
UICollectionView Class Reference

User Experience: Windows & Views



2009-11-17



Apple Inc.
© 2009 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

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

Apple, the Apple logo, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

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

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

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

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

Contents

UICollectionView Class Reference 5

Overview	5
Tasks	5
Modifying the Collection View Item	5
Working with the Responder Chain	6
Setting the Content	6
Setting the Selection Mode	6
Laying out the Collection View	6
Modifying the Background	7
Getting and Setting the Delegate	7
Drag and Drop Support	7
Getting a Collection Item and Its Frame	7
Instance Methods	7
allowsMultipleSelection	7
backgroundColors	8
content	8
delegate	8
draggingImageForItemsAtIndexes:withEvent:offset:	9
frameForItemAtIndex:	9
isFirstResponder	10
isSelectable	10
itemAtIndex:	10
itemPrototype	11
maxItemSize	11
maxNumberOfColumns	11
maxNumberOfRows	12
minItemSize	12
newItemForRepresentedObject:	12
selectionIndexes	13
setAllowsMultipleSelection:	13
setBackgroundColors:	13
setContent:	14
setDelegate:	14
setDraggingSourceOperationMask:forLocal:	15
setItemPrototype:	15
setMaxItemSize:	15
setMaxNumberOfColumns:	16
setMaxNumberOfRows:	16
setMinItemSize:	17
setSelectable:	17
setSelectionIndexes:	17

Constants 18
 NSCollectionViewDropOperation 18

Document Revision History 19

UICollectionView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.5 and later.
Companion guide	Collection View Programming Guide
Declared in	UICollectionView.h
Related sample code	IconCollection Reviews

Overview

`UICollectionView` class displays an array of content as a grid of views. The views are specified using the `UICollectionViewItem` class which makes loading nibs containing the view easy, and supports bindings.

Tasks

Modifying the Collection View Item

- [setItemPrototype:](#) (page 15)
Sets the receiver's item prototype to the specified collection view item.
- [itemPrototype](#) (page 11)
Returns the receiver's collection view item prototype.
- [newItemForRepresentedObject:](#) (page 12)
Returns the collection view item that is used for the specified object.

Working with the Responder Chain

- [isFirstResponder](#) (page 10)
Returns whether the receiver is the first responder.

Setting the Content

- [setContent:](#) (page 14)
Sets the receiver's content array.
- [content](#) (page 8)
Returns the receiver's content object.

Setting the Selection Mode

- [setSelectable:](#) (page 17)
Controls whether the receiver allows the user to select items.
- [isSelectable](#) (page 10)
Returns a Boolean value that indicates whether the receiver allows the user to select items, NO if it doesn't.
- [setAllowsMultipleSelection:](#) (page 13)
Controls whether the user can select multiple items at a time.
- [allowsMultipleSelection](#) (page 7)
Returns a Boolean value that indicates whether the receiver allows the user to select more than one item at a time.
- [setSelectionIndexes:](#) (page 17)
Sets the receiver's selection using the specified indexes.
- [selectionIndexes](#) (page 13)
Returns an index set containing the indexes of the receiver's currently selected objects in the content array.

Laying out the Collection View

- [setMaxNumberOfRows:](#) (page 16)
Sets the maximum number of rows the receiver will display.
- [maxNumberOfRows](#) (page 12)
Returns the maximum number of rows the receiver will display.
- [setMaxNumberOfColumns:](#) (page 16)
Sets the maximum number of columns the receiver will display
- [maxNumberOfColumns](#) (page 11)
Returns the maximum number of columns the receiver will display.
- [setMinItemSize:](#) (page 17)
Sets the minimum size used to display individual layout items in the grid.
- [minItemSize](#) (page 12)
Returns the minimum size used to display individual collection view items in the grid.

- [setMaxItemSize:](#) (page 15)
Sets the maximum size used to display individual collection view items in the grid.
- [maxItemSize](#) (page 11)
Returns the maximum size used to display individual collection view items in the grid

Modifying the Background

- [setBackgroundColors:](#) (page 13)
Sets the receiver's background colors to the specified array of colors.
- [backgroundColors](#) (page 8)
Return the receiver's background colors.

Getting and Setting the Delegate

- [delegate](#) (page 8)
Returns the receiver's delegate.
- [setDelegate:](#) (page 14)
Sets the receiver's delegate.

Drag and Drop Support

- [draggingImageForItemsAtIndexes:withEvent:offset:](#) (page 9)
This method computes and returns an image to use for dragging.
- [setDraggingSourceOperationMask:forLocal:](#) (page 15)
Configures the default value returned from `draggingSourceOperationMaskForLocal:`.

Getting a Collection Item and Its Frame

- [itemAtIndex:](#) (page 10)
Returns the collection view item for the represented object at the specified index.
- [frameForItemAtIndex:](#) (page 9)
Returns the frame of the collection view item at the specified index.

Instance Methods

allowsMultipleSelection

Returns a Boolean value that indicates whether the receiver allows the user to select more than one item at a time.

- (BOOL)allowsMultipleSelection

Return Value

YES if the receiver allows the user to select more than one column or row at a time, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

backgroundColors

Return the receiver's background colors.

- (NSArray *)backgroundColors

Return Value

Returns an array containing the receiver's background colors.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

content

Returns the receiver's content object.

- (NSArray *)content

Return Value

An array containing the receiver's content.

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

IconCollection

Declared In

NSCollectionView.h

delegate

Returns the receiver's delegate.

- (id < NSCollectionViewDelegate >)delegate

Return Value

The receiver's delegate object.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSCollectionView.h

draggingImageForItemsAtIndexes:withEvent:offset:

This method computes and returns an image to use for dragging.

```
- (NSImage *)draggingImageForItemsAtIndexes:(NSIndexSet *)indexes withEvent:(NSEvent *)event offset:(NSPointPointer)dragImageOffset
```

Parameters

indexes

The index set of the items to be dragged.

event

Mouse drag event.

dragImageOffset

An in/out parameter that will initially be set to `NSZeroPoint`. It can be modified to reposition the returned image. A *dragImageOffset* of `NSZeroPoint` will cause the image to be centered under the mouse.

Return Value

An image containing a rendering of the visible portions of the views for each item.

Discussion

You can override the default image by subclassing `NSCollectionView` and overriding this method, or by implementing the `collectionView:draggingImageForItemsAtIndexes:withEvent:offset:delegate` method, it will be preferred over this method.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSCollectionView.h

frameForItemAtIndex:

Returns the frame of the collection view item at the specified index.

```
- (NSRect)frameForItemAtIndex:(NSUInteger)index
```

Parameters

index

The index of the collection view item.

Return Value

Returns the frame calculated by the receiver where it intends to place the subview for the `NSCollectionViewItem` at the given index. The rectangle is returned in the receiver's coordinate system.

Discussion

You can use this method in the `collectionView:draggingImageForItemsAtIndexes:withEvent:offset:` method to determine which views are in the visible portion of the enclosing scroll view.

Overriding this method will have no effect on the receiver's subview layout.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSCollectionView.h

isFirstResponder

Returns whether the receiver is the first responder.

- (BOOL)isFirstResponder

Return Value

YES if the receiver is the first responder, otherwise NO.

Special Considerations

This method is fully key-value observing compliant.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

isSelectable

Returns a Boolean value that indicates whether the receiver allows the user to select items, NO if it doesn't.

- (BOOL)isSelectable

Return Value

YES if the receiver allows the user to select items, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

itemAtIndex:

Returns the collection view item for the represented object at the specified index.

- (NSCollectionViewItem *)itemAtIndex:(NSUInteger) *index*

Parameters

index

The index of the collection view item.

Return Value

An instance of `NSCollectionViewItem`.

Discussion

Rather than using the `NSCollectionViewItem` instance returned by this method to determine the frame of the collection item's view you should use [frameForItemAtIndex:](#) (page 9), it is significantly more efficient.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSCollectionView.h`

itemPrototype

Returns the receiver's collection view item prototype.

- (`NSCollectionViewItem *`)itemPrototype

Return Value

The receiver's collection view item prototype.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSCollectionView.h`

maxItemSize

Returns the maximum size used to display individual collection view items in the grid

- (`NSSize`)maxItemSize

Return Value

The maximum size, measured in points, used to display individual collection view items.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSCollectionView.h`

maxNumberOfColumns

Returns the maximum number of columns the receiver will display.

- (`NSUInteger`)maxNumberOfColumns

Return Value

The maximum number of columns the receiver will display.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

numberOfColumns

Returns the maximum number of columns the receiver will display.

- (NSUInteger)numberOfColumns

Return Value

The maximum number of rows the receiver will display.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

numberOfRows

Returns the minimum size used to display individual collection view items in the grid.

- (NSSize)numberOfRows

Return Value

The minimum size, measured in points, used to display individual collection view items.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

initWithItemForRepresentedObject:

Returns the collection view item that is used for the specified object.

- (NSCollectionViewItem *)initWithItemForRepresentedObject:(id)object

Parameters

object

The content object that the collection view item will represent.

Return Value

An initialized collection view item with the specified object and the appropriate view set. The collection view item should not be autoreleased.

Discussion

Subclasses can override this method if the collection view items are not generated from a prototype or if the prototype view needs to be modified. The subclass is responsible for setting the `view` and `representedObject` of the new collection view item.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

selectionIndexes

Returns an index set containing the indexes of the receiver's currently selected objects in the content array.

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

Return Value

An index set containing the indexes of the receiver's currently selected objects in the content array.

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setAllowsMultipleSelection:

Controls whether the user can select multiple items at a time.

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

Parameters

flag

YES to allow the user to select multiple items, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setBackgroundColors:

Sets the receiver's background colors to the specified array of colors.

```
- (void)setBackgroundColors:(NSArray *)colors
```

Parameters*colors*

An array containing the background colors for the receiver.

Discussion

Passing an empty array or *nil* resets the background colors to their default values provided by `controlAlternatingRowBackgroundColors`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setContent:

Sets the receiver's content array.

```
- (void)setContent:(NSArray *)content
```

Parameters*content*

An array containing the receiver's content.

Discussion

The content array can also be provided by creating a binding between the receiver's `NSContentBinding` and an array controller's `arrangedObjects` method.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id < NSCollectionViewDelegate >)aDelegate
```

Parameters*aDelegate*

The delegate object for the receiver. The delegate must conform to the `NSCollectionViewDelegate` protocol.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSCollectionView.h

setDraggingSourceOperationMask:forLocal:

Configures the default value returned from `draggingSourceOperationMaskForLocal:`.

```
- (void)setDraggingSourceOperationMask:(NSDragOperation)dragOperationMask
      forLocal:(BOOL)localDestination
```

Parameters

dragOperationMask

The types of drag operations allowed.

localDestination

If YES, mask applies when the drag destination object is in the same application as the receiver; if NO, mask applies when the destination object is outside the receiver's application.

Discussion

By default, this method returns `NSDragOperationEvery` when *localDestination* is YES and `NSDragOperationNone` when *localDestination* is NO. `NSCollectionView` will save the values you set for each *localDestination* value.

You typically will invoke this method, and not override it.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSCollectionView.h`

setItemPrototype:

Sets the receiver's item prototype to the specified collection view item.

```
- (void)setItemPrototype:(NSCollectionViewItem *)prototype
```

Parameters

prototype

The collection view item used as the prototype by the receiver.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSCollectionView.h`

setMaxItemSize:

Sets the maximum size used to display individual collection view items in the grid.

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

Parameters

size

The new maximum size, measured in points, with which to display individual collection view items.

Discussion

Setting the size to (0,0) specifies no maximum grid size. The default is (0,0). If the view in the receiver's collection view item prototype is resizable you should set this to the maximum size that the view should be displayed using.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setMaxNumberOfColumns:

Sets the maximum number of columns the receiver will display

```
- (void)setMaxNumberOfColumns:(NSUInteger)number
```

Parameters

number

The maximum number of columns the receiver will display.

Discussion

Setting to 0 specifies no maximum number of columns. Defaults to 0.

It is possible for a `NSCollectionView` instance to specify both the maximum number of rows and a maximum number of columns. If the number of content objects exceeds the number of displayable items ($n = \text{maxNumberOfRows} * \text{maxNumberOfColumns}$) only the first n items of the content array are displayed.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setMaxNumberOfRows:

Sets the maximum number of rows the receiver will display.

```
- (void)setMaxNumberOfRows:(NSUInteger)number
```

Parameters

number

The maximum number of rows the receiver can display.

Discussion

Setting to 0 specifies no maximum number of rows. Defaults to 0.

It is possible for a `NSCollectionView` instance to specify both the maximum number of rows and a maximum number of columns. If the number of content objects exceeds the number of displayable items ($n = \text{maxNumberOfRows} * \text{maxNumberOfColumns}$) only the first n items of the content array are displayed.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setMinItemSize:

Sets the minimum size used to display individual layout items in the grid.

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

Parameters*size*

The new minimum size, measured in points, with which to display individual layout items.

Discussion

The default is (0,0). If the view in the receiver's collection view item prototype is resizable you should set this to the minimum size that the view should be displayed using.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setSelectable:

Controls whether the receiver allows the user to select items.

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

Parameters*flag*

If flag is YES, the receiver allows the user to select items; if flag is NO, it doesn't.

Discussion

You can set selections programmatically regardless of this setting.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

setSelectionIndexes:

Sets the receiver's selection using the specified indexes.

```
- (void)setSelectionIndexes:(NSIndexSet *) indexes
```

Parameters*indexes*

The set of selection indexes for the receiver.

Discussion

To select all the receiver's objects, indexes should be an index set with indexes [0...count -1]. To deselect all indexes, pass an empty index set.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSCollectionView.h

Constants

NSCollectionViewDropOperation

These constants specify if acceptance of a drop should be at the item it is dropped on or before the item. These constants are used by the `collectionView:acceptDrop:index:dropOperation:` and `collectionView:validateDrop:proposedIndex:dropOperation:` methods in `NSCollectionViewDelegate` Protocol

```
enum {      NSCollectionViewDropOn = 0,      NSCollectionViewDropBefore = 1, };
typedef NSInteger NSCollectionViewDropOperation;
```

Constants

`NSCollectionViewDropOn`

The drop occurs at the collection view item to which the item was dragged.

Available in Mac OS X v10.6 and later.

Declared in `NSCollectionView.h`.

`NSCollectionViewDropBefore`

The drop occurs above the collection view item to which the item was dragged..

Available in Mac OS X v10.6 and later.

Declared in `NSCollectionView.h`.

Document Revision History

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

Date	Notes
2009-11-17	Corrected typos.
2009-05-26	Updated for Mac OS X v10.6. Added drag and drop support and moved delegates to <i>NSCollectionViewDelegate</i> protocol.
2007-12-11	Corrected typos.
2006-12-28	New document that describes the class used for creating groups of views from a prototype view.

REVISION HISTORY

Document Revision History