

keys

coerceValue:forKey:  
fetchObjects  
saveObjects  
createObject  
insertObject:  
canDelete  
deleteObject:  
updateObject:

EOMasterDataSources dataSourceQualifiedByKey:  
EOQualifiableDataSources qualifyWithRelationshipKey:ofObject:  
EORollbackDataSources rollback

initWithMasterDataSource:entity:  
Getting the master data source masterDataSource  
Getting the entity entity  
Getting the detail key detailKey  
Getting the objects masterObject  
array

(NSArray \*)array  
Returns the array of objects supplied by the receiver. This array contains the destination objects corresponding to the relationship key in the master enterprise object. An EODetailDatabaseDataSource's array is established when it's qualified with qualifyWithRelationshipKey:ofObject:.

coerceValue:aValue forKey:(NSString \*)key

Returns aValue as an instance of the value class defined for the attribute named key of the receiver. If aValue couldn't be coerced to the new type.

For example, suppose the

entity, valueClassName (EOAttribute)

createObject

Returns a new enterprise object with no values set, allocated from the data source's zone and initialized with the receiver's entity, or nil if the receiver's EOEntity is read-only. The sender of this message is responsible for setting the object's property values (especially its primary key) and for then inserting the newly created object into the data source.

If the new enterprise object responds to initWithPrimaryKey:entity: it's initialized with that method's primary key and with the data source's entity. If the enterprise object doesn't respond to initWithPrimaryKey:entity:, it's initialized with init.

(id <EOQualifiableDataSources>)dataSourceQualifiedByKey:(NSString \*)relName

Returns a new EODetailDatabaseDataSource that provides objects for the destination of the relationship named relName of the receiver's entity. Before using the returned detail data source you have to assign it a master data source with qualifyWithRelationshipKey:ofObject:.

Returns nil if the receiver can't supply a detail data source for relName (typically because its entity is read-only or that name).

entity

(BOOL)deleteObject:anEO

Sends deleteObject: to the receiver's master data source with anEO as the argument. Returns YES if successful, NO on failure, if the receiver's entity is read-only, if its relationship is to-one and the property in the master data source is not a valid object, or if it has no objects. For to-one relationships, deleteObject: sets the property in the master data source to nil. isReadOnly (EOEntity), canDelete

(NSString \*)detailKey

Returns the name of the relationship in the master object's entity. The EODetailDatabaseDataSource uses this key to get destination objects for this relationship. The detail key is established when the receiver is sent a qualifyWithRelationshipKey:ofObject: message.

The objects supplied by the master data source must have a class property matching the detail key, which the master data source uses to get the destination objects for its relationship.

masterObject

(NSArray \*)fetchObjects

Returns the array of objects supplied by the receiver, or nil if an error occurred. This array contains objects corresponding to the relationship key in the master enterprise object. EODetailDatabaseDataSource objects get their objects from their master object by using the key-value coding method valueForKey: with the relationship key, masterObject, detailKey

initWithMasterDataSource:(EODatabaseDataSource \*)masterDataSource  
entity:(EOEntity \*)anEntity

Initializes a newly allocated EODetailDatabaseDataSource with the given information. This is the designated initializer for the EODetailDatabaseDataSource class. Returns self.

Instances of this class are normally created by sending dataSourceQualifiedByKey: to an EODatabaseDataSource existing EODetailDatabaseDataSource. You should never allocate and initialize an EODetailDatabaseDataSource directly.

(BOOL)insertObject:anEO

Sends insertObject: to the receiver's master data source with anEO as the argument. Returns YES if successful, failure, if the receiver's entity is read-only, if its relationship is to-one and the property in the master is EONull, or if it has no objects.

isReadOnly (EOEntity)

(EODatabaseDataSource \*)masterDataSource

Returns the EODatabaseDataSource at the root of a series of EODetailDatabaseDataSources. Note that "master" in this method name differs from that in masterObject, which returns an object in the immediate data source.

masterObject

Returns the source object for the receiver's relationship. The objects provided by the receiver are the objects of this object's relationship. The master object is established by the qualifyWithRelationshipKey:ofObject: method with the relationship key, detailKey, fetchObjects

(void)qualifyWithRelationshipKey:(NSString \*)relName  
ofObject:anEO

(void)rollback

Sends rollback to the receiver's master data source. Returns YES on success, NO on failure.

(BOOL)saveObjects

If the receiver has any objects to save, updates its master object with the new array of objects for it. saveObjects to the receiver's master data source. Returns YES on success, NO on failure.

array

(BOOL)updateObject:anEO

Sends updateObject: to the receiver's master data source with anEO as the argument. Returns YES on success, NO on failure, if the receiver's entity is read-only, or if the receiver has no objects.

isReadOnly (EOEntity), array