

# NSOutlineViewDelegate Protocol Reference

# Contents

## **NSOutlineViewDelegate Protocol Reference** 4

Overview 4

Tasks 4

Expanding and Collapsing the Outline 4

Supporting Type Select 4

Working with Tooltips 5

Handling Selection 5

Displaying Cells 5

Moving and Resizing Columns 6

Working with the Outline Column 6

Editing Items 7

Working with Table Columns 7

Customizing Column and Row Sizes 7

Customizing Tracking Support 7

Grouping Rows 7

Working with NSView-Based Outline Views 8

Instance Methods 8

outlineView:dataCellForTableColumn:item: 8

outlineView:didAddRowView:forRow: 9

outlineView:didClickTableColumn: 10

outlineView:didDragTableColumn: 10

outlineView:didRemoveRowView:forRow: 11

outlineView:heightOfRowByItem: 11

outlineView:isGroupItem: 12

outlineView:mouseDownInHeaderOfTableColumn: 13

outlineView:nextTypeSelectMatchFromItem:toItem:forString: 13

outlineView:rowViewForItem: 14

outlineView:selectionIndexesForProposedSelection: 15

outlineView:shouldCollapseItem: 16

outlineView:shouldEditTableColumn:item: 16

outlineView:shouldExpandItem: 17

outlineView:shouldReorderColumn:toColumn: 18

outlineView:shouldSelectItem: 19

outlineView:shouldSelectTableColumn: 19

|   |    |
|---|----|
| outlineView:shouldShowCellExpansionForTableColumn:item:         | 20 |
| outlineView:shouldShowOutlineCellForItem:                       | 21 |
| outlineView:shouldTrackCell:forTableColumn:item:                | 21 |
| outlineView:shouldTypeSelectForEvent:withCurrentSearchString:   | 22 |
| outlineView:sizeToFitWidthOfColumn:                             | 23 |
| outlineView:toolTipForCell:rect:tableColumn:item:mouseLocation: | 24 |
| outlineView:typeSelectStringForTableColumn:item:                | 24 |
| outlineView:viewForTableColumn:item:                            | 25 |
| outlineView:willDisplayCell:forTableColumn:item:                | 26 |
| outlineView:willDisplayOutlineCell:forTableColumn:item:         | 27 |
| outlineViewColumnDidMove:                                       | 28 |
| outlineViewColumnDidResize:                                     | 28 |
| outlineViewItemDidCollapse:                                     | 29 |
| outlineViewItemDidExpand:                                       | 29 |
| outlineViewItemWillCollapse:                                    | 30 |
| outlineViewItemWillExpand:                                      | 30 |
| outlineViewSelectionDidChange:                                  | 31 |
| outlineViewSelectionIsChanging:                                 | 31 |
| selectionShouldChangeInOutlineView:                             | 32 |

## **Document Revision History** 33

# NSOutlineViewDelegate Protocol Reference

|                     |   |
|---------------------|---|
| Conforms to         | NSControlTextEditingDelegate  |
| Framework           | /System/Library/Frameworks/AppKit.framework                         |
| Availability        | Available in OS X v10.6 and later.                                  |
| Declared in         | NSOutlineView.h   |
| Companion guides    | Outline View Programming Topics<br>Drag and Drop Programming Topics |
| Related sample code | SidebarDemo   |

## Overview

The `NSOutlineViewDelegate` protocol defines the optional methods implemented by delegates of `NSOutlineView` objects.

## Tasks

### Expanding and Collapsing the Outline

- [outlineView:shouldExpandItem:](#) (page 17)  
Returns a Boolean value that indicates whether the outline view should expand a given item.
- [outlineView:shouldCollapseItem:](#) (page 16)  
Returns a Boolean value that indicates whether the outline view should collapse a given item.

### Supporting Type Select

- [outlineView:typeSelectStringForTableColumn:item:](#) (page 24)  
Returns the string that is used for type selection for a given column and item.

- [outlineView:nextTypeSelectMatchFromItem:toItem:forString:](#) (page 13)  
Returns the first item that matches the searchString from within the range of startItem to endItem
- [outlineView:shouldTypeSelectForEvent:withCurrentSearchString:](#) (page 22)  
Returns a Boolean value that indicates whether type select should proceed for a given event and search string.

## Working with Tooltips

---

- [outlineView:toolTipForCell:rect:tableColumn:item:mouseLocation:](#) (page 24)  
When the cursor pauses over a given cell, the value returned from this method is displayed in a tooltip.

## Handling Selection

---

- [outlineView:shouldSelectTableColumn:](#) (page 19)  
Returns a Boolean value that indicates whether the outline view should select a given table column.
- [outlineView:shouldSelectItem:](#) (page 19)  
Returns a Boolean value that indicates whether the outline view should select a given item.
- [outlineView:selectionIndexesForProposedSelection:](#) (page 15)  
Invoked to allow the delegate to modify the proposed selection.
- [selectionShouldChangeInOutlineView:](#) (page 32)  
Returns a Boolean value that indicates whether the outline view should change its selection.
- [outlineViewSelectionIsChanging:](#) (page 31)  
Invoked when notification is posted—that is, whenever the outline view’s selection changes.
- [outlineViewSelectionDidChange:](#) (page 31)  
Invoked when the selection did change notification is posted—that is, immediately after the outline view’s selection has changed.

## Displaying Cells

---

- [outlineView:willDisplayCell:forTableColumn:item:](#) (page 26)  
Informs the delegate that the cell specified by the column and item will be displayed.
- [outlineView:willDisplayOutlineCell:forTableColumn:item:](#) (page 27)  
Informs the delegate that an outline view is about to display a cell used to draw the expansion symbol.

- [outlineView:dataCellForTableColumn:item:](#) (page 8)  
Returns the cell to use in a given column for a given item.
- [outlineView:shouldShowOutlineCellForItem:](#) (page 21)  
Returns a whether the specified item should display the outline cell (the disclosure triangle).
- [outlineView:shouldShowCellExpansionForTableColumn:item:](#) (page 20)  
Invoked to allow the delegate to control cell expansion for a specific column and item.

## Moving and Resizing Columns

---

- [outlineView:shouldReorderColumn:toColumn:](#) (page 18)  
Sent to the delegate to allow or prohibit the specified column to be dragged to a new location.

## Working with the Outline Column

---

- [outlineViewColumnDidMove:](#) (page 28)  
Invoked whenever the user moves a column in the outline view.
- [outlineViewColumnDidResize:](#) (page 28)  
Invoked whenever the user resizes a column in the outline view.
- [outlineViewItemWillExpand:](#) (page 30)  
Invoked when notification is posted—that is, whenever the user is about to expand an item in the outline view.
- [outlineViewItemDidExpand:](#) (page 29)  
Invoked when notification is posted—that is, whenever the user expands an item in the outline view.
- [outlineViewItemWillCollapse:](#) (page 30)  
Invoked when notification is posted—that is, whenever the user is about to collapse an item in the outline view.
- [outlineViewItemDidCollapse:](#) (page 29)  
Invoked when the did collapse notification is posted—that is, whenever the user collapses an item in the outline view.

## Editing Items

---

- [outlineView:shouldEditTableColumn:item:](#) (page 16)  
Returns a Boolean value that indicates whether the outline view should allow editing of a given item in a given table column.

## Working with Table Columns

---

- [outlineView:mouseDownInHeaderOfTableColumn:](#) (page 13)  
Sent to the delegate whenever the mouse button is clicked in `outlineView` while the cursor is in a column header `tableColumn`.
- [outlineView:didClickTableColumn:](#) (page 10)  
Sent at the time the mouse button subsequently goes up in `outlineView` and `tableColumn` has been “clicked” without having been dragged anywhere.
- [outlineView:didDragTableColumn:](#) (page 10)  
Sent at the time the mouse button goes up in `outlineView` and `tableColumn` has been dragged during the time the mouse button was down.

## Customizing Column and Row Sizes

---

- [outlineView:heightOfRowByItem:](#) (page 11)  
Returns the height in points of the row containing `item`.
- [outlineView:sizeToFitWidthOfColumn:](#) (page 23)  
Invoked to allow the delegate to provide custom sizing behavior when a column’s resize divider is double clicked.

## Customizing Tracking Support

---

- [outlineView:shouldTrackCell:forTableColumn:item:](#) (page 21)  
Returns a Boolean value that indicates whether a given cell should be tracked.

## Grouping Rows

---

- [outlineView:isGroupItem:](#) (page 12)  
Returns a Boolean that indicates whether a given row should be drawn in the “group row” style.

## Working with NSView-Based Outline Views

---

- `outlineView:didAddRowView:forRow:` (page 9) *required method*  
Implemented to know when a new row view is added to the table. (required)
- `outlineView:didRemoveRowView:forRow:` (page 11) *required method*  
Implemented to know when a row view is removed from the table (required)
- `outlineView:rowViewForItem:` (page 14) *required method*  
implement this method to return a custom `NSTableViewRowView` for a particular item. (required)
- `outlineView:viewForTableColumn:item:` (page 25) *required method*  
Implemented to return the view used to display the specified item and column. (required)

## Instance Methods

### `outlineView:dataCellForTableColumn:item:`

---

*Returns the cell to use in a given column for a given item.*

```
– (NSCell *)outlineView:(NSOutlineView *)outlineView  
dataCellForTableColumn:(NSTableColumn *)tableColumn item:(id)item
```

#### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column for which the cell is required. This value may be `nil`.

`item`

The item for which the cell is required.

#### Return Value

The cell to use in column `tableColumn` for item `item`, or `nil`. The cell must properly implement `copyWithZone:` (since it may be copied by the outline view).

#### Discussion

You can return a different data cell for any particular combination of table column and item, or a cell that will be used for the entire row (a full-width cell). If `tableColumn` is non-`nil`, you should return a cell. Typically, you should default to returning the result from `[tableColumn dataCellForRow:row]`.



When each row (identified by the item) is being drawn, this method is first called with a `nil` value for `tableColumn`. At this time, you can return a cell that is used to draw the entire row, acting like a group. If you do return a cell for the `nil` table column, your implementations of the other corresponding data source and delegate methods must be prepared to be invoked with a `nil` value for `tableColumn`. If do not return a cell for the `nil` table column, the method is called once for each column in the outline view, as usual.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

NSOutlineView.h

---

## outlineView:didAddRowView:forRow:

---

*Implemented to know when a new row view is added to the table. (required)*

```
– (void)outlineView:(NSOutlineView *)outlineView didAddRowView:(NSTableRowView *)rowView forRow:(NSInteger)row
```

### Parameters

`outlineView`

The outline view that sent the message.

`rowView`

The new row view.

`row`

The row to which the view was added.

### Discussion

This delegate method is for `NSView`-based outline views. At this point, you can choose to add in extra views or modify any properties on `rowView`.

### Availability

Available in OS X v10.7 and later.

### Declared in

NSOutlineView.h

## **outlineView:didClickTableColumn:**

---

*Sent at the time the mouse button subsequently goes up in `outlineView` and `tableColumn` has been “clicked” without having been dragged anywhere.*

```
– (void)outlineView:(NSOutlineView *)outlineView didClickTableColumn:(NSTableColumn *)tableColumn
```

### **Parameters**

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column.

### **Availability**

Available in OS X v10.3 and later.

Available as part of an informal protocol prior to OS X v10.6.

### **Declared in**

`NSOutlineView.h`

## **outlineView:didDragTableColumn:**

---

*Sent at the time the mouse button goes up in `outlineView` and `tableColumn` has been dragged during the time the mouse button was down.*

```
– (void)outlineView:(NSOutlineView *)outlineView didDragTableColumn:(NSTableColumn *)tableColumn
```

### **Parameters**

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column.

### **Availability**

Available in OS X v10.3 and later.

Available as part of an informal protocol prior to OS X v10.6.

### **Declared in**

`NSOutlineView.h`

## outlineView:didRemoveRowView:forRow:

---

*Implemented to know when a row view is removed from the table (required)*

```
– (void)outlineView:(NSOutlineView *)outlineView didRemoveRowView:(NSTableRowView *)rowView forRow:(NSInteger)row
```

### Parameters

outlineView

The outline view that sent the message.

rowView

The row view that was removed.

row

The number of the row that was removed due to being moved offscreen, or –1 if the row was removed from the table so it is no longer valid.

### Discussion

The removed `rowView` may be reused by the table, so any additionally inserted views should be removed at this point.

### Availability

Available in OS X v10.7 and later.

### Declared in

NSOutlineView.h

## outlineView:heightOfRowByItem:

---

*Returns the height in points of the row containing item.*

```
– (CGFloat)outlineView:(NSOutlineView *)outlineView heightOfRowByItem:(id)item
```

### Parameters

outlineView

The outline view that sent the message.

item

The row item.

### Return Value

The height of the row.

## Discussion

Values returned by this method should not include intercell spacing and must be greater than 0.

Implement this method to support an outline view with varying row heights.

For large tables in particular, you should make sure that this method is efficient. `NSOutlineView` may cache the values this method returns, so if you would like to change a row's height make sure to invalidate the row height by calling `noteHeightOfRowsWithIndexesChanged:`. `NSOutlineView` automatically invalidates its entire row height cache in `reloadData` and `noteNumberOfRowsChanged`.

If you call `viewAtColumn:row:makeIfNecessary:` or `rowViewAtRow:makeIfNecessary:` within your implementation of this method, an exception is thrown.

**Important:** To avoid the possibility of a hang due to unexpected recursion, don't call geometry-calculating methods such as `bounds`, `rectOfColumn:`, or any `NSTableView` method that calls `tile` within your implementation of this method.

## Availability

Available in OS X v10.4 and later.

Available as part of an informal protocol prior to OS X v10.6.

## Declared in

`NSOutlineView.h`

---

## **outlineView:isGroupItem:**

*Returns a Boolean that indicates whether a given row should be drawn in the "group row" style.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView isGroupItem:(id)item

## Parameters

`outlineView`

The outline view that sent the message.

`item`

An item in the outline view.

## Return Value

YES to indicate a particular row should have the "group row" style drawn for that row, otherwise NO.

### Discussion

If the cell in that row is an instance of `NSTextFieldCell` and contains only a string value, the “group row” style attributes are automatically applied for that cell.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **outlineView:mouseDownInHeaderOfTableColumn:**

*Sent to the delegate whenever the mouse button is clicked in `outlineView` while the cursor is in a column header `tableColumn`.*

```
– (void)outlineView:(NSOutlineView *)outlineView  
mouseDownInHeaderOfTableColumn:(NSTableColumn *)tableColumn
```

### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column.

### Availability

Available in OS X v10.3 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **outlineView:nextTypeSelectMatchFromItem:toItem:forString:**

*Returns the first item that matches the `searchString` from within the range of `startItem` to `endItem`*

```
– (id)outlineView:(NSOutlineView *)outlineView  
nextTypeSelectMatchFromItem:(id)startItem toItem:(id)endItem forString:(NSString  
*)searchString
```

### Parameters

`outlineView`

The outline view that sent the message.

`startItem`

The first item to search.

`endItem`

The item before which to stop searching. It is possible for `endItem` to be less than `startItem` if the search will wrap.

`searchString`

The string for which to search.

### Return Value

The first item—from within the range of `startItem` to `endItem`—that matches the `searchString`, or `nil` if there is no match.

### Discussion

Implement this method if you want to control how type selection works. You should include `startItem` as a possible match, but do not include `endItem`.

It is not necessary to implement this method in order to support type select.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **`outlineView:rowViewForItem:`**

*implement this method to return a custom `NSTableView` for a particular item. (required)*

– `(NSTableView *)outlineView:(NSOutlineView *)outlineView rowViewForItem:(id)item`

### Parameters

`outlineView`

The outline view that sent the message.

`item`

The item displayed by the returned table row view.

### Return Value

An instance or subclass of `NSTableView`. If `nil` is returned, a `NSTableView` instance is created and used.

### Discussion

This method, if implemented, is only invoked for `NSView`-based outline views.

### Availability

Available in OS X v10.7 and later.

### Declared in

`NSOutlineView.h`

---

## **outlineView:selectionIndexesForProposedSelection:**

*Invoked to allow the delegate to modify the proposed selection.*

```
– (NSIndexSet *)outlineView:(NSOutlineView *)outlineView  
selectionIndexesForProposedSelection:(NSIndexSet *)proposedSelectionIndexes
```

### Parameters

`outlineView`

The outline view that sent the message.

`proposedSelectionIndexes`

An index set containing the indexes of the proposed selection.

### Return Value

An `NSIndexSet` instance containing the indexes of the new selection. Return `proposedSelectionIndexes` if the proposed selection is acceptable, or the value of the table view's existing selection to avoid changing the selection.

### Discussion

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of [outlineView:willDisplayOutlineCell:forTableColumn:item:](#) (page 27).

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

**Declared in**  
NSOutlineView.h

### **outlineView:shouldCollapseItem:**

---

*Returns a Boolean value that indicates whether the outline view should collapse a given item.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldCollapseItem:(id)item

#### **Parameters**

outlineView

The outline view that sent the message.

item

The item that should collapse.

#### **Return Value**

YES to permit outlineView to collapse item, NO to deny permission.

#### **Discussion**

The delegate can implement this method to disallow collapsing of specific items. For example, if the first row of your outline view should not be collapsed, your delegate method could contain this line of code:

```
return [outlineView rowForItem:item]!=0;
```

#### **Availability**

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

**Declared in**  
NSOutlineView.h

### **outlineView:shouldEditTableColumn:item:**

---

*Returns a Boolean value that indicates whether the outline view should allow editing of a given item in a given table column.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldEditTableColumn:(NSTableColumn \*)tableColumn item:(id)item



### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column.

`item`

The item.

### Return Value

YES to permit `outlineView` to edit the cell specified by `tableColumn` and `item`, NO to deny permission.

If this method returns YES, the cell may still not be editable—for example, if you have set up a custom `NSTextFieldCell` as a data cell, it must return YES for `isEditable` to allow editing.

### Discussion

The delegate can implement this method to disallow editing of specific cells.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

### See Also

`outlineView:setObjectValue:forTableColumn:byItem:` (NSOutlineViewDatasource Protocol)

### Declared in

`NSOutlineView.h`

---

## **`outlineView:shouldExpandItem:`**

*Returns a Boolean value that indicates whether the outline view should expand a given item.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldExpandItem:(id)item

### Parameters

`outlineView`

The outline view that sent the message.

`item`

The item that should expand.

### Return Value

YES to permit `outlineView` to expand `item`, NO to deny permission.

## Discussion

The delegate can implement this method to disallow expanding of specific items.

## Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

## Declared in

NSOutlineView.h

---

## outlineView:shouldReorderColumn:toColumn:

---

*Sent to the delegate to allow or prohibit the specified column to be dragged to a new location.*

```
– (BOOL)outlineView:(NSOutlineView *)outlineView  
shouldReorderColumn:(NSInteger)columnIndex toColumn:(NSInteger)newColumnIndex
```

## Parameters

outlineView

The outline view that sent the message.

columnIndex

The index of the column being dragged.

newColumnIndex

The proposed target index of the column.

## Return Value

YES if the column reordering should be allowed, otherwise NO.

## Discussion

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of `-1`. Returning NO will disallow that column from being reordered at all. Returning YES allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.

If this method is not implemented, all columns are considered reorderable.

## Availability

Available in OS X v10.6 and later.

**Declared in**  
NSOutlineView.h

### **outlineView:shouldSelectItem:**

---

*Returns a Boolean value that indicates whether the outline view should select a given item.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldSelectItem:(id)item

#### **Parameters**

outlineView

The outline view that sent the message.

item

The item.

#### **Return Value**

YES to permit outlineView to select item, NO to deny permission.

#### **Discussion**

You implement this method to disallow selection of particular items.

For better performance and finer grain control over the selection, use [outlineView:dataCellForTableColumn:item:](#) (page 8).

#### **Availability**

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

**Declared in**  
NSOutlineView.h

### **outlineView:shouldSelectTableColumn:**

---

*Returns a Boolean value that indicates whether the outline view should select a given table column.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView  
shouldSelectTableColumn:(NSTableColumn \*)tableColumn

#### **Parameters**

outlineView

The outline view that sent the message.

`tableColumn`

The table column.

### Return Value

YES to permit `outlineView` to select `tableColumn`, NO to deny permission.

### Discussion

The delegate can implement this method to disallow selection of specific columns.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **`outlineView:shouldShowCellExpansionForTableColumn:item:`**

*Invoked to allow the delegate to control cell expansion for a specific column and item.*

```
– (BOOL)outlineView:(NSOutlineView *)outlineView  
shouldShowCellExpansionForTableColumn:(NSTableColumn *)tableColumn item:(id)item
```

### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

A table column in the outline view.

`item`

An item in the outline view.

### Return Value

YES to allow an expansion tooltip to appear in the column `tableColumn` for item `item`, otherwise NO.

### Discussion

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns YES, the full cell contents will be shown in a special floating tool tip view, otherwise the content is truncated.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

**Declared in**  
NSOutlineView.h

---

### **outlineView:shouldShowOutlineCellForItem:**

---

*Returns a whether the specified item should display the outline cell (the disclosure triangle).*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldShowOutlineCellForItem:(id)item

#### **Parameters**

outlineView

The outline view that sent the message.

item

An item in the outline view.

#### **Return Value**

YES if the outline cell should be displayed, otherwise NO.

#### **Discussion**

Returning NO causes `frameOfOutlineCellAtRow:` to return `NSZeroRect`, hiding the cell. In addition, the row will not be collapsible by keyboard shortcuts.

This method is called only for expandable rows.

#### **Availability**

Available in OS X v10.6 and later.

**Declared in**  
NSOutlineView.h

---

### **outlineView:shouldTrackCell:forTableColumn:item:**

---

*Returns a Boolean value that indicates whether a given cell should be tracked.*

– (BOOL)outlineView:(NSOutlineView \*)outlineView shouldTrackCell:(NSCell \*)cell  
forTableColumn:(NSTableColumn \*)tableColumn item:(id)item

#### **Parameters**

outlineView

The outline view that sent the message.

`cell`

The cell used to display item `item` in column `tableColumn`

`tableColumn`

A table column in the outline view.

`item`

An item in the outline view.

### Return Value

YES if the cell should be tracked for the item `item` in column `tableColumn`, otherwise NO.

### Discussion

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked (and vice-versa). For example, this allows you to have a button cell in a table which does not change the selection, but can still be clicked on and tracked.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

NSOutlineView.h

---

## outlineView:shouldTypeSelectForEvent:withCurrentSearchString:

---

*Returns a Boolean value that indicates whether type select should proceed for a given event and search string.*

```
– (BOOL)outlineView:(NSOutlineView *)outlineView shouldTypeSelectForEvent:(NSEvent *)event withCurrentSearchString:(NSString *)searchString
```

### Parameters

`outlineView`

The outline view that sent the message.

`event`

The event that caused the message to be sent.

`searchString`

The string for which searching is to proceed. The search string is `nil` if no type select has begun.

### Return Value

YES if type select should proceed, otherwise NO.

### Discussion

Generally, this method will be called from `keyDown:` and the event will be a key event.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **outlineView:sizeToFitWidthOfColumn:**

*Invoked to allow the delegate to provide custom sizing behavior when a column's resize divider is double clicked.*

```
– (CGFloat)outlineView:(NSOutlineView *)outlineView  
sizeToFitWidthOfColumn:(NSInteger)column
```

### Parameters

`outlineView`

The outline view that sent the message.

`column`

The index of the column.

### Return Value

The width of the specified column.

### Discussion

By default, `NSOutlineView` iterates every row in the table, accesses a cell via `preparedCellAtColumn:row:`, and requests the `cellSize` to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

### Availability

Available in OS X v10.6 and later.

### Declared in

`NSOutlineView.h`

## **outlineView:toolTipForCell:rect:tableColumn:item:mouseLocation:**

---

*When the cursor pauses over a given cell, the value returned from this method is displayed in a tooltip.*

```
– (NSString *)outlineView:(NSOutlineView *)outlineView toolTipForCell:(NSCell *)cell  
  rect:(NSRectPointer)rect tableColumn:(NSTableColumn *)tc item:(id)item  
  mouseLocation:(NSPoint)mouseLocation
```

### **Parameters**

`outlineView`

The outline view that sent the message.

`cell`

The cell for which to generate a tooltip.

`rect`

The proposed active area of the tooltip. To control the default active area, you can modify the `rect` parameter. By default, `rect` is computed as `[cell drawingRectForBounds:cellFrame]`.

`tc`

The table column that contains `cell`.

`item`

The item for which to display a tooltip.

`mouseLocation`

The current mouse location in view coordinates.

### **Return Value**

If you don't want a tooltip at that location, return `nil` or the empty string.

### **Availability**

Available in OS X v10.4 and later.

Available as part of an informal protocol prior to OS X v10.6.

### **Declared in**

`NSOutlineView.h`

## **outlineView:typeSelectStringForTableColumn:item:**

---

*Returns the string that is used for type selection for a given column and item.*

```
– (NSString *)outlineView:(NSOutlineView *)outlineView  
  typeSelectStringForTableColumn:(NSTableColumn *)tableColumn item:(id)item
```



### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

A table column in the outline view.

`item`

An item in the outline view.

### Return Value

The string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return `nil` for that row and/or column to not be searched

### Discussion

Implement this method if you want to control the string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return `nil` to specify that the given row and/or column should not be searched. By default, all cells with text in them are searched.

The default value when this delegate method is not implemented is:

```
[[outlineView preparedCellAtColumn:tableColumn row:[outlineView rowForItem:item]]
 stringValue]
```

and you can return this value from the delegate method if you wish.

### Availability

Available in OS X v10.5 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **outlineView:viewForTableColumn:item:**

*Implemented to return the view used to display the specified item and column. (required)*

```
– (NSView *)outlineView:(NSOutlineView *)outlineView viewForTableColumn:(NSTableColumn *)tableColumn item:(id)item
```

### Parameters

`outlineView`

The outline view that sent the message.

`tableColumn`

The table column, or `nil` if the row is a group row.

`item`

The item displayed by the returned view.

### Return Value

The view to display the specified column and row. Returning `nil` is acceptable, in which case a view is not shown at that location.

### Discussion

This method is required if you wish to use `NSView` objects instead of `NSCell` objects for the cells within an outline view. Cells and views cannot be mixed within the same outline view.

It is recommended that the implementation of this method first call the `NSTableView` method `makeViewWithIdentifier:owner:` passing, respectively, the `tableColumn` parameter's identifier and `self` as the owner to attempt to reuse a view that is no longer visible. The frame of the view returned by this method is not important, and is automatically set by the outline view.

The view's properties should be properly set up before returning the result.

When using Cocoa bindings, this method is optional if at least one identifier has been associated with the table view at design time. If this method is not implemented, the outline view automatically calls `makeViewWithIdentifier:owner:` with the `tableColumn` parameter's identifier and the outline view's delegate as parameters, to attempt to reuse a previous view or automatically unarchive a prototype associated with the table view.

The `autoresizingMask` of the returned view is automatically set to `NSViewHeightSizable` to resize properly on row height changes.

### Availability

Available in OS X v10.7 and later.

### Declared in

`NSOutlineView.h`

---

## **`outlineView:willDisplayCell:forTableColumn:item:`**

*Informs the delegate that the cell specified by the column and item will be displayed.*

```
– (void)outlineView:(NSOutlineView *)outlineView willDisplayCell:(id)cell  
forTableColumn:(NSTableColumn *)tableColumn item:(id)item
```

### Parameters

`outlineView`

The outline view that sent the message.

`cell`

The cell.

`tableColumn`

The table column.

`item`

The item.

### Discussion

The delegate can implement this method to modify `cell` to provide further setup for the `cell` in `tableColumn` and `item`. It is not safe to do drawing inside this method—you should only set up state for `cell`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **`outlineView:willDisplayOutlineCell:forTableColumn:item:`**

---

*Informs the delegate that an outline view is about to display a cell used to draw the expansion symbol.*

```
– (void)outlineView:(NSOutlineView *)outlineView willDisplayOutlineCell:(id)cell  
forTableColumn:(NSTableColumn *)tableColumn item:(id)item
```

### Parameters

`outlineView`

The outline view that sent the message.

`cell`

The cell.

`tableColumn`

The table column.

`item`

The item.

### Discussion

Informs the delegate that `outlineView` is about to display `cell`—an expandable cell (a cell that has the expansion symbol)—for the column and item specified by `tableColumn` and `item`. The delegate can modify `cell` to alter its display attributes.

This method is not invoked when `outlineView` is about to display a non-expandable cell.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

### Declared in

`NSOutlineView.h`

---

## **outlineViewColumnDidMove:**

*Invoked whenever the user moves a column in the outline view.*

– (void)outlineViewColumnDidMove:(NSNotification \*)notification

### Parameters

`notification`

The posted notification.

### Discussion

This method is invoked as a result of posting an `NSOutlineViewColumnDidMoveNotification`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

### Declared in

`NSOutlineView.h`

---

## **outlineViewColumnDidResize:**

*Invoked whenever the user resizes a column in the outline view.*

– (void)outlineViewColumnDidResize:(NSNotification \*)notification

### Parameters

`notification`

The posted notification.

### Discussion

This method is invoked as a result of posting an `NSOutlineViewColumnDidResizeNotification`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

### Declared in

`NSOutlineView.h`

---

## **outlineViewItemDidCollapse:**

*Invoked when the did collapse notification is posted—that is, whenever the user collapses an item in the outline view.*

– (void)outlineViewItemDidCollapse:(NSNotification \*)notification

### Parameters

`notification`

The posted notification.

### Discussion

This method is invoked as a result of posting an `NSOutlineViewItemDidCollapseNotification`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

### Declared in

`NSOutlineView.h`

---

## **outlineViewItemDidExpand:**

*Invoked when notification is posted—that is, whenever the user expands an item in the outline view.*

– (void)outlineViewItemDidExpand:(NSNotification \*)notification

### Parameters

`notification`

The posted notification.

### Discussion

This method is invoked as a result of posting an `NSOutlineViewItemDidExpandNotification`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

### Declared in

`NSOutlineView.h`

---

## **outlineViewItemWillCollapse:**

*Invoked when `notification` is posted—that is, whenever the user is about to collapse an item in the outline view.*

– (void)outlineViewItemWillCollapse:(NSNotification \*)notification

### Parameters

`notification`

The posted notification.

### Discussion

This method is invoked as a result of posting an `NSOutlineViewItemWillCollapseNotification`.

### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

### Declared in

`NSOutlineView.h`

---

## **outlineViewItemWillExpand:**

*Invoked when `notification` is posted—that is, whenever the user is about to expand an item in the outline view.*

– (void)outlineViewItemWillExpand:(NSNotification \*)notification

#### Parameters

notification

The posted notification.

#### Discussion

This method is invoked as a result of posting an `NSOutlineViewItemWillExpandNotification`.

#### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

#### Declared in

`NSOutlineView.h`

---

### **outlineViewSelectionDidChange:**

*Invoked when the selection did change notification is posted—that is, immediately after the outline view’s selection has changed.*

– (void)outlineViewSelectionDidChange:(NSNotification \*)notification

#### Parameters

notification

The posted notification.

#### Discussion

This method is invoked as a result of posting an `NSOutlineViewSelectionDidChangeNotification`.

#### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

#### Declared in

`NSOutlineView.h`

---

### **outlineViewSelectionIsChanging:**

*Invoked when notification is posted—that is, whenever the outline view’s selection changes.*

– (void)outlineViewSelectionIsChanging:(NSNotification \*)notification

#### Parameters

notification

The posted notification.

#### Discussion

This method is invoked as a result of posting an `NSOutlineViewSelectionIsChangingNotification`.

#### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.7.

#### Declared in

`NSOutlineView.h`

---

### **selectionShouldChangeInOutlineView:**

---

*Returns a Boolean value that indicates whether the outline view should change its selection.*

– (BOOL)selectionShouldChangeInOutlineView:(NSOutlineView \*)outlineView

#### Parameters

outlineView

The outline view that sent the message.

#### Return Value

YES to permit `outlineView` to change its selection (typically a row being edited), NO to deny permission.

#### Discussion

For example, if the user is editing a cell and enters an improper value, the delegate can prevent the user from selecting or editing any other cells until a proper value has been entered into the original cell. The delegate can implement this method for complex validation of edited rows based on the values of any of their cells.

#### Availability

Available in OS X v10.0 and later.

Available as part of an informal protocol prior to OS X v10.6.

#### Declared in

`NSOutlineView.h`



# Document Revision History

This table describes the changes to *NSOutlineViewDelegate Protocol Reference*.

| Date       | Notes   |
|------------|---|
| 2013-12-16 | Added descriptions of methods for NSView-based outline views. Fixed typos.<br><br>Added information to discussion of <a href="#">outlineView:heightOfRowByItem:</a> (page 11) method. |
| 2013-10-22 | Updated for OS X v10.8.   |
| 2012-03-14 | Updated for OS X v10.7. Moved remaining informal delegates.   |
| 2010-01-25 | Expanded the description of the <code>outlineView:shouldShowCellExpansionForTableColumn:item:</code> method.  |
| 2009-05-29 | New document that describes the protocol that defines the methods for delegates of NSOutlineView objects.   |



Apple Inc.  
Copyright © 2013 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.

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.

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

Apple, the Apple logo, Cocoa, Mac, and OS X are trademarks of Apple Inc., registered in the U.S. and other countries.

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.