page 22) property is `IKCameraDeviceViewTransferModeFileBased` (page 25) then camera file will have been saved to *url*. The URL will be a complete path to save the file, including a 'sequence number' if the file already exists.

If the `transferMode` (page 22) is `IKCameraDeviceViewTransferModeMemoryBased` (page 25) then the *data* parameter contains the image data and can be handled as appropriate for your application.

In case of an error, the passed data (both *url* and *data*) will be `NULL` and *error* (which may come directly from the camera module / or the ImageCaptureCore framework) will describe why the download or save failed.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`IKCameraDeviceView.h`

cameraDeviceView:didEncounterError:

Invoked when the camera encounters an error.

```
- (void)cameraDeviceView:(IKCameraDeviceView *)cameraDeviceView
    didEncounterError:(NSError *)error
```

Parameters

cameraDeviceView

The camera device view that sent the message.

error

The error.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`IKCameraDeviceView.h`

cameraDeviceViewSelectionDidChange:

Invoked when the selection changed.

```
- (void)cameraDeviceViewSelectionDidChange:(IKCameraDeviceView *)cameraDeviceView
```

Parameters

cameraDeviceView

The camera device view that sent the message.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKCameraDeviceView.h

IKDeviceBrowserViewDelegate Protocol Reference

Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKDeviceBrowserView.h

Overview

The `IKDeviceBrowserViewDelegate` defines the methods that the delegate of the `IKDeviceBrowserView` class can implement. All the methods are optional.

Tasks

Responding to Selection Changes

`- deviceBrowserView:selectionDidChange:` (page 144)
Sent to the delegate when the selection changes in the browser view.

Responding to Errors

`- deviceBrowserView:didEncounterError:` (page 143)
Invoked when the device browser encounters an error.

Instance Methods

deviceBrowserView:didEncounterError:

Invoked when the device browser encounters an error.

```
- (void)deviceBrowserView:(IKDeviceBrowserView *)deviceBrowserView
    didEncounterError:(NSError *)error
```

Parameters

deviceBrowserView

The object that sent the message.

error

The error the device browser encountered.

Discussion

The user should handle the error in some fashion, for example, presenting an error panel to the user.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKDeviceBrowserView.h

deviceBrowserView:selectionDidChange:

Sent to the delegate when the selection changes in the browser view.

```
- (void)deviceBrowserView:(IKDeviceBrowserView *)deviceBrowserView
    selectionDidChange:(ICDevice *)device
```

Parameters

deviceBrowserView

The object that sent the message.

device

The newly selected device.

Availability

Available in Mac OS X v10.6 and later.

Declared In

IKDeviceBrowserView.h

IKFilterCustomUIProvider Protocol Reference

Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKFilterUI.h

Overview

The `IKFilterCustomUIProvider` protocol is an addition to the `CIFilter` class that defines a method for providing a view for a filter. This protocol is implemented by any filter that provides its own user interface.

Tasks

Providing a Custom View

`- provideViewForUIConfiguration:excludedKeys:` (page 145) *required method*
Provides a custom view for a filter. (required)

Instance Methods

provideViewForUIConfiguration:excludedKeys:

Provides a custom view for a filter. (required)

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

Parameters

inUIConfiguration
A dictionary that specifies the size of the controls. Provide the key `IKUISizeFlavor` and one of the following values: `IKUISizeMini`, `IKUISizeSmall`, or `IKUISizeRegular`. For more information on these constants, see *User Interface Options* in *CIFilter Image Kit Additions*.

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 or `nil` if the filter is unable to provide a view. If `nil`, the Image Kit framework will attempt to provide a user interface.

Discussion
This method overrides the method `viewForUIConfiguration:excludedKeys:` (page 12).

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKFilterUI.h`

IKImageBrowserDataSource Protocol Reference

(informal protocol)

Adopted by	IKImageBrowserView
Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Declared in	ImageKit/IKImageBrowserView.h

Overview

The `IKImageBrowserDataSource` informal protocol declares the methods that an instance of the `IKImageBrowserView` class uses to access the contents of its data source object.

Tasks

Providing Information About Items (Required)

- `numberOfItemsInImageBrowser:` (page 151) *required method*
  Returns the number of records managed by the data source object. (required)
- `imageBrowser:itemAtIndex:` (page 148) *required method*
  Returns an object for the item in an image browser view that corresponds to the specified index. (required)

Supporting Item Editing (Optional)

- `imageBrowser:removeItemsAtIndexes:` (page 149) *required method*
  Signals that a remove operation should be applied to the specified items. (required)
- `imageBrowser:moveItemsAtIndexes:toIndex:` (page 149) *required method*
  Signals that the specified items should be moved to the specified destination. (required)
- `imageBrowser:writeItemsAtIndexes:toPasteboard:` (page 150) *required method*
  Signals that a drag should begin. (required)

Providing Information About Groups (Optional)

- `numberOfGroupsInImageBrowser:` (page 150) *required method*
  Returns the number of groups in an image browser view. (required)

- `imageBrowser:groupAtIndex:` (page 148) *required method*
  Returns the group at the specified index. (required)

Instance Methods

imageBrowser:groupAtIndex:

Returns the group at the specified index. (required)

```
- (NSDictionary *) imageBrowser:(IKImageBrowserView *) aBrowser
    groupAtIndex:(NSUInteger) index;
```

Parameters

aBrowser
An image browser view.

index
The index of the group you want to retrieve.

Return Value
A dictionary that defines the group. The keys in this dictionary can be any of the following constants: `IKImageBrowserGroupStyle`, `IKImageBrowserGroupBackgroundColorKey`, `IKImageBrowserGroupTitleKey`, and `IKImageBrowserGroupRangeKey`. For more information on these constants, see *IKImageBrowserView Class Reference*.

Discussion
This method is optional.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

imageBrowser:itemAtIndex:

Returns an object for the item in an image browser view that corresponds to the specified index. (required)

```
- (id) imageBrowser:(IKImageBrowserView *) aBrowser itemAtIndex:(NSUInteger)index;
```

Parameters

aBrowser
An image browser view.

index
The index of the item you want to retrieve.

Return Value
An `IKImageBrowserItem` object.

Discussion
Your data source must implement this method. The returned object must implement the required methods of the `IKImageBrowserItem` protocol.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

imageBrowser:moveItemsAtIndexes:toIndex:

Signals that the specified items should be moved to the specified destination. (required)

```
- (BOOL) imageBrowser:(IKImageBrowserView *) aBrowser moveItemsAtIndexes: (NSIndexSet
    *)indexes toIndex:(NSUInteger)destinationIndex;
```

Parameters

aBrowser
An image browser view.

indexes
The indexes of the items that should be reordered.

destinationIndex
The starting index of the destination the items should be moved to.

Return Value
`YES` if successful; `NO` otherwise.

Discussion
This method is optional. It is invoked by the image browser view after Image Kit determines that a reordering operation should be applied. The data source should update itself by reordering its elements.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setAllowsReordering:` (page 75)

Declared In
`IKImageBrowserView.h`

imageBrowser:removeItemsAtIndexes:

Signals that a remove operation should be applied to the specified items. (required)

```
- (void) imageBrowser:(IKImageBrowserView *) aBrowser
    removeItemsAtIndexes:(NSIndexSet *) indexes;
```

Parameters

aBrowser
An image browser view.

indexes
The indexes of the items that should be removed.

Discussion
This method is optional. It is invoked by the image browser after Image Kit determines that a remove operation should be applied. In response, the data source should update itself by removing the specified items.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

imageBrowser:writeItemsAtIndexes:toPasteboard:

Signals that a drag should begin. (required)

```
- (NSUInteger) imageBrowser:(IKImageBrowserView *) aBrowser
    writeItemsAtIndexes:(NSIndexSet *) itemIndexes toPasteboard:(NSPasteboard
    *)pasteboard;
```

Parameters
aBrowser
An image browser view.
itemIndexes
The indexes of the items that should be dragged.
pasteboard
The pasteboard to copy the items to.

Return Value
The number of items written to the pasteboard.

Discussion
This method is optional. It is invoked after Image Kit determines that a drag should begin, but before the drag has been started.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

numberOfGroupsInImageBrowser:

Returns the number of groups in an image browser view. (required)

```
- (NSUInteger) numberOfGroupsInImageBrowser:(IKImageBrowserView *) aBrowser;
```

Parameters
aBrowser
An image browser view.

Return Value
The number of groups.

Discussion
This method is optional.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

numberOfItemsInImageBrowser:

Returns the number of records managed by the data source object. (required)

```
- (NSUInteger) numberOfItemsInImageBrowser:(IKImageBrowserView *) aBrowser;
```

Parameters

aBrowser

An image browser view.

Return Value
The number of records managed by the image browser view.

Discussion
Your data source must implement this method. An `IKImageView` object uses this method to determine how many cells it should create and display.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

IKImageBrowserDelegate Protocol Reference

(informal protocol)

Adopted by	IKImageBrowserView
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Declared in	ImageKit/IKImageBrowserView.h

Overview

The `IKImageBrowserDelegate` is an informal protocol for the delegate of an `IKImageBrowserView` object. You can implement these methods to perform custom tasks when in response to events in the image browser view.

Tasks

Performing Custom Tasks in Response to User Events

- `imageBrowser:backgroundWasRightClickedWithEvent:` (page 153) *required method*
  Performs custom tasks when the user right-clicks the image browser view background. (required)
- `imageBrowser:cellWasRightClickedAtIndex:withEvent:` (page 154) *required method*
  Performs custom tasks when the user right-clicks an item in the image browser view. (required)
- `imageBrowser:cellWasDoubleClickedAtIndex:` (page 154) *required method*
  Performs custom tasks when the user double-clicks an item in the image browser view. (required)
- `imageBrowserSelectionDidChange:` (page 155) *required method*
  Performs custom tasks when the selection changes. (required)

Instance Methods

imageBrowser:backgroundWasRightClickedWithEvent:

Performs custom tasks when the user right-clicks the image browser view background. (required)

```
- (void) imageBrowser:(IKImageBrowserView *) aBrowser
    backgroundWasRightClickedWithEvent:(NSEvent *) event;
```

Parameters

aBrowser

An image browser view.

event

The event that invoked the method.

Discussion

This method signals that the user either right-clicked the background or left-clicked it with the Alt key pressed. You can implement this method if you want to perform custom tasks at that time.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`IKImageBrowserView.h`

imageBrowser:cellWasDoubleClickedAtIndex:

Performs custom tasks when the user double-clicks an item in the image browser view. (required)

```
- (void) imageBrowser:(IKImageBrowserView *) aBrowser
    cellWasDoubleClickedAtIndex:(NSUInteger) index;
```

Parameters

aBrowser

An image browser view.

index

The index of the cell.

Discussion

This method signals that the user double-clicked an item in the image browser view. You can implement this method if you want to perform custom tasks at that time.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`IKImageBrowserView.h`

imageBrowser:cellWasRightClickedAtIndex:withEvent:

Performs custom tasks when the user right-clicks an item in the image browser view. (required)

```
- (void) imageBrowser:(IKImageBrowserView *) aBrowser
    cellWasRightClickedAtIndex:(NSUInteger) index withEvent:(NSEvent *) event;
```

Parameters

aBrowser

An image browser view.

index

The index of the cell.

event
The event that invoked the method.

Discussion
This method signals that the user either right-clicked an item in the browser or left-clicked the item with the Alt key pressed. You can implement this method if you want to perform custom tasks at that time.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

imageBrowserSelectionDidChange:

Performs custom tasks when the selection changes. (required)

```
- (void) imageBrowserSelectionDidChange:(IKImageBrowserView *) aBrowser;
```

Parameters
aBrowser
An image browser view.

Discussion
This method signals that the user changes the selection in the image browser view. You can implement this method if you want to perform custom tasks at that time.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

IKImageBrowserItem Protocol Reference

(informal protocol)

Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Declared in	ImageKit/IKImageBrowserView.h

Overview

The `IKImageBrowserItem` informal protocol declares the methods that an instance of the `IKImageBrowserView` class uses to access the contents of its data source for a given item. Some of the methods in this protocol are needed frequently, so you should implement them efficiently.

Tasks

Providing Required Information for an Image

- `imageUID` (page 159)
  Returns a unique string that identifies the data source item.
- `imageRepresentationType` (page 158)
  Returns the representation type of the image to display.
- `imageRepresentation` (page 158)
  Returns the image to display.

Providing Optional Information for an Image

- `imageVersion` (page 160)
  Returns the version of the item.
- `imageTitle` (page 159)
  Returns the display title of the image.
- `imageSubtitle` (page 159)
  Returns the display subtitle of the image.
- `isSelectable` (page 160)
  Returns whether this item is selectable.

Instance Methods

imageRepresentation

Returns the image to display.

```
- (id) imageRepresentation;
```

Return Value
The image to display; can return `nil` if the item has no image to display.

Discussion
Your data source must implement this method. This method is called frequently, so the receiver should cache the returned instance.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
CocoaCreateMovie
DesktopImage
ImageBrowser
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserView.h`

imageRepresentationType

Returns the representation type of the image to display.

```
- (NSString *) imageRepresentationType;
```

Return Value
A string that specifies the image representation type. The string can be any of the constants defined in "Image Representation Types" (page 161).

Discussion
Your data source must implement this method.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
DesktopImage
Image Kit with Core Data
ImageBrowser
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserView.h`

imageSubtitle

Returns the display subtitle of the image.

```
- (NSString *) imageSubtitle
```

Return Value
The display subtitle of the image.

Discussion
This method is optional.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
ImageBrowserViewAppearance

Declared In
`IKImageBrowserView.h`

imageTitle

Returns the display title of the image.

```
- (NSString *) imageTitle;
```

Return Value
The display title of the image.

Discussion
This method is optional.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
DesktopImage
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserView.h`

imageUID

Returns a unique string that identifies the data source item.

```
- (NSString *)  imageUID;
```

Return Value
The string that identifies the data source item

Discussion
Your data source must implement this method. The image browser view uses this identifier to associate the data source item and its cache.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
DesktopImage
Image Kit with Core Data
ImageBrowser
ImageBrowserViewAppearance
ImageKitDemo

Declared In
`IKImageBrowserView.h`

imageVersion

Returns the version of the item.

```
- (NSUInteger) imageVersion;
```

Return Value
The version of the item.

Discussion
This method is optional. The receiver can return a new version to let the image browser knows that it should not use its cache for the item.

Availability
Available in Mac OS X v10.5 and later.

Related Sample Code
Image Kit with Core Data

Declared In
`IKImageBrowserView.h`

isSelectable

Returns whether this item is selectable.

```
- (BOOL) isSelectable;
```

Return Value
`YES` if the item is selectable; `NO` otherwise.

Discussion
This method is optional. You can prevent selection of this item by returning `NO`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`IKImageBrowserView.h`

Constants

Image Representation Types

Representation types for images.

```
NSString * const IKImageBrowserPathRepresentationType;
NSString * const IKImageBrowserNSURLRepresentationType;
NSString * const IKImageBrowserNSImageRepresentationType;
NSString * const IKImageBrowserCGImageRepresentationType;
NSString * const IKImageBrowserCGImageSourceRepresentationType;
NSString * const IKImageBrowserNSDataRepresentationType;
NSString * const IKImageBrowserNSBitmapImageRepresentationType;
NSString * const IKImageBrowserQTMovieRepresentationType;
NSString * const IKImageBrowserQTMoviePathRepresentationType;
NSString * const IKImageBrowserQCCompositionRepresentationType;
NSString * const IKImageBrowserQCCompositionPathRepresentationType;
NSString * const IKImageBrowserQuickLookPathRepresentationType;
NSString * const IKImageBrowserIconRefPathRepresentationType;
NSString * const IKImageBrowserIconRefRepresentationType;
NSString * const IKImageBrowserPDFPageRepresentationType;
```

Constants

`IKImageBrowserPathRepresentationType`
A path representation (`NSString`).
Available in Mac OS X v10.5 and later.
Declared in `IKImageBrowserView.h`.

`IKImageBrowserNSURLRepresentationType`
An `NSURL`object.
Available in Mac OS X v10.5 and later.
Declared in `IKImageBrowserView.h`.

`IKImageBrowserNSImageRepresentationType`
An `NSImage` object.
Available in Mac OS X v10.5 and later.
Declared in `IKImageBrowserView.h`.

`IKImageBrowserCGImageRepresentationType`
A `CGImageRef` object.
Available in Mac OS X v10.5 and later.
Declared in `IKImageBrowserView.h`.

`IKImageBrowserCGImageSourceRepresentationType`
A `CGImageSourceRef` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserNSDataRepresentationType`
An `NSData` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserNSBitmapImageRepresentationType`
An `NSBitmapImageRep` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserQTMovieRepresentationType`
A `QTMovie` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserQTMoviePathRepresentationType`
A path (NSString) or URL (NSURL) to a QuickTime movie.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserQCCompositionRepresentationType`
A `QCComposition` object.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserQCCompositionPathRepresentationType`
A path (NSString) or URL (NSURL) to a Quartz Composer composition.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserQuickLookPathRepresentationType`
A path (NSString) or URL (NSURL) to load data using QuickLook.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserIconRefPathRepresentationType`
A path to an icon.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserIconRefRepresentationType`
An icon.

Available in Mac OS X v10.5 and later.

Declared in `IKImageBrowserView.h`.

`IKImageBrowserPDFPageRepresentationType`
A `PDFPage` instance or a `CGPDFPageRef`.

Available in Mac OS X v10.6 and later.

Declared in `IKImageBrowserView.h`.

Declared In
`IKImageBrowserView.h`

IKImageEditPanelDataSource Protocol Reference

Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKImageEditPanel.h
Companion guide	Image Kit Programming Guide

Overview

The `IKImageEditPanelDataSource` protocol describes the methods that an `IKImageEditPanel` object uses to access the contents of its data source object.

Tasks

Getting and Setting Image Properties

- `imageProperties` (page 167)
  Returns a dictionary of the image properties associated with the image in the image edit panel.
- `setImage:imageProperties:` (page 167) *required method*
  Sets an image with the specified properties. (required)

Getting Images From the Data Source

- `image` (page 167) *required method*
  Returns an image. (required)
- `thumbnailWithMaximumSize:` (page 168)
  Returns a thumbnail image whose size is no larger than the specified size.

New Methods

- `hasAdjustMode` (page 166)
  Returns whether the adjust mode view tab should be displayed.

Instance Methods

hasAdjustMode

Returns whether the adjust mode view tab should be displayed.

```
- (BOOL)hasAdjustMode
```

Return Value
YES if the tab should be displayed, otherwise NO.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageEditPanel.h

hasDetailsMode

Returns whether the details mode view tab should be displayed.

```
- (BOOL)hasDetailsMode
```

Return Value
YES if the tab should be displayed, otherwise NO.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageEditPanel.h

hasEffectsMode

Returns whether the effects mode view tab should be displayed.

```
- (BOOL)hasEffectsMode
```

Return Value
YES if the tab should be displayed, otherwise NO.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKImageEditPanel.h

image

Returns an image. (required)

```
- (CGImageRef)image
```

Return Value
An image.

Discussion
Your data source must implement this method.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
IKImageEditPanel.h

imageProperties

Returns a dictionary of the image properties associated with the image in the image edit panel.

```
- (NSDictionary *)imageProperties
```

Return Value
A dictionary that contains the properties of the image.

Availability
Available in Mac OS X v 10.5 and later.

See Also
- setImage:imageProperties (page ?)

Declared In
IKImageEditPanel.h

setImage:imageProperties:

Sets an image with the specified properties. (required)

```
- (void)setImage:(CGImageRef)image imageProperties:(NSDictionary *)metaData
```

Discussion
Your data source must implement this method.

Availability
Available in Mac OS X v 10.5 and later.

See Also
- imageProperties (page ?)

Declared In
`IKImageEditPanel.h`

thumbnailWithMaximumSize:

Returns a thumbnail image whose size is no larger than the specified size.

```
- (CGImageRef)thumbnailWithMaximumSize:(NSSize)size
```

Return Value
An image.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
`IKImageEditPanel.h`

IKScannerDeviceViewDelegate Protocol Reference

Framework	/System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.6 and later.
Declared in	ImageKit/IKScannerDeviceView.h

Overview

The `IKScannerDeviceViewDelegate` protocol defines the delegate protocol that the `IKScannerDeviceView` delegate must conform to.

Tasks

Scan Completed

`- scannerDeviceView:didScanToURL:fileData:error:` (page 170)
Invoked when the scan has completed and the data is available.

Scanner Encountered Error

`- scannerDeviceView:didEncounterError:` (page 169)
Invoked whenever the scanner encounters an error.

Instance Methods

scannerDeviceView:didEncounterError:

Invoked whenever the scanner encounters an error.

```
- (void)scannerDeviceView:(IKScannerDeviceView *)scannerDeviceView
    didEncounterError:(NSError *)error
```

Parameters

scannerDeviceView

The scanner device that sent the message.

error

The error.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKScannerDeviceView.h

scannerDeviceView:didScanToURL:fileData:error:

Invoked when the scan has completed and the data is available.

```
- (void)scannerDeviceView:(IKScannerDeviceView *)scannerDeviceView
    didScanToURL:(NSURL *)url fileData:(NSData *)data error:(NSError *)error
```

Parameters

scannerDeviceView

The scanner device that sent the message.

url

The URL to save the data.

data

The data from the scan.

error

Any error encountered during the scan.

Discussion
This method is called when the scan has completed..

If the *scannerDeviceView* `transferMode` (page 126) is `IKScannerDeviceViewTransferModeFileBased` (page 127), the scan will have been saved at the specified *url*. The URL will be in the download directory and be a complete path, including a 'sequence number' if the file already exists.

If the *scannerDeviceView*`transferMode` (page 126) is `IKScannerDeviceViewTransferModeMemoryBased` (page 127), the scanned data is contained in the data parameter. You can then take the action appropriate to your application.

In case of an error, the parameters (*url* and *data*) will be `NULL` and *error* (which may come directly from the scanner module / or the ImageCaptureCore framework) will describe why the scan or save failed.

Availability
Available in Mac OS X v10.6 and later.

Declared In
IKScannerDeviceView.h

IKSlideshowDataSource Protocol Reference

Adopted by	IKSlideshow
Framework	System/Library/Frameworks/Quartz.framework/ImageKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ImageKit/IKSlideShow.h

Overview

The `IKSlideshowDataSource` protocol describes the methods that an `IKSlideshow` object uses to access the contents of its data source object.

Important: Slide show data source methods may be called on secondary threads. When you implement these methods, you must ensure that they are safe to run on threads other than the main thread.

Tasks

Providing Slideshow Information

- `numberOfSlideshowItems` (page 172) *required method*
  Returns the number of items in a slideshow. (required)
- `slideshowItemAtIndex:` (page 174) *required method*
  Returns the item for a given index (required)
- `nameOfSlideshowItemAtIndex:` (page 172)
  Returns the display name for item at the specified index.
- `canExportSlideshowItemAtIndex:toApplication:` (page 172)
  Reports whether the export button should be enabled for a a slideshow item.

Performing Custom Tasks

- `slideshowWillStart` (page 174)
  Performs custom tasks when the slideshow is about to start.

Instance Methods

canExportSlideshowItemAtIndex:toApplication:

Reports whether the export button should be enabled for a a slideshow item.

```
- (BOOL)canExportSlideshowItemAtIndex:(NSUInteger)index toApplication:(NSString
  *)applicationBundleIdentifier
```

Return Value
YES if the export button should be enabled for an item; otherwise NO.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
IKSlideshow.h

nameOfSlideshowItemAtIndex:

Returns the display name for item at the specified index.

```
- (NSString *)nameOfSlideshowItemAtIndex:(NSUInteger)index
```

Parameters
index
The index for a slideshow item.

Return Value
The display name. For the best user experience, you should provide the localized name, because this string appears in the user interface.

Discussion
This method is optional.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
IKSlideshow.h

numberOfSlideshowItems

Returns the number of items in a slideshow. (required)

```
- (NSUInteger)numberOfSlideshowItems
```

Return Value
The number of items in the slideshow.

Discussion
Your data source must implement this method.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
IKSlideshow.h

slideshowDidChangeCurrentIndex:

Performs custom tasks when the slideshow changes to the item at the specified index.

```
- (void)slideshowDidChangeCurrentIndex:(NSUInteger)newIndex
```

Parameters
newIndex
The index of the current item.

Discussion
Image Kit invokes this method when the slideshow changes to the specified item. Implement this method to perform custom tasks at that time.

Availability
Available in Mac OS X v 10.5 and later.

Declared In
IKSlideshow.h

slideshowDidStop

Performs custom tasks when the slideshow stops.

```
- (void)slideshowDidStop
```

Discussion
TImage Kit invokes this method when the slideshow stops. Implement this method to perform custom tasks at that time.

Availability
Available in Mac OS X v 10.5 and later.

See Also
- slideshowWillStart (page ?)

Declared In
IKSlideshow.h

slideshowItemAtIndex:

Returns the item for a given index (required)

```
- (id)slideshowItemAtIndex:(NSUInteger)index
```

Parameters

index

An index of an item in the slideshow.

Return Value

The object that corresponds to the item at the specified index. The item can be any of the following objects: `NSImage`, `NSString` (to specify a path name), `NSURL`, `NSFileWrapper`, `CGImageRef`, or `PDFPage`.

Discussion

Your data source must implement this method.

Availability

Available in Mac OS X v 10.5 and later.

Declared In

`IKSlideshow.h`

slideshowWillStart

Performs custom tasks when the slideshow is about to start.

```
- (void)slideshowWillStart
```

Discussion

Image Kit invokes this method when the slideshow is about to start. Implement this method to perform custom tasks at that time.

Availability

Available in Mac OS X v 10.5 and later.

See Also

`- slideshowDidStop` (page ?)

Declared In

`IKSlideshow.h`

Document Revision History

This table describes the changes to *Image Kit Reference Collection*.

Date	Notes
2010-03-24	Added IKCameraDeviceView and delegate, IKScannerDeviceView and delegate, IKDeviceBrowserView and delegate.
2006-12-06	New collection that describes the Objective-C API for providing a user interface for images, image editing, and image processing.

Document Revision History