AppKit / NSCollectionViewDelegate

Protocol

NSCollectionViewDelegate

A set of methods that you use to manage the behavior of a collection view.

macOS

```
protocol NSCollectionViewDelegate : NSObjectProtocol
```

Overview

You use the methods of this protocol to facilitate the user-initiated selection and highlighting of items, to track changes to the collection view's visual elements, and to implement drag and drop support. The methods of this protocol are optional, but for some features, you must implement specific methods to support the feature.

Implement the methods of this protocol in an object that you use to manage your collection view. Typically, you implement delegate support in the view controller or window controller that manages the collection view itself, but you can implement these methods in another object if you prefer. Assign your delegate object to the collection view either programmatically (by setting the value of the collection view's `delegate` property) or at design time in Interface Builder.

To implement drag and drop support in your collection view, implement the following methods:

- To support the dragging of content from the collection view, implement either the `collectionView(_:pasteboardWriterForItemAt:)` or `collectionView(_:writeItemsAt:to:)` method.
- To support the dropping of content into the collection view, implement the `collectionView(_:validateDrop:proposedIndexPath:dropOperation:)` and `collectionView(_:acceptDrop:indexPath:dropOperation:)` methods.
- To support multi-image drag and drop, you must implement the `collectionView(_:pasteboardWriterForItemAt:)` and `collectionView(_:updateDraggingItemsFor

`Drag:)` methods.

For more information about handling drag and drop operations, see Drag and Drop Programming Topics.

Legacy Support

Before OS X 10.11, collection views supported only a single section of items organized in a grid layout. The drag and drop methods of this protocol include variants that take a single index or an `NSIndexSet` as a parameter. Although you can use those methods to implement your drag and drop support, it is recommended that you use the newer methods that take `NSIndexPath` objects instead.

Topics

Managing the Selection

```
func collectionView(NSCollectionView, shouldSelectItemsAt: Set<IndexPath>) -> Set<IndexPath>
```

Asks the delegate to approve the pending selection of items.

```
func collectionView(NSCollectionView, didSelectItemsAt: Set<IndexPath>)
```

Notifies the delegate object that one or more items were selected.

```
func collectionView(NSCollectionView, shouldDeselectItemsAt: Set<IndexPath>) -> Set<IndexPath>
```

Asks the delegate object to approve the pending deselection of items.

```
func collectionView(NSCollectionView, didDeselectItemsAt: Set<IndexPath>)
```

Notifies the delegate object that one or more items were deselected.

Managing Item Highlighting

```
func collectionView(NSCollectionView, shouldChangeItemsAt: Set<IndexPath>, to: NSCollectionViewItem.HighlightState) -> Set<IndexPath>
```

Asks the delegate to approve the pending highlighting of the specified items.

```
func collectionView(NSCollectionView, didChangeItemsAt: Set<IndexPath>, to: NSCollectionViewItem.HighlightState)
```

Notifies the delegate that the highlight state of the specified items changed.

Tracking the Addition and Removal of Items

```
func collectionView(NSCollectionView, willDisplay: NSCollectionViewItem, forRepresentedObjectAt: IndexPath)
```

Notifies the delegate that the specified item is about to be displayed by the collection view.

```
func collectionView(NSCollectionView, didEndDisplaying: NSCollectionViewItem, forRepresentedObjectAt: IndexPath)
```

Notifies the delegate that the specified item was removed from the collection view.

```
func collectionView(NSCollectionView, willDisplaySupplementaryView: NSView, forElementKind: NSCollectionView.SupplementaryElementKind, at: IndexPath)
```

Notifies the delegate that the specified supplementary view is about to be displayed by the collection view.

```
func collectionView(NSCollectionView, didEndDisplayingSupplementaryView: NSView, forElementOfKind: NSCollectionView.SupplementaryElementKind, at: IndexPath)
```

Notifies the delegate that the specified supplementary view was removed from the collection view.

Handling Layout Changes

```
func collectionView(NSCollectionView, transitionLayoutForOldLayout: NSCollectionViewLayout, newLayout: NSCollectionViewLayout) -> NSCollectionViewTransitionLayout
```

Returns the transition layout object to use when performing an animated change between different layouts.

Drag and Drop Support

{} Supporting Collection View Drag and Drop Through File Promises

Share data between macOS apps during drag and drop by using an item provider.

```
func collectionView(NSCollectionView, canDragItemsAt: Set<IndexPath>, with: NSEvent) -> Bool
```

Returns a Boolean indicating whether a drag operation involving the specified items can begin.

`func collectionView(NSCollectionView, pasteboardWriterForItemAt: IndexPath) -> (any NSPasteboardWriting)?`

Provides the pasteboard writer for the item at the specified index path.

~~`func collectionView(NSCollectionView, writeItemsAt: Set<IndexPath>, to: NSPasteboard) -> Bool`~~

Places the data for the drag operation on the pasteboard.

Deprecated

~~`func collectionView(NSCollectionView, namesOfPromisedFilesDroppedAtDestination: URL, forDraggedItemsAt: Set<IndexPath>) -> [String]`~~

Returns the names of the promised files that you created for a drag operation.

Deprecated

`func collectionView(NSCollectionView, draggingImageForItemsAt: Set<IndexPath>, with: NSEvent, offset: NSPointPointer) -> NSImage`

Creates and returns a drag image to represent the specified items during a drag.

`func collectionView(NSCollectionView, draggingSession: NSDraggingSession, willBeginAt: NSPoint, forItemsAt: Set<IndexPath>)`

Notifies your delegate that a drag session is about to begin.

`func collectionView(NSCollectionView, draggingSession: NSDraggingSession, endedAt: NSPoint, dragOperation: NSDragOperation)`

Notifies your delegate that a drag session ended.

`func collectionView(NSCollectionView, updateDraggingItemsForDrag: any NSDraggingInfo)`

Asks your delegate to update the dragging items during a drag operation.

`func collectionView(NSCollectionView, validateDrop: any NSDraggingInfo, proposedIndexPath: AutoreleasingUnsafeMutablePointer<NSIndexPath>, dropOperation: UnsafeMutablePointer<NSCollectionView.DropOperation>) -> NSDragOperation`

Validates whether a drop operation is possible at the specified location.

`func collectionView(NSCollectionView, acceptDrop: any NSDraggingInfo, indexPath: IndexPath, dropOperation: NSCollectionView.DropOperation) -> Bool`

Incorporates the dropped content into the collection view.

Legacy Collection View Support

These methods are called only on a collection view that has only one section. They are not called for collection views with two or more sections.

```
func collectionView(NSCollectionView, canDragItemsAt: IndexSet, with: NSEvent) -> Bool
```

Returns a Boolean indicating whether the collection view can begin dragging the specified items.

```
func collectionView(NSCollectionView, pasteboardWriterForItemAt: Int) -> (any NSPasteboardWriting)?
```

Provides the pasteboard writer for the item at the specified index

~~func collectionView(NSCollectionView, writeItemsAt: IndexSet, to: NSPasteboard) -> Bool~~

Invoked after it has been determined that a drag should begin, but before the drag has been started.

Deprecated

~~func collectionView(NSCollectionView, namesOfPromisedFilesDroppedAtDestination: URL, forDraggedItemsAt: IndexSet) -> [String]~~

Invoked to return an array of filenames that the receiver promises to create.

Deprecated

```
func collectionView(NSCollectionView, draggingImageForItemsAt: IndexSet, with: NSEvent, offset: NSPointPointer) -> NSImage
```

Creates and returns a drag image to represent the specified items during a drag.

```
func collectionView(NSCollectionView, draggingSession: NSDraggingSession, willBeginAt: NSPoint, forItemsAt: IndexSet)
```

Notifies your delegate that a drag session is about to begin.

```
func collectionView(NSCollectionView, validateDrop: any NSDraggingInfo, proposedIndex: UnsafeMutablePointer<Int>, dropOperation: UnsafeMutablePointer<NSCollectionView.DropOperation>) -> NSDragOperation
```

Validates the specified location to see if it is a valid drop target.

```
func collectionView(NSCollectionView, acceptDrop: any NSDraggingInfo, index: Int, dropOperation: NSCollectionView.DropOperation) -> Bool
```

Invoked when the mouse is released over a collection view that previously allowed a drop.

Relationships

Inherits From

`NSObjectProtocol`

Inherited By

`NSCollectionViewDelegateFlowLayout`

See Also

Data

`protocol NSCollectionViewDataSource`

A set of methods that a data source object implements to provide the information and view objects that a collection view requires to present content.

`class NSCollectionViewDiffableDataSource`

The object you use to manage data and provide items for a collection view.

`struct NSDiffableDataSourceSnapshot`

A representation of the state of the data in a view at a specific point in time.