EOSortOrdering

Inherits From:	NSObject
Conforms To:	NSCoding NSObject (NSObject)
Declared In:	EOControl/EOSortOrdering.h

Class Description

An EOSortOrdering records the way that a group of objects should be sorted, using a property key and a method selector for comparing values of that property. EOSortOrderings are used both to generate SQL when fetching rows from a database server, and to sort objects in memory. Both the EOFetchSpecification class and the added NSArray sorting methods accept an array of EOSortOrderings, which are applied in series to perform sorts by more than one property.

Sorting with SQL

When an EOSortOrdering is used to fetch data from a relational database, it's rendered into a SQL ORDER BY clause for a SELECT statement. The Framework predefines symbols for four comparison selectors, rendered into SQL as:

Defined Name	SQL Expression
EOCompareAscending	(*key*) asc
EOCompareDescending	(*key*) desc
EOCompareCaseInsensitiveAscending	upper(*key*) asc
EOCompareCaseInsensitiveDescending	upper(*key*) desc

This sort ordering:

```
NSArray *nameOrdering = [NSArray arrayWithObjects:
    [EOSortOrdering sortOrderingWithKey:@"lastName" selector:EOCompareAscending],
    [EOSortOrdering sortOrderingWithKey:@"firstName" selector:EOCompareAscending],
    nil];
```

results in this ORDER BY clause:

```
order by (lastName) asc, (firstName) asc
```

In-Memory Sorting

Enterprise Objects Framework adds a method each to NSArray and NSMutableArray for sorting objects in memory. NSArray's **sortedArrayUsingKeyOrderArray:** returns a new NSArray sorted using an array of EOSortOrderings. Similarly, NSMutableArray's **sortUsingKeyOrderArray:** sorts the receiver's contents. This code fragment, for example, sorts an array of Employee objects by last name, then first name using the EOSortOrdering created above:

```
NSArray *sortedEmployees = [employees sortedArrayUsingKeyOrderArray:nameOrdering];
```

The predefined comparison selectors are:

Defined Name	Selector
EOCompareAscending	compareAscending:
EOCompareDescending	compareDescending:
EOCompareCaseInsensitiveAscending	compareCaseInsensitiveAscending:
EOCompareCaseInsensitiveDescending	compareCaseInsensitiveDescending:

The first two can be used with any value class; the second two with NSString objects only. The NSArray sorting methods extract property values using the key-value coding protocol and apply the selectors to the values. If you use custom value classes, you should be sure to implement the appropriate comparison methods to avoid exceptions when sorting objects.

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:

Method Types

Creating instances	+ sortOrderingWithKey:selector: – initWithKey:selector:
Examining a sort ordering	– key – selector

Class Methods

sortOrderingWithKey:selector:

+ (EOSortOrdering *)**sortOrderingWithKey:**(NSString *)*key* **selector:**(SEL)*aSelector*

Creates and returns an EOSortOrdering based on *key* and *selector.*

See also: – **initWithKey:selector:**

Instance Methods

initWithKey:selector:

– (id)**initWithKey:**(NSString *)*key* **selector:**(SEL)*aSelector*

Initializes a newly allocated EOSortOrdering based on *key* and *selector* and returns **self**. This is the designated initializer for the EOSortOrdering class.

See also: + **sortOrderingWithKey:selector:**

key

– (NSString *)**key**

Returns the key by which the receiver orders items.

See also: – **selector**

selector

– (SEL)**selector**

Returns the method selector used to compare values when sorting.

See also: – **key**