
EODatabase

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	EOAccess/EODatabase.h

Class Description

An EODatabase represents a single database server. It contains an EOAdaptor which is capable of communicating with the server, a list of EOModels that describe the server's schema, a list of EODatabaseContexts that are connected to the server, and a set of snapshots representing the state of all objects stored in the server. An EODatabase's EODatabaseContexts automatically use it to record snapshots. There's little need for your code to interact directly with it.

Each of an EODatabase's EODatabaseContexts forms a separate transaction scope, and is in effect a separate logical user to the server. An EODatabaseContext uses one or more pairs of EODatabaseChannel and EOAdaptorChannel objects to manage data operations (insert, update, delete, and fetch). Adaptors may support a limited number of contexts per database or channels per context, but an application is guaranteed at least one of each.

The EODatabase, EODatabaseContext, and EODatabaseChannel classes form the *database level* of the Enterprise Objects Framework, which operates in terms of enterprise objects. The database level is a client of the *adaptor level*, which operates in terms of dictionary objects and which is defined by the adaptor classes: EOAdaptor, EOAdaptorContext, and EOAdaptorChannel. The objects of the database level create their own corresponding adaptor objects, so you don't have to use adaptor API when you're working at the database level. You do have access to the adaptor objects, however, should you need a feature available only in the adaptor level.

Snapshots

EODatabase's most significant responsibility is to record snapshots for its EODatabaseContexts. The snapshots made by an EODatabase form the global view of data for nearly every other part of the application, representing the current view of data in the server as far as the application is concerned (though other applications may have made changes). This global view is only overridden locally by EODatabaseContexts, which form their own snapshots as they make changes during a transaction. When an EODatabaseContext commits its top-level transaction, it reconciles all changed snapshots with the global view of the database object, so that other database contexts (except those with open transactions) immediately use the new snapshots as well.

Method Types

Creating instances	<ul style="list-style-type: none">– initWithModel:– initWithAdaptor:– adaptor
Adding and removing EOModels	<ul style="list-style-type: none">– addModel:– addModelIfCompatible:– removeModel:– models
Getting EOEntities	<ul style="list-style-type: none">– entityForObject:– entityNamed:
Recording snapshots	<ul style="list-style-type: none">– recordSnapshot:forGlobalID:– recordSnapshots:– forgetSnapshotForGlobalID:– forgetSnapshotsForGlobalIDs:– forgetAllSnapshots– snapshotForGlobalID:– snapshots
Registering EODatabaseContexts	<ul style="list-style-type: none">– registerContext:– unregisterContext:– registeredContexts
Managing the result cache	<ul style="list-style-type: none">– invalidateResultCache– invalidateResultCacheForEntityNamed:– resultCacheForEntityNamed:– setResultCache:forEntityNamed:

Class Methods

adaptor

– (EOAdaptor *)**adaptor**

Returns the EOAdaptor used by the EODatabase for communication with the database server. Your application can interact directly with the EOAdaptor, but should avoid altering its state (for example, by starting a transaction with one of its adaptor contexts).

addModel:

– (void)**addModel:**(EOModel *)*aModel*

Adds *aModel* to the receiver’s list of EOModels. This allows EODatabases to load entities and their properties only as they’re needed, by dividing them among separate EOModels. *aModel* must use the same EOAdaptor as the receiver and use the same connection dictionary as the receiver’s other EOModels.

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

addModelIfCompatible:

– (BOOL)**addModelIfCompatible:**(EOModel *)*aModel*

Adds *aModel* to the receiver’s list of EOModels, checking first to see whether it’s compatible with those other EOModels. Returns YES if *aModel* is already in the list or if it’s successfully added. Returns NO if *aModel*’s adaptor name differs from that of the receivers or if the receiver’s adaptor returns NO to a **canServiceModel:** message.

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

entityForObject:

– (EOEntity *)**entityForObject:**(id)*anObject*

Returns the EOEntity mapped to *anObject* in the receiver’s EOModels, or **nil** if there is no such EOEntity. This method works by sending **entityForObject:** messages to each of the receiver’s EOModels and returning the first one found.

See also: – **entityNamed:**

entityNamed:

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

Returns the EOEntity named *entityName* in the receiver’s EOModels, or **nil** if there is no such EOEntity. This method works by sending **entityNamed:** messages to each of the receiver’s EOModels and returning the first one found.

See also: – **entityForObject:**

forgetAllSnapshots

– (void)**forgetAllSnapshots**

Clears all of the receiver's snapshots.

See also: – **forgetSnapshotForGlobalID:**, – **forgetSnapshotsForGlobalIDs:**,
– **recordSnapshot:forGlobalID:**, – **recordSnapshots:**

forgetSnapshotForGlobalID:

– (void)**forgetSnapshotForGlobalID:**(EOGlobalID *)*globalID*

Clears the snapshot made for the enterprise object identified by *globalID*.

See also: – **forgetSnapshotsForGlobalIDs:**, – **forgetAllSnapshots**, – **recordSnapshot:forGlobalID:**

forgetSnapshotsForGlobalIDs:

– (void)**forgetSnapshotsForGlobalIDs:**(NSArray *)*globalIDs*

Clears the snapshots made for the enterprise objects identified by each of the EOGlobalIDs in *globalIDs*.

See also: – **forgetSnapshotForGlobalID:**, – **forgetAllSnapshots**, – **recordSnapshots:**

initWithAdaptor:

– **initWithAdaptor:**(EOAdaptor *)*anAdaptor*

Initializes a newly allocated EODatabase with *anAdaptor* as its adaptor and returns **self**. You should never associate more than one EODatabase with a given EOAdaptor. In general, use **initWithModel:**, which automatically selects the adaptor. This is the designated initializer for the EODatabase class.

initWithModel:

– **initWithModel:**(EOModel *)*aModel*

Initializes a newly allocated EODatabase by creating an instance of EOAdaptor named in *aModel* and invoking **initWithAdaptor:**. Returns **self**.

See also: + **adaptorWithModel:** (EOAdaptor), – **adaptorName** (EOModel)

invalidateResultCache

– (void)**invalidateResultCache**

Invalidates all of the arrays in the receiver’s result cache, regardless of the associated entities. See the method description for **setResultCache:forEntityNamed:** for more discussion of this topic.

See also: – **invalidateResultCacheForEntityNamed:**, – **resultCacheForEntityNamed:**

invalidateResultCacheForEntityNamed:

– (void)**invalidateResultCacheForEntityNamed:**(NSString *)*entityName*

Invalidates the result cache containing an array of globalIDs for the objects associated with the entity *entityName*. See the method description for **setResultCache:forEntityNamed:** for more discussion of this topic.

See also: – **invalidateResultCache**, – **resultCacheForEntityNamed:**

models

– (NSArray *)**models**

Returns the receiver’s EOModels.

See also: – **initWithModel:**, – **addModel:**, – **addModelIfCompatible:**, – **removeModel:**

recordSnapshot:forGlobalID:

– (void)**recordSnapshot:**(NSDictionary *)*aSnapshot* **forGlobalID:**(EOGlobalID *)*globalID*

Records *aSnapshot* under *globalID*.

See also: – **globalIDForRow:** (EOEntity), – **recordSnapshots:**, – **forgetSnapshotForGlobalID:**

recordSnapshots:

– (void)**recordSnapshots:**(NSDictionary *)*snapshots*

Records the snapshots in *snapshots*. *snapshots* is a dictionary whose keys are EOGlobalIDs and whose values are the snapshots for those global IDs.

See also: – **recordSnapshot:forGlobalID:**, – **forgetSnapshotsForGlobalIDs:**

registerContext:

– (void)**registerContext:**(EODatabaseContext *)*aContext*

Records *aContext* as one of the receiver’s EODatabaseContexts, without retaining it. *aContext* must have been created with the receiver using EODatabaseContext’s **initWithDatabase:** method, which invokes this method automatically. You should never need to invoke this method directly.

See also: – **unregisterContext:**, – **registeredContexts**

registeredContexts

– (NSArray *)**registeredContexts**

Returns all EODatabaseContexts that have been registered with the receiver.

– **registerContext:**, – **unregisterContext:**

removeModel:

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

Removes *aModel* from the receiver’s list of EOModels.

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

resultCacheForEntityNamed:

– (NSArray *)**resultCacheForEntityNamed:**(NSString *)*entityName*

Returns an array containing the globalIDs of the objects associated with *entityName*. See the method description for **setResultCache:forEntityNamed:** for more discussion of this topic.

See also: – **invalidateResultCache**, – **invalidateResultCacheForEntityNamed:**

setResultCache:forEntityNamed:

– (void)**setResultCache:**(NSArray *)*cache* **forEntityNamed:**(NSString *)*entityName*

Sets to *cache* the array of globalIDs for the objects associated with the entity *entityName*. This cache can be used by the receiver’s EODatabaseContexts to evaluate qualifiers in memory rather than in the database.

See also: – **invalidateResultCache**, – **invalidateResultCacheForEntityNamed:**,
– **resultCacheForEntityNamed:**

snapshotForGlobalID:

– (NSDictionary *)**snapshotForGlobalID:**(EOGlobalID *)*globalID*

Returns the snapshot associated with *globalID* if there is one; otherwise returns **nil**.

See also: – **recordSnapshot:forGlobalID:**, – **forgetSnapshotForGlobalID:**

snapshots

– (NSDictionary *)**snapshots**

Returns all of the receiver’s snapshots, stored in an NSDictionary under their EOGlobalIDs.

unregisterContext:

– (void)**unregisterContext:**(EODatabaseContext *)*aContext*

Removes *aContext* as one of the receiver’s EODatabaseContexts, without releasing it. EODatabaseContext automatically invokes this method when deallocated; you should never need to invoke it directly.

See also: – **registerContext:**, – **registeredContexts**