
NSArrayController Class Reference

Data Management: Data Types & Collections





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

Aperture and Finder are trademarks of Apple Inc.

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

NSArrayController Class Reference 5

Overview	5
Tasks	5
Managing Sort Descriptors	5
Arranging Objects	6
Managing Content	6
Selection Attributes	6
Managing Selections	7
Inserting	8
Adding and Removing Objects	8
Filtering Content	8
Automatic Content Rearranging	9
Instance Methods	9
add:	9
addObject:	9
addObjects:	10
addSelectedObjects:	10
addSelectionIndexes:	11
alwaysUsesMultipleValuesMarker	11
arrangedObjects	11
arrangeObjects:	12
automaticallyPreparesContent	12
automaticallyRearrangesObjects	13
automaticRearrangementKeyPaths	13
avoidsEmptySelection	14
canInsert	14
canSelectNext	15
canSelectPrevious	15
clearsFilterPredicateOnInsertion	15
didChangeArrangementCriteria	16
filterPredicate	16
insert:	17
insertObject:atArrangedObjectIndex:	17
insertObjects:atArrangedObjectIndexes:	18
preservesSelection	18
rearrangeObjects	19
remove:	19
removeObject:	20
removeObjectAtArrangedObjectIndex:	20
removeObjects:	21
removeObjectsAtArrangedObjectIndexes:	21

- removeSelectedObjects: 21
- removeSelectionIndexes: 22
- selectedObjects 22
- selectionIndex 23
- selectionIndexes 23
- selectNext: 24
- selectPrevious: 24
- selectsInsertedObjects 25
- setAlwaysUsesMultipleValuesMarker: 25
- setAutomaticallyPreparesContent: 25
- setAutomaticallyRearrangesObjects: 26
- setAvoidsEmptySelection: 26
- setClearsFilterPredicateOnInsertion: 27
- setFilterPredicate: 27
- setPreservesSelection: 28
- setSelectedObjects: 28
- setSelectionIndex: 28
- setSelectionIndexes: 29
- setSelectsInsertedObjects: 30
- setSortDescriptors: 30
- sortDescriptors 30

Document Revision History 33

Index 35

NSArrayController Class Reference

Inherits from	NSObjectController : NSController : NSObject
Conforms to	NSCoding (NSController) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.3 and later.
Declared in	NSArrayController.h
Companion guides	Cocoa Bindings Programming Topics Predicate Programming Guide Core Data Programming Guide
Related sample code	DemoMonkey Departments and Employees GridCalendar LightTable XMLBrowser

Overview

`NSArrayController` is a bindings compatible class that manages a collection of objects. Typically the collection is an array, however, if the controller manages a relationship of a managed object (see `NSManagedObject`) the collection may be a set. `NSArrayController` provides selection management and sorting capabilities.

Tasks

Managing Sort Descriptors

- [setSortDescriptors:](#) (page 30)
Sets the sort descriptors for the receiver.
- [sortDescriptors](#) (page 30)
Returns the receiver's array of sort descriptors.

Arranging Objects

- [arrangeObjects:](#) (page 12)
Returns a given array, appropriately sorted and filtered.
- [arrangedObjects](#) (page 11)
Returns an array containing the receiver's content objects arranged using [arrangeObjects:](#) (page 12).
- [rearrangeObjects](#) (page 19)
Triggers filtering of the receiver's content.

Managing Content

- [add:](#) (page 9)
Creates and adds a new object to the receiver's content and arranged objects.
- [setAutomaticallyPreparesContent:](#) (page 25)
Sets whether the receiver automatically creates and inserts new content objects automatically.
- [automaticallyPreparesContent](#) (page 12)
Returns a Boolean value that indicates whether the receiver automatically prepares its content when it is loaded from a nib.

Selection Attributes

- [setAvoidsEmptySelection:](#) (page 26)
Sets whether the receiver attempts to avoid an empty selection.
- [avoidsEmptySelection](#) (page 14)
Returns a Boolean value that indicates whether if the receiver requires that the content array attempt to maintain a selection.
- [setClearsFilterPredicateOnInsertion:](#) (page 27)
Sets whether the receiver automatically clears an existing filter predicate when a new object is inserted or added to the content array.
- [preservesSelection](#) (page 18)
Returns a Boolean value that indicates whether the receiver will attempt to preserve the current selection when the content changes.
- [setPreservesSelection:](#) (page 28)
Sets whether the receiver attempts to preserve selection when the content changes.
- [alwaysUsesMultipleValuesMarker](#) (page 11)
Returns a Boolean value that indicates whether the receiver always returns the multiple values marker when multiple objects are selected.
- [setAlwaysUsesMultipleValuesMarker:](#) (page 25)
Sets whether the receiver always returns the multiple values marker when multiple objects are selected.

Managing Selections

- [selectionIndex](#) (page 23)
Returns the index of the first object in the receiver's selection.
- [setSelectionIndex:](#) (page 28)
Sets the receiver's selection to the given index, and returns a Boolean value that indicates whether the selection was changed.
- [setSelectsInsertedObjects:](#) (page 30)
Sets whether the receiver will automatically select objects as they are inserted.
- [selectsInsertedObjects](#) (page 25)
Returns whether the receiver automatically selects inserted objects.
- [setSelectionIndexes:](#) (page 29)
Sets the receiver's selection indexes and returns a Boolean value that indicates whether the selection changed.
- [selectionIndexes](#) (page 23)
Returns an index set containing the indexes of the receiver's currently selected objects in the content array.
- [addSelectionIndexes:](#) (page 11)
Adds the objects at the specified indexes in the receiver's content array to the current selection, returning YES if the selection was changed.
- [removeSelectionIndexes:](#) (page 22)
Removes the object as the specified *indexes* from the receiver's current selection, returning YES if the selection was changed.
- [setSelectedObjects:](#) (page 28)
Sets *objects* as the receiver's current selection, returning YES if the selection was changed.
- [selectedObjects](#) (page 22)
Returns an array containing the receiver's selected objects.
- [addSelectedObjects:](#) (page 10)
Adds *objects* from the receiver's content array to the current selection, returning YES if the selection was changed.
- [removeSelectedObjects:](#) (page 21)
Removes *objects* from the receiver's current selection, returning YES if the selection was changed.
- [selectNext:](#) (page 24)
Selects the next object, relative to the current selection, in the receiver's arranged content.
- [canSelectNext](#) (page 15)
Returns YES if the next object, relative to the current selection, in the receiver's content array can be selected.
- [selectPrevious:](#) (page 24)
Selects the previous object, relative to the current selection, in the receiver's arranged content.
- [canSelectPrevious](#) (page 15)
Returns YES if the previous object, relative to the current selection, in the receiver's content array can be selected.

Inserting

- [canInsert](#) (page 14)
Returns a Boolean value that indicates whether an object can be inserted into the receiver's content collection.
- [insert:](#) (page 17)
Creates a new object and inserts it into the receiver's content array.

Adding and Removing Objects

- [addObject:](#) (page 9)
Adds *object* to the receiver's content collection and the arranged objects array.
- [addObjects:](#) (page 10)
Adds *objects* to the receiver's content collection.
- [insertObject:atArrangedObjectIndex:](#) (page 17)
Inserts *object* into the receiver's arranged objects array at the location specified by *index*, and adds it to the receiver's content collection.
- [insertObjects:atArrangedObjectIndexes:](#) (page 18)
Inserts *objects* into the receiver's arranged objects array at the locations specified in *indexes*, and adds it to the receiver's content collection.
- [removeObjectAtArrangedObjectIndex:](#) (page 20)
Removes the object at the specified *index* in the receiver's arranged objects from the receiver's content array.
- [removeObjectsAtArrangedObjectIndexes:](#) (page 21)
Removes the objects at the specified *indexes* in the receiver's arranged objects from the content array.
- [remove:](#) (page 19)
Removes the receiver's selected objects from the content collection.
- [removeObject:](#) (page 20)
Removes *object* from the receiver's content collection.
- [removeObjects:](#) (page 21)
Removes *objects* from the receiver's content collection.

Filtering Content

- [clearsFilterPredicateOnInsertion](#) (page 15)
Returns a Boolean value that indicates whether the receiver automatically clears an existing filter predicate when new items are inserted or added to the content.
- [filterPredicate](#) (page 16)
Returns the predicate used by the receiver to filter the array controller contents.
- [setFilterPredicate:](#) (page 27)
Sets the predicate used to filter the contents of the receiver.

Automatic Content Rearranging

- [setAutomaticallyRearrangesObjects:](#) (page 26)
Sets whether or not the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.
- [automaticallyRearrangesObjects](#) (page 13)
Returns a Boolean that indicates if the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.
- [automaticRearrangementKeyPaths](#) (page 13)
Returns an array of key paths that trigger automatic content sorting or filtering.
- [didChangeArrangementCriteria](#) (page 16)
Invoked when any criteria for arranging objects change.

Instance Methods

add:

Creates and adds a new object to the receiver's content and arranged objects.

- (void)add:(id)sender

Parameters

sender

Typically the object that invoked this method.

Special Considerations

Beginning with Mac OS X v10.4 the result of this method is deferred until the next iteration of the runloop so that the error presentation mechanism (see Error Responders and Error Recovery) can provide feedback as a sheet.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSArrayController.h

addObject:

Adds *object* to the receiver's content collection and the arranged objects array.

- (void)addObject:(id)object

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addObjects:](#) (page 10)
- [insertObject:atArrangedObjectIndex:](#) (page 17)

- [removeObject:](#) (page 20)

Related Sample Code

Aperture Edit Plugin - Borders & Titles

QTCompressionOptionsWindow

XMLBrowser

Declared In

NSArrayController.h

addObjects:

Adds *objects* to the receiver's content collection.

```
- (void)addObjects:(NSArray *)objects
```

Discussion

If [selectsInsertedObjects](#) (page 25) returns YES (the default), the added objects are selected in the array controller.

It is important to note that inserting many objects with `selectsInsertedObjects` on can cause a significant performance penalty. In this case it is more efficient to use the `setContent:` method to set the array, or to set `selectsInsertedObjects` to NO before adding the objects with `addObjects:`.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addObject:](#) (page 9)

- [insertObjects:atArrangedObjectIndexes:](#) (page 18)

- [removeObjects:](#) (page 21)

Declared In

NSArrayController.h

addSelectedObjects:

Adds *objects* from the receiver's content array to the current selection, returning YES if the selection was changed.

```
- (BOOL)addSelectedObjects:(NSArray *)objects
```

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeSelectedObjects:](#) (page 21)

- [setSelectedObjects:](#) (page 28)

Declared In

NSArrayController.h

addSelectionIndexes:

Adds the objects at the specified indexes in the receiver's content array to the current selection, returning YES if the selection was changed.

- (BOOL)addSelectionIndexes:(NSIndexSet *)*indexes*

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeSelectionIndexes:](#) (page 22)

Declared In

NSArrayController.h

alwaysUsesMultipleValuesMarker

Returns a Boolean value that indicates whether the receiver always returns the multiple values marker when multiple objects are selected.

- (BOOL)alwaysUsesMultipleValuesMarker

Return Value

YES if the receiver always returns the multiple values marker when multiple objects are selected—even if the selected items have the same value, otherwise NO.

Discussion

The default is NO.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAlwaysUsesMultipleValuesMarker:](#) (page 25)

Declared In

NSArrayController.h

arrangedObjects

Returns an array containing the receiver's content objects arranged using [arrangeObjects:](#) (page 12).

- (id)arrangedObjects

Return Value

An array containing the receiver’s content objects arranged using [arrangeObjects:](#) (page 12).

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [arrangeObjects:](#) (page 12)

Related Sample Code

- CoreRecipes
- DemoMonkey
- QTCompressionOptionsWindow
- With and Without Bindings

Declared In

NSArrayController.h

arrangeObjects:

Returns a given array, appropriately sorted and filtered.

- (NSArray *)arrangeObjects:(NSArray *)objects

Return Value

An array containing *objects* filtered using the receiver's filter predicate (see [filterPredicate](#) (page 16)) and sorted according to the receiver’s [sortDescriptors](#) (page 30).

Discussion

Subclasses should override this method to use a different sort mechanism, provide custom object arrangement, or (typically only prior to Mac OS X version 10.4, which provides a filter predicate) filter the objects.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [arrangedObjects](#) (page 11)
- [rearrangeObjects](#) (page 19)
- [sortDescriptors](#) (page 30)

Declared In

NSArrayController.h

automaticallyPreparesContent

Returns a Boolean value that indicates whether the receiver automatically prepares its content when it is loaded from a nib.

- (BOOL)automaticallyPreparesContent

Return Value

YES if the receiver automatically prepares its content when loaded from a nib, otherwise NO.

Discussion

See [setAutomaticallyPreparesContent:](#) (page 25) for a full explanation of "automatically prepares content."

The default is YES.

See Also

- [setAutomaticallyPreparesContent:](#) (page 25)
- [prepareContent](#)

automaticallyRearrangesObjects

Returns a Boolean that indicates if the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.

- (BOOL)automaticallyRearrangesObjects

Return Value

YES if the receiver automatically rearranges objects upon changes to the content, NO if the content does not automatically rearrange.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSArrayController.h

automaticRearrangementKeyPaths

Returns an array of key paths that trigger automatic content sorting or filtering.

- (NSArray *)automaticRearrangementKeyPaths

Return Value

An array of key paths that trigger automatic content sorting or filtering.

Discussion

Subclasses can override this method to customize the default behavior of the sort descriptors and filtering predicates, for example, if additional arrangement criteria are used in a custom implementation of [rearrangeObjects](#) (page 19).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [rearrangeObjects](#) (page 19)

Declared In

NSArrayController.h

avoidsEmptySelection

Returns a Boolean value that indicates whether if the receiver requires that the content array attempt to maintain a selection.

- (BOOL)avoidsEmptySelection

Return Value

YES if the receiver requires that the content array attempt to maintain a selection at all times, otherwise NO.

Discussion

The default is YES.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setAvoidsEmptySelection:](#) (page 26)

Declared In

NSArrayController.h

canInsert

Returns a Boolean value that indicates whether an object can be inserted into the receiver's content collection.

- (BOOL)canInsert

Return Value

YES if an object can be inserted into the receiver's content collection, otherwise NO.

Discussion

The result of this method can be used by a binding to enable user interface items.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [insert:](#) (page 17)

Declared In

NSArrayController.h

canSelectNext

Returns YES if the next object, relative to the current selection, in the receiver's content array can be selected.

- (BOOL)canSelectNext

Discussion

The result of this method can be used by a binding to enable user interface items.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectNext:](#) (page 24)
- [canSelectPrevious](#) (page 15)

Declared In

NSArrayController.h

canSelectPrevious

Returns YES if the previous object, relative to the current selection, in the receiver's content array can be selected.

- (BOOL)canSelectPrevious

Discussion

The result of this method can be used by a binding to enable user interface items.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [canSelectNext](#) (page 15)
- [selectPrevious:](#) (page 24)

Declared In

NSArrayController.h

clearsFilterPredicateOnInsertion

Returns a Boolean value that indicates whether the receiver automatically clears an existing filter predicate when new items are inserted or added to the content.

- (BOOL)clearsFilterPredicateOnInsertion

Return Value

YES if the receiver automatically clears an existing filter predicate when new items are inserted or added to the content, otherwise NO.

Discussion

The default is YES.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setClearsFilterPredicateOnInsertion:](#) (page 27)

Declared In

NSArrayController.h

didChangeArrangementCriteria

Invoked when any criteria for arranging objects change.

```
- (void)didChangeArrangementCriteria
```

Discussion

This method is invoked by the controller itself when any criteria for arranging objects change (sort descriptors or filter predicates) to reset the key paths for automatic rearranging.

Special Considerations

If you implement a subclass of `NSArrayController` and override [rearrangeObjects](#) (page 19) to use additional arrangement criteria, you should invoke this method if those criteria change.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [rearrangeObjects](#) (page 19)

Declared In

NSArrayController.h

filterPredicate

Returns the predicate used by the receiver to filter the array controller contents.

```
- (NSPredicate *)filterPredicate
```

Return Value

The predicate used by the receiver to filter the array controller contents. Returns `nil` if no filter predicate is set.

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setClearsFilterPredicateOnInsertion:](#) (page 27)
- [setFilterPredicate:](#) (page 27)

Declared In

NSArrayController.h

insert:

Creates a new object and inserts it into the receiver's content array.

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

Parameters

sender

Typically the object that invoked this method.

Discussion

If an entity name is specified (see `entityName`), this method creates an instance of the of the class specified by the entity, otherwise this method creates an instance of the class specified by `objectClass`.

Special Considerations

Beginning with Mac OS X v10.4 the result of this method is deferred until the next iteration of the runloop so that the error presentation mechanism (see Error Responders and Error Recovery) can provide feedback as a sheet.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [canInsert](#) (page 14)

Declared In

NSArrayController.h

insertObject:atArrangedObjectIndex:

Inserts *object* into the receiver's arranged objects array at the location specified by *index*, and adds it to the receiver's content collection.

```
- (void)insertObject:(id)object atArrangedObjectIndex:(NSInteger)index
```

Discussion

Subclasses can override this method to provide customized arranged objects support.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [insertObjects:atArrangedObjectIndexes:](#) (page 18)
- [addObject:](#) (page 9)
- [removeObjectAtArrangedObjectIndex:](#) (page 20)

Related Sample Code

With and Without Bindings

Declared In

NSArrayController.h

insertObjects:atArrangedObjectIndexes:

Inserts *objects* into the receiver's arranged objects array at the locations specified in *indexes*, and adds it to the receiver's content collection.

```
- (void)insertObjects:(NSArray *)objects atArrangedObjectIndexes:(NSIndexSet *)indexes
```

Availability

Available in Mac OS X v10.3 and later.

See Also

- [insertObject:atArrangedObjectIndex:](#) (page 17)
- [addObjects:](#) (page 10)
- [removeObjectsAtArrangedObjectIndexes:](#) (page 21)

Related Sample Code

DemoMonkey

Declared In

NSArrayController.h

preservesSelection

Returns a Boolean value that indicates whether the receiver will attempt to preserve the current selection when the content changes.

```
- (BOOL)preservesSelection
```

Return Value

YES if the receiver attempts to preserve the current selection when the content changes, otherwise NO.

Discussion

The default is YES.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setClearsFilterPredicateOnInsertion:](#) (page 27)

Declared In

NSArrayController.h

rearrangeObjects

Triggers filtering of the receiver's content.

- (void)rearrangeObjects

Discussion

This method invokes [arrangeObjects:](#) (page 12).

When you detect that filtering criteria change (such as when listening to the text sent by an `NSSearchField` instance), invoke this method on `self`.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [arrangeObjects:](#) (page 12)
- [didChangeArrangementCriteria](#) (page 16)
- [automaticRearrangementKeyPaths](#) (page 13)

Related Sample Code

CoreRecipes

Departments and Employees

iSpend

Declared In

NSArrayController.h

remove:

Removes the receiver's selected objects from the content collection.

- (void)remove:(id)sender

Parameters

sender

Typically the object that invoked this method.

Discussion

See [removeObject:](#) (page 20) for a discussion of the semantics of removing objects when using Core Data.

Special Considerations

Beginning with Mac OS X v10.4 the result of this method is deferred until the next iteration of the runloop so that the error presentation mechanism (see Error Responders and Error Recovery) can provide feedback as a sheet.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [removeObjects:](#) (page 21)
- [removeObjectAtIndex:](#) (page 20)
- [addObject:](#) (page 9)

Related Sample Code

Aperture Edit Plugin - Borders & Titles

CoreRecipes

Declared In

NSArrayController.h

removeObject:Removes *object* from the receiver's content collection.- (void)removeObject:(id)*object***Special Considerations**

If you are using Core Data, the exact semantics of this method differ depending on the settings for the array controller. If the receiver's content is fetched automatically, removed objects are marked for deletion by the managed object context (and hence removal from the object graph). If, however, the receiver's `contentSet` is bound to a relationship, `removeObject:` by default only removes the object from the relationship (not from the object graph). You can, though, set the "Deletes Object on Remove" option for the `contentSet` binding, in which case objects are marked for deletion as well as being removed from the relationship.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeObjects:](#) (page 21)
- [removeObjectAtArrangedObjectIndex:](#) (page 20)
- [addObject:](#) (page 9)

Related Sample Code

CameraBrowser

ScannerBrowser

Declared In

NSArrayController.h

removeObjectAtArrangedObjectIndex:Removes the object at the specified *index* in the receiver's arranged objects from the receiver's content array.- (void)removeObjectAtArrangedObjectIndex:(NSUInteger)*index***Discussion**See [removeObject:](#) (page 20) for a discussion of the semantics of removing objects when using Core Data.**Availability**

Available in Mac OS X v10.3 and later.

See Also

- [removeObjectsAtArrangedObjectIndexes:](#) (page 21)
- [insertObject:atArrangedObjectIndex:](#) (page 17)

- [removeObject:](#) (page 20)

Declared In

NSArrayController.h

removeObjects:

Removes *objects* from the receiver's content collection.

- (void)removeObjects:(NSArray *)*objects*

Special Considerations

See [removeObject:](#) (page 20) for a discussion of the semantics of removing objects when using Core Data.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeObject:](#) (page 20)
- [removeObjectsAtArrangedObjectIndexes:](#) (page 21)
- [addObjects:](#) (page 10)

Related Sample Code

QTCompressionOptionsWindow

Declared In

NSArrayController.h

removeObjectsAtArrangedObjectIndexes:

Removes the objects at the specified *indexes* in the receiver's arranged objects from the content array.

- (void)removeObjectsAtArrangedObjectIndexes:(NSIndexSet *)*indexes*

Discussion

See [removeObject:](#) (page 20) for a discussion of the semantics of removing objects when using Core Data.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeObjectAtArrangedObjectIndex:](#) (page 20)
- [insertObjects:atArrangedObjectIndexes:](#) (page 18)
- [removeObjects:](#) (page 21)

Declared In

NSArrayController.h

removeSelectedObjects:

Removes *objects* from the receiver's current selection, returning YES if the selection was changed.

- (BOOL)removeSelectedObjects:(NSArray *)*objects*

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addSelectedObjects:](#) (page 10)

Declared In

NSArrayController.h

removeSelectionIndexes:

Removes the object as the specified *indexes* from the receiver's current selection, returning YES if the selection was changed.

- (BOOL)removeSelectionIndexes:(NSIndexSet *)*indexes*

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addSelectionIndexes:](#) (page 11)

Declared In

NSArrayController.h

selectedObjects

Returns an array containing the receiver's selected objects.

- (NSArray *)selectedObjects

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSelectedObjects:](#) (page 28)

Related Sample Code

CalendarItems

CoreRecipes

QTCompressionOptionsWindow
ScannerBrowser

Declared In

NSArrayController.h

selectionIndex

Returns the index of the first object in the receiver’s selection.

- (NSInteger)selectionIndex

Return Value

The index of the first object in the receiver’s selection, or `NSNotFound` if there is no selection.

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSelectionIndex:](#) (page 28)
- [selectionIndexes](#) (page 23)

Related Sample Code

- ClipboardViewer
- CoreRecipes
- DemoMonkey
- SBSystemPrefs
- ScriptingBridgeFinder

Declared In

NSArrayController.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 v10.3 and later.

See Also

- [setSelectionIndexes:](#) (page 29)

- [selectedIndex](#) (page 23)

Declared In

NSArrayController.h

selectNext:

Selects the next object, relative to the current selection, in the receiver's arranged content.

- (void)selectNext:(id)sender

Discussion

The *sender* is typically the object that invoked this method.

Special Considerations

Beginning with Mac OS X v10.4 the result of this method is deferred until the next iteration of the runloop so that the error presentation mechanism (see Error Responders and Error Recovery) can provide feedback as a sheet.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectPrevious:](#) (page 24)

- [canSelectNext](#) (page 15)

Declared In

NSArrayController.h

selectPrevious:

Selects the previous object, relative to the current selection, in the receiver's arranged content.

- (void)selectPrevious:(id)sender

Discussion

The *sender* is typically the object that invoked this method.

Special Considerations

Beginning with Mac OS X v10.4 the result of this method is deferred until the next iteration of the runloop so that the error presentation mechanism (see Error Responders and Error Recovery) can provide feedback as a sheet.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectNext:](#) (page 24)

- [canSelectPrevious](#) (page 15)

Declared In

NSArrayController.h

selectsInsertedObjects

Returns whether the receiver automatically selects inserted objects.

- (BOOL)selectsInsertedObjects

Return Value

YES if the receiver automatically selects inserted objects, otherwise NO.

Discussion

The default is YES.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSelectsInsertedObjects:](#) (page 30)

Declared In

NSArrayController.h

setAlwaysUsesMultipleValuesMarker:

Sets whether the receiver always returns the multiple values marker when multiple objects are selected.

- (void)setAlwaysUsesMultipleValuesMarker:(BOOL)flag

Parameters

flag

If YES, the receiver always returns the multiple values marker when multiple objects are selected, even if they have the same value.

Discussion

Setting *flag* to YES can increase performance if your application doesn't allow editing multiple values. The default is NO.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [alwaysUsesMultipleValuesMarker](#) (page 11)

Declared In

NSArrayController.h

setAutomaticallyPreparesContent:

Sets whether the receiver automatically creates and inserts new content objects automatically.

- (void)setAutomaticallyPreparesContent:(BOOL)flag

Parameters*flag*

If YES, the receiver automatically prepares its content.

Discussion

If *flag* is YES and the receiver is not using a managed object context, `prepareContent` is used to create the content object.

If *flag* is YES and a managed object context is set, the initial content is fetched from the managed object context using the current fetch predicate. The controller also registers as an observer of its managed object context. It then tracks insertions and deletions of its entity using the context's notifications, and updates its content array as appropriate.

Setting *flag* to YES is the same as checking the “Automatically Prepares Content” option in the Interface Builder controller inspector.

See Also

- [automaticallyPreparesContent](#) (page 12)
- `prepareContent`

setAutomaticallyRearrangesObjects:

Sets whether or not the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.

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

Parameters*flag*

A Boolean value that indicates whether the receiver automatically rearranges its content (YES) or not (NO).

Discussion

The default is NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSArrayController.h`

setAvoidsEmptySelection:

Sets whether the receiver attempts to avoid an empty selection.

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

Parameters*flag*

If YES, the receiver maintains a selection unless there are no objects in the content array.

Discussion

The default is YES.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [avoidsEmptySelection](#) (page 14)

Declared In

NSArrayController.h

setClearsFilterPredicateOnInsertion:

Sets whether the receiver automatically clears an existing filter predicate when a new object is inserted or added to the content array.

- (void)setClearsFilterPredicateOnInsertion:(BOOL)*flag*

Parameters

flag

If YES, the receiver automatically clears an existing filter predicate when a new object is inserted or added to the content array.

Discussion

The default is YES.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [clearsFilterPredicateOnInsertion](#) (page 15)

Declared In

NSArrayController.h

setFilterPredicate:

Sets the predicate used to filter the contents of the receiver.

- (void)setFilterPredicate:(NSPredicate *)*filterPredicate*

Parameters

filterPredicate

The predicate used to filter the contents of the receiver.

Discussion

If *filterPredicate* is nil, any existing filter predicate is removed.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [filterPredicate](#) (page 16)

Declared In

NSArrayController.h

setPreservesSelection:

Sets whether the receiver attempts to preserve selection when the content changes.

- (void)setPreservesSelection:(BOOL)*flag*

Parameters

flag

If YES, the receiver attempts to preserve selection when the content changes.

Discussion

The default is YES.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [preservesSelection](#) (page 18)

Declared In

NSArrayController.h

setSelectedObjects:

Sets *objects* as the receiver's current selection, returning YES if the selection was changed.

- (BOOL)setSelectedObjects:(NSArray *)*objects*

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectedObjects](#) (page 22)
- [addSelectedObjects:](#) (page 10)

Related Sample Code

CoreRecipes

Declared In

NSArrayController.h

setSelectionIndex:

Sets the receiver's selection to the given index, and returns a Boolean value that indicates whether the selection was changed.

- (BOOL)setSelectionIndex:(NSUInteger)*index*

Parameters*index*

The index for the selection.

Return Value

YES if the selection was changed, otherwise NO.

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectionIndex](#) (page 23)
- [setSelectionIndexes:](#) (page 29)

Related Sample Code

CoreRecipes

DemoMonkey

GridCalendar

QTCompressionOptionsWindow

Declared In

`NSArrayController.h`

setSelectionIndexes:

Sets the receiver's selection indexes and returns a Boolean value that indicates whether the selection changed.

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

Parameters*indexes*

The set of selection indexes for the receiver.

Return Value

YES if the selection was changed, otherwise NO.

Discussion

Attempting to change the selection may cause a `commitEditing` message which fails, thus denying the selection change.

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 v10.3 and later.

See Also

- [selectionIndexes](#) (page 23)
- [setSelectionIndex:](#) (page 28)

Related Sample Code

DemoMonkey

Declared In

NSArrayController.h

setSelectsInsertedObjects:

Sets whether the receiver will automatically select objects as they are inserted.

- (void)setSelectsInsertedObjects:(BOOL)flag

Parameters*flag*

If YES then items will be selected upon insertion.

Discussion

The default is YES.

Availability

Available in Mac OS X v10.3 and later.

See Also- [selectsInsertedObjects](#) (page 25)**Declared In**

NSArrayController.h

setSortDescriptors:

Sets the sort descriptors for the receiver.

- (void)setSortDescriptors:(NSArray *)sortDescriptors

Parameters*sortDescriptors*An array of `NSSortDescriptor` objects, used by the receiver to arrange its content.**Availability**

Available in Mac OS X v10.3 and later.

See Also- [sortDescriptors](#) (page 30)- [arrangeObjects:](#) (page 12)**Declared In**

NSArrayController.h

sortDescriptors

Returns the receiver's array of sort descriptors.

- (NSArray *)sortDescriptors

Return Value

The array of `NSSortDescriptor` objects used by the receiver to arrange its content.

Discussion

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSortDescriptors:](#) (page 30)
- [arrangeObjects:](#) (page 12)

Related Sample Code

iSpend

Declared In

NSArrayController.h

Document Revision History

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

Date	Notes
2009-11-17	Added links to description of error presentation mechanism.
2009-08-18	Corrected definition of <code>didChangeArrangementCriteria</code> .
2007-04-09	Updated to include new API introduced in Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

add: [instance method 9](#)
addObject: [instance method 9](#)
addObjects: [instance method 10](#)
addSelectedObjects: [instance method 10](#)
addSelectionIndexes: [instance method 11](#)
alwaysUsesMultipleValuesMarker [instance method 11](#)
arrangedObjects [instance method 11](#)
arrangeObjects: [instance method 12](#)
automaticallyPreparesContent [instance method 12](#)
automaticallyRearrangesObjects [instance method 13](#)
automaticRearrangementKeyPaths [instance method 13](#)
avoidsEmptySelection [instance method 14](#)

C

canInsert [instance method 14](#)
canSelectNext [instance method 15](#)
canSelectPrevious [instance method 15](#)
clearsFilterPredicateOnInsertion [instance method 15](#)

D

didChangeArrangementCriteria [instance method 16](#)

F

filterPredicate [instance method 16](#)

I

insert: [instance method 17](#)
insertObject:atArrangedObjectIndex: [instance method 17](#)
insertObjects:atArrangedObjectIndexes: [instance method 18](#)

P

preservesSelection [instance method 18](#)

R

rearrangeObjects [instance method 19](#)
remove: [instance method 19](#)
removeObjectAtArrangedObjectIndex: [instance method 20](#)
removeObject: [instance method 20](#)
removeObjectsAtArrangedObjectIndexes: [instance method 21](#)
removeObjects: [instance method 21](#)
removeSelectedObjects: [instance method 21](#)
removeSelectionIndexes: [instance method 22](#)

S

selectedObjects [instance method 22](#)
selectionIndex [instance method 23](#)
selectionIndexes [instance method 23](#)
selectNext: [instance method 24](#)
selectPrevious: [instance method 24](#)
selectsInsertedObjects [instance method 25](#)
setAlwaysUsesMultipleValuesMarker: [instance method 25](#)
setAutomaticallyPreparesContent: [instance method 25](#)

setAutomaticallyRearrangesObjects: **instance method 26**
setAvoidsEmptySelection: **instance method 26**
setClearsFilterPredicateOnInsertion: **instance method 27**
setFilterPredicate: **instance method 27**
setPreservesSelection: **instance method 28**
setSelectedObjects: **instance method 28**
setSelectionIndex: **instance method 28**
setSelectionIndexes: **instance method 29**
setSelectsInsertedObjects: **instance method 30**
setSortDescriptors: **instance method 30**
sortDescriptors **instance method 30**