

---

# EOColumnAssociation

<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 EOColumnAssociation links an NSTableColumn to a single attribute of all displayed objects in an EODisplayGroup. The value of each object is displayed in its corresponding row.

### Usable With

NSTableColumn

### Aspects

value	An attribute of the objects, displayed in each row of the NSTableColumn.
enabled	A boolean attribute of the objects, which determines whether each object's value cell is editable. Note that because EOTableViewAssociation 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.

### Object Keys Taken

identifier	An EOColumnAssociations sets itself as the identifier of its NSTableColumn. (Note: This key isn't formally reserved by the <b>objectKeysTaken</b> method, as Interface Builder doesn't treat it as an outlet.)
------------	--

---

## Class Description

EOColumnAssociations cooperate with an EOTableViewAssociation to display values in the columns of an NSTableView. The EOTableViewAssociation receives target, delegate, and data source messages from the NSTableView, and forwards them as needed to the appropriate EOColumnAssociation. The EOColumnAssociations provide values for the cells of each NSTableColumn, and also accept edited values to set in their EODisplayGroups.

EOColumnAssociations provide values using NSTableView's **tableView:setObjectValue:forTableColumn:row:** and **tableView:objectValueForTableColumn:** methods. This allows values with non-string representations to be displayed. For example, if an NSImageCell is used as an NSTableColumn's data cell, an EOColumnAssociation can be used to display NSImages in the NSTableView.

### Example

To display the last and first names of objects in the Talent EODisplayGroup, bind two NSTableColumns to the "lastName" and "firstName" keys of the EODisplayGroup (this automatically creates an EOTableViewAssociation to manage the individual columns). When you run the application, the last and first names of each Talent object in the EODisplayGroup's **displayedObjects** array are put in the corresponding row.

## Instance Methods

### setSortingSelector:

– (void)**setSortingSelector:(SEL)aSelector**

Sets the method selector used to sort rows to *aSelector*, one of:

Defined Name	Selector
EOCompareAscending	compareAscending:
EOCompareDescending	compareDescending:
EOCompareCaseInsensitiveAscending	compareCaseInsensitiveAscending:
EOCompareCaseInsensitiveDescending	compareCaseInsensitiveDescending:
<b>nil</b>	(none; the column isn't sorted)

If the EOTableViewAssociation for the receiver's NSTableView sorts its rows, it applies this method as needed to sort them. The default sorting selector is EOCompareAscending. See the **compare...** methods in the Value Class Additions specification for details on the sorting methods.

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

---

## sortingSelector

– (SEL)sortingSelector

Returns the method selector used to sort rows, or **nil** if the column isn't sorted.

**See also:** – **sortsByColumnOrder** (NSTableViewAssociation), – **setSortingSelector:**

## Delegate and Data Source Methods

These methods are forwarded by the NSTableView's EOTableViewAssociation to the appropriate EOColumnAssociation.

### control:isValidObject:

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

Saves the value of any cell being edited using **setValue:forAspect:**, and if successful sends an **associationDidEndEditing:** message to the receiver's EODisplayGroup. Returns YES if successful (or if no changes need be saved), NO if unsuccessful.

### control:didFailToFormatString:errorDescription:

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

Invokes **shouldEndEditingForAspect:invalidInput:errorDescription:** (defined by EOAssociation) and returns the result.

### control:textShouldBeginEditing:

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

Sends an **associationDidBeginEditing:** message to the receiver's EODisplayGroup and returns YES.

### tableView:objectValueForTableColumn:row:

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

Returns the value of the property of the object at *rowIndex* bound to the **value** aspect.

---

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

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

Sets the property of the object at *rowIndex* bound to the **value** aspect to *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 returns YES. Note that because the **enabled** aspects of EOTableViewAssociation and EOColumnAssociation 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** aspect of the object at *rowIndex*.

**See also:** – **tableView:willDisplayCell:forTableColumn:row:**