

fetchObjects

Editing objects insertObject:

deleteObject:
updateObject:
saveObjects

Creating a new object createObject

Converting values to proper types

coerceValue:forKey:

Accessing attributes of the data source

canDelete
keys

(BOOL)canDelete

Returns YES if the data source allows objects to be deleted, NO if it doesn't.

coerceValue:aValue forKey:(NSString *)key

Returns aValue as an instance of a value class appropriate for the property named by key. For example, EODatabaseDataSource converts values to the value classes defined in an EOModel. Returns nil if aValue couldn't be coerced to the new type.

You should convert any value whose type isn't known to conform to those expected by an enterprise object before passing the value to the object. For example, the Interface Layer's EOController class uses its data source to convert strings received from user-interface objects. The code excerpt below shows how to convert a string representation of an NSDate for updating by a data-source object:

property values and for then inserting the newly created object into the data source. For example, if using a relational database as its external store, the sender must assign the newly created object a value for its primary key. `EODatabaseDataSource` and `EODetailDatabaseDataSource` initialize a newly created enterprise object with `initWithPrimaryKey:entity:` if the enterprise object responds to that method.

`insertObject:`

`(BOOL)deleteObject:anEO`

Deletes anEO from the data source. Returns YES on success, NO on failure.

`canDelete`

`(NSArray *)fetchObjects`

Returns the data-bearing objects provided by the data source, possibly including those that have been saved. The fetched objects are allocated from the data source's `zone`, if possible. If you need to specify which the fetched objects are to be allocated, retrieve them using `EODatabaseChannel's fetchWithZone:`

Note that if a data source has been qualified in some manner, it may provide fewer objects than it advertises. If an error occurs during the fetch, `fetchObjects` returns nil.

`(BOOL)insertObject:anEO`

Inserts anEO into the data source. Returns YES on success, NO on failure.

`(NSArray *)keys`

Returns the keys that name the properties of the data source's enterprise objects, for example, `^lastModifiedDate`, `^hireDate`, and so on. These are also the names of the class properties for the data source's entity. For more information from an `EOModel` (see the `EODatabaseDataSource` class specification).

`classPropertyNames (EOEntity)`

`(BOOL)saveObjects`

Saves modified objects to persistent storage (for example, an `EODatabaseDataSource` does this by committing a transaction). Returns YES on success, NO on failure.

`(BOOL)updateObject:anEO`

Updates anEO in the data source. Returns YES on success, NO on failure.