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

EOMasterDataSources dataSourceQualifiedByKey:

EOQualifiableDataSources qualifyWithRelationshipKey:ofObject:

EORollbackDataSources rollback

initWithDatabaseChannel:entityNamed:
initWithModelName:entityName:
initWithModelName:entityName:databaseName:
contextName:channelName:

Naming a database channel+ registerChannel:forRendezvousWithDatabaseName:
contextName:channelName:

Getting the database channel databaseChannel
+ databaseChannelWithDatabaseName:contextName:
channelName:

Releasing a database channel+ releaseObjectsWithDatabaseName:contextName:
channelName:

Getting the root entity entity

isFetchEnabled

Setting the fetch order setFetchOrder:
fetchOrder

(EOQualifier *)auxiliaryQualifier

Returns the auxiliary qualifier used with the primary qualifier when fetching objects. This qualifier
conditions to the primary qualifier and is useful for narrowing the scope of a data source without a
qualifier. This is especially useful for setting a qualifier on a qualified peer data source, since the m
primary qualifier to specify the relationship it fetches for.

auxiliaryQualifier, qualifierForFetch, qualifier (EOEntity)

(BOOL)beginsTransactionsAutomatically

Returns YES if the data source automatically starts a transaction if needed upon the first insert, del
NO if it doesn't. If this method returns NO you must control transactions explicitly through the dat
EODatabaseContext. EODatabaseDataSources by default do begin transactions automatically.

Returns aValue as an instance of the value class defined for the attribute named key of the receiver
aValue couldn't be coerced to the new type. Returns aValue passed in if the key does not correspo

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 initia below, or nil if the receiver's EOEntity is read-only. The sender of this message is responsible for object's property values (especially its primary key) and for then inserting the newly created objec

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

(EODatabaseChannel *)databaseChannel

Returns the EODatabaseChannel that the data source uses to communicate with the database serve channel to control transactions if you've disabled automatic transactions with the setBeginsTransa method, but should be careful about altering other state, as it can cause errors in the database data s

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

Returns an EODetailDatabaseDataSource object that provides objects for the destination of the rel relName of the receiver's entity. Before using the detail data source you have to give it a master ob qualifyWithRelationshipKey:ofObject:.

Returns nil if the entity has no relationship named relName or if that relationship has no destinatio

entity

(BOOL)deleteObject:anEO

Sends deleteObject: to the receiver's database channel with anEO as the argument. Returns YES o failure.

canDelete (EODataSources protocol)

(EOEntity *)entity

Returns the entity that the EODatabaseDataSource fetches objects for. This is the entity of the EO qualifier.

isFetchEnabled, qualifierForFetch, fetchOrder

(NSArray *)fetchOrder

Returns the fetch order used when supplying objects. See the EOAttributeOrdering class specificat information.

initWithDatabaseChannel:(EODatabaseChannel *)aChannel
entityNamed:(NSString *)entityName

Initializes a newly allocated EODatabaseDataSource with aChannel as its communication channel with the entity in aChannel's model named entityName as its root entity. Doesn't use the rendezvo initWithModelName:entityName:databaseName:contextName:channelName:. This is the designat EODatabaseDataSource class. Returns self.

initWithModelName:(NSString *)modelName
entityName:(NSString *)entityName

Initializes a newly allocated EODatabaseDataSource with the model named modelName and with entityName as its root entity. Invoking this method is equivalent to invoking initWithModelName: databaseName:contextName:channelName: with the model's name, the entity's name, and with ni component rendezvous names. Returns self.

initWithModelName:(NSString *)modelName
entityName:(NSString *)entityName
databaseName:(NSString *)databaseName
contextName:(NSString *)contextName
channelName:(NSString *)channelName

Initializes a newly allocated EODatabaseDataSource with the model named modelName and with entityName as its root entity. Returns self.

This method uses the last three name arguments to share database-level objects with other instance class description under ªRendezvous on Database-level objects.º

(BOOL)insertObject:anEO

Sends insertObject: to the receiver's database channel with anEO as the argument. anEO must hav primary key. Returns YES on success, NO on failure.

Returns the primary qualifier used when fetching objects. This qualifier minimally specifies the EO
entity, and may contain more restrictive information. It's set by the qualifyWithRelationshipKey:o
declared in the EOQualifiableDataSources protocol.

auxiliaryQualifier, qualifierForFetch, qualifier (EOEntity)

(EOQualifier *)qualifierForFetch

Returns the qualifier used when fetching objects. This qualifier is the conjunction of the primary q
auxiliary qualifier.

auxiliaryQualifier, qualifier, conjoinWithQualifier: (EOQualifier)

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

Qualifies the receiver so that it supplies objects based on relName, which is the name of an EORel
EOEntity for anEO. This results in the receiver's entity and primary qualifier being changed.

If anEO is nil, the detail data source disables fetching by invoking setFetchEnabled: this is useful f
offº a qualified peer data source when no records are selected in its master.

(void)rollback

If the receiver automatically started the current transaction, sends rollbackTransaction to the databa
receiver's database channel. Doesn't affect the data source's enterprise objects. Returns YES on su

(BOOL)saveObjects

If the receiver automatically started the current transaction, sends commitTransaction to the databa
receiver's database channel. Returns YES on success, NO on failure.

(void)setAuxiliaryQualifier:(EOQualifier *)aQualifier

Sets to aQualifier the auxiliary qualifier used with the primary qualifier when fetching objects. Thi
conditions to the primary qualifier and is useful for narrowing the scope of a data source without a
qualifier.

Raises NSInvalidArgumentException if aQualifier's entity isn't the same as the primary qualifier's

auxiliaryQualifier, qualifierForFetch

(void)setFetchEnabled:(BOOL)flag

Enables fetching according to flag. If flag is NO, then fetchObjects returns no objects until fetching

(void)setFetchOrder:(NSArray *)fetchOrder

Sets to fetchOrder the fetch order used for retrieving objects from the database. See the EOAttribu specification for more information.

(void)setQualifier:(EOQualifier *)aQualifier

Sets to aQualifier the primary qualifier used with the auxiliary qualifier when fetching objects. Thi specifies the EODatabaseDataSource's entity, and may contain more restrictive information (thoug the auxiliary qualifier). You must supply a valid qualifier to this method if aQualifier is nil, subseq retrive objects from the database.

This method is invoked by qualifyWithRelationshipKey:ofObject: when an EODatabaseDataSourc qualified peer.

setAuxiliaryQualifier:, qualifierForFetch, qualifier (EOEntity)

(BOOL)updateObject:anEO

Sends updateObject: to the receiver's database channel with anEO as the argument. Returns YES o failure.