

---

# EODetailSelectionAssociation

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

---

## Class at a Glance

### Purpose

An EODetailSelectionAssociation binds the selection in one EODisplayGroup (the destination) to a relationship in another (the source), so that the destination EODisplayGroup acts as an editor for that relationship. The selection in the destination EODisplayGroup reflects the value of the relationship for the object selected in the source EODisplayGroup. Changing the selection in the destination EODisplayGroup changes the value of the relationship in the source EODisplayGroup.

### Usable With

EODisplayGroup

### Aspects

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

### Object Keys Taken

None

---

## Class Description

An EODetailSelectionAssociation binds two EODisplayGroups together by a relationship, where the destination EODisplayGroup shows all possible values for the relationship, but 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.

## Example

Suppose that in your business model, an employee can be assigned any number of projects. A custom application shows this by having all employees listed in one table view, and all projects in another. When an employee is selected in the first table view, his assigned projects are selected in the other. To set up this kind of display, you use an EODetailSelectionAssociation between the Employee and Project EODisplayGroups, binding the **selectedObjects** aspect to the “projects” key. This causes the Projects table view to display all available projects and to highlight the ones assigned to the selected employee. The user can then add and remove projects from the selected employee by highlighting and unhighlighting them.