

---

# EOModelGroup

<b>Inherits From:</b>	NSObject
<b>Conforms To:</b>	NSObject (NSObject)
<b>Declared In:</b>	EOAccess/EOModelGroup.h

## Class Description

An EOModelGroup represents an aggregation of related models (see the EOModel class specification for more information on models). When a model in the group needs to resolve a foreign relationship to another model, it looks for that model in its group.

The default EOModelGroup contains all models for an application, as well as any frameworks the application references. The entity name space among all of these models is global; consequently, the same entity name shouldn't appear in any two of the models. All cross-model information is represented in the models by entity name only. Binding the entity name to an actual entity is done at run-time within the EOModelGroup.

In the majority of applications, the default EOModelGroup should be sufficient. However, if your particular application requires different model grouping semantics, you can create your own EOModelGroup instance, add the appropriate models, and then use that instance to replace the default EOModelGroup.

## Accessing Models Within a Model Group

Each model lives within a group and can form connections to other models in its group. A model can find a related model using the statement:

```
[[self modelGroup] modelNamed:name];
```

A data source can locate a model using the statement:

```
[[EOModelGroup defaultManager] modelNamed:name];
```

Putting models with identical names in separate groups allows you to load two models with the same name into EOModeler.

## Assigning EOModelGroup Delegates

Your EOModelGroup object should have a delegate which can influence how it finds and loads models. EOModelGroup's class object has its own delegate in addition to the delegate you assign to instances of the EOModelGroup class. The class delegate implements a single method—**defaultModelGroup**—while the

---

instance delegate can implement the remaining methods listed in the “Methods Implemented By the Delegate” section. Note that the following delegate methods are set on `EOModelGroup`, rather than `EOEntity`, to provide a single point in the code where you can alter the database-to-objects mapping:

```
entity:relationshipForRow:relationship:
subEntityForEntity:primaryKey:isFinal:
entity:failedToLookupClassNamed:
entity:classForObjectWithGlobalID:
```

## Method Types

Getting the models in a group	<ul style="list-style-type: none"><li>– <code>addModel:</code></li><li>– <code>modelName:</code></li><li>– <code>modelNames</code></li><li>– <code>models</code></li><li>– <code>modelWithPath:</code></li><li>– <code>removeModel:</code></li></ul>
Getting the default group	<ul style="list-style-type: none"><li>+ <code>defaultGroup</code></li><li>+ <code>setDefaultGroup:</code></li></ul>
Constructing a global model group	+ <code>globalModelGroup</code>
Searching a group for an entity	– <code>entityNamed:</code>
Searching a group for an object	– <code>entityForObject:</code>
Loading all of a group’s objects	– <code>loadAllModelObjects</code>
Assigning a delegate	<ul style="list-style-type: none"><li>+ <code>delegate</code></li><li>– <code>delegate</code></li><li>+ <code>setDelegate:</code></li><li>– <code>setDelegate:</code></li></ul>

## Class Methods

### **defaultGroup**

+ (EOModelGroup \*)**defaultGroup**

Returns the default `EOModelGroup`. Unless you’ve either specified a default model group with **setDefaultGroup:** or implemented the **defaultModelGroup** class delegate method to return a non-**nil** value, this method is equivalent to **globalModelGroup**.

**See also:** + `delegate`

---

## **delegate**

+ (id)**delegate**

Returns the EOModelGroup’s class delegate. This delegate optionally implements the **defaultModelGroup** method (see “Methods Implemented By the Delegate,” below, for more information).

**See also:** + **setDelegate**

## **globalModelGroup**

+ (EOModelGroup \*)**globalModelGroup**

Returns an EOModelGroup composed of all models in the resource directory of the main bundle, as well as all bundles and frameworks loaded into the application.

**See also:** + **defaultGroup**

## **setDefaultGroup:**

+ (void)**setDefaultGroup:**(EOModelGroup \*)*group*

Sets the default EOModelGroup normally returned by **defaultGroup** to *group*.

**See also:** + **setDelegate:**

## **setDelegate:**

+ (void)**setDelegate:**(id)*anObject*

Sets to *anObject* the EOModelGroup’s class delegate. The class delegate is optional; if allows you to implement the **defaultModelGroup** method (see “Methods Implemented By the Delegate,” below, for more information).

**See also:** + **delegate**, + **setDefaultGroup:**

## **Instance Methods**

### **addModel:**

– (void)**addModel:**(EOModel \*)*model*

Adds *model* to the receiver. If the receiver already contains an EOModel with the same name as *model*, the existing EOModel is released and *model* takes its place.

**See also:** – **models**, – **removeModel:**

---

## **delegate**

– (id)**delegate**

Returns the receiver's delegate.

**See also:** – **setDelegate:**

## **entityForObject:**

– (EOEntity \*)**entityForObject:(id)***object*

Returns the EOEntity associated with *object* from any of the models in the receiver that handle *object*, or **nil** if none of the entities in the receiver handles *object*.

## **entityNamed:**

– (EOEntity \*)**entityNamed:(NSString \*)***entityName*

Searches each of the EOModels in the receiver for the entity specified by *entityName*, and returns the entity if found. Returns **nil** if it is unable to find the specified entity.

## **loadAllModelObjects**

– (void)**loadAllModelObjects**

Sends **loadAllModelObjects** to each of the receiver's EOModels, thereby loading any EOEntities, EOAttributes, EORelationships, and EOStoredProcedures that haven't yet been loaded from each of the EOModels in the receiver.

**See also:** – **loadAllModelObjects (EOModel)**

## **modelName:**

– (EOModel \*)**modelName:(NSString \*)***modelName*

Returns the EOModel named *modelName* if it's part of the receiver, or **nil** if the receiver doesn't contain an EOModel with the specified name.

**See also:** – **modelName:**, – **models**

---

## **modelNameNames**

– (NSArray \*)**modelNameNames**

Returns an array containing the names of all of the EOModels in the receiver, or an empty array if the receiver contains no EOModels. The order of the model names in the array isn't defined.

**See also:** – **modelNameNamed:**, – **models**

## **modelWithPath:**

– (EOModel \*)**modelWithPath:**(NSString \*)*path*

If the receiver contains an EOModel whose path (as determined by sending **path** to the EOModel object) is equal to *path*, that EOModel is returned. Otherwise, returns **nil**. NSString's **isEqual:** method is used to compare the paths, and each path is standardized (with **stringByStandardizingPath**) before comparison.

**See also:** – **modelNameNamed:**, – **path** (EOModel)

## **models**

– (NSArray \*)**models**

Returns an array containing the receiver's EOModels, or an empty array if the receiver contains no EOModels. The order of the models in the array isn't defined.

**See also:** – **modelNameNamed:**, – **modelNameNames**

## **removeModel:**

– (void)**removeModel:**(EOModel \*)*aModel*

Removes *aModel* from the receiver, and unbinds any connections to *aModel* from other EOModels in the receiver. Posts an EOModelInvalidatedNotification to the default notification center after removing *aModel* from the receiver.

**See also:** – **addModel:**, – **models**

## **setDelegate:**

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

Sets the receiver's delegate to *anObject*.

**See also:** – **delegate**

---

## Methods Implemented By the Delegate

### **defaultModelGroup**

– (EOModelGroup \*)**defaultModelGroup**

If implemented by the class delegate, this method should return the EOModelGroup to be returned in response to the message **defaultModelGroup**. If this delegate method returns **nil**, EOModelGroup's **defaultModelGroup** class method will simply pass on the return value from EOModelGroup's **globalModelGroup** class method.

**Note:** This is the only method implemented by the class delegate.

**See also:** + **delegate**, + **setDelegate**:

### **modelGroup:entityNamed:**

– (EOModel \*)**modelGroup:**(EOModelGroup \*)*group* **entityNamed:**(NSString \*)*name*

If implemented by the delegate, this method should search the *group* for the entity named *name* and return the entity's EOModel. Return **nil** if *name* is not an entity in *group*.

### **entity:relationshipForRow:relationship:**

– (EORelationship \*)**entity:**(EOEntity \*)*entity* **relationshipForRow:**(NSDictionary \*)*row*  
**relationship:**(EORelationship \*)*relationship*

Invoked when relationships are instantiated for a newly fetched object. The delegate can use the information in *row* to determine which entity the target enterprise object should be associated with, and replace the relationship appropriately.

### **subEntityForEntity:primaryKey:isFinal:**

– (EOEntity \*)**subEntityForEntity:**(EOEntity \*)*entity* **primaryKey:**(NSDictionary \*)*primaryKeys* **isFinal:**(BOOL \*)*flag*

Allows the delegate to fine-tune inheritance by indicating from which sub-entity an object should be fetched based on its *primaryKey*. The entity returned must be a sub-entity of *entity*. If the delegate knows that the object should be fetched from the returned entity and not one of its sub-entities, it should set *flag* to YES.

---

### **entity:failedToLookupClassNameed:**

– (Class)**entity:**(EOEntity \*)*entity* **failedToLookupClassNameed:**(NSString \*)*className*

Invoked when the class name specified for *entity* cannot be found at run-time. The delegate can take action (such as loading a bundle) to provide *entity* with a class corresponding to *className*. If the delegate cannot provide anything, or if there is no delegate, EOGenericRecord is used.

### **entity:classForObjectWithGlobalID:**

– (Class)**entity:**(EOEntity \*)*entity* **classForObjectWithGlobalID:**(EOGlobalID \*)*globalID*

Used to fine-tune inheritance. The delegate can use *globalID* to determine a subclass to be used in place of the one specified in *entity*.

### **relationship:failedToLookupDestinationNamed:**

– (EOEntity \*)**relationship:**(EORelationship \*)*relationship*  
**failedToLookupDestinationNamed:**(NSString \*)*entityName*

Invoked when loading *relationship* and the destination *entityName* specified in the model file cannot be found in the model group. This most often occurs when a model references entities in another model file that can't be found. If the delegate doesn't implement this method, an exception is raised. If the delegate does implement this method, the method's return value is set as the destination entity. if the delegate returns **nil**, the destination entity is set to **nil**.

## **Notifications**

EOModelGroup declares and posts the following notification.

### **EOModelInvalidatedNotification**

<b>Notification Object</b>	The invalidated model.
----------------------------	------------------------

This notification is sent out by an EOModelGroup when an EOModel is removed from group. This notification is sent, for instance, inside Interface Builder when the user has saved changes to a model in EOModeler and the objects in Interface Builder must be brought back in sync. The old model is flushed and receivers of the notification (like data sources) can invoke **modelName:** to re-fetch their models.