

---

# EOTableViewAssociation

<b>Inherits From:</b>	EOAssociation : EODelayedObserver : NSObject
<b>Conforms To:</b>	NSCoding (EOAssociation) EOObserving (EODelayedObserver) NSObject (NSObject)
<b>Declared In:</b>	EOInterface/EOColumnAssociation.h

---

## Class at a Glance

### Purpose

An EOTableViewAssociation coordinates the individual EOColumnAssociations bound to an NSTableView. The first EOColumnAssociation to be bound to an NSTableView automatically creates the EOTableViewAssociation; you should rarely need to do so yourself.

### Usable With

NSTableView

### Aspects

source	Bound to the EODisplayGroup providing objects. This aspect doesn't use a key.
enabled	A boolean attribute of the objects, which determines whether each object's row is editable. Note that because EOColumnAssociation also uses this aspect, you can use it with different keys to limit editability to the whole row or to an individual cell (column) in that row.
textColor	An NSColor attribute of the objects, which determines the color of text for each object's row in the NSTableView.
bold	A boolean attribute of the objects, which determines whether each objects row is displayed in bold or regular weight text.
italic	A boolean attribute of the objects, which determines whether each objects row is displayed in italic or normal angle text.

---

**Object Keys Taken**

dataSource	An EOTableViewAssociation responds to some data source messages and forwards others to the appropriate EOColumnAssociation.
delegate	An EOTableViewAssociation forwards delegate messages to the appropriate EOColumnAssociations.
target	Reserved, but not used.

---

## Class Description

An EOTableViewAssociation manages the individual EOColumnAssociations between an NSTableView and an EODisplayGroup. It receives data source and delegate messages from the NSTableView, some of which it handles itself, and some of which it forwards to the appropriate EOColumnAssociations. In addition to this, an EOTableViewAssociation can sort the objects in the EODisplayGroup by the left-to-right order of the table columns.

## Class Methods

### **bindToTableView:displayGroup:**

+ (void)**bindToTableView:(NSTableView \*)aTableView**  
**displayGroup:(EODisplayGroup \*)aDisplayGroup**

Creates an EOTableViewAssociation binding *aTableView* to *aDisplayGroup*, if there isn't already one. EOColumnAssociation's **establishConnection** invokes this method to guarantee the presence of a coordinating EOTableViewAssociation.

## Instance Methods

### **editingAssociation**

– (EOColumnAssociation \*)**editingAssociation**

Returns the EOColumnAssociation for the NSTableView cell being edited, or **nil** if no cell is being edited.

---

### **setSortsByColumnOrder:**

– (void)**setSortsByColumnOrder:(BOOL)***flag*

Controls whether the receiver applies a sort ordering to its EODisplayGroup. If *flag* is YES, it builds EOSortOrderings for each of the EOColumnAssociations, collects them into an NSArray based on the left-to-right order of the columns, and assigns them to the EODisplayGroup with **setSortOrdering:.** If *flag* is NO, it doesn't alter the sort ordering of the EODisplayGroup.

An EOTableViewAssociation assigns sort orderings based on the left to right order of the table columns, and reassigns them whenever the user moves a column.

**See also:** – **sortingSelector** (EOColumnAssociation), – **sortsByColumnOrder**

### **sortsByColumnOrder**

– (BOOL)**sortsByColumnOrder**

Returns YES if the receiver assigns EOSortOrderings to its EODisplayGroup based on the sorting selectors of its EOColumnAssociations, NO if it doesn't alter the EODisplayGroup's sort ordering.

An EOTableViewAssociation assigns sort orderings based on the left to right order of the table columns, and reassigns them whenever the user moves a column.

**See also:** – **sortingSelector** (EOColumnAssociation), – **setSortsByColumnOrder:**

## **NSTableView Delegate and Data Source Methods**

### **control:isValidObject:**

– (BOOL)**control:(NSControl \*)***aTableView* **isValidObject:(id)***anObject*

Forwards the message to the receiver's editing association.

**See also:** – **editingAssociation**

### **control:didFailToFormatString:errorDescription:**

– (BOOL)**control:(NSControl \*)***aTableView*  
**didFailToFormatString:(NSString \*)***aString*  
**errorDescription:(NSString \*)***errorDescription*

Forwards the message to the receiver's editing association.

**See also:** – **editingAssociation**

---

**control:textShouldBeginEditing:**

– (BOOL)**control:**(NSControl \*)*aTableView* **textShouldBeginEditing:**(NSText \*)*fieldEditor*

Forwards the message to the receiver’s editing association.

**See also:** – **editingAssociation**

**numberOfRowsInTableView:**

– (int)**numberOfRowsInTableView:**(NSTableView \*)*aTableView*

Returns the number of displayed objects in the receiver’s EODisplayGroup.

**See also:** – **displayedObjects** (EODisplayGroup)

**tableView:objectValueForTableColumn:row:**

– (id)**tableView:**(NSTableView \*)*aTableView*  
    **objectValueForTableColumn:**(NSTableColumn \*)*aTableColumn*  
    **row:**(int)*rowIndex*

Forwards the message to *aTableColumn*’s identifier—assumed to be the EOColumnAssociation bound to that column—so that it can provide the value.

**tableView:setObjectValue:forTableColumn:row:**

– (void)**tableView:**(NSTableView \*)*aTableView*  
    **setObjectValue:**(id)*value*  
    **forTableColumn:**(NSTableColumn \*)*aTableColumn*  
    **row:**(int)*rowIndex*

Forwards the message to *aTableColumn*’s identifier—assumed to be the EOColumnAssociation bound to that column—so that it can set the value.

**tableView:shouldEditTableColumn:row:**

– (BOOL)**tableView:**(NSTableView \*)*aTableView*  
    **shouldEditTableColumn:**(NSTableColumn \*)*aTableColumn*  
    **row:**(int)*rowIndex*

Returns NO if the “enabled” aspect is bound and its value for the object at *rowIndex* is NO. Otherwise forwards the message to *aTableColumn*’s identifier—assumed to be the EOColumnAssociation bound to that column—and returns its response. Note that because the two associations’ “enabled” aspects can be

---

bound to different keys, you can limit editability to the whole row or to an individual cell (column) in that row.

**tableView:willDisplayCell:forTableColumn:row:**

– (void)**tableView:(NSTableView \*)aTableView**  
    **willDisplayCell:(id)aCell**  
    **forTableColumn:(NSTableColumn \*)aTableColumn**  
    **row:(int)rowIndex**

Alters the display characteristics for *aCell* according to the values for the “enabled”, “textColor”, “bold”, and “italic” aspects of the object at *rowIndex*. Then forwards the message to *aTableColumn*’s identifier—assumed to be the EOColumnAssociation bound to that column—allowing it to adjust *aCell* based on its own “enabled” aspect.

**tableViewSelectionDidChange:**

– (void)**tableViewSelectionDidChange:(NSNotification \*)aNotification**

Updates the receiver’s EODisplayGroup based on the new selection in the table view.

**See also:** – **setSelectionIndexes:** (EODisplayGroup)