

---

# EODatabaseDataSource

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

## Class Description

EODatabaseDataSource is a concrete subclass of EODataSource that fetches objects based on an EOModel, using the EODatabaseContext of its EOEditingContext. An EODatabaseDataSource can be set up to fetch all objects for its root entity, to fetch objects matching a particular EOFetchSpecification, and to further filter its fetching with an auxiliary EOQualifier.

EODatabaseDataSource implements all the functionality defined by EODataSource. In addition to fetching objects, it can insert and delete them (provided the entity isn't read-only), can provide a detail data source, and can be set up in a master-peer configuration. See the EODataSource class specification for more information on these topics.

## Method Types

Creating instances	– initWithEditingContext:entityName:
Setting selection criteria	– setFetchSpecification: – fetchSpecification – setAuxiliaryQualifier: – auxiliaryQualifier – fetchSpecificationForFetch
Getting objects used for fetching	– entity – databaseContext
Enabling fetching	– setFetchEnabled: – isFetchEnabled

---

## Instance Methods

### **auxiliaryQualifier**

– (EOQualifier \*)**auxiliaryQualifier**

Returns the EOQualifier used to further filter the objects fetched by the receiver's EOFetchSpecification.

**See also:** – **fetchSpecificationForFetch**, – **fetchSpecification**, – **setAuxiliaryQualifier:**

### **databaseContext**

– (EODatabaseContext \*)**databaseContext**

Returns the EODatabaseContext that the receiver uses to access the external database. This is either the root EOObjectStore for the receiver's EOEditingContext, or if the root is an EOCooperatingObjectStore, it's the EODatabaseContext under that EOCooperatingObjectStore that services the EOModel containing the EOEntity for the receiver.

### **entity**

– (EOEntity \*)**entity**

Returns the EOEntity from which the receiver fetches objects.

### **fetchSpecification**

– (EOFetchSpecification \*)**fetchSpecification**

Returns the receiver's basic EOFetchSpecification. Its EOQualifier is conjoined with the receiver's auxiliary EOQualifier when the receiver fetches objects. The sender of this message can alter the EOFetchSpecification directly, or replace it using **setFetchSpecification:**.

**See also:** – **fetchSpecificationForFetch**, – **auxiliaryQualifier**

### **fetchSpecificationForFetch**

– (EOFetchSpecification \*)**fetchSpecificationForFetch**

Returns a copy of the EOFetchSpecification that the receiver uses to fetch. This is constructed by conjoining the EOQualifier of the receiver's EOFetchSpecification with its auxiliary EOQualifier. Modifying the returned EOFetchSpecification doesn't affect the receiver's fetching behavior; use **setFetchSpecification:** and **setAuxiliaryQualifier:** for that purpose.

**See also:** – **fetchSpecification**, – **auxiliaryQualifier**

---

### **initWithEditingContext:entityName:**

- (id)**initWithEditingContext:**(EOEditingContext \*)*anEditingContext*  
**entityName:**(NSString \*)*anEntityName*

Initializes a newly allocated EODatabaseDataSource to fetch objects into *anEditingContext* for the EOEntity named by *anEntityName*. This method checks *anEditingContext*'s EOObjectStoreCoordinator for an EODatabaseChannel that services the EOModel containing the named EOEntity. If none exists, this method creates one. This is the designated initializer for the EODatabaseDataSource class. Returns **self**.

### **isFetchEnabled**

- (BOOL)**isFetchEnabled**

Returns YES if the receiver's **fetchObjects** method actually fetches objects, NO if it returns an empty array without fetching. Fetching is typically disabled in a master-peer configuration when no object is selected in the master.

**See also:** – **setFetchEnabled:**

### **setAuxiliaryQualifier:**

- (void)**setAuxiliaryQualifier:**(EOQualifier \*)*aQualifier*

Sets the EOQualifier used to further filter the objects fetched by the receiver's EOFetchSpecification to *aQualifier*. This is useful for altering the basic EOFetchSpecification.

**See also:** – **fetchSpecificationForFetch**, – **fetchSpecification**, – **auxiliaryQualifier**

### **setFetchEnabled:**

- (void)**setFetchEnabled:**(BOOL)*flag*

Controls whether the receiver can fetch. If *flag* is YES the receiver's **fetchObjects** method actually fetches objects, if NO it returns an empty array without fetching. Fetching is typically disabled in a master-peer configuration when no object is selected in the master. For example, EODatabaseDataSource's implementation of **qualifyWithRelationshipKey:ofObject:** invokes this method to enable or disable fetching based on whether a master object is provided.

**See also:** – **setFetchEnabled:**

---

**setFetchSpecification:**

– (void)**setFetchSpecification:**(EOFetchSpecification \*)*aFetchSpecification*

Sets the receiver's basic EOfetchSpecification to *aFetchSpecification*. Its EOQualifier is conjoined with the receiver's auxiliary EOQualifier when the receiver fetches objects.

**See also:** – **setAuxiliaryQualifier:**, – **fetchSpecificationForFetch**, – **fetchSpecification**