initWithName:
Setting the name setName:
name
+ isValidName:
Getting the model model
Getting the qualifier qualifier
Accessing attributes addAttribute:

primaryKeyAttributes
primaryKeyAttributeNames
isValidPrimaryKeyAttribute:

Getting primary keys primaryKeyForRow:

Setting class properties setClassProperties:
classProperties
classPropertyNames
isValidClassProperty:

Setting locking attributes setAttributesUsedForLocking:
attributesUsedForLocking
isValidAttributeUsedForLocking:

Setting the enterprise object class
setClassName:
className

Setting external information setExternalName:
externalName

Setting the external query setExternalQuery:
externalQuery

Setting read-only status setReadOnly:
isReadOnly

Setting the user dictionary setUserDictionary:
userDictionary

(BOOL)addAttribute:(EOAttribute *)anAttribute

Adds anAttribute to the entity and returns YES on success, NO if anAttribute's name is already in or relationship. Sets anAttribute's entity to self.

removeAttributeNamed:, attributeNamed:, setEntity: (EOAttribute)

(EOAttribute *)attributeNamed:(NSString *)name

Returns the attribute named name, or nil if no such attribute exists.

attributes, relationshipNamed:

(NSArray *)attributes

Returns the entity's attributes, or nil if the entity has none.

(NSArray *)attributesUsedForLocking

Returns the attributes whose values are compared when a database-level object performs an update the EOUpdateWithPessimisticLocking or EOUpdateWithOptimisticLocking update strategies are u database-level classes fetch an enterprise object, they cache these attributes' values in a snapshot. I enterprise object is updated, the values of these attributes in the object are checked with those in th differ, the update fails. See the EODatabaseContext class specification for more information.

(const char *)className

Returns the name of the enterprise object class associated with the entity. When a row is fetched fo EODatabaseChannel, it's returned as an instance of this class. This class might not be present in th in fact your application may have to load it on demand. If your application doesn't load a class, EO used.

An enterprise object class other than EOGenericRecord can be mapped to only one entity.

databaseChannel:failedToLookupClassNamed: (EODatabaseChannel, ªMethods Implemented by

(NSArray *)classProperties

Returns the properties bound to the entity's enterprise object class. An entity may have many prop properties are actually passed to enterprise objects (and therefore made visible in your application)

classPropertyNames

(NSArray *)classPropertyNames

Returns the names of all the properties returned by classProperties.

(NSString *)externalName

Returns the name of the entity as understood by the database server.

initWithName:(NSString *)name

Initializes a newly allocated EOEntity with name as its internal name. To be usable the EOEntity n
attributes and belong to a model. This is the designated initializer for the EOEntity class. Returns s

addAttribute:, addEntity: (EOModel)

(BOOL)isReadOnly

Returns YES if the entity can't be modified, NO if it can. If an entity can't be modified, then enter
that entity also can't be modified (inserted, deleted, or updated).

(BOOL)isValidAttributeUsedForLocking:(EOAttribute *)anAttribute

Returns NO if anAttribute isn't an EOAttribute, if the EOAttribute doesn't belong to the entity, or
derived. Otherwise returns YES. An attribute that isn't valid for locking will cause setAttributesUs

(BOOL)isValidClassProperty:aProperty

Returns NO if either aProperty isn't an EOAttribute or EORelationship, or if aProperty doesn't bel
Otherwise returns YES.

(BOOL)isValidPrimaryKeyAttribute:(EOAttribute *)anAttribute

Returns NO if anAttribute isn't an EOAttribute or if anAttribute isn't a simple attribute of the entit
or flattened). Otherwise returns YES.

(EOModel *)model

Returns the model that contains the entity.

addEntity: (EOModel)

(NSString *)name

Returns the entity's name.

(NSArray *)primaryKeyAttributes

(NSDictionary *)primaryKeyForRow:(NSDictionary *)aRow

Returns the primary key for aRow, or nil if the primary key can't be computed. The primary key is whose keys are attribute names and whose values are values for those attributes (note that "key" is here).

(EOQualifier *)qualifier

Returns a qualifier for the entity that can be used in EOAdaptorChannel's selectAttributes:describe fetchOrder:lock: method or EODatabaseChannel's selectObjectsDescribedByQualifier:fetchOrder: qualifier will select all rows for the entity.

(BOOL)referencesProperty:aProperty

Returns YES if any of the entity's attributes or relationships reference aProperty, NO otherwise. A referenced by a flattened attribute or by a relationship. For example, suppose a model has an

removeAttributeNamed:, removeRelationshipNamed:

(EORelationship *)relationshipNamed:(NSString *)name

Returns the relationship named name, or nil if the entity has no such relationship.

relationships, attributeNamed:

(NSArray *)relationships

Returns the entity's relationships, or nil if the entity has none.

(void)removeAttributeNamed:(NSString *)name

Removes the attribute named name if it exists. You should always use referencesProperty: to chec referenced by another property before removing it.

(void)removeRelationshipNamed:(NSString *)name

Removes the relationship named name if it exists. You should always use referencesProperty: to c isn't referenced by another property before removing it.

Sets the name of the class associated with the entity to name. This class need not be present in the
this message is sent. When an EODatabaseChannel fetches objects for the entity, they're created as
Your application may have to load the class on demand if it isn't present in the run-time system if
EOGenericRecord will be used.

databaseChannel:failedToLookupClassNamed: (EODatabaseChannel, ªMethods Implemented by

(BOOL)setClassProperties:(NSArray *)properties

Sets the entity's class properties to the EOAttributes and EORelationships in properties. Returns Y
the entity responds NO to isValidClassProperty: for any of the objects in the array.

(void)setExternalName:(NSString *)name

Sets the name of the entity used with the database server to name. For example, though your applic
entity as ªJobTitleº the database may require a form such as ªJOB_TTLº. An adaptor uses the exte
communicate with the database your application should never need to use the external name.

(void)setExternalQuery:(NSString *)aQuery

Sets aQuery as the SQL SELECT statement that an EOAdaptorChannel uses to select rows for the
is unrestricted. An unrestricted qualifier is one that specifies only the entity, and would thus fetch a
that entity.

External queries are useful for hiding records or invoking database-specific features such as stored
application attempts to select all records for an entity.

An external query is sent unaltered to the database server, and so must contain the external (colum
names of EOAttributes. However, to work properly with the adaptor the external query must use th
alphabetical order by their corresponding EOAttributes' names.

isEmpty (EOQualifier), selectAttributes:describedByQualifier:fetchOrder:lock: (EOAdaptorChan

(BOOL)setName:(NSString *)name

Sets the entity's name to name and returns YES on success, NO if name is in use by another entity

(BOOL)setPrimaryKeyAttributes:(NSArray *)keys

Sets the primary key attributes to the attributes in keys. Returns YES on success, or NO if the entit
isValidPrimaryKeyAttribute: for any of the objects in the array.

(void)setUserDictionary:(NSDictionary *)aDictionary

Sets the dictionary of auxiliary data, which your application can use for whatever it needs. The dic
property list (that is, it must contain only NSString, NSData, NSArray, and NSString objects).

(NSDictionary *)userDictionary

Returns a dictionary of user data. Your application can use this to store any auxiliary information i
is a property list (that is, it contains only NSString, NSData, NSArray, and NSString objects).