NSOutlineView Class Reference

Developer

Contents

NSOutlineView Class Reference

Inherits from	NSTableView : NSControl : NSView : NSResponder : NSObject
Conforms to	NSUserInterfaceValidations (NSTableView) NSTextViewDelegate (NSTableView) NSDraggingSource (NSTableView) NSAnimatablePropertyContainer (NSView) NSUserInterfaceItemIdentification (NSView) NSDraggingDestination (NSView) NSAppearanceCustomization (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in OS X v10.0 and later.
Declared in	NSOutlineView.h
Companion guides	Outline View Programming Topics Drag and Drop Programming Topics
Related sample code	DragNDropOutlineView ImageBackground OutlineView SidebarDemo TableViewPlayground

Overview

`NSOutlineView` is a subclass of `NSTableView` that uses a row-and-column format to display hierarchical data that can be expanded and collapsed, such as directories and files in a file system. A user can expand and collapse rows, edit values, and resize and rearrange columns.

Like a table view, an outline view does not store its own data, instead it retrieves data values as needed from a data source to which it has a weak reference (see "Delegates and Data Sources"). See `NSOutlineViewDataSource Protocol`, which declares the methods that an `NSOutlineView` object uses to access the contents of its data source object.

An outline view has the following features:

- A user can expand and collapse rows.
- Each item in the outline view must be unique. In order for the collapsed state to remain consistent between reloads the item's pointer must remain the same and the item must maintain `isEqual:` sameness.
- The view gets data from a data source (see `NSOutlineViewDataSource Protocol`).
- The view retrieves only the data that needs to be displayed.

Important: It is possible that your data source methods for populating the outline view may be called before `awakeFromNib` if the data source is specified in Interface Builder. You should defend against this by having the data source's `outlineView:numberOfChildrenOfItem:` method return `0` for the number of items when the data source has not yet been configured. In `awakeFromNib`, when the data source is initialized you should always call `reloadData`.

Subclassing

Subclassing `NSOutlineView` is not recommended. Customization can be accomplished in your data source class implementation (conforming to `NSOutlineViewDataSource Protocol`) or your delegate class implementation (conforming to `NSOutlineViewDelegate Protocol`).

Tasks

Setting the Data Source

– `setDataSource:` (page 26)

Sets the receiver's data source to a given object.

– `dataSource` (page 12)

Returns the object that provides the data displayed by the receiver.

Working with Expandability

– `isExpandable:` (page 16)

Returns a Boolean value that indicates whether a given item is expandable.

– `isItemExpanded:` (page 17)

Returns a Boolean value that indicates whether a given item is expanded.

Expanding and Collapsing the Outline

– `expandItem:` (page 13)

Expands a given item.

– `expandItem:expandChildren:` (page 13)

Expands a specified item and, optionally, its children.

– `collapseItem:` (page 10)

Collapses a given item.

– `collapseItem:collapseChildren:` (page 11)

Collapses a given item and, optionally, its children.

Redisplaying Information

– `reloadItem:` (page 21)

Reloads and redisplays the data for the given item.

– `reloadItem:reloadChildren:` (page 22)

Reloads a given item and, optionally, its children.

Converting Between Items and Rows

– `itemAtRow:` (page 17)

Returns the item associated with a given row.

– `rowForItem:` (page 24)

Returns the row associated with a given item.

Working with the Outline Column

– `setOutlineTableColumn:` (page 28)

Sets the table column in which hierarchical data is displayed.

– `outlineTableColumn` (page 20)

Returns the table column in which hierarchical data is displayed.

– `autoresizesOutlineColumn` (page 9)

Returns a Boolean value that indicates whether the receiver automatically resizes its outline column when the user expands or collapses items.

– `setAutoresizesOutlineColumn:` (page 24)

Sets whether the receiver automatically resizes its outline column when the user expands or collapses an item.

Working with Indentation

– `levelForItem:` (page 18)

Returns the indentation level for a given item.

– `levelForRow:` (page 19)

Returns the indentation level for a given row.

– `setIndentationPerLevel:` (page 28)

Sets the per-level indentation.

– `indentationPerLevel` (page 15)

Returns the current indentation per level.

– `setIndentationMarkerFollowsCell:` (page 27)

Sets whether the indentation marker symbol displayed in the outline column should be indented along with the cell contents, or always displayed left-justified in the column.

– `indentationMarkerFollowsCell` (page 14)

Returns a Boolean value that indicates whether the indentation marker symbol displayed in the outline column should be indented along with the cell contents, or always displayed left-justified in the column.

Working with Persistence

– `autosaveExpandedItems` (page 10)

Returns a Boolean value that indicates whether the expanded items in the receiver are automatically saved across launches of the app containing the outline view.

– `setAutosaveExpandedItems:` (page 25)
    Sets whether the expanded items in the receiver are automatically saved across launches of the app containing the outline view.

Supporting Drag and Drop

– `setDropItem:dropChildIndex:` (page 27)
    Used to “retarget” a proposed drop.

– `shouldCollapseAutoExpandedItemsForDeposited:` (page 29)
    Returns a Boolean value that indicates whether auto-expanded items should return to their original collapsed state.

Getting the Parent for an Item

– `parentForItem:` (page 21)
    Returns the parent for a given item.

Getting the Frame for a Cell

– `frameOfOutlineCellAtRow:` (page 14)
    Returns the frame of the outline cell for a given row.

Getting and Setting the Delegate

– `delegate` (page 12)
    Returns the receiver’s delegate.

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

Manipulating Items

– `insertItemsAtIndexes:inParent:withAnimation:` (page 15)
    Inserts new items at the given indexes in the given parent with the specified optional animations.

– `moveItemAtIndex:inParent:toIndex:inParent:` (page 19)
Moves an item at a given index in the given parent to a new index in a new parent.

– `removeItemsAtIndexes:inParent:withAnimation:` (page 23)
Removes items at the given indexes in the given parent with the specified optional animations.

User Interface Layout Direction

– `userInterfaceLayoutDirection` (page 30)
Returns the user interface layout direction.

– `setUserInterfaceLayoutDirection:` (page 29)
Sets the user interface layout direction.

Instance Methods

autoresizesOutlineColumn

Returns a Boolean value that indicates whether the receiver automatically resizes its outline column when the user expands or collapses items.

```
- (BOOL)autoresizesOutlineColumn
```

Return Value
`YES` if the outline column is automatically resized, otherwise `NO`.

Discussion
The outline column contains the cells with the expansion symbols and is generally the first column. The default is `YES`.

Availability
Available in OS X v10.0 and later.

See Also
– `setAutoresizesOutlineColumn:` (page 24)
– `outlineTableColumn` (page 20)

Declared in
`NSOutlineView.h`

autosaveExpandedItems

Returns a Boolean value that indicates whether the expanded items in the receiver are automatically saved across launches of the app containing the outline view.

```
- (BOOL)autosaveExpandedItems
```

Return Value

`YES` if when an item is expanded, the outline view displays the previous expanded state of its contained items, otherwise `NO`.

Discussion

The outline view information is saved separately for each user and for each application that user uses. Note that if `autosaveName` returns `nil`, this setting is ignored, and outline information isn't saved.

Special Considerations

Starting in OS X version 10.5, the value for `autosaveExpandedItems` is saved out in the nib file. The default value is `NO`.

Availability

Available in OS X v10.0 and later.

See Also

`autosaveName` (`NSTableView`)
`autosaveTableColumns` (`NSTableView`)
`– setAutosaveExpandedItems:` (page 25)

Declared in

`NSOutlineView.h`

collapseItem:

Collapses a given item.

```
- (void)collapseItem:(id)item
```

Parameters

`item`

An item in the receiver.

Discussion

If `item` is not expanded or not expandable, does nothing

If collapsing takes place, posts item collapse notification.

Availability

Available in OS X v10.0 and later.

See Also

– `expandItem:` (page 13)

Related Sample Code
DragNDropOutlineView

Declared in

`NSOutlineView.h`

collapseItem:collapseChildren:

Collapses a given item and, optionally, its children.

```
- (void)collapseItem:(id)item collapseChildren:(BOOL)collapseChildren
```

Parameters

`item`

An item in the receiver.

Starting in OS X version 10.5, passing `'nil'` will collapse each item under the root in the outline view.

`collapseChildren`

If `YES`, recursively collapses `item` and its children. If `NO`, collapses `item` only (identical to `collapseItem:` (page 10)).

Discussion

For example, this method is invoked with the `collapseChildren` parameter set to `YES` when a user Option-clicks the disclosure triangle for an item in the outline view (to collapse the item and all its contained items).

For each item collapsed, posts an item collapsed notification.

Availability

Available in OS X v10.0 and later.

See Also

– `collapseItem:` (page 10)
– `expandItem:expandChildren:` (page 13)

Declared in

`NSOutlineView.h`

dataSource

Returns the object that provides the data displayed by the receiver.

```
- (id<NSOutlineViewDataSource>)dataSource
```

Return Value

The object that provides the data displayed by the receiver.

Discussion

See “Writing an Outline View Data Source” and the NSOutlineViewDataSource Protocol informal protocol specification for more information.

Availability

Available in OS X v10.6 and later.

See Also

– `setDataSource:` (page 26)

Declared in

`NSOutlineView.h`

delegate

Returns the receiver’s delegate.

```
- (id<NSOutlineViewDelegate>)delegate
```

Return Value

The receiver’s delegate.

Availability

Available in OS X v10.6 and later.

See Also

– `setDelegate:` (page 26)

Declared in

`NSOutlineView.h`

expandItem:

Expands a given item.

```
- (void)expandItem:(id)item
```

Parameters

`item`

An item in the receiver.

Discussion

If `item` is not expandable or is already expanded, does nothing.

If expanding takes place, posts an item expanded notification.

Availability

Available in OS X v10.0 and later.

See Also

`– collapseItem:` (page 10)

Declared in

`NSOutlineView.h`

expandItem:expandChildren:

Expands a specified item and, optionally, its children.

```
- (void)expandItem:(id)item expandChildren:(BOOL)expandChildren
```

Parameters

`item`

An item in the receiver.

Starting in OS X version 10.5, passing `'nil'` will expand each item under the root in the outline view.

`expandChildren`

If `YES`, recursively expands `item` and its children. If `NO`, expands `item` only (identical to `expandItem:` (page 13)).

Discussion

For example, this method is invoked with the `expandChildren` parameter set to `YES` when a user Option-clicks the disclosure triangle for an item in the outline view (to expand the item and all its contained items).

For each item expanded, posts an item expanded notification.

Availability

Available in OS X v10.0 and later.

See Also

– `collapseItem:collapseChildren:` (page 11)
– `expandItem:` (page 13)

Declared in

`NSOutlineView.h`

frameOfOutlineCellAtRow:

Returns the frame of the outline cell for a given row.

```
- (NSRect)frameOfOutlineCellAtRow:(NSInteger)row
```

Parameters

`row`

The index of the row for which to return the frame.

Return Value

The frame of the outline cell for the row at index `row`, considering the current indentation and the value returned by `indentationMarkerFollowsCell` (page 14). If the row at index `row` is not an expandable row, returns `NSZeroRect`.

Discussion

You can override this method in a subclass to return a custom frame for the outline button cell. If your override returns an empty rect, no outline cell is drawn for that row. You might do that, for example, so that the disclosure triangle will not be shown for a row that should never be expanded.

Availability

Available in OS X v10.5 and later.

Declared in

`NSOutlineView.h`

indentationMarkerFollowsCell

Returns a Boolean value that indicates whether the indentation marker symbol displayed in the outline column should be indented along with the cell contents, or always displayed left-justified in the column.

```
- (BOOL)indentationMarkerFollowsCell
```

Return Value

YES if the indentation marker is indented along with the cell contents, otherwise NO.

Discussion

The default is YES.

Availability

Available in OS X v10.0 and later.

See Also

– setIndentationMarkerFollowsCell: (page 27)

Declared in

NSOutlineView.h

indentationPerLevel

Returns the current indentation per level.

```
- (CGFloat)indentationPerLevel
```

Return Value

The current indentation per level, in points.

Availability

Available in OS X v10.0 and later.

See Also

– setIndentationPerLevel: (page 28)

Declared in

NSOutlineView.h

insertItemsAtIndexes:inParent:withAnimation:

Inserts new items at the given indexes in the given parent with the specified optional animations.

```
- (void)insertItemsAtIndexes:(NSIndexSet *)indexes inParent:(id)parent
withAnimation:(NSTableViewAnimationOptions)animationOptions
```

Parameters

indexes

Indexes at which to insert items.

`parent`

The parent for the items, or `nil` if the parent is the root.

`animationOptions`

Animated slide effects used when inserting items.

Discussion

This method parallels the `NSTableView` method `insertRowsAtIndexes:withAnimation:` and is used in a way similar to `NSMutableArray` method `insertObjects:atIndexes:`. The method does nothing if `parent` is not expanded. The actual item values are determined by the `NSOutlineViewDataSource` method `outlineView:child:ofItem:` (which is called only after `endUpdates` to ensure data source integrity).

> **Note:** NSCell-based outline views must first call `beginUpdates` before calling this method.

You can call this method multiple times within the same `beginUpdates/endUpdates` block; new insertions move previously inserted new items, just like modifying an array. Inserting an index beyond what is available throws an exception.

Availability

Available in OS X v10.7 and later.

See Also

– `moveItemAtIndex:inParent:toIndex:inParent:` (page 19)
– `removeItemsAtIndexes:inParent:withAnimation:` (page 23)

Related Sample Code

DragNDropOutlineView

TableViewPlayground

Declared in

`NSOutlineView.h`

isExpandable:

Returns a Boolean value that indicates whether a given item is expandable.

```
- (BOOL)isExpandable:(id)item
```

Parameters

`item`

An item in the receiver.

Return Value

YES if item is expandable—that is, item can contain other items, otherwise NO.

Availability

Available in OS X v10.0 and later.

See Also

– expandItem: (page 13)
– isItemExpanded: (page 17)

Declared in

NSOutlineView.h

isItemExpanded:

Returns a Boolean value that indicates whether a given item is expanded.

```
– (BOOL)isItemExpanded:(id)item
```

Parameters

item

An item in the receiver.

Return Value

YES if item is expanded, otherwise NO.

Availability

Available in OS X v10.0 and later.

See Also

– expandItem: (page 13)
– isExpandable: (page 16)

Declared in

NSOutlineView.h

itemAtRow:

Returns the item associated with a given row.

```
– (id)itemAtRow:(NSInteger)row
```

Parameters

`row`

The index of a row in the receiver.

Return Value

The item associated with `row`.

Availability

Available in OS X v10.0 and later.

See Also

– `rowForItem:` (page 24)

Related Sample Code

DragNDropOutlineView

TableViewPlayground

Declared in

`NSOutlineView.h`

levelForItem:

Returns the indentation level for a given item.

```
– (NSInteger)levelForItem:(id)item
```

Parameters

`item`

An item in the receiver.

Return Value

The indentation level for `item`. If `item` is `nil` (which is the root item), returns –1.

Discussion

The levels are zero-based—that is, the first level of displayed items is level `0`.

Availability

Available in OS X v10.0 and later.

See Also

– `indentationPerLevel` (page 15)

– `levelForRow:` (page 19)

Declared in

`NSOutlineView.h`

levelForRow:

Returns the indentation level for a given row.

```
- (NSInteger)levelForRow:(NSInteger)row
```

Parameters

`row`

The index of a row in the receiver.

Return Value

The indentation level for `row`. For an invalid row, returns –1.

Discussion

The levels are zero-based—that is, the first level of displayed items is level `0`.

Availability

Available in OS X v10.0 and later.

See Also

– `indentationPerLevel` (page 15)
– `levelForItem:` (page 18)

Declared in

`NSOutlineView.h`

moveItemAtIndex:inParent:toIndex:inParent:

Moves an item at a given index in the given parent to a new index in a new parent.

```
- (void)moveItemAtIndex:(NSInteger)fromIndex inParent:(id)oldParent
toIndex:(NSInteger)toIndex inParent:(id)newParent
```

Parameters

`fromIndex`

Index of the item to be moved.

`oldParent`

The parent of the item to be moved.

`toIndex`
: Index in the new parent to which the item is moved.

`newParent`
: The parent of the item after it is moved.

Discussion

This method parallels the `NSTableView` method `moveRowAtIndex:toIndex:`. The `newParent` can be the same as `oldParent` to reorder an item within the same parent.

> **Note:** NSCell-based outline views must first call `beginUpdates` before calling this method.

You can call this method multiple times within the same `beginUpdates/endUpdates` block. Moving from an invalid index, or to an invalid index, throws an exception.

Availability

Available in OS X v10.7 and later.

See Also

– `removeItemsAtIndexes:inParent:withAnimation:` (page 23)
– `insertItemsAtIndexes:inParent:withAnimation:` (page 15)

Related Sample Code

TableViewPlayground

Declared in

`NSOutlineView.h`

outlineTableColumn

Returns the table column in which hierarchical data is displayed.

```
- (NSTableColumn *)outlineTableColumn
```

Return Value

The table column in which hierarchical data is displayed. A `nil` outline table column is silently ignored.

Discussion

Each level of hierarchical data is indented by the amount specified by `indentationPerLevel` (page 15) (the default is 16.0), and decorated with the indentation marker (disclosure triangle) on rows that are expandable.

Special Considerations

Starting in OS X version 10.5, outline table column data is saved in `encodeWithCoder:` and restored in `initWithCoder:`.

Availability

Available in OS X v10.0 and later.

See Also

– `setOutlineTableColumn:` (page 28)
– `autoresizesOutlineColumn` (page 9)

Declared in

`NSOutlineView.h`

parentForItem:

Returns the parent for a given item.

```
- (id)parentForItem:(id)item
```

Parameters

`item`

The item for which to return the parent.

Return Value

The parent for `item`, or `nil` if the parent is the root.

Availability

Available in OS X v10.5 and later.

Related Sample Code

SidebarDemo

TableViewPlayground

Declared in

`NSOutlineView.h`

reloadItem:

Reloads and redisplays the data for the given item.

```
- (void)reloadItem:(id)item
```

Parameters

`item`

The item to reload and display.

Discussion

This method may cause the outline view to change its selection without invoking the `outlineViewSelectionDidChange:` delegate method.

Availability

Available in OS X v10.0 and later.

See Also

– `reloadItem:reloadChildren:` (page 22)

Declared in

`NSOutlineView.h`

reloadItem:reloadChildren:

Reloads a given item and, optionally, its children.

```
– (void)reloadItem:(id)item reloadChildren:(BOOL)reloadChildren
```

Parameters

`item`

An item in the receiver.

Starting in OS X version 10.5, passing `'nil'` will reload everything under the root in the outline view.

`reloadChildren`

If `YES`, recursively reloads `item` and its children. If `NO`, reloads `item` only (identical to `reloadItem:` (page 21)).

It is not necessary, or efficient, to reload children if the item is not expanded.

Availability

Available in OS X v10.0 and later.

See Also

– `reloadItem:` (page 21)

Declared in

`NSOutlineView.h`

removeItemsAtIndexes:inParent:withAnimation:

Removes items at the given indexes in the given parent with the specified optional animations.

```
- (void)removeItemsAtIndexes:(NSIndexSet *)indexes inParent:(id)parent
withAnimation:(NSTableViewAnimationOptions)animationOptions
```

Parameters

`indexes`

Indexes of the items to be removed.

`parent`

The parent of the items to be removed.

`animationOptions`

Animated slide effects used when removing items.

Discussion

This method parallels the `NSTableView` method `removeRowsAtIndexes:withAnimation:` and is used in a way similar to `NSMutableArray` method `removeObjectsAtIndexes:`. The method does nothing if `parent` is not expanded. If any of the child items is expanded, then all of its child rows are also be removed.

> **Note:** NSCell-based outline views must first call `beginUpdates` before calling this method.

You can call this method multiple times within the same `beginUpdates/endUpdates` block; changes work just like modifying an array. Removing an item at an index beyond what is available throws an exception.

Availability

Available in OS X v10.7 and later.

See Also

– `moveItemAtIndex:inParent:toIndex:inParent:` (page 19)
– `insertItemsAtIndexes:inParent:withAnimation:` (page 15)

Related Sample Code

DragNDropOutlineView

TableViewPlayground

Declared in

`NSOutlineView.h`

rowForItem:

Returns the row associated with a given item.

```
- (NSInteger)rowForItem:(id)item
```

Parameters

`item`

An item in the receiver.

Return Value

The row associated with `item`, or –1 if `item` is `nil` or cannot be found.

Availability

Available in OS X v10.0 and later.

See Also

– `itemAtRow:` (page 17)

Related Sample Code

DragNDropOutlineView

Declared in

`NSOutlineView.h`

setAutoresizesOutlineColumn:

Sets whether the receiver automatically resizes its outline column when the user expands or collapses an item.

```
- (void)setAutoresizesOutlineColumn:(BOOL)resize
```

Parameters

`resize`

`YES` if the outline column is automatically resized, otherwise `NO`.

Discussion

The outline column contains the cells with the expansion symbols and is generally the first column. The default is `YES`.

The outline column is resized based on how many indentation levels are exposed or hidden (for example, if expanding a row exposes a single indentation level, the outline column width is increased by one `indentationPerLevel` (page 15)).

Availability

Available in OS X v10.0 and later.

See Also

– `autoresizesOutlineColumn` (page 9)
– `outlineTableColumn` (page 20)

Declared in

`NSOutlineView.h`

setAutosaveExpandedItems:

Sets whether the expanded items in the receiver are automatically saved across launches of the app containing the outline view.

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

Discussion

If `flag` is different from the current value, this method also reads in the saved information and sets the outline view's options to match. `YES` indicates that when an item is expanded, the outline view displays the previous expanded state of its contained items.

The outline information is saved separately for each user and for each application that user uses.

If `autosaveName` returns `nil` or if you haven't implemented the data source methods `outlineView:itemForPersistentObject:` and `outlineView:persistentObjectForItem:`, this setting is ignored, and expanded item information isn't saved.

Note that you can have separate settings for `autosaveExpandedItems` (page 10) and `autosaveTableColumns`, so you could, for example, save expanded item information, but not table column positions.

Special Considerations

Starting in OS X version 10.5, the value for `autosaveExpandedItems` is saved out in the nib file. The default value is `NO`.

Availability

Available in OS X v10.0 and later.

See Also

– `autosaveExpandedItems` (page 10)
`setAutosaveTableColumns:` (NSTableView)

Declared in

`NSOutlineView.h`

setDataSource:

Sets the receiver's data source to a given object.

```
- (void)setDataSource:(id<NSOutlineViewDataSource>)anObject
```

Parameters

`anObject`

The data source for the receiver. The object must implement the appropriate methods of NSOutlineViewDataSource Protocol.

Discussion

The receiver maintains a weak reference to the data source (see "Encapsulating Data"). After setting the data source, this method invokes `tile`.

This method raises an `NSInternalInconsistencyException` if `anObject` doesn't respond to all of `outlineView:child:ofItem:`, `outlineView:isItemExpandable:`, `outlineView:numberOfChildrenOfItem:`, and `outlineView:objectValueForTableColumn:byItem:`.

Availability

Available in OS X v10.6 and later.

See Also

– `dataSource` (page 12)

Declared in

`NSOutlineView.h`

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id<NSOutlineViewDelegate>)anObject
```

Parameters

`anObject`

The delegate for the receiver. The delegate must conform to the `NSOutlineViewDelegate Protocol` protocol.

Availability

Available in OS X v10.6 and later.

See Also

– `delegate` (page 12)

Declared in

`NSOutlineView.h`

setDropItem:dropChildIndex:

Used to "retarget" a proposed drop.

```
- (void)setDropItem:(id)item dropChildIndex:(NSInteger)index
```

Parameters

`item`

The target item.

`index`

The drop index.

Discussion

For example, to specify a drop on `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as `NSOutlineViewDropOnItemIndex`. To specify a drop between child `2` and `3` of `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as `3` (children are a zero-based index). To specify a drop on an un-expandable `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as `NSOutlineViewDropOnItemIndex`.

Availability

Available in OS X v10.0 and later.

Declared in

`NSOutlineView.h`

setIndentationMarkerFollowsCell:

Sets whether the indentation marker symbol displayed in the outline column should be indented along with the cell contents, or always displayed left-justified in the column.

```
- (void)setIndentationMarkerFollowsCell:(BOOL)drawInCell
```

Discussion

The default is `YES`, the indentation marker is indented along with the cell contents.

Availability

Available in OS X v10.0 and later.

See Also

– `indentationMarkerFollowsCell` (page 14)

Declared in

`NSOutlineView.h`

setIndentationPerLevel:

Sets the per-level indentation.

```
– (void)setIndentationPerLevel:(CGFloat)newIndentLevel
```

Parameters

`newIndentLevel`

The indentation per level, in points.

Availability

Available in OS X v10.0 and later.

See Also

– `indentationPerLevel` (page 15)

Declared in

`NSOutlineView.h`

setOutlineTableColumn:

Sets the table column in which hierarchical data is displayed.

```
– (void)setOutlineTableColumn:(NSTableColumn *)outlineTableColumn
```

Parameters

`outlineTableColumn`

The table column in which hierarchical data is displayed.

Special Considerations

Starting in OS X version 10.5, outline table column data is saved in `encodeWithCoder:` and restored in `initWithCoder:`.

Availability

Available in OS X v10.0 and later.

See Also

– `outlineTableColumn` (page 20)
– `autoresizesOutlineColumn` (page 9)

Declared in

`NSOutlineView.h`

setUserInterfaceLayoutDirection:

Sets the user interface layout direction.

```
- (void)setUserInterfaceLayoutDirection:(NSUserInterfaceLayoutDirection)value
```

Parameters

`value`

The new user interface layout direction.

Discussion

When set to `NSUserInterfaceLayoutDirectionRightToLeft`, the outline view displays the disclosure triangle to the right of the cell instead of the left. The default value is `NSUserInterfaceLayoutDirectionLeftToRight`.

Availability

Available in OS X v10.7 and later.

See Also

– `userInterfaceLayoutDirection` (page 30)

Declared in

`NSOutlineView.h`

shouldCollapseAutoExpandedItemsForDeposited:

Returns a Boolean value that indicates whether auto-expanded items should return to their original collapsed state.

```
- (BOOL)shouldCollapseAutoExpandedItemsForDeposited:(BOOL)deposited
```

Parameters

`deposited`

If `YES`, the drop terminated successfully; if `NO` the drop failed.

Return Value

`YES` if auto-expanded items should return to their original collapsed state; otherwise `NO`.

Discussion

Override this method to provide custom behavior. If the target of a drop is not auto-expanded (by hovering long enough) the drop target still gets expanded after a successful drop unless this method returns `YES`. The default implementation returns `NO` after a successful drop.

This method is called in a variety of situations. For example, it is sent shortly after `outlineView:acceptDrop:item:childIndex:` is processed and also if the drag exits the outline view (exiting the view is treated the same as a failed drop). The return value of `outlineView:acceptDrop:item:childIndex:` determines the incoming value of the `deposited` parameter.

Availability

Available in OS X v10.0 and later.

Declared in

`NSOutlineView.h`

userInterfaceLayoutDirection

Returns the user interface layout direction.

```
- (NSUserInterfaceLayoutDirection)userInterfaceLayoutDirection
```

Return Value

The current user interface layout direction.

Discussion

When set to `NSUserInterfaceLayoutDirectionRightToLeft`, the outline view displays the disclosure triangle to the right of the cell instead of the left. The default value is `NSUserInterfaceLayoutDirectionLeftToRight`.

Availability

Available in OS X v10.7 and later.

See Also

– `setUserInterfaceLayoutDirection:` (page 29)

Declared in

`NSOutlineView.h`

Constants

Drop on Item Index

This constant defines an index that allows you to drop an item directly on a target.

```
enum {
   NSOutlineViewDropOnItemIndex = -1
};
```

Constants

`NSOutlineViewDropOnItemIndex`

May be used as a valid child index of a drop target item.

In this case, the drop will happen directly on the target item.

Available in OS X v10.0 and later.

Declared in `NSOutlineView.h`.

Outline View Button Keys

These keys are used by the outline view to create disclosure buttons that collapse and expand items.

```
NSString *const NSOutlineViewDisclosureButtonKey;
NSString *const NSOutlineViewShowHideButtonKey;
```

Constants

`NSOutlineViewDisclosureButtonKey`

The normal triangle disclosure button.

Available in OS X v10.9 and later.

Declared in `NSOutlineView.h`.

`NSOutlineViewShowHideButtonKey`

The Show/Hide button.

Available in OS X v10.9 and later.

Declared in `NSOutlineView.h`.

Discussion

The outline view creates these buttons by calling its inherited `makeViewWithIdentifier:owner:` method, passing in the key as the identifier and the delegate as the owner.

> **Note:** These keys are backwards compatible to OS X v10.7, however, the symbol is not exported prior to v10.9 and the string value (`@"NSOutlineViewDisclosureButtonKey"`) must be used.

Notifications

NSOutlineViewColumnDidMoveNotification

Posted whenever a column is moved by user action in an `NSOutlineView` object.

The notification object is the `NSOutlineView` object in which a column moved. The `userInfo` dictionary contains the following information:

Key	Value
`@"NSOldColumn"`	An `NSNumber` object containing the integer value of the column’s original index
`@"NSNewColumn"`	An `NSNumber` object containing the integer value of the column’s present index

Availability

Available in OS X v10.0 and later.

See Also

`moveColumn:toColumn:` (`NSTableView`)

Declared in

`NSOutlineView.h`

NSOutlineViewColumnDidResizeNotification

Posted whenever a column is resized in an `NSOutlineView` object.

The notification object is the `NSOutlineView` object in which a column was resized. The `userInfo` dictionary contains the following information:

Key	Value
`@"NSTableColumn"`	The column that was resized.
`@"NSOldWidth"`	An `NSNumber` object containing the column's original width

Availability
Available in OS X v10.0 and later.

Declared in
`NSOutlineView.h`

NSOutlineViewItemDidCollapseNotification

Posted whenever an item is collapsed in an `NSOutlineView` object.

The notification object is the `NSOutlineView` object in which an item was collapsed. A collapsed item's children lose their status as being selected. The `userInfo` dictionary contains the following information:

Key	Value
`@"NSObject"`	The item that was collapsed (an `id`)

Availability
Available in OS X v10.0 and later.

Declared in
`NSOutlineView.h`

NSOutlineViewItemDidExpandNotification

Posted whenever an item is expanded in an `NSOutlineView` object.

The notification object is the NSOutlineView object in which an item was expanded. The userInfo dictionary contains the following information:

Key	Value
@"NSObject"	The item that was expanded (an id)

Availability
Available in OS X v10.0 and later.

Declared in
NSOutlineView.h

NSOutlineViewItemWillCollapseNotification

Posted before an item is collapsed (after the user clicks the arrow but before the item is collapsed).

The notification object is the NSOutlineView object that contains the item about to be collapsed. A collapsed item's children will lose their status as being selected. The userInfo dictionary contains the following information:

Key	Value
@"NSObject"	The item about to be collapsed (an id)

Availability
Available in OS X v10.0 and later.

Declared in
NSOutlineView.h

NSOutlineViewItemWillExpandNotification

Posted before an item is expanded (after the user clicks the arrow but before the item is collapsed).

The notification object is the outline view that contains an item about to be expanded. The userInfo dictionary contains the following information:

Key	Value
`@"NSObject"`	The item that is to be expanded (an `id`)

Availability
Available in OS X v10.0 and later.

Declared in
`NSOutlineView.h`

NSOutlineViewSelectionDidChangeNotification

Posted after the outline view's selection changes.

The notification object is the outline view whose selection changed. This notification does not contain a `userInfo` dictionary.

Availability
Available in OS X v10.0 and later.

Declared in
`NSOutlineView.h`

NSOutlineViewSelectionIsChangingNotification

Posted as the outline view's selection changes (while the mouse button is still down).

The notification object is the outline view whose selection is changing. This notification does not contain a `userInfo` dictionary.

Availability
Available in OS X v10.0 and later.

Declared in
`NSOutlineView.h`

Document Revision History

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

Date	Notes
2013-12-16	Updated for OS X v10.9. Clarified abstracts of autosaveExpandedItems (page 10) and setAutosaveExpandedItems: (page 25) methods. Clarified descriptions of setAutoresizesOutlineColumn: (page 24), reloadItem: (page 21), and shouldCollapseAutoExpandedItemsForDeposited: (page 29) methods.
2011-01-13	Updated for OS X v10.7. Delegate methods moved to NSOutlineViewDelegate protocol.
2010-05-13	Edited setDropItem:dropChildIndex: description.
2009-11-17	Clarified that delegates are mixed with the class and protocol. Added warning to reloadItem: about selection bug.
2009-05-29	Updated for OS X v10.6. Delegate methods moved to NSOutlineViewDelegate Protocol Reference.
2009-02-04	Added documentation for method outlineView:selectionIndexesForProposedSelection:.
2007-07-26	Clarified definitions for several methods.

Date	Notes
	Noted changes for Mac OS version 10.5 in the following methods: autosaveExpandedItems (page 10), setAutosaveExpandedItems: (page 25), collapseItem:collapseChildren: (page 11), expandItem:expandChildren: (page 13), reloadItem:reloadChildren: (page 22), outlineTableColumn (page 20), setOutlineTableColumn: (page 28), frameOfOutlineCellAtRow: (page 14),outlineView:shouldCollapseItem: and setDelegate: (page 26).
2006-11-07	Added note about equality of items in an NSOutlineView object.
2006-06-28	Added a definition of data source methods.
2006-05-23	First publication of this content as a separate document. Added information regarding how items are expanded and collapsed to the following method descriptions: autosaveExpandedItems (page 10), setAutosaveExpandedItems: (page 25), collapseItem:collapseChildren: (page 11), and expandItem:expandChildren: (page 13).

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

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.