EODetailDataSource

Inherits From:	EODataSource : NSObject
Conforms To:	NSObject (NSObject)
Declared In:	EOControl/EODetailDataSource.h

Class Description

EODetailDataSource defines a data source for use in master-detail configurations, where operations in the detail data source are applied directly to properties of a master object. EODetailDataSource implements the standard **fetchObjects**, **insertObject:**, and **deleteObject:** methods to operate on a relationship property of its master object, so it works for any concrete subclass of EODataSource, including another EODetailDataSource (for a chain of three master and detail data sources).

To set up an EODetailDataSource programmatically, you typically create it by sending a **dataSourceQualifiedByKey:** message to the master data source, then establish the master object with a **qualifyWithRelationshipKey:ofObject:** message. The latter method records the name of a relationship for a particular object to resolve in **fetchObjects**, and to modify in **insertObject:**, and **deleteObject:**. These three methods then manipulate the relationship property of the master object to perform the operations requested. See the individual method descriptions for more information.

Method Types

Creating and qualifying instances	– initWithMasterDataSource:detailKey: – qualifyWithRelationshipKey:ofObject:
Examining instances	– masterDataSource – detailKey – masterObject
Getting the objects	– fetchObjects
Inserting and deleting objects	– insertObject: – deleteObject:

Instance Methods

deleteObject:

– (void)**deleteObject:**(id)*anObject*

Sends a **removeObject:fromBothSidesOfRelationshipWithKey:** message to the master object with *anObject* and the receiver's detail key as the arguments. Raises an NSInternalInconsistencyException if there's no master object or no detail key set.

See also: – **masterObject**, – **detailKey**

detailKey

– (NSString *)detailKey

Returns the name of the relationship for which the receiver provides objects, as set in either **initWithMasterDataSource:detailKey:** or **qualifyWithRelationshipKey:ofObject:**. If none has been set yet, returns **nil**.

See also: – **masterObject**

fetchObjects

– (NSArray *)**fetchObjects**

Sends **valueForKey:** to the master object with the receiver's detail key as the argument, constructs an NSArray for the returned object or objects, and returns it. Returns an empty array if there's no master object, or an array containing the master object itself if no detail key is set.

See also: – **masterObject**, – **detailKey**

initWithMasterDataSource:detailKey:

– (id)**initWithMasterDataSource:**(EODataSource *)*masterDataSource*
detailKey:(NSString *)*relationshipKey*

Initializes a newly allocated EODetailDataSource to provide objects based on a relationship of objects in *masterDataSource* named by *relationshipKey.* The receiver initially has no master object selected; to select one, use **qualifyWithRelationshipKey:ofObject:**.This is the designated initializer for the EODetailDataSource class. Returns **self**.

insertObject:

– (void)**insertObject:**(id)*object*

Sends an **addObject:toBothSidesOfRelationshipWithKey:** message to the master object with *anObject* and the receiver's detail key as the arguments. Raises an NSInternalInconsistencyException if there's no master object or no detail key set.

See also: – **masterObject**, – **detailKey**

masterDataSource

– (EODataSource *)**masterDataSource**

Returns the receiver's master data source.

See also: – **masterObject**, – **detailKey**

masterObject

– (id)masterObject

Returns the object in the master data source for which the receiver provides objects. You can change this with a **qualifyWithRelationshipKey:ofObject:** message.

See also: – **masterDataSource**, – **detailKey**

qualifyWithRelationshipKey:ofObject:

– (void)**qualifyWithRelationshipKey:**(NSString *)*relationshipKey* **ofObject:**(id)*masterObject*

Configures the receiver to provide objects based on the relationship of *masterObject* named by *relationshipKey*. *relationshipKey* can be different from the one used with **initWithMasterDataSource:detailKey:**, which changes the relationship for the receiver. If *masterObject* is **nil**, this method causes the receiver to return an empty array when sent a **fetchObjects** message.

See also: – **detailKey**, – **masterObject**