EOActionAssociation

Inherits from:	EOActionWidgetAssociation: EOWidgetAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOActionAssociation object allows you to set up an interface object, such as a button, to send a message to the objects selected in the association's display group when the interface object is acted on.

Usable With

com.webobjects.eointerface.swing:
Any object that implements the method `addActionListener` (javax.swing.JButton and javax.swing.JMenuItem, for example).

com.webobjects.eointerface.cocoa:
NSControl, NSActionCell, and their subclasses.

Aspects

`action`	Bound to a key that names the method to invoke on the selected objects. If the `argument` aspect isn't bound, the method must take no arguments. If the `argument` aspect is bound, then the method must take exactly one argument.
`argument`	An object attribute or relationship of the selected object, passed as an argument to the action method. (Usually bound to a different EODisplayGroup than the one bound to `action`.)
`enabled`	A boolean attribute of the selected object, which determines whether the display object is enabled.

Object Keys Taken

`target`	On receiving an action message from the display object, an EOActionAssocation sends its action to the selected objects.

Examples

Suppose you have an application that manages member accounts, each of which has a restriction on the outstanding balance allowed. You want a user to be able to increase the restriction limit by selecting one or more members and then clicking a button. To do this, you define a `boostRestrictions` method in the Member class that increases the limit by 20%. In Interface Builder, control-drag a connection from the button to the Member display group. Select EOActionAssociation in the Connections inspector, and bind the association's `action` aspect to the "boostRestrictions" key.

In another scenario, one EODisplayGroup shows Members, while another shows video tapes available for rent. Here, you want a user to be able to select a member, select a video tape, and then click a Rent button that checks the selected tape out to the selected member. To do this, define a `rentVideoTape` method in the Member class that takes a VideoTape as an argument and handles the accounting involved in a video rental. Then, in Interface Builder, control-drag a connection from the button to the Members display group. Select EOActionAssociation in the Connections inspector, and bind the association's `action` aspect to Member's `rentVideoTape` action. Similarly, control-drag a connection from the button to the VideoTape display group. Select EOActionAssociation in the Connections inspector, and bind the association's `argument` aspect to the VideoTape display group. Now, when the user selects a Member, selects a VideoTape, and clicks the button, the selected Member is sent a `rentVideoTape` message with the selected VideoTape.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOActionAssociation`

`displayGroupSelectionsAllowEnabled`

`invokeAction`

`primaryAspect`

Constructors

EOActionAssociation

```
public EOActionAssociation(Object aDisplayObject)
```

Creates a new EOActionAssociation to monitor and update the value in `aDisplayObject`, typically a button or menu item.

You normally set up associations in Interface Builder, in which case you don’t need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` (EOAssociation), `establishConnection` **(EOAssociation)**

Instance Methods

displayGroupSelectionsAllowEnabled

```
protected boolean displayGroupSelectionsAllowEnabled()
```

Description forthcoming.

invokeAction

```
public void invokeAction()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Description forthcoming.

EOActionInsertionAssociation

Inherits from:	EOActionWidgetAssociation: EOWidgetAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOActionInsertionAssociation object inserts objects from one display group into another.

Usable With

com.webobjects.eointerface.swing:
Any object that implements the method `addActionListener` (javax.swing.JButton and javax.swing.JMenuItem, for example).

com.webobjects.eointerface.cocoa:
Any object that responds to `setAction`, typically an NSControl.

Aspects

source	Bound to the EODisplayGroup containing objects to insert. This aspect doesn’t use a key.
destination	A relationship of the selected object into which objects from the source EODisplayGroup are inserted. Usually bound to a different EODisplayGroup than source.
enabled	A boolean attribute of the selected object (usually in the destination EODisplayGroup), which determines whether the NSControl is enabled.

Object Keys Taken

target	On receiving an action message from the display object, an EOActionInsertionAssociation inserts objects from the source EODisplayGroup into the destination EODisplayGroup.

Example

Suppose an application shows Talent in one display group and Movies in another. You want a user to be able to select a talent, select a movie, and then click an Assign Director button that assigns the selected talent as one of the movie’s directors. To do this, in Interface Builder, control-drag a connection from the button to the Talent display group. Select EOActionInsertionAssociation in the Connections inspector, and double-click the association’s source aspect, binding it to the Talent display group. Similarly, control-drag a connection from the button to the Movie display group. Select EOActionAssociation in the Connections inspector, and bind the association’s destinationaspect to the “directors” key. Now, when the user clicks the button, the selected Talent is added to the directors relationship of the selected Movie. If more than one talent is selected, both are added to the relationship. If more than one Movie is selected, the selected talent are added to the relationship of the first Movie in the selection.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

- `EOActionInsertionAssociation`
- `displayGroupSelectionsAllowEnabled`
- `invokeAction`
- `primaryAspect`

Constructors

EOActionInsertionAssociation

```
public EOActionInsertionAssociation(Object anObject)
```

Description forthcoming.

Instance Methods

displayGroupSelectionsAllowEnabled

```
protected boolean displayGroupSelectionsAllowEnabled()
```

Description forthcoming.

invokeAction

```
public void invokeAction()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`SourceAspect`.

EOActionWidgetAssociation

Inherits from:	EOWidgetAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOActionWidgetAssociation`

`displayGroupSelectionsAllowEnabled`

`invokeAction`

`isEnabled`

`isEnabledAtIndex`

`subjectChanged`

`widgetPluginClass`

Constructors

EOActionWidgetAssociation

`public EOActionWidgetAssociation(Object anObject)`

Description forthcoming.

Instance Methods

displayGroupSelectionsAllowEnabled

```
protected boolean displayGroupSelectionsAllowEnabled()
```

Description forthcoming.

invokeAction

```
public abstract void invokeAction()
```

Description forthcoming.

isEnabled

```
protected boolean isEnabled()
```

Description forthcoming.

isEnabledAtIndex

```
protected boolean isEnabledAtIndex(int index)
```

Description forthcoming.

subjectChanged

```
public void subjectChanged()
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

EOActionWidgetAssociation.ActionPlugin

Inherits from:	EOWidgetAssociation.WidgetPlugin : Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

```
EOActionWidgetAssociation.ActionPlugin
setEnabled
```

Constructors

EOActionWidgetAssociation.ActionPlugin

```
public EOActionWidgetAssociation.ActionPlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object anObject)
```

Description forthcoming.

Instance Methods

setEnabled

```
public abstract void setEnabled(boolean aBoolean)
```

Description forthcoming.

EOAssociation

Inherits from:	EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class at a Glance

An EOAssociation maintains a two-way binding between the properties of a display object, such as a text field or combo box, and the properties of one or more enterprise objects contained in one or more EODisplayGroups. You typically create and configure associations in Interface Builder, using the programmatic interface only when you write your own EOAssociation subclasses.

Principal Attributes

- A display object (such as a text field or combo box)
- Aspects that control different parameters of the display object (such as `value` and `enabled`)
- One or more EODisplayGroups (no more than one per aspect)
- One or more keys (enteprise object properties) (as many as one key per aspect)

Class Description

EOAssociation defines the mechanism that transfers values between EODisplayGroups and the user interface of an application. An EOAssociation instance is tied to a single display object, a user interface object or other kind of object that manages values intended for display. The EOAssociation takes over certain outlets of the display object and sets its value according to the selection in the EODisplayGroup. An EOAssociation also has various aspects, which define the different parameters of the display object that it controls, such as the value or values displayed and whether the display object is enabled or editable. Each aspect can be bound to an EODisplayGroup with a key denoting a property of the enterprise objects in the EODisplayGroup. The value or values of this property determine the value for the EOAssociation's aspect.

EOAssociation is an abstract class, defining only the general mechanism for binding display objects to EODisplayGroups. You always create instances of its various subclasses, which define behavior specific to different kinds of display objects. For information on the different EOAssociation subclasses you can use, see the following subclass specifications:

com.webobjects.eointerface.cocoa

EOCocoaButtonPlugin	EOCocoaSimpleTextPlugin
EOCocoaCheckBoxPlugin	EOCocoaTableColumnPlugin
EOCocoaComboBoxPlugin	EOCocoaTableViewPlugin
EOCocoaImageViewPlugin	EOCocoaTextFieldPlugin
EOCocoaPopUpButtonPlugin	EOCocoaTextPlugin
EOCocoaRadioMatrixPlugin	

com.webobjects.eointerface.swing

EOSwingButtonPlugin	EOSwingQuickTimeViewPlugin
EOSwingCheckBoxPlugin	EOSwingTableColumnPlugin

com.webobjects.eointerface.swing

EOSwingComboBoxPlugin	EOSwingTablePlugin
EOSwingImageViewPlugin	EOSwingTextPlugin

• [is everything in .cocoa and .swing an association? how can I find out? •

You normally set up EOAssociations using Interface Builder; each of the class specifications for EOAssociation's subclasses provide an example using Interface Builder to set them up. EOAssociation's programmatic interface is more important when defining custom EOAssociation subclasses. For more information on EOAssociations, see the sections:

- "How EOAssociations Work" (page 41)
- "Setting up an EOAssociation Programmatically" (page 43)
- "Creating a Subclass of EOAssociation" (page 44)

Constants

EOAssociation defines the following String constants to identify the names of association aspects:

ActionAspect	NullAspectSignature
ArgumentAspect	ParentAspect
AttributeAspectSignature	SelectedIndexAspect
AttributeToManyAspectSignature	SelectedObjectAspect
AttributeToOneAspectSignature	SelectedTitleAspect
AttributeToOneToManyAspectSignature	SourceAspect
BackgroundColorAspect	TextColorAspect
BoldAspect	TitlesAspect

DestinationAspect	ToManyAspectSignature
EnabledAspect	ToOneAspectSignature
ItalicAspect	ToOneToManyAspectSignature
MatchKey1Aspect	URLAspect
MatchKey2Aspect	ValueAspect
MatchKey3Aspect	

Interfaces Implemented

NSDisposable

dispose

EOObserving

Method Types

All methods

EOAssociation

associationClassesForObject

registerAssociationClass

aspectSignatures

aspects

bindAspect

breakConnection

copyMatchingBindingsFromAssociation

displayGroupForAspect

displayGroupKeyForAspect

endEditing

establishConnection

isConnected

isEnabled

isEnabledAtIndex

isExplicitlyDisabled

isUsableWithObject

object

objectKeysTaken

primaryAspect

priority

setExplicitlyDisabled

setObject

setValueForAspect

setValueForAspectAtIndex

shouldEndEditing

shouldEndEditingAtIndex

subjectChanged

valueForAspect

valueForAspectAtIndex

Constructors

EOAssociation

```
public EOAssociation(Object anObject)
```

Description forthcoming.

Static Methods

associationClassesForObject

```
public static NSArray associationClassesForObject(Object anObject)
```

Description forthcoming.

registerAssociationClass

```
public static void registerAssociationClass(Class aClass)
```

Description forthcoming.

Instance Methods

aspectSignatures

```
public NSArray aspectSignatures()
```

Overridden by subclasses to return the signatures of the receiver's aspects, an array of string objects matching its aspects array index for index. The signature strings can be any of:

Constant	The Aspect Can Be Bound to
`AttributeAspectSignature`	Attributes
`AttributeToOneAspectSignature`	Attributes and to-one relationships
`AttributeToManyAspectSignature`	Attributes and to-many relationships
`AttributeToOneToManyAspectSignature`	Attributes, to-one relationships, and to-many relationships
`ToOneAspectSignature`	To-one relationships
`ToOneToManyAspectSignature`	To-one and to-many relationships
`ToManyAspectSignature`	To-many relationships
`NullAspectSignature`	An EODisplayGroup without a key (the key is irrelevant).

Interface Builder uses aspect signatures to enable and disable keys in its Connections inspectors.

EOAssociation's implementation of this method returns an array of `AttributeToOneToManyAspectSignature` strings.

aspects

```
public NSArray aspects()
```

Overridden by subclasses to return the names of the receiving class's aspects as an array of string objects. Subclasses should include their superclass's aspects and add their own when overriding this method.

bindAspect

```
public void bindAspect(
    String aspectName,
    EODisplayGroup anEODisplayGroup,
    String key )
```

Defines the receiver's link between its display object and `aEODisplayGroup`. `aspectName` is the name of the aspect it observes in its display object, and `key` is the name of the property it observes in `aEODisplayGroup`. Invoke `establishConnection` after this method to finish setting up the binding. See "Setting up an EOAssociation Programmatically" (page 43) in the class description for more information.

breakConnection

```
public void breakConnection()
```

Removes the receiver from its EODisplayGroup and display object. Subclasses should override this method to remove the receiver from any outlets of the display object and invoke `super`'s implementation at the end.

See Also: `establishConnection`

copyMatchingBindingsFromAssociation

```
public void copyMatchingBindingsFromAssociation(EOAssociation anEOAssociation)
```

Description forthcoming.

displayGroupForAspect

`public EODisplayGroup displayGroupForAspect(String aspectName)`

Returns the EODisplayGroup bound to the receiver for aspectName, or `null` if there's no such object.

See Also: `displayGroupKeyForAspect`

displayGroupKeyForAspect

`public String displayGroupKeyForAspect(String aspectName,)`

Returns the EODisplayGroup key bound to the receiver for aspectName, or `null` if there's no EODisplayGroup.

See Also: `displayGroupForAspect`

dispose

`public void dispose()`

Description forthcoming.

endEditing

`public boolean endEditing()`

Overridden by subclasses to pass the value of the receiver's display object to the EODisplayGroup, by invoking `setValueForAspect` with the display object's value and the appropriate aspect (typically "value"). Returns `true` if successful, `false` if not—specifically if `setValueForAspect` returns `false`.

Subclasses whose display objects immediately pass their changes back to the EOAssociation—such as a button or pop-up list—need not override this method. It's only needed when the display object's value is edited rather than simply set.

EOAssociation's implementation does nothing but return `true`.

establishConnection

`public void establishConnection()`

Overridden by subclasses to attach the receiver to the outlets of its display object, and to otherwise configure the display object (such as by setting its action method). EOAssociation's implementation subscribes the receiver as an observer of its EODisplayGroups. Subclasses should invoke `super`'s implementation after establishing their own connections.

See "Setting up an EOAssociation Programmatically" (page 43) in the class description for more information.

isConnected

`public boolean isConnected()`

Description forthcoming.

isEnabled

`protected boolean isEnabled()`

Returns `false` if the receiver has explicitly disabled its display object or if the receiver's `EnabledAspect` (if bound) resolves to `false`; `true` otherwise.

isEnabledAtIndex

`protected boolean isEnabledAtIndex(int index)`

Returns `false` if the receiver has explicitly disabled its display object or if the receiver's `EnabledAspect` (if bound) resolves to `false` for **index**; `true` otherwise.

isExplicitlyDisabled

`public boolean isExplicitlyDisabled()`

Returns `true` if the receiver has explicitly disabled its display object, `false` otherwise.

isUsableWithObject

```
public boolean isUsableWithObject(Object anObject)
```

Overridden by subclasses to return `true` if instances of the receiving class are usable with *anObject* `false` if they aren't. The receiving class can examine any relevant characteristic of *anObject*—its class, configuration (such as whether an NSMatrix operates in radio mode), and so on.

object

```
public Object object()
```

Description forthcoming.

objectKeysTaken

```
public NSArray objectKeysTaken()
```

Overridden by subclasses to return the names of display object outlets that instances assume control of. Interface Builder uses this information to disable connections from these outlets in its Connections Inspector.

primaryAspect

```
public String primaryAspect()
```

Overridden by subclasses to return the default aspect, usually one denoting the displayed value, which by convention is named "value". EOAssociation's implementation returns `null`.

priority

```
public int priority()
```

Returns the receiver's change notification priority. For more information, see the EODelayedObserver class specification (EOControl).

setExplicitlyDisabled

```
public void setExplicitlyDisabled(boolean flag)
```

Sets according to `flag` whether or not the association is explicitly disabled. An association is “explicitly disabled” when the display object shouldn’t be editable, such as in the case where the display object simply displays the results of a search.

setObject

```
public void setObject(Object anObject)
```

Description forthcoming.

setValueForAspect

```
public boolean setValueForAspect(
    Object value,
    String aspectName)
```

Sets a value of the selected enterprise object in the EODisplayGroup bound to `aspectName`. Retrieves the display group and key bound to `aspectName`, and sends the display group a `setSelectedObjectValue` message with `value` and the key as arguments. Returns `true` if successful, or if there’s no display group bound to `aspectName`. Returns `false` if there’s an display group and it doesn’t accept the new value.

See Also: `valueForAspect`

setValueForAspectAtIndex

```
public boolean setValueForAspectAtIndex(
    Object value,
    String aspectName,
    int index)
```

Sets a value of the enterprise object at index in the EODisplayGroup bound to aspectName. Retrieves the display group and key bound to aspectName, and sends the display group a setValueForObjectAtIndex message with value, index, and the key as arguments. Returns true if successful, or if there's no display group bound to aspectName. Returns false if there's a display group and it doesn't accept the new value.

See Also: valueForAspectAtIndex

shouldEndEditing

```
public boolean shouldEndEditing(
    String aspectName,
    String inputString,
    String errorDescription)
```

Invoked by subclasses when the display object fails to validate its input, this method informs the EODisplayGroup bound to aspectName with an associationFailedToValidateValue message, using the display group's selected object. Returns the result of that message, or true if there's no display group.

For example, an association bound to an NSControl object (Cocoa) receives a controlDidFailToFormatStringErrorDescription delegate message when the control's formatter fails to format the input string. Its implementation of that method invokes shouldEndEditing.

shouldEndEditingAtIndex

```
public boolean shouldEndEditingAtIndex(
    String aspectName,
    String inputString,
    String errorDescription,
    int index)
```

Works in the same manner as shouldEndEditing, but allows you to specify a particular object by index rather than implicitly specifying the selected object.

subjectChanged

```
public void subjectChanged()
```

Overridden by subclasses to update state based when an EODisplayGroup's selection or contents changes. This method is invoked automatically anytime a display group that's bound to the receiver changes. The receiver can query its display group with `selectionChanged` and `contentsChanged` messages to determine how it needs to update.

valueForAspect

```
public Object valueForAspect(String aspectName)
```

Returns a value of the selected enterprise object in the EODisplayGroup bound to **aspectName**. Retrieves the display group and key bound to **aspectName**, and sends the display group a `selectedObjectValueForKey` message with the key. Returns `null` if there's no display group or key bound to **aspectName**.

See Also: `setValueForAspect`

valueForAspectAtIndex

```
public Object valueForAspectAtIndex(
    String aspectName,
    int index)
```

Returns a value of the enterprise object at **index** in the EODisplayGroup bound to **aspectName**. Retrieves the display group and key bound to **aspectName**, and sends the display group a `valueForObjectAtIndex` message with **index** and the key. Returns `null` if there's no display group or key bound to **aspectName**.

See Also: `valueForAspectAtIndex`

EODetailSelectionAssociation

Inherits from:	EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EODetailSelectionAssociation binds two EODisplayGroups together through a relationship, so that the destination display group acts as an editor for that relationship.

The destination display group shows all possible values for the relationship and indicates the actual members of the relationship by selecting them. The user can change the objects included in the relationship of the source by selecting and deselecting them in the destination.

EODetailSelectionAssociation is a useful alternative to EOMasterDetailAssociation and EOMasterPeerAssociation when it's more important to add and remove objects from a relationship than it is to edit the attributes of those objects.

Usable With
EODisplayGroup

Aspects

`selectedObjects`	A relationship from objects in the source EODisplayGroup.

Object Keys Taken

None

Example

Suppose that an employee can be assigned any number of projects. Your application displays employees in one table view and projects in another. When an employee is selected in the first table view, the employee's assigned projects are selected in the other. To change the employee's project assignments, a user changes the selection in the project table view: to add a project to the set, the user selects it, and to remove a project from the set, the user deselects it. To do this, in Interface Builder control-drag a connection from the Projects display group to the Employee display group. Choose EODetailSelectionAssociation in the Connections inspector, and bind the `selectedObjects` aspect to the "projects" key.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EODetailSelectionAssociation`

`isUsableWithObject`

`primaryAspect`

`subjectChanged`

Constructors

EODetailSelectionAssociation

`public EODetailSelectionAssociation(Object aDisplayObject)`

Creates a new EODetailSelectionAssociation to monitor and update the value in `aDisplayObject`, an EODisplayGroup.

You normally set up associations in Interface Builder, in which case you don't need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` (EOAssociation), `establishConnection` **(EOAssociation)**

Instance Methods

isUsableWithObject

`public boolean isUsableWithObject(Object anObject)`

Description forthcoming.

primaryAspect

`public String primaryAspect()`

Returns EOAssociation.`SourceAspect`.

subjectChanged

```
public void subjectChanged()
```

Description forthcoming.

EODisplayGroup

Inherits from:	Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class at a Glance

An EODisplayGroup collects an array of objects from an EODataSource, and works with a group of EOAssociation objects to display and edit the properties of those objects.

Principal Attributes

- Array of objects supplied by an EODataSource
- EOQualifier and EOSortOrderings to filter the objects for display
- Array of selection indexes
- Delegate

Commonly Used Methods

`allObjects`	Returns all objects in the EODisplayGroup.
`displayedObjects`	Returns the subset of all objects made available for display.
`selectedObjects`	Returns the selected objects.
`setQualifier`	Sets a filter that limits the objects displayed.
`setSortOrderings`	Sets the ordering used to sort the objects.
`updateDisplayedObjects`	Filters, sorts, and redisplays the objects.
`insertNewObjectAtIndex`	Creates a new object and inserts it into the EODataSource.

Class Description

An EODisplayGroup is the basic user interface manager for an Enterprise Objects Framework or Java Client application. It collects objects from an EODataSource, filters and sorts them, and maintains a selection in the filtered subset. It interacts with user interface objects and other display objects through EOAssociations, which bind the values of objects to various aspects of the display objects.

An EODisplayGroup manipulates its EODataSource by sending it `fetchObjects`, `insertObject`, and other messages, and registers itself as an editor and message handler of the EODataSource's EOEditingContext. The EOEditingContext allows the EODisplayGroup to intercede in certain operations, as described in the EOEditingContext.Editor and EOEditingContext.MessageHandler interface specifications (both interfaces are defined in EOControl). EODisplayGroup implements all the methods of these informal protocols; see their specifications for more information.

Most of an EODisplayGroup's interactions are with its associations, its EODataSource, and its EOEditingContext. See the EOAssociation, EODataSource, and EOEditingContext class specifications for more information on these interactions.

Creating an EODisplayGroup

You create most EODisplayGroups in Interface Builder, by dragging an entity icon from the EOModeler application, which creates an EODisplayGroup with an EODatabaseDataSource (EODistributedDataSource, for Java Client applications), or by dragging an EODisplayGroup with no EODataSource from the EOPalette. EODisplayGroups with EODataSources operate independent of other EODisplayGroups, while those without EODataSources must be set up in a master-detail association with another EODisplayGroup.

To create an EODisplayGroup programmatically, simply initialize it and set its EODataSource:

```
EODistributedDataSource dataSource;    /* Assume this exists. */
EODisplayGroup displayGroup;

displayGroup = new EODisplayGroup();
displayGroup.setDataSource(dataSource);
```

After creating the EODisplayGroup, you can add associations as described in the EOAssociation class specification.

Getting Objects

Since an EODisplayGroup isn't much use without objects to manage, the first thing you do with an EODisplayGroup is send it a fetch message. You can use the basic `fetch` method or you can configure the EODisplayGroup in Interface Builder to fetch automatically when its nib file is loaded. These methods all ask the EODisplayGroup's EODataSource to fetch from its persistent store with a `fetchObjects` message.

Filtering and Sorting

An EODisplayGroup's fetched objects are available through its `allObjects` method. These objects are treated only as candidates for display, however. The array of objects actually displayed is filtered and sorted by the EODisplayGroup's delegate, or by a qualifier and sort ordering array. You set the qualifier and sort orderings using the `setQualifier` and `setSortOrderings` methods. The `displayedObjects` method returns this filtered and sorted array; index arguments to other EODisplayGroup methods are defined in terms of this array.

If the EODisplayGroup has a delegate that responds to `displayGroupDisplayArrayForObjects`, it invokes this method rather than using its own qualifier and sort ordering array. The delegate is then responsible for filtering the objects and returning a sorted array. If the delegate only needs to perform one of these steps, it can get the qualifier or sort orderings from the EODisplayGroup and apply either itself using EOQualifier's `filteredArrayUsingQualifier` and EOSortOrdering's `sortedArrayUsingKeyOrderArray` methods, which are added by the control layer.

If you change the qualifier or sort ordering, or alter the delegate in a way that changes how it filters and sorts the EODisplayGroup's objects, you can send `updateDisplayedObjects` to the EODisplayGroup to get it to refilter and resort its objects. Note that this doesn't cause the EODisplayGroup to refetch.

Changing and Examining the Selection

An EODisplayGroup keeps a selection in terms of indexes into the array of displayed objects. EOAssociations that display values for multiple objects are responsible for updating the selection in their EODisplayGroups according to user actions on their display objects. This is typically done with the `setSelectionIndexes` method. Other methods available for indirect manipulation of the selection are the action methods `selectNext` and `selectPrevious`, as well as `selectObjectsIdenticalTo` and `selectObjectsIdenticalToSelectFirstOnNoMatch`.

To get the selection, you can use the `selectionIndexes` method, which returns an array of NSNumbers, or `selectedObjects`, which returns an array containing the selected objects themselves. Another method, `selectedObject`, returns the first selected object if there is one.

The Delegate

EODisplayGroup offers a number of methods for its delegate to implement; if the delegate does, it invokes them as appropriate. Besides the aforementioned `displayGroupDisplayArrayForObjects`, there are methods that inform the delegate that the EODisplayGroup has fetched, created an object (or failed to create one), inserted or deleted an object, changed the selection, or set a value for a property. There are also methods that request permission from the delegate to perform most of these same actions. The delegate can return `true` to permit the action or `false` to deny it. For more information, see each method's description in the EODisplayGroup.Delegate interface specification.

Methods for Use by EOAssociations

While most of your application code interacts with objects directly, EODisplayGroup also defines methods for its associations to access properties of individual objects without having to know anything about which methods they implement. Accessing properties through the EODisplayGroup offers associations the benefit of automatic validation, as well.

Associations access objects by index into the displayed objects array, or by object identifier. `valueForObjectAtIndex` returns the value of a named property for the object at a given index, and `setValueForObjectAtIndex` sets it. Similarly, `valueForObject` and `setValueForObject`access the objects by object identifier. EOAssociations can also get and set values for the first object in the selection using `selectedObjectValueForKey` and `setSelectedObjectValue`.

Interfaces Implemented

NSDisposable

`dispose`

Method Types

Configuring behavior

`defaultStringMatchFormat`

`defaultStringMatchOperator`

`fetchesOnLoad`

`queryBindingValues`

`queryOperatorValues`

`selectsFirstObjectAfterFetch`

`setDefaultStringMatchFormat`

`setDefaultStringMatchOperator`

`setFetchesOnLoad`

`setQueryBindingValues`

`setQueryOperatorValues`

`setSelectedObject`

setSelectedObjects

setSelectsFirstObjectAfterFetch

setUsesOptimisticRefresh

setValidatesChangesImmediately

usesOptimisticRefresh

validatesChangesImmediately

Setting the data source

setDataSource

dataSource

Setting the qualifier and sort ordering

setQualifier

qualifier

setSortOrderings

sortOrderings

Managing queries

qualifierFromQueryValues

setEqualToQueryValues

equalToQueryValues

setGreaterThanQueryValues

greaterThanQueryValues

setLessThanQueryValues

lessThanQueryValues

qualifyDataSource

qualifyDisplayGroup

enterQueryMode

inQueryMode

setInQueryMode

`enabledToSetSelectedObjectValueForKey`

Fetching objects from the data source

`fetch`

Getting the objects

`allObjects`

`displayedObjects`

Updating display of values

`redisplay`

`updateDisplayedObjects`

Setting the objects

`setObjectArray`

Changing the selection

`setSelectionIndexes`

`selectObjectsIdenticalTo`

`selectObject`

`clearSelection`

`selectNext`

`selectPrevious`

Examining the selection

`selectionIndexes`

`selectedObject`

`selectedObjects`

Adding keys

`setLocalKeys`

`localKeys`

Getting the associations

`observingAssociations`

Setting the delegate

setDelegate

delegate

Changing values from associations

setSelectedObjectValue

selectedObjectValueForKey

setValueForObject

valueForObject

setValueForObjectAtIndex

valueForObjectAtIndex

Editing by associations

associationDidBeginEditing

associationDidEndEditing

associationFailedToValidateValue

editingAssociation

endEditing

Querying changes for associations

contentsChanged

selectionChanged

updatedObjectIndex

Interacting withthe EOEditingContext

editorHasChangesForEditingContext

editingContextWillSaveChanges

editingContextPresentErrorMessage

Other methods

EODisplayGroup

globalDefaultForValidatesChangesImmediately globalDefaultStringMatchFormat

globalDefaultStringMatchOperator

setGlobalDefaultForValidatesChangesImmediately

setGlobalDefaultStringMatchFormat

setGlobalDefaultStringMatchOperator

awakeFromNib

delete

deleteObjectAtIndex

deleteSelection

editingContextShouldContinueFetching

insert

insertNewObjectAtIndex

insertObjectAtIndex

insertedObjectDefaultValues

objectsChangedInEditingContext

objectsInvalidatedInEditingContext

selectObjectsIdenticalToSelectFirstOnNoMatch

setInsertedObjectDefaultValues

undoManager

willChange

Constructors

EODisplayGroup

`public EODisplayGroup()`

Creates a new EODisplayGroup. The new display group needs to have an EODataSource set with `setDataSource`.

See Also: `bindAspect` **(EOAssociation)**

Static Methods

globalDefaultForValidatesChangesImmediately

`public static boolean globalDefaultForValidatesChangesImmediately()`

Returns `true` if the default behavior for new display group instances is to immediately handle validation errors, or `false` if the default behavior leaves errors for the EOEditingContext to handle when saving changes.

See Also: `validatesChangesImmediately`

globalDefaultStringMatchFormat

`public static String globalDefaultStringMatchFormat()`

Returns the default string match format string used by display group instances.

See Also: `defaultStringMatchFormat`

globalDefaultStringMatchOperator

`public static String globalDefaultStringMatchOperator()`

Returns the default string match operator used by display group instances.

See Also: `defaultStringMatchOperator`

setGlobalDefaultForValidatesChangesImmediately

`public static void setGlobalDefaultForValidatesChangesImmediately(boolean flag)`

Sets the default behavior display group instances use when they encounter a validation error. If `flag` is `true`, the default behavior is for display groups to immediately present an attention panel indicating a validation error. If `flag` is `false`, the default behavior if for display groups to leave validation errors to be handled when changes are saved. By default, display groups don't validate changes immediately.

See Also: `setValidatesChangesImmediately`

setGlobalDefaultStringMatchFormat

`public static void setGlobalDefaultStringMatchFormat(String format)`

Sets the default string match format to be used by display group instances. The default format string for pattern matching is "`%@*`".

See Also: `setDefaultStringMatchFormat`

setGlobalDefaultStringMatchOperator

`public static void setGlobalDefaultStringMatchOperator(String op)`

Sets the default string match operator to be used by display group instances. The default operator is case insensitive like.

See Also: `setDefaultStringMatchOperator`

Instance Methods

allObjects

```
public NSArray allObjects()
```

Returns all of the objects collected by the receiver.

See Also: displayedObjects, fetch

associationDidBeginEditing

```
public void associationDidBeginEditing(EOAssociation anEOAssociation)
```

Invoked by anAssociation when its display object begins editing to record that EOAssociation as the editing association.

See Also: editingAssociation, endEditing, associationFailedToValidateValue

associationDidEndEditing

```
public void associationDidEndEditing(EOAssociation anEOAssociation)
```

Invoked by anAssociation to clear the editing association. If anAssociation is the receiver's editing association, clears the editing association. Otherwise does nothing.

See Also: editingAssociation, endEditing, associationFailedToValidateValue

associationFailedToValidateValue

```
public boolean associationFailedToValidateValue(
    EOAssociation anEOAssociation,
    String value,
    String key,
    Object anObject,
    String errorDescription)
```

Invoked by anAssociation from its shouldEndEditingAtIndex method to let the receiver handle a validation error. This method opens an attention panel with errorDescription as the message and returns false.

See Also: displayGroupShouldDisplayAlert (EODisplayGroup.Delegate)

awakeFromNib

```
public void awakeFromNib()
```

Invoked when the receiver is unarchived from a nib file to prepare it for use in an application. You should never invoke this method directly. Finishes initializing the receiver and updates the display.

See Also: redisplay

clearSelection

```
public boolean clearSelection()
```

Invokes setSelectionIndexes to clear the selection, returning true on success and false on failure.

contentsChanged

```
public boolean contentsChanged()
```

Returns true if the receiver's array of objects has changed and not all observers have been notified, false otherwise. EOAssociations use this in their subjectChanged methods to determine what they need to update.

See Also: selectionChanged, updatedObjectIndex

dataSource

`public com.webobjects.eocontrol.EODataSource dataSource()`

Returns the receiver’s EODataSource.

See Also: `setDataSource`

defaultStringMatchFormat

`public String defaultStringMatchFormat()`

Returns the format string that specifies how pattern matching will be performed on string values in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`). If a key in the `queryMatch` dictionary does not have an associated operator in the `queryOperatorValues` dictionary, then its value is matched using pattern matching, and the format string returned by this method specifies how it will be matched.

See Also: `defaultStringMatchOperator`, `setDefaultStringMatchFormat`

defaultStringMatchOperator

`public String defaultStringMatchOperator()`

Returns the operator used to perform pattern matching for string values in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`). If a key in one of the query dictionaries does not have an associated operator in the `queryOperatorValues` dictionary, then the operator returned by this method is used to perform pattern matching.

See Also: `defaultStringMatchFormat`, `setDefaultStringMatchOperator`

delegate

`public Object delegate()`

Returns the receiver’s delegate.

See Also: `setDelegate`

delete

`public void delete()`

Deprecated. Use `deleteSelection`. instead.

deleteObjectAtIndex

`public boolean deleteObjectAtIndex(int` **index**`)`

Attempts to delete the object at **index**, returning `true` if successful and `false` if not. Checks with the delegate using `displayGroupShouldDeleteObject`. If the delegate returns `false`, this method fails and returns `false`. If successful, sends the delegate a `displayGroupDidDeleteObject` message.

This method performs the delete by sending `deleteObject` to the EODataSource. If that message throws an exception, this method fails and returns `false`.

deleteSelection

`public boolean deleteSelection()`

Attempts to delete the selected objects, returning `true` if successful and `false` if not.

displayedObjects

`public NSArray displayedObjects()`

Returns the objects that should be displayed or otherwise made available to the user, as filtered by the receiver's delegate or by its qualifier and sort ordering.

See Also: `allObjects`, `updateDisplayedObjects`, `displayGroupDisplayArrayForObjects` (EODisplayGroup.Delegate), `qualifier`, `sortOrderings`

dispose

`public void dispose()`

See the method description in the documentation for NSDisposable.

editingAssociation

```
public EOAssociation editingAssociation()
```

Returns the EOAssociation editing a value if there is one, `false` if there isn't.

See Also: `associationDidBeginEditing`, `associationDidEndEditing`

editingContextPresentErrorMessage

```
public void editingContextPresentErrorMessage(
    com.webobjects.eocontrol.EOEditingContext anEOEditingContext,
    String errorMessage)
```

Invoked by `anEditingContext` as part of the EOEditingContext.MessageHandlers interface, this method presents an attention panel with `errorMessage` as the message to display.

editingContextShouldContinueFetching

```
public boolean editingContextShouldContinueFetching(
    com.webobjects.eocontrol.EOEditingContext anEOEditingContext,
    int count,
    int limit,
    com.webobjects.eocontrol.EOObjectStore anEOObjectStore)
```

Invoked by `anEditingContext` as part of the EOEditingContext.MessageHandlers interface, this method presents an attention panel prompting the user about whether or not to continue fetching the current result set.

editingContextWillSaveChanges

```
public void editingContextWillSaveChanges(
    com.webobjects.eocontrol.EOEditingContext anEOEditingContext)
```

Invoked by anEditingContext in its `saveChanges` method as part of the EOEditors informal protocol, this method allows the EODisplayGroup to prohibit a save operation. EODisplayGroup's implementation of this method invokes `endEditing`, and throws an exception if it returns `false`. Thus, if there's an association that refuses to end editing, anEditingContext doesn't save changes.

editorHasChangesForEditingContext

```
public boolean editorHasChangesForEditingContext(
    com.webobjects.eocontrol.EOEditingContext anEOEditingContext)
```

Invoked by `anEditingContext` as part of the EOEditors interface, this method returns `false` if any association is editing, `true` otherwise.

See Also: `editingAssociation`, `associationDidBeginEditing`, `associationDidEndEditing`

enabledToSetSelectedObjectValueForKey

```
public boolean enabledToSetSelectedObjectValueForKey(String key)
```

Returns `true` to indicate that a single value association (such as an EOControlAssociation for a NSTextField) should be enabled for setting `key`, `false` otherwise. Normally this is the case if the receiver has a selected object. However, if `key` is a special query key (for example, "@query=.name"), then the control should be enabled even without a selected object.

endEditing

```
public boolean endEditing()
```

Attempts to end any editing taking place. If there's no editing association or if the editing association responds `true` to an `endEditing` message, returns `true`. Otherwise returns `false`.

See Also: `editingAssociation`

enterQueryMode

```
public void enterQueryMode()
```

This action method invokes `setInQueryMode` with an argument of `true`.

equalToQueryValues

`public NSDictionary equalToQueryValues()`

Returns the receiver's dictionary of equalTo query values. This dictionary is typically manipulated by associations bound to keys of the form @query=.propertyName. The `qualifierFromQueryValues` method uses this dictionary along with the lessThan and greaterThan dictionaries to construct qualifiers.

See Also: `setEqualToQueryValues`, `greaterThanQueryValues`, `lessThanQueryValues`

fetch

`public boolean fetch()`

Attempts to fetch objects from the EODataSource, returning `true` on success and `false` on failure.

Before fetching, invokes `endEditing` and sends `displayGroupShouldFetch` to the delegate, returning `false` if either of these methods does. If both return `true`, sends a `fetchObjects` message to the receiver's EODataSource to replace the object array, and if successful sends the delegate a `displayGroupDidFetchObjects` message.

fetchesOnLoad

`public boolean fetchesOnLoad()`

Returns `true` if the receiver fetches automatically after being loaded from a nib file, `false` if it must be told explicitly to fetch. The default is `false`. You can set this behavior in Interface Builder using the Inspector panel.

See Also: `fetch`, `fetchesOnLoad`

greaterThanQueryValues

```
public NSDictionary greaterThanQueryValues()
```

Returns the receiver’s dictionary of greaterThan query values. This dictionary is typically manipulated by associations bound to keys of the form @query>.propertyName. The `qualifierFromQueryValues` method uses this dictionary along with the lessThan and equalTo dictionaries to construct qualifiers.

See Also: `setGreaterThanQueryValues`,, `equalToQueryValues`

inQueryMode

```
public boolean inQueryMode()
```

Returns `true` to indicate that the receiver is in query mode, `false` otherwise. In query mode, user interface controls that normally display values become empty, allowing users to type queries directly into them (this is also known as a “Query By Example” interface). In effect, the receiver’s “displayedObjects” are replaced with an empty equalTo query values dictionary. When `qualifyDisplayGroup` or `qualifyDataSource` is subsequently invoked, the query is performed and the display reverts to displaying values—this time, the objects returned by the query.

See Also: `setInQueryMode`, `enterQueryMode`

insert

```
public void insert()
```

This action method invokes `insertObjectAtIndex` with an index just past the first index in the selection, or 0 if there’s no selection.

insertNewObjectAtIndex

```
public Object insertNewObjectAtIndex(int anIndex)
```

Asks the receiver’s EODataSource to create a new object by sending it a `createObject` message, then inserts the new object using `insertObjectAtIndex`. The EODataSource createObject method has the effect of inserting the object into the EOEditingContext.

If a new object can't be created, this method sends the delegate a `displayGroupCreateObjectFailed` message or, if the delegate doesn't respond, opens an attention panel to inform the user of the error.

See Also: `insert`

insertObjectAtIndex

```
public boolean insertObjectAtIndex(
    Object anObject,
    int index)
```

Inserts **anObject** into the receiver's EODataSource and `displayedObjects` array at **index**, if possible. This method checks with the delegate before actually inserting, using `displayGroupShouldInsertObject`. If the delegate refuses, **anObject** isn't inserted. After successfully inserting the object, this method informs the delegate with a `displayGroupDidInsertObject` message, and selects the newly inserted object. Throws an exception if **index** is out of bounds.

Unlike the `insertNewObjectAtIndex` method, this method does not insert the object into the EOEditingContext. If you use this method, you're responsible for inserting the object into the EOEditingContext yourself.

insertedObjectDefaultValues

```
public NSDictionary insertedObjectDefaultValues()
```

Returns the default values to be used for newly inserted objects. The keys into the dictionary are the properties of the entity that the display group manages. If the dictionary returned by this method is empty, the `insert...` method adds an object that is initially empty. Because the object is empty, the display group has no value to display on the HTML page for that object, meaning that there is nothing for the user to select and modify. Use the `setInsertedObjectDefaultValues` method to set up a default value so that there is something to display on the page.

lessThanQueryValues

`public NSDictionary lessThanQueryValues()`

Returns the receiver’s dictionary of lessThan query values. This dictionary is typically manipulated by associations bound to keys of the form @query<.propertyName. The `qualifierFromQueryValues` method uses this dictionary along with the greaterThan and equalTo dictionaries to construct qualifiers.

See Also: `setLessThanQueryValues`, `greaterThanQueryValues`, `equalToQueryValues`

localKeys

`public NSArray localKeys()`

Returns the additional keys that EOAssociations can be bound to. An EODisplayGroup’s basic keys are typically those of the attributes and relationships of its objects, as defined by their EOClassDescription through an EOEntity in the model. Local keys are typically used to form associations with key paths, with arbitrary methods of objects, or with properties of objects not associated with an EOEntity. Interface Builder allows the user to add and remove local keys in the EODisplayGroup Attributes Inspector panel.

See Also: `setLocalKeys`

objectsChangedInEditingContext

`public void objectsChangedInEditingContext(NSNotification` **aNSNotification**`)`

Description forthcoming.

objectsInvalidatedInEditingContext

`public void objectsInvalidatedInEditingContext(NSNotification` **aNSNotification**`)`

Description forthcoming.

observingAssociations

```
public NSArray observingAssociations()
```

Returns all EOAssociations that observe the receiver’s objects.

qualifier

```
public com.webobjects.eocontrol.EOQualifier qualifier()
```

Returns the receiver’s qualifier, which it uses to filter its array of objects for display when the delegate doesn’t do so itself.

See Also: `updateDisplayedObjects`, `displayedObjects`, `setQualifier`

qualifierFromQueryValues

```
public com.webobjects.eocontrol.EOQualifier qualifierFromQueryValues()
```

Builds a qualifier constructed from entries in the three query dictionaries: equalTo, greaterThan, and lessThan. These, in turn, are typically manipulated by associations bound to keys of the form @query=.firstName, @query>.budget, @query<.budget.

See Also: `qualifyDisplayGroup`, `qualifyDataSource`

qualifyDataSource

```
public void qualifyDataSource()
```

Takes the result of `qualifierFromQueryValues` and applies to the receiver's data source. The receiver then sends itself a `fetch` message. If the receiver is in query mode, query mode is exited. This method differs from `qualifyDisplayGroup` as follows: whereas `qualifyDisplayGroup` performs in-memory filtering of already fetched objects, `qualifyDataSource` triggers a new qualified fetch against the database.

qualifyDisplayGroup

`public void qualifyDisplayGroup()`

Takes the result of `qualifierFromQueryValues` and applies to the receiver using `setQualifier`. The method `updateDisplayedObjects` is invoked to refresh the display. If the receiver is in query mode, query mode is exited.

queryBindingValues

`public NSDictionary queryBindingValues()`

Returns a dictionary containing the actual values that the user wants to query upon. You use this method to perform a query stored in the model file. Bind keys in this dictionary to elements on your component that specify query values, then pass this dictionary to the fetch specification that performs the fetch.

queryOperatorValues

`public NSDictionary queryOperatorValues()`

Returns a dictionary of operators to use on items in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`). If a key in a query dictionary also exists in `queryOperatorValues`, that operator for that key is used.

See Also: `qualifierFromQueryValues`

redisplay

`public void redisplay()`

Notifies all observing associations to redisplay their values.

See Also: `observingAssociations`

selectNext

```
public boolean selectNext()
```

Attempts to select the object just after the currently selected one, returning `true` if successful and `false` if not. The selection is altered in this way:

- If there are no objects, does nothing and returns `false`.
- If there's no selection, selects the object at index zero and returns `true`.
- If the first selected object is the last object in the displayed objects array, selects the first object and returns `true`.
- Otherwise selects the object after the first selected object.

selectObject

```
public boolean selectObject(Object anObject)
```

Returns `true` to indicate that the receiver has found and selected **anObject**, `false` if it can't find a match for **anObject** (in which case it clears the selection). The selection is performed on the receiver's `displayedObjects`, not on `allObjects`.

selectObjectsIdenticalTo

```
public boolean selectObjectsIdenticalTo(NSArray objectSelection)
```

Attempts to select the objects in the receiver's displayed objects array which are equal to those of **objects**, returning `true` if successful and `false` otherwise.

selectObjectsIdenticalToSelectFirstOnNoMatch

```
public boolean selectObjectsIdenticalToSelectFirstOnNoMatch(
    NSArray objectSelection,
    boolean flag)
```

Selects the objects in the receiver's displayed objects array that are equal to those of **objects**, returning `true` if successful and `false` otherwise. If no objects in the displayed objects array match **objects** and **flag** is `true`, attempts to select the first object in the displayed objects array.

See Also: `setSelectionIndexes`

selectPrevious

```
public boolean selectPrevious()
```

Attempts to select the object just before the presently selected one, returning `true` if successful and `false` if not. The selection is altered in this way:

- If there are no objects, does nothing and returns `false`.
- If there's no selection, selects the object at index zero and returns `true`.
- If the first selected object is at index zero, selects the last object and returns `true`.
- Otherwise selects the object before the first selected object.

selectedObject

```
public Object selectedObject()
```

Returns the first selected object in the displayed objects array, or `null` if there's no such object.

See Also: `displayedObjects`, `selectionIndexes`

selectedObjectValueForKey

```
public Object selectedObjectValueForKey(String key)
```

Returns the value corresponding to *key* for the first selected object in the receiver's displayed objects array, or `null` if exactly one object isn't selected.

See Also: `valueForObjectAtIndex`

selectedObjects

```
public NSArray selectedObjects()
```

Returns the objects selected in the receiver's displayed objects array.

See Also: `displayedObjects`, `selectionIndexes`

selectionChanged

`public boolean selectionChanged()`

Returns `true` if the selection has changed and not all observers have been notified, `false` otherwise. EOAssociations use this in their `subjectChanged` methods to determine what they need to update.

See Also: `contentsChanged`

selectionIndexes

`public NSArray selectionIndexes()`

Returns the indexes of the receiver's selected objects as Numbers, in terms of its displayed objects array.

See Also: `displayedObjects`, `selectedObjects`, `selectedObject`, `setSelectionIndexes`

selectsFirstObjectAfterFetch

`public boolean selectsFirstObjectAfterFetch()`

Returns `true` if the receiver automatically selects its first displayed object after a fetch if there was no selection, `false` if it leaves an empty selection as-is.

See Also: `displayedObjects`, `fetch`, `setSelectsFirstObjectAfterFetch`

setDataSource

`public void setDataSource(com.webobjects.eocontrol.EODataSource anEODataSource)`

Sets the receiver's EODataSource to **aDataSource**. In the process, it performs these actions:

- Unregisters `self` as an editor and message handler for the previous EODataSource's EOEditingContext, if necessary, and registers `self` with **aDataSource**'s editing context. If the new editing context already has a message handler, however, the receiver doesn't assume that role.
- Registers `self` for `ObjectsChangedInEditingContextNotification` and `InvalidatedAllObjectsInStoreNotification` from the new editing context.

- Clears the receiver's array of objects.
- Sends `displayGroupDidChangeDataSource` to the delegate if there is one.

See Also: `dataSource`

setDefaultStringMatchFormat

```
public void setDefaultStringMatchFormat(String format)
```

Sets how pattern matching will be performed on String values in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`). This format is used for query dictionary properties that have String values and that do not have an associated entry in the `queryOperatorValues` dictionary. In these cases, the value is matched using pattern matching and format specifies how it will be matched.

The default format string for pattern matching is "`%@*`" which means that the string value in the `queryMatch` dictionary is used as a prefix. For example, if the query dictionary contains a value "Jo" for the key "Name", the query returns all records whose name values begin with "Jo".

See Also: `defaultStringMatchFormat`, `setDefaultStringMatchOperator`

setDefaultStringMatchOperator

```
public void setDefaultStringMatchOperator(String matchOperator)
```

Sets the operator used to perform pattern matching for String values in the `queryMatch` dictionary. This operator is used for properties listed in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`) that have String values and that do not have an associated entry in the `queryOperatorValues` dictionary. In these cases, the operator **matchOperator** is used to perform pattern matching.

The default value for the query match operator is `caseInsensitiveLike`, which means that the query does not consider case when matching letters. The other possible value for this operator is `like`, which matches the case of the letters exactly.

See Also: `defaultStringMatchOperator`, `setDefaultStringMatchFormat`

setDelegate

`public void setDelegate(Object anObject)`

Sets the receiver's delegate to `anObject`.

See Also: `delegate`

setEqualToQueryValues

`public void setEqualToQueryValues(NSDictionary values)`

Sets to `values` the receiver's dictionary of equalTo query values. The `qualifierFromQueryValues` method uses this dictionary along with the lessThan and greaterThan dictionaries to construct qualifiers.

See Also: `equalToQueryValues`, `setLessThanQueryValues`, `setGreaterThanQueryValues`

setFetchesOnLoad

`public void setFetchesOnLoad(boolean flag)`

Controls whether the receiver automatically fetches its objects after being loaded from a nib file. If `flag` is `true` it does; if `flag` is `false` the receiver must be told explicitly to fetch. The default is `false`. You can also set this behavior in Interface Builder using the Inspector panel.

See Also: `fetch`, `fetchesOnLoad`

setGreaterThanQueryValues

`public void setGreaterThanQueryValues(NSDictionary values)`

Sets to `values` the receiver's dictionary of greaterThan query values. The `qualifierFromQueryValues` method uses this dictionary along with the lessThan and equalTo dictionaries to construct qualifiers.

See Also: `greaterThanQueryValues`, `setLessThanQueryValues`, `setEqualToQueryValues`

setInQueryMode

`public void setInQueryMode(boolean flag)`

Sets according to `flag` whether the receiver is in query mode.

See Also: `inQueryMode`, `enterQueryMode`

setInsertedObjectDefaultValues

`public void setInsertedObjectDefaultValues(NSDictionary defaultValues)`

Sets default values to be used for newly inserted objects. When you use the `insert...` method to add an object, that object is initially empty. Because the object is empty, there is no value to be displayed on the HTML page, meaning there is nothing for the user to select and modify. You use this method to provide at least one field that can be displayed for the newly inserted object. The possible keys into the dictionary are the properties of the entity managed by this display group.

See Also: `insertedObjectDefaultValues`

setLessThanQueryValues

`public void setLessThanQueryValues(NSDictionary values)`

Sets to values the receiver's dictionary of lessThan query values. The `qualifierFromQueryValues` method uses this dictionary along with the greaterThan and equalTo dictionaries to construct qualifiers.

See Also: `lessThanQueryValues`, `setGreaterThanQueryValues`, `setEqualToQueryValues`

setLocalKeys

`public void setLocalKeys(NSArray keys)`

Sets the additional keys to which EOAssociations can be bound to the strings in keys. Instead of invoking this method programmatically, you can use Interface Builder to add and remove local keys in the EODisplayGroup Attributes Inspector panel.

See Also: `localKeys`

setObjectArray

```
public void setObjectArray(NSArray objects)
```

Sets the receiver's objects to `objects`, regardless of what its EODataSource provides. This method doesn't affect the EODataSource's objects at all; specifically, it results in neither inserts or deletes of objects in the EODataSource. `objects` should contain objects with the same property names or methods as those accessed by the receiver. This method is used by `fetch` to set the array of fetched objects; you should rarely need to invoke it directly.

After setting the object array, this method restores as much of the original selection as possible by invoking `selectObjectsIdenticalTo`. If there's no match and the receiver selects after fetching, then the first object is selected.

See Also: `allObjects`, `displayedObjects`, `selectsFirstObjectAfterFetch`

setQualifier

```
public void setQualifier(com.webobjects.eocontrol.EOQualifier anEOQualifier)
```

Sets the receiver's qualifier to `aQualifier`. This qualifier is used to filter (in memory) the receiver's array of objects for display when the delegate doesn't do so itself. Use `updateDisplayedObjects` to apply the qualifier.

Note: To set the qualifier used to fetch objects from the database, set the qualifier of the display group's `dataSource` (assuming that the data source is an EODatabaseDataSource).

If the receiver's delegate responds to `displayGroupDisplayArrayForObjects`, that method is used instead of the qualifier to filter the objects.

See Also: `displayedObjects`, `qualifier`, `qualifierFromQueryValues`

setQueryBindingValues

```
public void setQueryBindingValues(NSDictionary values)
```

Sets the dictionary of values that a user wants to query on. You use this method to perform a query stored in the model file. Bind keys in the `queryBindingValues` dictionary to elements of your component that specify query values.

setQueryOperatorValues

```
public void setQueryOperatorValues(NSDictionary values)
```

Sets the dictionary of operators to use on items in the query dictionaries (`equalToQueryValues`, `greaterThanQueryValues`, and `lessThanQueryValues`). If a key in a query dictionary also exists in `queryOperatorValues`, that operator for that key is used.

setSelectedObject

```
public void setSelectedObject(Object anObject)
```

Sets the selected objects to **anObject**.

setSelectedObjectValue

```
public boolean setSelectedObjectValue(
    Object value,
    String key)
```

Invokes `setValueForObject` with the first selected object, returning `true` if successful and `false` otherwise. This method should be invoked only by EOAssociation objects to propagate changes from display objects.

See Also: `setValueForObjectAtIndex`, `valueForObject`

setSelectedObjects

```
public void setSelectedObjects(NSArray objects)
```

Sets the selected objects to **objects**.

setSelectionIndexes

```
public boolean setSelectionIndexes(NSArray indexes)
```

Selects the objects at **indexes** in the receiver's array if possible, returning `true` if successful and `false` if not (in which case the selection remains unaltered). **indexes** is an array of Numbers. This method is the primitive method for altering the selection; all other such methods invoke this one to make the change.

This method invokes `endEditing` to wrap up any changes being made by the user. If `endEditing` returns `false`, this method fails and returns `false`. This method then checks the delegate with a `displayGroupShouldChangeSelection` message. If the delegate returns `false`, this method also fails and returns `false`. If the receiver successfully changes the selection, its observers (typically EOAssociations) each receive a `subjectChanged` message.

setSelectsFirstObjectAfterFetch

`public void setSelectsFirstObjectAfterFetch(boolean flag)`

Controls whether the receiver automatically selects its first displayed object after a fetch when there were no selected objects before the fetch. If **flag** is `true` it does; if **flag** is `false` then no objects are selected. By default, display groups select the first object after a fetch when there was no previous selection.

See Also: `displayedObjects`, `fetch`, `selectsFirstObjectAfterFetch`

setSortOrderings

`public void setSortOrderings(NSArray orderings)`

Sets the EOSortOrdering objects that `updateDisplayedObjects` uses to sort the displayed objects to **orderings**. Use `updateDisplayedObjects` to apply the sort orderings.

If the receiver's delegate responds to `displayGroupDisplayArrayForObjects`, that method is used instead of the sort orderings to order the objects.

See Also: `displayedObjects`, `sortOrderings`

setUsesOptimisticRefresh

`public void setUsesOptimisticRefresh(boolean flag)`

Controls how the receiver redisplays on changes to objects. If **flag** is `true` it redisplays only when elements of its displayed objects array change; if **flag** is `false` it redisplays on any change in its EOEditingContext. Because changes to other objects can affect the displayed objects (through flattened attributes or custom methods, for example), EODisplayGroups by default use the more pessimistic refresh technique of redisplaying on any change in the EOEditingContext. If you know that none of the EOAssociations for a particular EODisplayGroup display derived values, you can turn on optimistic refresh to reduce redisplay time.

The default is `false`. You can also change this setting in Interface Builder's Inspector panel using the Refresh All check box.

See Also: `usesOptimisticRefresh`

setValidatesChangesImmediately

```
public void setValidatesChangesImmediately(boolean flag)
```

Controls the receiver's behavior on encountering a validation error. Whenever an EODisplayGroup sets a value in an object, it sends the object a `validateValueForKey` message, allowing the object to coerce the value's type to a more appropriate one or to return an exception indicating that the value isn't valid. If this method is invoked with a **flag** of `true`, the receiver immediately presents an attention panel indicating the validation error. If this method is invoked with a **flag** of `false`, the receiver leaves validation errors to be handled when changes are saved. By default, display groups don't validate changes immediately.

See Also: – `saveChanges` (EOEditingContext), `validatesChangesImmediately`

setValueForObject

```
public boolean setValueForObject(
    Object value,
    Object anObject,
    String key)
```

Sets a property of **anObject**, identified by **key**, to **value**. Returns `true` if successful and `false` otherwise. If a new value is set, sends the delegate a `displayGroupDidSetValueForObject` message.

This method should be invoked only by EOAssociation objects to propagate changes from display objects. Other application code should interact with the objects directly.

If the receiver validates changes immediately, it sends **anObject** a `validateValueForKey` message, returning `false` if the object refuses to validate **value**. Otherwise, validation errors are checked by the EOEditingContext when it attempts to save changes

See Also: `setValueForObjectAtIndex`, `setSelectedObjectValue`, `valueForObject`, `validatesChangesImmediately`

setValueForObjectAtIndex

```
public boolean setValueForObjectAtIndex(
    Object value,
    int index,
    String key)
```

Invokes `setValueForObject` with the object at **index**, returning `true` if successful and `false` otherwise. This method should be invoked only by EOAssociation objects to propagate changes from display objects.

See Also: `setSelectedObjectValue`, `valueForObjectAtIndex`

sortOrderings

```
public NSArray sortOrderings()
```

Returns an array of EOSortOrdering objects that `updateDisplayedObjects` uses to sort the displayed objects, as returned by the `displayedObjects` method.

See Also: `setSortOrderings`

undoManager

```
public NSUndoManager undoManager()
```

Returns the receiver's undo manager.

updateDisplayedObjects

```
public void updateDisplayedObjects()
```

Recalculates the receiver's displayed objects array and redisplays. If the receiver's delegate responds to `displayGroupDisplayArrayForObjects`, it's sent this message and the returned array is set as the display group's displayed object. Otherwise, the receiver applies its qualifier and sort ordering to its array of objects. In either case, any objects that were selected before remain selected in the new displayed objects array.

See Also: `redisplay`, `displayedObjects`, `selectedObjects`, `qualifier`, `sortOrderings`

updatedObjectIndex

```
public int updatedObjectIndex()
```

Returns the index in the displayed objects array of the most recently updated object, or –1 if more than one object has changed. The return value is meaningful only when `contentsChanged` returns `true`. EOAssociations can use this method to optimize redisplay of their user interface objects.

usesOptimisticRefresh

```
public boolean usesOptimisticRefresh()
```

Returns `true` if the receiver redisplays only when its displayed objects change, `false` if it redisplays on any change in its EOEditingContext.

See Also: `setUsesOptimisticRefresh`

validatesChangesImmediately

```
public boolean validatesChangesImmediately()
```

Returns `true` if the receiver immediately handles validation errors, or `false` if it leaves errors for the EOEditingContext to handle when saving changes.

See Also: `setValidatesChangesImmediately`

valueForObject

```
public Object valueForObject(
    NSKeyValueCodingAdditions anObject,
    String key)
```

Returns `anObject`'s value for the property identified by `key`.

valueForObjectAtIndex

```
public Object valueForObjectAtIndex(
    int index,
    String key)
```

Returns the value of the object at `index` for the property identified by `key`.

willChange

```
public void willChange()
```

Notifies observers that the receiver will change.

Notifications

DisplayGroupWillFetchNotification

```
public static final String DisplayGroupWillFetchNotification
```

Posted whenever an EODisplayGroup receives a `fetch` message. The notification contains:

Notification Object	The EODisplayGroup that received the `fetch` message.
Userinfo	None

EOMasterAssociation

Inherits from:	EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOMasterAssociation`

`isUsableWithObject`

`primaryAspect`

`priority`

Constructors

EOMasterAssociation

`public EOMasterAssociation(Object` **`anObject`**`)`

Description forthcoming.

Instance Methods

isUsableWithObject

```
public boolean isUsableWithObject(Object anObject)
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`SourceAspect`.

priority

```
public int priority()
```

Description forthcoming.

EOMasterCopyAssociation

Inherits from:	EOMasterAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOMasterCopyAssociation object synchronizes two EODisplayGroups that share the same data source but have different qualifiers.

By binding two display groups with an EOMasterCopyAssociation, any changes performed in one display group are immediately reflected in the other. Similarly, changing the selection in one display group immediately changes it in the other one.

Usable With
EODisplayGroup

Aspects

`parent`	An EODisplayGroup with which the association's display group should be synchronized.

Object Keys Taken

None

Examples

Suppose you have an EODisplayGroup for displaying Talent objects (actors and directors) and another display group for displaying the pictures of the Talents who are actors. When a Talent is selected in the first display group, you want the "actor" display group to select that Talent's picture if the selected Talent is an actor. Since both display groups manage Talent objects, they can share the same EODataSource. However, the first display group is unqualified—it fetches all Talent objects; the second display group is qualified to fetch only the Talents who are actors.

To do this, in Interface Builder, start with an unqualified display group for displaying all the Talents. Drag a second display group from the Enterprise Objects palette into your nib. Control-drag a connection from the new display group to the unqualified Talent display group. In the Connections inspector, choose EOMasterCopyAssociation, select the `parent` aspect, and click Connect. This action automatically sets the second display group's data source. Initially, the data source is set to an EODetailDataSource—that's what you'll see in Interface Builder. However, at runtime, the association switches the second display group's data source to that of the `parent` display group.

Now when you run the application, the display groups will be synchronized with one another. (You'll programmatically assign a qualifier to the second display group so that it filters out non-actor Talents.)

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOMasterCopyAssociation`

`subjectChanged`

Constructors

EOMasterCopyAssociation

`public EOMasterCopyAssociation(Object aDisplayObject)`

Creates a new EOMasterCopyAssociation to monitor and update the value in **aDisplayObject**, an EODisplayGroup.

You normally set up associations with the Interface Builder application, in which case you don't need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` **(EOAssociation),** `establishConnection` **(EOAssociation)**

Instance Methods

subjectChanged

```
public void subjectChanged()
```

Description forthcoming.

EOMasterDetailAssociation

Inherits from:	EOMasterAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOMasterDetailAssociation object binds one EODisplayGroup (the detail) to a relationship in another (the master), so that the detail display group contains the destination objects for the object selected in the master. The display groups' data sources also operate in a master-detail arrangement, meaning changes to one are immediately reflected in the other. In this arrangement, the detail EODisplayGroup's data source must be an EODetailDataSource. The detail objects are taken directly from the selected object in the master EODisplayGroup, so that changes to the objects in one EODisplayGroup are instantly reflected in the other.

In com.webobjects.eointerface.cocoa, by contrast, with an EOMasterPeerAssociation, the two EODisplayGroups are independent of each other. In a master-peer setup, insertions and deletions in the detail EODisplayGroup don't affect the corresponding relationship property of the selected object in the master EODisplayGroup. Master-peer setups are more appropriate when no insertions or deletions will be made in the detail EODisplayGroup. See the EOMasterPeerAssociation class specification for more information.

Example

Suppose you have a master EODisplayGroup displaying Movie objects and a detail display group displaying Talent objects. The two display groups are bound to one another through Movie's `directors` relationship—a to-many relationship from Movie to Talent. When a Movie is selected, you want the Talent display group to display the Talents who directed the Movie. Inserting a new director into the Talent display group should add the director to the selected Movie's `directors` relationship; and similarly, deleting a director from the Talent display group should remove the director from the selected Movie's `directors` relationship.

To do this, in Interface Builder, control-drag a connection from the Talent display group to the Movie display group. In the Connections inspector, choose EOMasterDetailAssociation, and bind `parent` aspect to the "directors" key.

Interfaces Implemented

NSDisposable

`dispose`

EOObserving

Method Types

All methods

`EOMasterDetailAssociation`

`isUsableWithObject`

`subjectChanged`

Constructors

EOMasterDetailAssociation

`public EOMasterDetailAssociation(Object` **`aDisplayObject`**`)`

Creates a new EOMasterDetailAssociation to monitor and update the value in **`aDisplayObject`**, an EODisplayGroup.

You normally set up associations with the Interface Builder application, in which case you don't need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` (EOAssociation), `establishConnection` **(EOAssociation)**

Instance Methods

dispose

`public void dispose()`

See the description in the documentation for NSDisposable.

isUsableWithObject

`public boolean isUsableWithObject(Object` **`aDisplayObject`**`)`

Returns `true` if **`aDisplayObject`** is an instance of EODisplayGroup and its `dataSource` is either `null` or an EODetailDataSource (EOControl).

See Also: `isUsableWithObject` (EOAssociation)

subjectChanged

```
public void subjectChanged()
```

See the `subjectChanged` method description in the superclass EOAssociation.

EOMasterPeerAssociation

Inherits from:	EOMasterDetailAssociation: EOMasterAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOMasterPeerAssociation binds two EODisplayGroups together in a master-detail relationship, where the detail EODisplayGroup shows the destination objects for the relationship of the master EODisplayGroup.

In a master-peer arrangement, the detail display group's data source is independent. Detail objects are fetched independently from the detail's data source, which means that changes to one display group aren't automatically reflected in the other. To update the other display group, it's necessary to save the changes made and then have the other display group fetch its objects anew.

Contrast this with a master-detail setup using an EOMasterDetailAssociation. With an EOMasterDetailAssociation, the display groups' data sources also operate in a master-detail arrangement, meaning changes to one are immediately reflected in the other. The detail objects are taken directly from the selected object in the master display group, so that changes to the objects in one display group are instantly reflected in the other. Master-peer setups display these advantages over master-detail setups:

- You can use them to display the destination objects for relationships that are defined in the model but not declared as class properties. This is typically done for rarely accessed information—or information that's costly to access. By not defining the relationship as a class property, the destination objects aren't stored as instance variables in the source objects, which saves memory and the cost of constructing faults for the relationship.
- Because the detail display group fetches objects with its own data source, you can configure the detail data source with an auxiliary EOQualifier to limit the objects fetched. This further reduces the cost of fetching data.
- You can use an EOMasterPeerAssociation to fetch detail information that may be updated in another editing context or even in another application; thus this association helps you to remain "up to date" with the database.

Generally, master-peer setups are only appropriate when no insertions or deletions will be made in the detail display group. For a master-detail relationship that reflects changes between two display groups, including insertions and deletions, use an EOMasterDetailAssociation.

Usable With

EODisplayGroups whose data sources are not EODetailDataSources

Aspects

`parent`	A relationship from the master EODisplayGroup.

Object Keys Taken

None

Example

Suppose you have a database of salesmen and their associated sales. Each salesman has a city ID. The sales are related to the salesmen by salesman ID, but also have a city ID. You want a list of all the sales in a salesman's city so you could evaluate it against other salesmen. For this, you create a relationship between salesman and sales based on city ID (the relationship is not a class property). You can then display that information using an EOMasterPeerAssociation.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOMasterPeerAssociation`

`isUsableWithObject`

Constructors

EOMasterPeerAssociation

`public EOMasterPeerAssociation(Object anObject)`

Description forthcoming.

Instance Methods

isUsableWithObject

```
public boolean isUsableWithObject(Object anObject)
```

Description forthcoming.

EOPickTextAssociation

Inherits from:	EOValueAssociation : EOWidgetAssociation : EOAssociation : EODelayedObserver (EOControl) : Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOPickTextAssociation takes the value of its display object, such as an NSTextField, and uses it to form a qualifier with up to three LIKE operators, each compared to a different key of the EODisplayGroup. This allows the user to perform a similarity search based on whole or partial values.

EOPickTextAssociations are most often used with a table view to qualify a list of fetched objects that is too long for convenient scrolling.

Usable With
Any NSControl

Aspects

`matchKey1`	An attribute to match using a LIKE qualifier.
`matchKey2`	An attribute to match using a LIKE qualifier.
`matchKey3`	An attribute to match using a LIKE qualifier.

Object Keys Taken

`target`	The EOPickTextAssociation applies its qualifier when sent an action message from the NSControl.
`delegate`	The EOPickTextAssociation applies its qualifier when sent a `controlTextDidChange` message, causing dynamic update as the user types.

Example

Make an EOPickTextAssociation between an NSTextField and an EODisplayGroup of People objects. Bind the `matchKey1` and `matchKey2` aspects to the “lastName” and “firstName” keys. If the user types “Bi” in the field, the EOPickTextAssociation applies the following qualifier to the EODisplayGroup:

```
(lastName like “*Bi*“) OR (firstName like “*Bi*“)
```

which matches names like “Bill Smith” and “Joe Biggs”. The list of objects displayed in the display group is restricted to those that match the qualifier.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOPickTextAssociation`

`displayValueAspect`

`primaryAspect`

Constructors

EOPickTextAssociation

`public EOPickTextAssociation(Object` **`aDisplayObject`**`)`

Creates a new EOPickTextAssociation to monitor and update the row values in `aDisplayObject`, an NSControl (Cocoa) which has a text as an attribute.

You normally set up associations with the Interface Builder application, in which case you don't need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` **(EOAssociation),** `establishConnection` **(EOAssociation)**

Instance Methods

displayValueAspect

```
protected String displayValueAspect()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`SourceAspect`.

EOTableAssociation

Inherits from:	EOWidgetAssociation : EOAssociation : EODelayedObserver (EOControl) : Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

EOTableAssociation associates the contents of its `SourceAspect`'s display group with an EOTableAssociation.TablePlugin . In general use, it should never be necessary to explicitly instantiate this class, as EOTableColumnAssociation's `setTable` assures that an instance exists for its `table`.

Usable With
EOTable• **and some cocoa class too, right? •**

Aspects
`EOAssociation.EnabledAspect`
`EOAssociation.SourceAspect`

Interfaces Implemented

NSDisposable

`dispose`

EOObserving

Method Types

Table attributes

`boldStateAtColumnAndRow`

`italicAtColumnAndRow`

`textColorAtColumnAndRow`

`valueAtColumnAndRow`

Other methods

`EOTableAssociation`

`editingTableColumnAssociation`

`isEditableAtColumnAndRow`

`numberOfDisplayedObjects`

`primaryAspect`

`setSortOrderingByTableColumnOrder`

`setSortsByColumnOrder`

`setValueAtColumnAndRow`

`sortsByColumnOrder`

`subjectChanged`

`tableDidChangeColumns`

`tableDidChangeSelection`

`widgetPluginClass`

Constructors

EOTableAssociation

```
public EOTableAssociation(Object aDisplayObject)
```

Creates a new EOTableAssociation to monitor and update the value in `aDisplayObject`, an EOTable.

In general use, it should never be necessary to explicitly instantiate this class, as EOTableColumnAssociation's `setTable` assures that an instance exists for its `table`.

See Also: `bindAspect` (EOAssociation), `establishConnection` (EOAssociation)

Instance Methods

boldStateAtColumnAndRow

```
public int boldStateAtColumnAndRow(
    int columnIndex,
    int rowIndex)
```

Description forthcoming.

dispose

```
public void dispose()
```

See the description in the documentation for NSDisposable.

editingTableColumnAssociation

```
public EOTableColumnAssociation editingTableColumnAssociation()
```

Description forthcoming.

isEditableAtColumnAndRow

```
public boolean isEditableAtColumnAndRow(
    int columnIndex,
    int rowIndex)
```

Description forthcoming.

italicAtColumnAndRow

```
public int italicAtColumnAndRow(
    int columnIndex,
    int rowIndex)
```

Description forthcoming.

numberOfDisplayedObjects

```
public int numberOfDisplayedObjects()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`SourceAspect`.

See Also: `primaryAspect` (EOAssociation)

setSortOrderingByTableColumnOrder

```
public void setSortOrderingByTableColumnOrder()
```

Description forthcoming.

setSortsByColumnOrder

```
public void setSortsByColumnOrder(boolean aBoolean)
```

Description forthcoming.

setValueAtColumnAndRow

```
public boolean setValueAtColumnAndRow(
    Object value,
    int columnIndex,
    int rowIndex)
```

Description forthcoming.

sortsByColumnOrder

```
public boolean sortsByColumnOrder()
```

Description forthcoming.

subjectChanged

```
public void subjectChanged()
```

See the `subjectChanged` method description in the superclass EOAssociation.

tableDidChangeColumns

```
public void tableDidChangeColumns()
```

Description forthcoming.

tableDidChangeSelection

```
public void tableDidChangeSelection()
```

Description forthcoming.

textColorAtColumnAndRow

```
public Object textColorAtColumnAndRow(
    int columnIndex,
    int rowIndex)
```

Description forthcoming.

valueAtColumnAndRow

```
public Object valueAtColumnAndRow(
    int anInt,
    int anInt)
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

EOTableAssociation.TablePlugin

Inherits from:	EOWidgetAssociation.WidgetPlugin : Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

`EOTableAssociation.TablePlugin`

`editingColumnIndex`

`editingRowIndex`

`existingTableAssociation`

`numberOfColumns`

`selectionIndexes`

`tableColumnAssociationForColumnAtIndex`

`updateSelectionIndexes`

`updateTableContents`

Constructors

EOTableAssociation.TablePlugin

```
public EOTableAssociation.TablePlugin(
    EOWidgetAssociation association,
    Object widget)
```

Description forthcoming.

Instance Methods

editingColumnIndex

```
public abstract int editingColumnIndex()
```

Description forthcoming.

editingRowIndex

```
public abstract int editingRowIndex()
```

Description forthcoming.

existingTableAssociation

```
public abstract EOTableAssociation existingTableAssociation()
```

Description forthcoming.

numberOfColumns

```
public abstract int numberOfColumns()
```

Description forthcoming.

selectionIndexes

```
public abstract int[] selectionIndexes()
```

Description forthcoming.

tableColumnAssociationForColumnAtIndex

```
public abstract EOTableColumnAssociation tableColumnAssociationForColumnAtIndex(int columnIndex)
```

Description forthcoming.

updateSelectionIndexes

```
public abstract void updateSelectionIndexes(int[] selectedRowIndexes[])
```

Description forthcoming.

updateTableContents

```
public abstract void updateTableContents(int numberOfRows)
```

Description forthcoming.

EOTableColumnAssociation

Inherits from:	EOWidgetAssociation : EOAssociation : EODelayedObserver (EOControl) : Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

An EOTableColumnAssociation associates a single attribute of all enterprise objects in its `ValueAspect`'s EODisplayGroup with a Swing JTable TableColumn . The value of each object's attribute is displayed in its corresponding row.

By far the easiest way to configure EOTableColumnAssociations is in Interface Builder, but they may also be instantiated programmatically. Because Swing's TableColumn maintains no reference to its containing JTable, this relationship must be explicitly specified via `setTable` before `establishConnection` is invoked.

Usable With

javax.swing.table.TableColumn **• and a cocoa class too? •**

Aspects

`BoldAspect`	Description forthcoming.
`EnabledAspect`	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.
`ItalicAspect`	Description forthcoming.
`ValueAspect`	An attribute of the objects, displayed in each row of the TableColumn.

Interfaces Implemented

NSDisposable

`dispose`

EOObserving

Method Types

All methods

`EOTableColumnAssociation`

`boldStateAtRow`

`endEditing`

`isEditableAtRow`

`italicStateAtRow`

`primaryAspect`

`setObject`

setSortingSelector

setTable

setValueAtRow

sortingSelector

table

textColorAtRow

valueAtRow

widgetDidBeginEditing

widgetDidEndEditing

widgetPluginClass

Constructors

EOTableColumnAssociation

```
public EOTableColumnAssociation(Object anObject)
```

Description forthcoming.

```
public EOTableColumnAssociation(
    Object object,
    Object table)
```

Description forthcoming.

Instance Methods

boldStateAtRow

`public int boldStateAtRow(int rowIndex)`

Description forthcoming.

dispose

`public void dispose()`

See the method description in the documentation for NSDisposable.

endEditing

`public boolean endEditing()`

Description forthcoming.

isEditableAtRow

`public boolean isEditableAtRow(int rowIndex)`

Returns whether or not the property bound to the receiver's `ValueAspect` is editable at `row`, as determined by the `EnabledAspect`. If this aspect is bound, a non-zero value at `row` indicates that the property may be edited. If the `EnabledAspect` is unbound all rows are considered editable.

italicStateAtRow

`public int italicStateAtRow(int rowIndex)`

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`ValueAspect`.

setObject

```
public void setObject(Object anObject)
```

Description forthcoming.

setSortingSelector

```
public void setSortingSelector(NSSelector selector)
```

Description forthcoming.

setTable

```
public void setTable(Object table)
```

Because TableColumn maintains no reference to its containing JTable, the consumer must explicitly specify this relationship by invoking `setTable` **before** `establishConnection`. This method also assures that an instance of EOTableAssociation exists for **table**.

setValueAtRow

```
public boolean setValueAtRow(
    Object value,
    int rowIndex)
```

Description forthcoming.

sortingSelector

```
public NSSelector sortingSelector()
```

Description forthcoming.

table

```
public Object table()
```

Description forthcoming.

textColorAtRow

```
public Object textColorAtRow(int rowIndex)
```

Description forthcoming.

valueAtRow

```
public Object valueAtRow(int rowIndex)
```

Description forthcoming.

widgetDidBeginEditing

```
public boolean widgetDidBeginEditing()
```

Description forthcoming.

widgetDidEndEditing

```
public boolean widgetDidEndEditing()
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

EOTableColumnAssociation.TableColumnPlugin

Inherits from:	EOWidgetAssociation.WidgetPlugin: Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

```
EOTableColumnAssociation.TableColumnPlugin
columnIndexInTable
displayValueForValue
endEditing
isEditable
table
tableAssociation
valueForDisplayValue
```

Constructors

EOTableColumnAssociation.TableColumnPlugin

```
public EOTableColumnAssociation.TableColumnPlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object widget)
```

Description forthcoming.

Instance Methods

columnIndexInTable

```
public abstract int columnIndexInTable()
```

Description forthcoming.

displayValueForValue

```
public abstract Object displayValueForValue(Object value)
```

Description forthcoming.

endEditing

```
public abstract boolean endEditing()
```

Description forthcoming.

isEditable

```
public abstract boolean isEditable()
```

Description forthcoming.

table

```
public abstract Object table()
```

Description forthcoming.

tableAssociation

```
public abstract EOTableAssociation tableAssociation()
```

Description forthcoming.

valueForDisplayValue

```
public abstract Object valueForDisplayValue(Object displayValue)
```

Description forthcoming.

EOTextAssociation

Inherits from:	EOValueAssociation: EOWidgetAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

In a Java Client application (using Swing), an EOTextAssociation object displays a plain text attribute in an EOTextField, EOTextArea, or EOFormCell by binding the text object to a string. Text is written back to the object as a String.

In a Cocoa application, an EOTextAssociation object displays a plain or rich text attribute in an NSText object by binding the text object to a string or NSData attribute. It determines the kind of text received from an object by examining the beginning for signature codes specific to RTF and RTFD. When writing text back to the object, the association examines the configuration of the NSText object to determine the type to use according to the following table:

Multiple Fonts	Allows Graphics	Type Written to Object
NO	NO	NSString text
YES	NO	NSData containing RTF
YES	YES	NSData containing RTFD

Usable With

com.webobjects.eointerface.swing:
EOTextField, EOTextArea, EOFormCell

com.webobjects.eointerface.cocoa:
NSText, NSTextView

Aspects

`value`	A text attribute of the selected object.
`URL`	Description forthcoming.
`enabled`	Description forthcoming.
`textColor`	Description forthcoming.
`backgroundColor`	Description forthcoming.
`bold`	Description forthcoming.
`italic`	Description forthcoming.

Object Keys Taken • how do I know what object keys are taken? •

`delegate`	An EOTextAssociation accepts delegate messages related to the editing and validation of text; see the NSText and NSTextView class specifications for more information.

Interfaces Implemented

NSDisposable

`dispose`

EOObserving

Method Types

All methods

`EOTextAssociation`

`defaultDisabledBackgroundColor`

`defaultEnabledBackgroundColor`

`setDefaultBackgroundColors`

`displayValueFromURL`

`setUsesDefaultBackgroundColors`

`usesDefaultBackgroundColors`

Constructors

EOTextAssociation

```
public EOTextAssociation(Object aDisplayObject)
```

Creates a new EOTextAssociation to monitor and update the value in `aDisplayObject`, which is typically Cocoa NSActionCell or, in Swing applications, an EOFormCell.

You normally set up associations with the Interface Builder application, in which case you don't need to create them programmatically. However, if you do create them up programmatically, setting them up is a multi-step process. After creating an association, you must bind its aspects and establish its connections.

See Also: `bindAspect` (EOAssociation), `establishConnection` **(EOAssociation)**

Static Methods

defaultDisabledBackgroundColor

```
public static Object defaultDisabledBackgroundColor()
```

Description forthcoming.

defaultEnabledBackgroundColor

```
public static Object defaultEnabledBackgroundColor()
```

Description forthcoming.

setDefaultBackgroundColors

```
public static void setDefaultBackgroundColors(
    Object enabledColor,
    Object disabledColor)
```

Description forthcoming.

Instance Methods

displayValueFromURL

```
protected Object displayValueFromURL(String URLString)
```

Description forthcoming.

dispose

```
public void dispose()
```

See the description in the documentation for NSDisposable.

setUsesDefaultBackgroundColors

```
public void setUsesDefaultBackgroundColors(boolean flag)
```

Description forthcoming.

usesDefaultBackgroundColors

```
public boolean usesDefaultBackgroundColors()
```

Description forthcoming.

EOTextAssociation.TextPlugin

Inherits from:	EOValueAssociation.ValuePlugin : EOWidgetAssociation.WidgetPlugin : Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

`EOTextAssociation.TextPlugin`

`setColors`

`setFontProperties`

Constructors

EOTextAssociation.TextPlugin

```
public EOTextAssociation.TextPlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object widget)
```

Description forthcoming.

Instance Methods

setColors

```
public abstract void setColors(
    Object textColor,
    Object bgColor)
```

Description forthcoming.

setFontProperties

```
public abstract void setFontProperties(
    int boldState,
    int italicState)
```

Description forthcoming.

EOValueAssociation

Inherits from:	EOWidgetAssociation : EOAssociation : EODelayedObserver (EOControl) : Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOValueAssociation`

`bindAspect`

`displayValueAspect`

`displayValueFromURL`

`endEditing`

`primaryAspect`

`subjectChanged`

`widgetDidBeginEditing`

`widgetDidChange`

`widgetDidEndEditing`

`widgetPluginClass`

Constructors

EOValueAssociation

```
public EOValueAssociation(Object anObject)
```

Description forthcoming.

Instance Methods

bindAspect

```
public void bindAspect(
    String aspect,
    EODisplayGroup anEODisplayGroup,
    String key)
```

Description forthcoming.

displayValueAspect

```
protected String displayValueAspect()
```

Description forthcoming.

displayValueFromURL

```
protected Object displayValueFromURL(String URLString)
```

Description forthcoming.

endEditing

```
public boolean endEditing()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`ValueAspect`.

subjectChanged

```
public void subjectChanged()
```

Description forthcoming.

widgetDidBeginEditing

```
public boolean widgetDidBeginEditing()
```

Description forthcoming.

widgetDidChange

```
public boolean widgetDidChange()
```

Description forthcoming.

widgetDidEndEditing

```
public boolean widgetDidEndEditing()
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

EOValueAssociation.ValuePlugin

Inherits from:	EOWidgetAssociation.WidgetPlugin: Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

```
EOValueAssociation.ValuePlugin

setValue

useURLAsValue

value
```

Constructors

EOValueAssociation.ValuePlugin

```
public EOValueAssociation.ValuePlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object anObject)
```

Description forthcoming.

Instance Methods

setValue

```
public abstract void setValue(
    Object anObject,
    boolean aBoolean)
```

Description forthcoming.

useURLAsValue

```
public boolean useURLAsValue()
```

Description forthcoming.

value

```
public abstract Object value()
```

Description forthcoming.

EOValueSelectionAssociation

Inherits from:	EOWidgetAssociation: EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

EOObserving

Method Types

All methods

`EOValueSelectionAssociation`

`endEditing`

`primaryAspect`

`subjectChanged`

`widgetPluginClass`

`widgetSelectionDidChange`

Constructors

EOValueSelectionAssociation

```
public EOValueSelectionAssociation(Object anObject)
```

Description forthcoming.

Instance Methods

endEditing

```
public boolean endEditing()
```

Description forthcoming.

primaryAspect

```
public String primaryAspect()
```

Returns EOAssociation.`SelectedTitleAspect`.

subjectChanged

```
public void subjectChanged()
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

widgetSelectionDidChange

```
public boolean widgetSelectionDidChange()
```

Description forthcoming.

C L A S S

EOValueSelectionAssociation. ValueSelectionPlugin

Inherits from:	EOWidgetAssociation.WidgetPlugin: Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

Method Types

All methods

```
EOValueSelectionAssociation.ValueSelectionPlugin

selectionIndex

setSelectionIndex

setTitlesFromObjects

titles
```

Constructors

EOValueSelectionAssociation.ValueSelectionPlugin

```
public EOValueSelectionAssociation.ValueSelectionPlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object widget)
```

Description forthcoming.

Instance Methods

selectionIndex

```
public abstract int selectionIndex()
```

Description forthcoming.

setSelectionIndex

```
public abstract void setSelectionIndex(
    int selectionIndex,
    boolean isEnabled)
```

Description forthcoming.

setTitlesFromObjects

```
public abstract void setTitlesFromObjects(Object[] objects[])
```

Description forthcoming.

titles

```
public abstract String[] titles()
```

Description forthcoming.

EOWidgetAssociation

Inherits from:	EOAssociation: EODelayedObserver (EOControl): Object
Implements:	NSDisposable EOObserving (EOControl)
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

`dispose`

EOObserving

Method Types

All methods

`EOWidgetAssociation`

`defaultPrefersContinuousChangeNotification`

`setDefaultPrefersContinuousChangeNotification`

`canSupportValueFormatter`

`isUsableWithObject`

`objectKeysTaken`

`prefersContinuousChangeNotification`

`setObject`

`setPrefersContinuousChangeNotification`

`setValueFormatter`

`valueFormatter`

`widgetPlugin`

`widgetPluginClass`

Constructors

EOWidgetAssociation

`public EOWidgetAssociation(Object anObject)`

Description forthcoming.

Static Methods

defaultPrefersContinuousChangeNotification

`public static boolean defaultPrefersContinuousChangeNotification()`

Description forthcoming.

setDefaultPrefersContinuousChangeNotification

`public static void setDefaultPrefersContinuousChangeNotification(boolean flag)`

Description forthcoming.

Instance Methods

canSupportValueFormatter

`public boolean canSupportValueFormatter()`

Description forthcoming.

dispose

```
public void dispose()
```

See the description in the documentation for NSDisposable.

isUsableWithObject

```
public boolean isUsableWithObject(Object anObject)
```

Description forthcoming.

objectKeysTaken

```
public NSArray objectKeysTaken()
```

Description forthcoming.

prefersContinuousChangeNotification

```
public boolean prefersContinuousChangeNotification()
```

Description forthcoming.

setObject

```
public void setObject(Object anObject)
```

Description forthcoming.

setPrefersContinuousChangeNotification

```
public void setPrefersContinuousChangeNotification(boolean flag)
```

Description forthcoming.

setValueFormatter

```
public void setValueFormatter(Object format)
```

Description forthcoming.

valueFormatter

```
public Object valueFormatter()
```

Description forthcoming.

widgetPlugin

```
public EOWidgetAssociation.WidgetPlugin widgetPlugin()
```

Description forthcoming.

widgetPluginClass

```
protected Class widgetPluginClass()
```

Description forthcoming.

EOWidgetAssociation. WidgetPlugin

Inherits from:	Object
Implements:	NSDisposable
Package:	com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Interfaces Implemented

NSDisposable

`dispose`

Method Types

All methods

`EOWidgetAssociation.WidgetPlugin`

`association`

`breakConnection`

`establishConnection`

`unacceptableAspects`

`widget`

`widgetKeysTaken`

Constructors

EOWidgetAssociation.WidgetPlugin

```
public EOWidgetAssociation.WidgetPlugin(
    EOWidgetAssociation anEOWidgetAssociation,
    Object widget)
```

Description forthcoming.

Instance Methods

association

```
public EOWidgetAssociation association()
```

Description forthcoming.

breakConnection

```
public void breakConnection()
```

Description forthcoming.

dispose

```
public void dispose()
```

See the description in the documentation for NSDisposable.

establishConnection

```
public void establishConnection()
```

Description forthcoming.

unacceptableAspects

```
public String[] unacceptableAspects()
```

Description forthcoming.

widget

```
public Object widget()
```

Description forthcoming.

widgetKeysTaken

```
public String[] widgetKeysTaken()
```

Description forthcoming.

C L A S S

EOWidgetPluginRegistry

Inherits from: Object

Package: com.webobjects.eointerface

Class Description

Documentation for this class is forthcoming.

Method Types

All methods

`EOWidgetPluginRegistry`

`findWidgetPluginClass`

`newWidgetPluginForAssociation`

`registerWidgetPluginClass`

`setWidgetSetPlugin`

`widgetSetPlugin`

Constructors

EOWidgetPluginRegistry

```
public EOWidgetPluginRegistry()
```

Description forthcoming.

Static Methods

findWidgetPluginClass

```
public static Class findWidgetPluginClass(
    Class associationClass,
    Class widgetClass)
```

Description forthcoming.

newWidgetPluginForAssociation

```
public static EOWidgetAssociation.WidgetPlugin newWidgetPluginForAssociation(
    EOWidgetAssociation association,
    Object widget)
```

Description forthcoming.

registerWidgetPluginClass

```
public static void registerWidgetPluginClass(
    Class associationClass,
    Class widgetClass,
    Class widgetPluginClass)
```

Description forthcoming.

setWidgetSetPlugin

```
public static void setWidgetSetPlugin(EOWidgetPluginRegistry.WidgetSetPlugin aWidgetSetPlugin)
```

Description forthcoming.

widgetSetPlugin

```
public static EOWidgetPluginRegistry.WidgetSetPlugin widgetSetPlugin()
```

Description forthcoming.

EODisplayGroup.Delegate

Package: com.webobjects.eointerface

Interface Description

Documentation for this interface is forthcoming.

Method Types

All methods

`displayGroupCreateObjectFailed`

`displayGroupDidChangeDataSource`

`displayGroupDidChangeSelectedObjects`

`displayGroupDidChangeSelection`

`displayGroupDidDeleteObject`

`displayGroupDidFetchObjects`

`displayGroupDidInsertObject`

`displayGroupDidSetValueForObject`

```
displayGroupDisplayArrayForObjects

displayGroupShouldChangeSelection

displayGroupShouldDeleteObject

displayGroupShouldDisplayAlert

displayGroupShouldFetch

displayGroupShouldInsertObject

displayGroupShouldRedisplay

displayGroupShouldRefetch
```

Instance Methods

displayGroupCreateObjectFailed

```
public abstract void displayGroupCreateObjectFailed(
    EODisplayGroup anEODisplayGroup,
    com.webobjects.eocontrol.EODataSource anEODataSource)
```

Description forthcoming.

displayGroupDidChangeDataSource

```
public abstract void displayGroupDidChangeDataSource(EODisplayGroup anEODisplayGroup)
```

Description forthcoming.

displayGroupDidChangeSelectedObjects

```
public abstract void displayGroupDidChangeSelectedObjects(EODisplayGroup anEODisplayGroup)
```

Description forthcoming.

displayGroupDidChangeSelection

```
public abstract void displayGroupDidChangeSelection(EODisplayGroup anEODisplayGroup)
```

Description forthcoming.

displayGroupDidDeleteObject

```
public abstract void displayGroupDidDeleteObject(
    EODisplayGroup anEODisplayGroup,
    Object anObject)
```

Description forthcoming.

displayGroupDidFetchObjects

```
public abstract void displayGroupDidFetchObjects(
    EODisplayGroup anEODisplayGroup,
    NSArray objects)
```

Description forthcoming.

displayGroupDidInsertObject

```
public abstract void displayGroupDidInsertObject(
    EODisplayGroup anEODisplayGroup,
    Object anObject)
```

Description forthcoming.

displayGroupDidSetValueForObject

```
public abstract void displayGroupDidSetValueForObject(
    EODisplayGroup anEODisplayGroup,
    Object value,
    Object anObject,
    String key)
```

Description forthcoming.

displayGroupDisplayArrayForObjects

```
public abstract NSArray displayGroupDisplayArrayForObjects(
    EODisplayGroup anEODisplayGroup,
    NSArray objects)
```

Description forthcoming.

displayGroupShouldChangeSelection

```
public abstract boolean displayGroupShouldChangeSelection(
    EODisplayGroup anEODisplayGroup,
    NSArray newIndexes)
```

Description forthcoming.

displayGroupShouldDeleteObject

```
public abstract boolean displayGroupShouldDeleteObject(
    EODisplayGroup anEODisplayGroup,
    Object anObject)
```

Description forthcoming.

displayGroupShouldDisplayAlert

```
public abstract boolean displayGroupShouldDisplayAlert(
    EODisplayGroup anEODisplayGroup,
    String title,
    String message)
```

Description forthcoming.

displayGroupShouldFetch

```
public abstract boolean displayGroupShouldFetch(EODisplayGroup anEODisplayGroup)
```

Description forthcoming.

displayGroupShouldInsertObject

```
public abstract boolean displayGroupShouldInsertObject(
    EODisplayGroup anEODisplayGroup,
    Object anObject,
    int index)
```

Description forthcoming.

displayGroupShouldRedisplay

```
public abstract boolean displayGroupShouldRedisplay(
    EODisplayGroup anEODisplayGroup,
    NSNotification aNSNotification)
```

Description forthcoming.

displayGroupShouldRefetch

```
public abstract boolean displayGroupShouldRefetch(
    EODisplayGroup anEODisplayGroup,
    NSNotification aNSNotification)
```

Description forthcoming.

EOWidgetAssociation.WidgetPlugin.Formatting

Package: com.webobjects.eointerface

Interface Description

Documentation for this interface is forthcoming.

Method Types

All methods

`setValueFormatter`

`valueFormatter`

Instance Methods

setValueFormatter

`public abstract void setValueFormatter(Object anObject)`

Description forthcoming.

valueFormatter

`public abstract Object valueFormatter()`

Description forthcoming.