UICollectionViewLayout Class Reference

|  |  |
| --- | --- |
| **Inherits from** | [NSObject](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSObject_Class/Reference/Reference.html#//apple_ref/occ/cl/NSObject) |
| **Conforms to** | [NSCoding](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Protocols/NSCoding_Protocol/Reference/Reference.html#//apple_ref/occ/intf/NSCoding) [NSObject (NSObject)](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Protocols/NSObject_Protocol/Reference/NSObject.html#//apple_ref/occ/intf/NSObject) |
| **Framework** | /System/Library/Frameworks/[UIKit.framework](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIKit_Framework/_index.html" \l "//apple_ref/doc/uid/TP40006955) |
| **Availability** | Available in iOS 6.0 and later. |
| **Declared in** | UICollectionViewLayout.h |

## Overview

The UICollectionViewLayout class is an abstract base class that you subclass and use to generate layout information for a collection view. The job of a layout object is to determine the placement of cells, supplementary views, and decoration views inside the collection view’s bounds and to report that information to the collection view when asked. The collection view then applies the provided layout information to the corresponding views so that they can be presented onscreen.

类是一个用于生成collection view的布局信息的抽象基类。布局对象的工作是决定collection view边界以内的cell，补充视图和装饰视图的布局，并在collection view请求时报告这些信息。然后Collection view会将所提供的布局信息应用到相关的视图上，以至于它们可以显示在屏幕上。

You must subclass UICollectionViewLayout in order to use it. Before you consider subclassing, though, you should look at the [UICollectionViewFlowLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewFlowLayout_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewFlowLayout" \t "_self)class to see if it can be adapted to your layout needs.

你必须创建UICollectionViewLayout的子类以使用它。在你决定创建子类之前，你应该看看[UICollectionViewFlowLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewFlowLayout_class/Reference/Reference.html#//apple_ref/occ/cl/UICollectionViewFlowLayout)类是否适用于你的布局需求。

### Subclassing Notes

The main job of a layout object is to provide information about the position and visual state of items in the collection view. The layout object does not create the views for which it provides the layout. Those views are created by the collection view’s data source. Instead, the layout object defines the position and size of visual elements based on the design of the layout.

布局对象的主要工作是提供Collection view中的item的位置和可视化状态信息。布局对象不会创建其布局的视图。这些视图由Collection view的数据源创建。取而代之的，布局视图决定给予布局设计的可视元素的位置和大小。

Collection views have three types of visual elements that need to be laid out:

Collection view有三种可视元素需要布局：

* **Cells** are the main elements positioned by the layout. Each cell represents a single data item in the collection. A collection view can have a single group of cells or it can divide those cells into multiple sections. The layout object’s main job is to arrange the cells in the collection view’s content area.
* **Cell**是布局定位的主要元素。每一个cell代表集合中一个单一的数据项。Collection view可以有一组cell或者可以将这些cell分成多段。布局对象的主要工作是排列Collection view的内容区域内的所有cell。
* **Supplementary views** present data but are different than cells. Unlike cells, supplementary views cannot be selected by the user. Instead, you use supplementary views to implement things like header and footer views for a given section or for the entire collection view. Supplementary views are optional and their use and placement is defined by the layout object.
* **Supplementary view**展示数据但是与cell不同。不像cell，supplementary view不能被用户选中。相反，一般用supplementary view实现指定段落或整个collection view的头部和脚部视图之类的东西。Supplementary view是可选的，他们的使用和定位由布局对象决定。
* **Decoration views** are visual adornments that cannot be selected and are not inherently tied to the data of the collection view. Decoration views are another type of supplementary view. Like supplementary views, they are optional and their use and placement is defined by the layout object.
* **Decoration view是无法选中且与collection view的数据无关的可视化装饰物。Decoration view是另一种supplementary view。像supplementary view一样，他们是可选，其使用和定位由布局对象决定。**

The collection view asks its layout object to provide layout information for these elements at many different times. Every cell and view that appears on screen is positioned using information from the layout object. Similarly, every time items are inserted into or deleted from the collection view, additional layout occurs for the items being added or removed. However, the collection view always limits layout to the objects that are visible onscreen.

Collection view在许多不同的时候请求布局对象提供这些元素的布局信息。出现在屏幕上的每一个cell和视图使用来自布局对象的信息定位。类似的，每次有项目插入或从collection view中删除，这些项目对应的布局也需要添加或删除。但是，collection view总是限制只对屏幕内可见的对象布局。

#### Methods to Override

Every layout object should implement the following methods:

每一个布局对象都应该实现以下方法：

* [collectionViewContentSize](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/collectionViewContentSize)
* [layoutAttributesForElementsInRect:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForElementsInRect:)
* [layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForItemAtIndexPath:)
* [layoutAttributesForSupplementaryViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForSupplementaryViewOfKind:atIndexPath:) (if your layout supports supplementary views)
* [layoutAttributesForDecorationViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForDecorationViewOfKind:atIndexPath:) (if your layout supports decoration views)
* [shouldInvalidateLayoutForBoundsChange:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/shouldInvalidateLayoutForBoundsChange:)

These methods provide the fundamental layout information that the collection view needs to place contents on the screen. Of course, if your layout does not support supplementary or decoration views, do not implement the corresponding methods.

这些方法提供collection view用于在屏幕上防止内容的基本布局信息。当然，如果你的布局不支持supplementary或decoration视图，不要实现相关的方法。

When the data in the collection view changes and items are to be inserted or deleted, the collection view asks its layout object to update the layout information. Specifically, any item that is moved, added, or deleted must have its layout information updated to reflect its new location. For moved items, the collection view uses the standard methods to retrieve the item’s updated layout attributes. For items being inserted or deleted, the collection view calls some different methods, which you should override to provide the appropriate layout information:

当collection view中的数据改变，项目被插入或删除，collection view会请求其布局对象更新布局信息。特别的，任何项目移动、添加或删除都必须更新其布局信息到新的位置。对于移动的项目，collection view使用标准的方法检索其更新的布局属性。对于插入或删除的项目，collection view调用不同的方法，你应该重载这些方法以提供恰当的布局信息：

* [initialLayoutAttributesForAppearingItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingItemAtIndexPath:)
* [initialLayoutAttributesForAppearingSupplementaryElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingSupplementaryElementOfKind:atIndexPath:)
* [initialLayoutAttributesForAppearingDecorationElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingDecorationElementOfKind:atIndexPath:)
* [finalLayoutAttributesForDisappearingItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingItemAtIndexPath:)
* [finalLayoutAttributesForDisappearingSupplementaryElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingSupplementaryElementOfKind:atIndexPath:)
* [finalLayoutAttributesForDisappearingDecorationElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingDecorationElementOfKind:atIndexPath:)

In addition to these methods, you can also override the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) to handle any layout-related preparation. You can also override the[finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method and use it to add animations to the overall animation block or to implement any final layout-related tasks.

除了这些方法以为，你也可以重载[prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:)方法来控制布局相关的展示。你也可以重载[finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates)方法，并使用其添加动画到整个动画块，或者实现任何最终与布局相关的任务。

#### Optimizing Layout Performance Using Invalidation Contexts

When designing your custom layouts, you can improve performance by invalidating only those parts of your layout that actually changed. When you change items, calling the invalidateLayout method forces the collection view to recompute all of its layout information and reapply it. A better solution is to recompute only the layout information that changed, which is exactly what invalidation contexts allow you to do. An invalidation context lets you specify which parts of the layout changed. The layout object can then use that information to minimize the amount of data it recomputes.

To define a custom invalidation context for your layout, subclass the [UICollectionViewLayoutInvalidationContext](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutInvalidationContext_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayoutInvalidationContext" \t "_self) class. In your subclass, define custom properties that represent the parts of your layout data that can be recomputed independently. When you need to invalidate your layout at runtime, create an instance of your invalidation context subclass, configure the custom properties based on what layout information changed, and pass that object to your layout’s[invalidateLayoutWithContext:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/invalidateLayoutWithContext:) method. Your custom implementation of that method can use the information in the invalidation context to recompute only the portions of your layout that changed.

If you define a custom invalidation context class for your layout object, you should also override the [invalidationContextClass](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/clm/UICollectionViewLayout/invalidationContextClass) method and return your custom class. The collection view always creates an instance of the class you specify when it needs an invalidation context. Returning your custom subclass from this method ensures that your layout object always has the invalidation context it expects.

## Tasks

### Getting the Collection View Information

* [collectionView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instp/UICollectionViewLayout/collectionView)*property*
* [– collectionViewContentSize](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/collectionViewContentSize)

### Providing Layout Attributes

* [+ layoutAttributesClass](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/clm/UICollectionViewLayout/layoutAttributesClass)
* [– prepareLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareLayout)
* [– layoutAttributesForElementsInRect:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForElementsInRect:)
* [– layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForItemAtIndexPath:)
* [– layoutAttributesForSupplementaryViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForSupplementaryViewOfKind:atIndexPath:)
* [– layoutAttributesForDecorationViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForDecorationViewOfKind:atIndexPath:)
* [– targetContentOffsetForProposedContentOffset:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/targetContentOffsetForProposedContentOffset:)
* [– targetContentOffsetForProposedContentOffset:withScrollingVelocity:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/targetContentOffsetForProposedContentOffset:withScrollingVelocity:)

### Responding to Collection View Updates

* [– prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:)
* [– finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates)
* [– indexPathsToInsertForSupplementaryViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/indexPathsToInsertForSupplementaryViewOfKind:)
* [– indexPathsToInsertForDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/indexPathsToInsertForDecorationViewOfKind:)
* [– initialLayoutAttributesForAppearingItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingItemAtIndexPath:)
* [– initialLayoutAttributesForAppearingSupplementaryElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingSupplementaryElementOfKind:atIndexPath:)
* [– initialLayoutAttributesForAppearingDecorationElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/initialLayoutAttributesForAppearingDecorationElementOfKind:atIndexPath:)
* [– indexPathsToDeleteForSupplementaryViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/indexPathsToDeleteForSupplementaryViewOfKind:)
* [– indexPathsToDeleteForDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/indexPathsToDeleteForDecorationViewOfKind:)
* [– finalLayoutAttributesForDisappearingItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingItemAtIndexPath:)
* [– finalLayoutAttributesForDisappearingSupplementaryElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingSupplementaryElementOfKind:atIndexPath:)
* [– finalLayoutAttributesForDisappearingDecorationElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalLayoutAttributesForDisappearingDecorationElementOfKind:atIndexPath:)

### Invalidating the Layout

* [– invalidateLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/invalidateLayout)
* [– invalidateLayoutWithContext:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/invalidateLayoutWithContext:)
* [+ invalidationContextClass](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/clm/UICollectionViewLayout/invalidationContextClass)
* [– shouldInvalidateLayoutForBoundsChange:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/shouldInvalidateLayoutForBoundsChange:)
* [– invalidationContextForBoundsChange:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/invalidationContextForBoundsChange:)

### Coordinating Animated Changes

* [– prepareForAnimatedBoundsChange:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForAnimatedBoundsChange:)
* [– finalizeAnimatedBoundsChange](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeAnimatedBoundsChange)

### Transitioning Between Layouts

* [– prepareForTransitionFromLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForTransitionFromLayout:)
* [– prepareForTransitionToLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForTransitionToLayout:)
* [– finalizeLayoutTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeLayoutTransition)

### Registering Decoration Views

* [– registerClass:forDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/registerClass:forDecorationViewOfKind:)
* [– registerNib:forDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/registerNib:forDecorationViewOfKind:)

## Properties

### collectionView

The collection view object currently using this layout object. (read-only)

@property (nonatomic, readonly) [UICollectionView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/UICollectionView) \*collectionView

##### Discussion

The collection view object sets the value of this property when a new layout object is assigned to it.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

## Class Methods

### invalidationContextClass

Returns the class to use when creating an invalidation context for the layout.

+ (Class)invalidationContextClass

##### Discussion

If you subclass UICollectionViewLayout and use a custom invalidation context object to improve the performance of your layout updates, override this method and return your [UICollectionViewLayoutInvalidationContext](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutInvalidationContext_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayoutInvalidationContext" \t "_self) subclass. When the collection view needs to invalidate your layout, it uses the class you provide to create an appropriate invalidation context object.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### layoutAttributesClass

Returns the class to use when creating layout attributes objects.

+ (Class)layoutAttributesClass

##### Return Value

The class to use for layout attributes objects.

##### Discussion

If you subclass [UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayoutAttributes" \t "_self) in order to manage additional layout attributes, you should override this method and return your custom subclass. The methods for creating layout attributes use this class when creating new layout attributes objects.

This method is intended for subclassers only and does not need to be called by your code.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForItemAtIndexPath:)
* [– layoutAttributesForSupplementaryViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForSupplementaryViewOfKind:atIndexPath:)
* [– layoutAttributesForDecorationViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForDecorationViewOfKind:atIndexPath:)

##### Declared In

UICollectionViewLayout.h

## Instance Methods

### collectionViewContentSize

Returns the width and height of the collection view’s contents.

- ([CGSize](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/doc/c_ref/CGSize))collectionViewContentSize

##### Return Value

The width and height of the collection view’s contents.

##### Discussion

Subclasses must override this method and use it to return the width and height of the collection view’s content. These values represent the width and height of all the content, not just the content that is currently visible. The collection view uses this information to configure its own content size for scrolling purposes.

The default implementation of this method returns [CGSizeZero](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/c/data/CGSizeZero" \t "_self).

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalizeAnimatedBoundsChange

Cleans up after any animated changes to the view’s bounds or after the insertion or deletion of items.

- (void)finalizeAnimatedBoundsChange

##### Discussion

The collection view calls this method after creating the animations for changing the view’s bounds or after the animated insertion or deletion of items. This method is the layout object’s opportunity to do any cleanup related to those operations.

You can also use this method to perform additional animations. Any animations you create are added to the animation block used to handle the insertions, deletions, and bounds changes.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalizeCollectionViewUpdates

Performs any additional animations or clean up needed during a collection view update.

- (void)finalizeCollectionViewUpdates

##### Discussion

The collection view calls this method as the last step before preceding to animate any changes into place. This method is called within the animation block used to perform all of the insertion, deletion, and move animations so you can create additional animations using this method as needed. Otherwise, you can use it to perform any last minute tasks associated with managing your layout object’s state information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalizeLayoutTransition

Tells the layout object to perform any final steps before the transition animations occur.

- (void)finalizeLayoutTransition

##### Discussion

The collection view calls this method after it has gathered all of the layout attributes needed to perform a transition from one layout to another. You can use this method to clean up any data structures or caches created by your implementations of the [prepareForTransitionFromLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForTransitionFromLayout:) or [prepareForTransitionToLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForTransitionToLayout:)methods.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalLayoutAttributesForDisappearingDecorationElementOfKind:atIndexPath:

Returns the final layout information for a decoration view that is about to be removed from the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)finalLayoutAttributesForDisappearingDecorationElementOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)elementKind atIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)elementIndexPath

##### Parameters

elementKind

A string that identifies the type of decoration view.

elementIndexPath

The index path of the view being deleted.

##### Return Value

A layout attributes object that describes the position of the decoration view to use as the end point for animating its removal.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any decoration views that are about to be deleted. Your implementation should return the layout information that describes the final position and state of the view. The collection view uses this information as the end point for any animations. (The starting point of the animation is the view’s current location.) If you return nil, the layout object uses the same attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalLayoutAttributesForDisappearingItemAtIndexPath:

Returns the final layout information for an item that is about to be removed from the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)finalLayoutAttributesForDisappearingItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)itemIndexPath

##### Parameters

itemIndexPath

The index path of the item being deleted.

##### Return Value

A layout attributes object that describes the position of the cell to use as the end point for animating its removal.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any items that are about to be deleted. Your implementation should return the layout information that describes the final position and state of the item. The collection view uses this information as the end point for any animations. (The starting point of the animation is the item’s current location.) If you return nil, the layout object uses the same attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### finalLayoutAttributesForDisappearingSupplementaryElementOfKind:atIndexPath:

Returns the final layout information for a supplementary view that is about to be removed from the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)finalLayoutAttributesForDisappearingSupplementaryElementOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)elementKindatIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)elementIndexPath

##### Parameters

elementKind

A string that identifies the type of supplementary view.

elementIndexPath

The index path of the view being deleted.

##### Return Value

A layout attributes object that describes the position of the supplementary view to use as the end point for animating its removal.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any supplementary views that are about to be deleted. Your implementation should return the layout information that describes the final position and state of the view. The collection view uses this information as the end point for any animations. (The starting point of the animation is the view’s current location.) If you return nil, the layout object uses the same attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### indexPathsToDeleteForDecorationViewOfKind:

Returns an array of index paths representing the decoration views to remove.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html#//apple_ref/doc/c_ref/NSArray) \*)indexPathsToDeleteForDecorationViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind

##### Parameters

kind

The specific type of decoration view.

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects indicating the decoration views you want to remove or an empty array if you do not want to remove any views of the given kind.

##### Discussion

The collection view calls this method whenever you delete cells or sections to the collection view. Implementing this method gives your layout object an opportunity to remove any decoration views that are no longer needed.

The collection view calls this method between its calls to [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) and [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates).

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### indexPathsToDeleteForSupplementaryViewOfKind:

Returns an array of index paths representing the supplementary views to remove.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html#//apple_ref/doc/c_ref/NSArray) \*)indexPathsToDeleteForSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind

##### Parameters

kind

The specific type of supplementary view.

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects indicating the supplementary views you want to remove or an empty array if you do not want to remove any views of the given kind.

##### Discussion

The collection view calls this method whenever you delete cells or sections to the collection view. Implementing this method gives your layout object an opportunity to remove any supplementary views that are no longer needed.

The collection view calls this method between its calls to [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) and [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates).

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### indexPathsToInsertForDecorationViewOfKind:

Returns an array of index paths representing the decoration views to add.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html#//apple_ref/doc/c_ref/NSArray) \*)indexPathsToInsertForDecorationViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind

##### Parameters

kind

The specific type of decoration view.

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects indicating the location of the new decoration views or an empty array if you do not want to add any decoration views.

##### Discussion

The collection view calls this method whenever you add cells or sections to the collection view. Implementing this method gives your layout object an opportunity to add new decoration views to complement the additions.

The collection view calls this method between its calls to [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) and [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates).

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### indexPathsToInsertForSupplementaryViewOfKind:

Returns an array of index paths for the supplementary views you want to add to the layout.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html#//apple_ref/doc/c_ref/NSArray) \*)indexPathsToInsertForSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind

##### Parameters

kind

The specific type of supplementary view.

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects indicating the location of the new supplementary views or an empty array if you do not want to add any supplementary views.

##### Discussion

The collection view calls this method whenever you add cells or sections to the collection view. Implementing this method gives your layout object an opportunity to add new supplementary views to complement the additions.

The collection view calls this method between its calls to [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) and [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates).

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### initialLayoutAttributesForAppearingDecorationElementOfKind:atIndexPath:

Returns the starting layout information for a decoration view being inserted into the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)initialLayoutAttributesForAppearingDecorationElementOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)elementKind atIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)elementIndexPath

##### Parameters

elementKind

A string that identifies the type of decoration view.

elementIndexPath

The index path of the item being inserted.

##### Return Value

A layout attributes object that describes the position at which to place the corresponding decoration view.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any decoration views that are about to be inserted. Your implementation should return the layout information that describes the initial position and state of the view. The collection view uses this information as the starting point for any animations. (The end point of the animation is the view’s new location in the collection view.) If you return nil, the layout object uses the item’s final attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### initialLayoutAttributesForAppearingItemAtIndexPath:

Returns the starting layout information for an item being inserted into the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)initialLayoutAttributesForAppearingItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)itemIndexPath

##### Parameters

itemIndexPath

The index path of the item being inserted. You can use this path to locate the item in the collection view’s data source.

##### Return Value

A layout attributes object that describes the position at which to place the corresponding cell.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any items that are about to be inserted. Your implementation should return the layout information that describes the initial position and state of the item. The collection view uses this information as the starting point for any animations. (The end point of the animation is the item’s new location in the collection view.) If you return nil, the layout object uses the item’s final attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### initialLayoutAttributesForAppearingSupplementaryElementOfKind:atIndexPath:

Returns the starting layout information for a supplementary view being inserted into the collection view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)initialLayoutAttributesForAppearingSupplementaryElementOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)elementKindatIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)elementIndexPath

##### Parameters

elementKind

A string that identifies the type of supplementary view.

elementIndexPath

The index path of the item being inserted.

##### Return Value

A layout attributes object that describes the position at which to place the corresponding supplementary view.

##### Discussion

This method is called after the [prepareForCollectionViewUpdates:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareForCollectionViewUpdates:) method and before the [finalizeCollectionViewUpdates](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/finalizeCollectionViewUpdates) method for any supplementary views that are about to be inserted. Your implementation should return the layout information that describes the initial position and state of the view. The collection view uses this information as the starting point for any animations. (The end point of the animation is the view’s new location in the collection view.) If you return nil, the layout object uses the item’s final attributes for both the start and end points of the animation.

The default implementation of this method returns nil.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### invalidateLayout

Invalidates the current layout and triggers a layout update.

- (void)invalidateLayout

##### Discussion

You can call this method at any time to update the layout information. This method invalidates the layout of the collection view itself and returns right away. Thus, you can call this method multiple times from the same block of code without triggering multiple layout updates. The actual layout update occurs during the next view layout update cycle.

If you override this method, you must call super at some point in your implementation.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### invalidateLayoutWithContext:

Invalidates the current layout using the information in the provided context object.

- (void)invalidateLayoutWithContext:([UICollectionViewLayoutInvalidationContext](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutInvalidationContext_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutInvalidationContext) \*)context

##### Parameters

context

The context object indicating which parts of the layout to refresh.

##### Discussion

The default implementation of this method optimizes the layout process using the base properties of the [UICollectionViewLayoutInvalidationContext](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutInvalidationContext_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayoutInvalidationContext" \t "_self) class. If you define a custom context object for your layout, override this method and apply any custom properties of the context object to your layout computations.

If you override this method, you must call super at some point in your implementation.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### invalidationContextForBoundsChange:

Returns a context object that defines the portions of the layout that should change when a bounds change occurs.

- ([UICollectionViewLayoutInvalidationContext](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutInvalidationContext_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutInvalidationContext) \*)invalidationContextForBoundsChange:([CGRect](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGRect))newBounds

##### Parameters

newBounds

The new bounds for the collection view.

##### Return Value

An invalidation context that describes the changes that need to be made.

##### Discussion

The default implementation of this method creates an instance of the class provided by the invalidationContextClass class method and returns it. If you want to use a custom invalidation context object with your layout, always override that method and return your custom class.

You can override this method if you want to create and configure your custom invalidation context in response to a bounds change. If you override this method, you must call super first to get the invalidation context object to return. After getting this object, set any custom properties and return it.

##### Availability

* Available in iOS 7.0 and later.

##### See Also

* [+ invalidationContextClass](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/clm/UICollectionViewLayout/invalidationContextClass)

##### Declared In

UICollectionViewLayout.h

### layoutAttributesForDecorationViewOfKind:atIndexPath:

Returns the layout attributes for the specified decoration view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)layoutAttributesForDecorationViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString)\*)decorationViewKind atIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath)\*)indexPath

##### Parameters

decorationViewKind

A string that identifies the type of the decoration view.

indexPath

The index path of the decoration view.

##### Return Value

A layout attributes object containing the information to apply to the decoration view.

##### Discussion

If your layout object defines any decoration views, you must override this method and use it to return layout information for those views.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForItemAtIndexPath:)
* [– layoutAttributesForSupplementaryViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForSupplementaryViewOfKind:atIndexPath:)

##### Declared In

UICollectionViewLayout.h

### layoutAttributesForElementsInRect:

Returns the layout attributes for all of the cells and views in the specified rectangle.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)layoutAttributesForElementsInRect:([CGRect](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/doc/c_ref/CGRect))rect

##### Parameters

rect

The rectangle (specified in the collection view’s coordinate system) containing the target views.

##### Return Value

An array of [UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayoutAttributes" \t "_self) objects representing the layout information for the cells and views. The default implementation returns nil.

##### Discussion

Subclasses must override this method and use it to return layout information for all items whose view intersects the specified rectangle. Your implementation should return attributes for all visual elements, including cells, supplementary views, and decoration views.

When creating the layout attributes, always create an attributes object that represents the correct element type (cell, supplementary, or decoration). The collection view differentiates between attributes for each type and uses that information to make decisions about which views to create and how to manage them.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### layoutAttributesForItemAtIndexPath:

Returns the layout attributes for the item at the specified index path.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)layoutAttributesForItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath

##### Parameters

indexPath

The index path of the item.

##### Return Value

A layout attributes object containing the information to apply to the item’s cell.

##### Discussion

Subclasses must override this method and use it to return layout information for items in the collection view. You use this method to provide layout information only for items that have a corresponding cell. Do not use it for supplementary views or decoration views.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– layoutAttributesForSupplementaryViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForSupplementaryViewOfKind:atIndexPath:)
* [– layoutAttributesForDecorationViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForDecorationViewOfKind:atIndexPath:)

##### Declared In

UICollectionViewLayout.h

### layoutAttributesForSupplementaryViewOfKind:atIndexPath:

Returns the layout attributes for the specified supplementary view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)layoutAttributesForSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind atIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath

##### Parameters

kind

A string that identifies the type of the supplementary view.

indexPath

The index path of the view.

##### Return Value

A layout attributes object containing the information to apply to the supplementary view.

##### Discussion

If your layout object defines any supplementary views, you must override this method and use it to return layout information for those views.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForItemAtIndexPath:)
* [– layoutAttributesForDecorationViewOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForDecorationViewOfKind:atIndexPath:)

##### Declared In

UICollectionViewLayout.h

### prepareForAnimatedBoundsChange:

Prepares the layout object for animated changes to the view’s bounds or the insertion or deletion of items.

- (void)prepareForAnimatedBoundsChange:([CGRect](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/doc/c_ref/CGRect))oldBounds

##### Parameters

oldBounds

The current bounds of the collection view.

##### Discussion

The collection view calls this method before performing any animated changes to the view’s bounds or before the animated insertion or deletion of items. This method is the layout object’s opportunity to perform any calculations needed to prepare for those animated changes. Specifically, you might use this method to calculate the initial or final positions of inserted or deleted items so that you can return those values when asked for them.

You can also use this method to perform additional animations. Any animations you create are added to the animation block used to handle the insertions, deletions, and bounds changes.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### prepareForCollectionViewUpdates:

Notifies the layout object that the contents of the collection view are about to change.

- (void)prepareForCollectionViewUpdates:([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)updateItems

##### Parameters

updateItems

An array of [UICollectionViewUpdateItem](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewUpdateItem_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewUpdateItem" \t "_self) objects that identify the changes being made.

##### Discussion

When items are inserted or deleted, the collection view notifies its layout object so that it can adjust the layout as needed. The first step in that process is to call this method to let the layout object know what changes to expect. After that, additional calls are made to gather layout information for inserted, deleted, and moved items that are going to be animated around the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### prepareForTransitionFromLayout:

Tells the layout object to prepare to be installed as the layout for the collection view.

- (void)prepareForTransitionFromLayout:(UICollectionViewLayout\*)oldLayout

##### Parameters

oldLayout

The layout object installed in the collection view at the beginning of the transition. You might use this object to provide different ending attributes based on the starting layout object.

##### Discussion

Prior to performing a layout transition, the collection view calls this method so that your layout object can perform any initial calculations needed to generate layout attributes.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### prepareForTransitionToLayout:

Tells the layout object that it is about to be removed as the layout for the collection view.

- (void)prepareForTransitionToLayout:(UICollectionViewLayout\*)newLayout

##### Parameters

newLayout

The layout object to be installed in the collection view at the end of the transition. You might use this object to provide different starting attributes depending on the final layout object.

##### Discussion

Prior to performing a layout transition, the collection view calls this method so that your layout object can perform any initial calculations needed to generate layout attributes.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### prepareLayout

Tells the layout object to update the current layout.

- (void)prepareLayout

##### Discussion

Layout updates occur the first time the collection view presents its content and whenever the layout is invalidated explicitly or implicitly because of a change to the view. During each layout update, the collection view calls this method first to give your layout object a chance to prepare for the upcoming layout operation.

The default implementation of this method does nothing. Subclasses can override it and use it to set up data structures or perform any initial computations needed to perform the layout later.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### registerClass:forDecorationViewOfKind:

Registers a class for use in creating decoration views for a collection view.

- (void)registerClass:(Class)viewClass forDecorationViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)decorationViewKind

##### Parameters

viewClass

The class to use for the supplementary view.

decorationViewKind

The element kind of the decoration view. You can use this string to distinguish between decoration views with different purposes in the layout. This parameter must not be nil and must not be an empty string.

##### Discussion

This method gives the layout object a chance to register a decoration view for use in the collection view. Decoration views provide visual adornments to a section or to the entire collection view but are not otherwise tied to the data provided by the collection view’s data source.

You do not need to create decoration views explicitly. After registering one, it is up to the layout object to decide when a decoration view is needed and return the corresponding layout attributes from its [layoutAttributesForElementsInRect:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForElementsInRect:) method. For layout attributes that specify a decoration view, the collection view creates (or reuses) a view and displays it automatically based on the registered information.

If you previously registered a class or nib file with the same kind string, the class you specify in the viewClass parameter replaces the old entry. You may specify nil forviewClass if you want to unregister the decoration view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### registerNib:forDecorationViewOfKind:

Registers a nib file for use in creating decoration views for a collection view.

- (void)registerNib:([UINib](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UINib_Ref/Reference/Reference.html#//apple_ref/doc/c_ref/UINib) \*)nib forDecorationViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)decorationViewKind

##### Parameters

nib

The nib object containing the cell definition. The nib file must contain only one top-level object and that object must be of the type [UICollectionReusableView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionReusableView" \t "_self).

decorationViewKind

The element kind of the decoration view. You can use this string to distinguish between decoration views with different purposes in the layout. This parameter must not be nil and must not be an empty string.

##### Discussion

This method gives the layout object a chance to register a decoration view for use in the collection view. Decoration views provide visual adornments to a section or to the entire collection view but are not otherwise tied to the data provided by the collection view’s data source.

You do not need to create decoration views explicitly. After registering one, it is up to the layout object to decide when a decoration view is needed and return the corresponding layout attributes from its [layoutAttributesForElementsInRect:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/layoutAttributesForElementsInRect:) method. For layout attributes that specify a decoration view, the collection view creates (or reuses) a view and displays it automatically based on the registered information.

If you previously registered a class or nib file with the same kind string, the class you specify in the viewClass parameter replaces the old entry. You may specify nil forviewClass if you want to unregister the decoration view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h

### shouldInvalidateLayoutForBoundsChange:

Asks the layout object if the new bounds require a layout update.

- (BOOL)shouldInvalidateLayoutForBoundsChange:([CGRect](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/doc/c_ref/CGRect))newBounds

##### Parameters

newBounds

The new bounds of the collection view.

##### Return Value

YES if the collection view requires a layout update or NO if the layout does not need to change.

##### Discussion

The default implementation of this method returns NO. Subclasses can override it and return an appropriate value based on whether changes in the bounds of the collection view require changes to the layout of cells and supplementary views.

If the bounds of the collection view change and this method returns YES, the collection view invalidates the layout by calling the [invalidateLayoutWithContext:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/invalidateLayoutWithContext:)method.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– invalidationContextForBoundsChange:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/invalidationContextForBoundsChange:)

##### Declared In

UICollectionViewLayout.h

### targetContentOffsetForProposedContentOffset:

Returns the content offset to use after an animated layout update or change.

- ([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGPoint))targetContentOffsetForProposedContentOffset:([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGPoint))proposedContentOffset

##### Parameters

proposedContentOffset

The proposed point (in the coordinate space of the collection view’s content view) for the upper-left corner of the visible content. This represents the point that the collection view has calculated as the most likely value to use at the end of the animation.

##### Return Value

The content offset that you want to use instead. The default implementation of this method returns the value in the proposedContentOffset parameter.

##### Discussion

During layout updates, or when transitioning between layouts, the collection view calls this method to give you the opportunity to change the proposed content offset to use at the end of the animation. You might override this method if the animations or transition might cause items to be positioned in a way that is not optimal for your design.

The collection view calls this method after calling the [prepareLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/prepareLayout) and [collectionViewContentSize](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionViewLayout/collectionViewContentSize) methods.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionViewLayout.h

### targetContentOffsetForProposedContentOffset:withScrollingVelocity:

Returns the point at which to stop scrolling.

- ([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGPoint))targetContentOffsetForProposedContentOffset:([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGPoint))proposedContentOffset withScrollingVelocity:([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGPoint))velocity

##### Parameters

proposedContentOffset

The proposed point (in the collection view’s content view) at which to stop scrolling. This is the value at which scrolling would naturally stop if no adjustments were made. The point reflects the upper-left corner of the visible content.

velocity

The current scrolling velocity along both the horizontal and vertical axes. This value is measured in points per second.

##### Return Value

The content offset that you want to use instead. This value reflects the adjusted upper-left corner of the visible area. The default implementation of this method returns the value in the proposedContentOffset parameter.

##### Discussion

If you want the scrolling behavior to snap to specific boundaries, you can override this method and use it to change the point at which to stop. For example, you might use this method to always stop scrolling on a boundary between items, as opposed to stopping in the middle of an item.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionViewLayout.h