# IKImageBrowserView Class Reference

Graphics & Animation: 2D Drawing

2010-03-24

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

# IKImageBrowserView Class Reference

| | |
|---|---|
| **Inherits from** | NSView : NSResponder : NSObject |
| **Conforms to** | NSAnimatablePropertyContainer (NSView)<br>NSCoding (NSResponder)<br>NSObject (NSObject) |
| **Framework** | /System/Library/Frameworks/Quartz.framework/ImageKit.framework |
| **Availability** | Available in Mac OS X v10.5 and later. |
| **Declared in** | IKImageBrowserView.h |
| **Related sample code** | AutomatorHandsOn<br>DesktopImage<br>ImageBrowser<br>ImageBrowserViewAppearance<br>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:` and `imageBrowser:itemAtIndex:` 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 17)
Initializes a newly allocated image browser view with the provided frame rectangle.

## Updating the Display of the Content

– `reloadData` (page 20)
Marks the receiver as needing its data reloaded.

## Getting and Setting the Delegate

– `setDelegate:` (page 26)
Sets the delegate of the receiver.

– `delegate` (page 15)
Returns the delegate of the receiver.

## Getting and Setting the Data Source

– `setDataSource:` (page 26)
Sets the data source of the receiver.

– `dataSource` (page 14)
Returns the data source of the receiver.

## Setting the Appearance

– `setCellsStyleMask:` (page 25)
Defines the appearance style of the cells.

– `cellsStyleMask` (page 13)
Returns the appearance style mask for the cell.

– `setConstrainsToOriginalSize:` (page 25)
Sets whether the receiver constrains the cell's image to its original size.

– `constrainsToOriginalSize` (page 14)
Returns whether the receiver constrains the cell's image to its original size.

– `setIntercellSpacing:` (page 28)
Sets the spacing between cells in the view.

– `intercellSpacing` (page 18)
Returns the spacing between cells in the view.

## Creating a Custom Cell for an Item

– `newCellForRepresentedItem:` (page 19)
Returns the cell to use for the specified item.

## Zooming and Resizing

- `setZoomValue:` (page 29)
  Sets the zoom value.
- `zoomValue` (page 30)
  Returns the current zoom value.
- `setContentResizingMask:` (page 25)
  Determines how the receiver resizes its content when zooming.
- `contentResizingMask` (page 14)
  Returns the receiver's content resizing mask, which determines how its content is resized while zooming.

## Scrolling

- `scrollIndexToVisible:` (page 21)
  Scrolls the receiver to the item at the specified index.

## Setting and Getting Cell Size

- `setCellSize:` (page 24)
  Sets the cell size.
- `cellSize` (page 12)
  Returns the cell size.

## Getting Item Information

- `indexOfItemAtPoint:` (page 17)
  Returns the index of the item at the specified location.
- `itemFrameAtIndex:` (page 18)
  Returns the frame rectangle for the item located at the specified index.
- `visibleItemIndexes` (page 29)
  Returns the indexes of the view's currently visible items.
- `cellForItemAtIndex:` (page 12)
  Returns the browser cell for the item at the specified index.

## Reordering and Groups Items

- `selectionIndexes` (page 21)
  Returns the indexes of the selected cells.
- `setSelectionIndexes:byExtendingSelection:` (page 28)
  Selects cells at the specified indexes.
- `setAllowsMultipleSelection:` (page 22)
  Controls whether the user can select more than one cell at a time.

- `allowsMultipleSelection` (page 10)
  Returns whether multiple selections are allowed.
- `setAllowsEmptySelection:` (page 22)
  Controls whether an empty selection is allowed.
- `allowsEmptySelection` (page 10)
  Returns whether an empty selection is allowed.
- `setAllowsReordering:` (page 23)
  Controls whether the user can reorder items.
- `allowsReordering` (page 10)
  Returns whether the user can reorder items.
- `setAnimates:` (page 23)
  Controls whether the receiver animates reordering and changes of the data source.
- `animates` (page 11)
  Returns whether the receiver animates reordering and changes of the data source.
- `expandGroupAtIndex:` (page 16)
  Expands a group at the specified index.
- `collapseGroupAtIndex:` (page 13)
  Collapses a group at the specified index.
- `isGroupExpandedAtIndex:` (page 18)
  Returns whether the group at the provided index is expanded.

## Supporting Drag and Drop

- `setDraggingDestinationDelegate:` (page 27)
  Sets the dragging destination delegate of the receiver.
- `draggingDestinationDelegate` (page 15)
  Returns the dragging destination delegate of the receiver.
- `setDropIndex:dropOperation:` (page 27)
  Allows the class to retarget the drop action.
- `indexAtLocationOfDroppedItem` (page 17)
  Returns the index of the cell where the drop operation occurred.
- `setAllowsDroppingOnItems:` (page 22)
  Specifies whether the user can drop on items.
- `allowsDroppingOnItems` (page 9)
  Returns whether the user can drop on items.
- `dropOperation` (page 15)
  Returns the current drop operation.

## Core Animation Layer Integration

- `setForegroundLayer:` (page 27)
  The Core Animation layer used as the foreground overlay.

- `foregroundLayer` (page 16)
  Returns the foreground Core Animation layer
- `setBackgroundLayer:` (page 23)
  The Core Animation layer used as the view's background.
- `backgroundLayer` (page 11)
  Returns the foreground Core Animation layer

### QuickLook Support

- `setCanControlQuickLookPanel:` (page 24)
  Specifies whether the view can automatically take control of the QuickLook panel.
- `canControlQuickLookPanel` (page 11)
  Returns whether the view can automatically take control of the QuickLook panel.

### Getting Columns and Rows Information

- `numberOfColumns` (page 19)
  Returns the current number of columns.
- `numberOfRows` (page 19)
  Returns the current number of rows.
- `rectOfColumn:` (page 20)
  Returns the rectangle containing the specified column.
- `columnIndexesInRect:` (page 13)
  Returns the column indexes in the specified rectangle.
- `rectOfRow:` (page 20)
  Returns the rectangle containing the specified row.
- `rowIndexesInRect:` (page 21)
  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 22)

**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 22)

**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 23)

**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 23)

**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 23)

**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 24)

**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 24)

**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 25)

**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 16)
`- isGroupExpandedAtIndex:` (page 18)

**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 25)

### 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 25)

### 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 26)

**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 26)

**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 27)

**Declared In**
IKImageBrowserView.h

## dropOperation

Returns the current drop operation.

```
- (IKImageBrowserDropOperation)dropOperation
```

**Return Value**
IKImageBrowserDropOn (page 34) if the drop occurs on an item, otherwise IKImageBrowserDropBefore (page 34).

**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 34) 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 34) 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 13)
`- isGroupExpandedAtIndex:` (page 18)

**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 27)

**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 16)
- `– collapseGroupAtIndex:` (page 13)

### 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 28)

**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 10)

### 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 10)

**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 10)

**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 11)

**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 11)

**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 11)

**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 29), 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 12)

- setZoomValue: (page 29)

### 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 30) using the C bitwise `OR` operator.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- cellsStyleMask (page 13)

### 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 14)

### 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 14)

**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 14)

**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 15)

**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 15)

### 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 33).

### Discussion

For example, To specify a drop on the second item, one would specify index as `1`, and operation as `IKImageBrowserDropOn` (page 34). To specify a drop after the last item, one would specify index as the number of items and operation as `IKImageBrowserDropBefore` (page 34).

Passing a value of `-1` for *index*, and `IKImageBrowserDropOn` (page 34) 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 16)

**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 21)

**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 24), 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 30)
`- setCellSize:` (page 24)

**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 29)

**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 31).

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 27).

```
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`

# Document Revision History

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

| Date | Notes |
|---|---|
| 2010-03-24 | Updated for Mac OS X v 10.6. |
| 2006-12-13 | New document that describes the class that can display and browse a large amount of images and movies in an efficient manner. |