initWithDatabaseContext:

Opening and closing a channel openChannel

refetchObject:
cancelFetch
setCurrentEntity:

Getting the adaptor channel adaptorChannel

Getting the database context databaseContext

Setting the delegate setDelegate:
delegate

(EOAdaptorChannel *)adaptorChannel

Returns the EOAdaptorChannel used by the EODatabaseChannel for communication with the data
ªWorking with the Adaptor Channelº in the class description for information on safely using an EO
adaptor channel.

(void)cancelFetch

Cancels any fetch in progress.

isFetchInProgress, selectObjectsDescribedByQualifier:fetchOrder:, fetchWithZone:

(void)closeChannel

Closes the EODatabaseChannel so that it can't perform operations with the server. Any fetch in pr
the channel's database context has outstanding transactions, however, the result depends on the se
servers immediately roll back all outstanding transactions while others wait until the user logs out

openChannel, nestedTransactions (EODatabaseContext)

(EODatabaseContext *)databaseContext

Returns the EODatabaseContext that controls transactions for the EODatabaseChannel.

delegate

Returns the EODatabaseChannel's delegate.

(BOOL)deleteObject:anEO

Deletes the row in the database server corresponding to anEO. Returns YES if successful, NO if no
deletion may fail are:

·anEO is nil.

fetchWithZone:(NSZone *)zone

Fetches and returns the next object in the result set produced by a selectObjectsDescribedByQualif
message returns nil if there are no more objects in the current result set or if an error occurs. If the
EODatabase performs uniquing and the object fetched already exists, it's simply sent the new valu
channel creates a new enterprise object.

To create a new enterprise object, the database channel allocates an instance of the enterprise objec
being fetched. The channel then initializes the object with initWithPrimaryKey:entity: if it implem
otherwise simply with an init message. To set the object's values the channel sends it a takeValues
message with the fetched row as the argument, and finishes off with an awakeForDatabaseChannel
the EODatabaseChannelNotification informal protocol specification).

If an enterprise object matching the next selected object has already been uniqued by the EODatab
EODatabase, that object is sent takeValuesFromDictionary: and awakeForDatabaseChannel: mess
values with those just fetched. If the uniqued object's values differ from those fetched, an ambigui
description of the databaseChannel:willRefetchConflictingObject:withSnapshot: delegate message
how such a situation is handled.

This method invokes the delegate methods databaseChannel:willFetchObjectOfClass:withZone: an
didFetchObject:, and may invoke either databaseChannel:willRefetchObject:fromSnapshot: or data
willRefetchConflictingObject:withSnapshot:.

objectForPrimaryKey:entity: (EODatabase and EODatabaseContext), takeValuesFromDictionary
informal protocol)

initWithDatabaseContext:(EODatabaseContext *)aDatabaseContext

Initializes a newly allocated EODatabaseChannel with aDatabaseContext as the EODatabaseConte
automatically open the channel. The new EODatabaseChannel retains aDatabaseContext. This is th
for the EODatabaseChannel class. Returns self, or nil if no more channels can be associated with a

openChannel

(BOOL)insertObject:anEO

Attempts to insert anEO's simple (nonflattened, nonderived) attribute values as a row into the data
YES if successful and NO if not. If anEO is successfully inserted, the receiver's database context u
snapshot with recordObject:primaryKey:snapshot:. Raises NSInvalidArgumentException if anEO
EOFault class specification for information on faults).

Some of the reasons insertion may fail are:

·anEO is nil.
·The receiver's EODatabaseContext has no transaction in progress.
·anEO's EOEntity has a read-only attribute that the server requires be set.
·A primary key can't be determined for anEO.
·The database doesn't allow insertion.
·The receiver's delegate disallows insertion.

Note that read-only attributes are silently unmodified, but if the server requires such an attribute to
will occur.

This method invokes the delegate methods databaseChannel:willInsertObject: and databaseChanne

(BOOL)isOpen

Returns YES if the channel has been successfully opened with openChannel, NO if not.

(BOOL)lockObject:anEO

Locks anEO for update. Fetches the properties composing anEO's snapshot and compares the fetch
values stored in the snapshot. Returns YES if all values are the same fails and returns NO if any of
returns NO under the following conditions:

·anEO's entity is read-only.
·The channel's database context has no transaction in progress.
·The database server or its adaptor doesn't support locking.

This method invokes the delegate methods databaseChannel:willLockObject: and databaseChanne

isReadOnly (EOEntity)

(BOOL)openChannel

Puts the channel and both its context and database into a state where they are ready to perform ope
on success and NO on failure. You shouldn't attempt to open an already open channel.

closeChannel

(BOOL)refetchObject:anEO

Refetches the object for anEO's primary key from the database server. anEO is modified by this m
success, NO on failure for any reason (specifically, if there's a fetch in progress). May also raise
NSInternalInconsistencyException under the conditions described for the delegate method databas
willRefetchConflictingObject:withSnapshot:.

Some possible reasons for failure are:

·anEO is nil.
·No snapshot for anEO exists.
·The receiver's EODatabaseContext has no transaction in progress.
·The database channel has a fetch in progress.

This method invokes the delegate methods databaseChannel:willRefetchObject: and databaseChan
and may invoke databaseChannel:willRefetchConflictingObject:withSnapshot:.

isFetchInProgress, takeValuesFromDictionary: (EOKeyValueCoding informal protocol)

(BOOL)selectObjectsDescribedByQualifier:(EOQualifier *)aQualifier
fetchOrder:(NSArray *)fetchOrder

databaseChannel:didSelectObjectsDescribedByQualifier:fetchOrder:.

fetchWithZone:

(void)setCurrentEntity:(EOEntity *)anEntity

Sets the entity used when fetching enterprise objects. Subsequent fetchWithZone: messages during
create an object of the class associated with anEntity. If you perform a select operation in the datab
database channel's adaptor channel, you should set the proper entity before having the EODatabas
the selected rows. See ªWorking with the Adaptor Channelº in the class description for an exampl

This method is invoked automatically when you use selectObjectsDescribedByQualifier:fetchOrd

(void)setDelegate:anObject

Sets the EODatabaseChannel's delegate to anObject.

(BOOL)updateObject:anEO

Updates the row in the database corresponding to anEO. Returns YES if successful, NO if not. So
failure are:

·anEO is nil.
·The receiver's EODatabaseContext has no transaction in progress.
·A primary key can't be determined for anEO (because no snapshot was recorded).
·No record for anEO's primary key exists in the database.
·anEO's EOEntity is read-only.
·anEO's EOEntity has a read-only attribute that the server requires be set.
·The update strategy of the receiver's EODatabaseContext doesn't permit the update.
·The receiver's delegate disallows the update.

Note that read-only attributes are silently unmodified, but if the server requires such an attribute to
will occur.

This method invokes the delegate methods databaseChannel:willUpdateObject: and databaseChan

isReadOnly (EOEntity)

(void)databaseChannel:channel didDeleteObject:anEO

Invoked after channel has deleted anEO from the database.

(void)databaseChannel:channel didFetchObject:anEO

Invoked after channel has fetched anEO from the database.

(void)databaseChannel:channel didSelectObjectsDescribedByQualifier:(EOQualifier *)aQuali
    (NSArray *)fetchOrder

Invoked after channel has selected objects described by aQualifier with fetchOrder.

(void)databaseChannel:channel didUpdateObject:anEO

Invoked after channel has updated anEO in the database.

(Class)databaseChannel:channel failedToLookupClassNamed:(const char *)name

Invoked when channel has failed to find a class with the given name in the run-time system. The d
external module if needed and return the named class. If the delegate returns nil or doesn't implem
EOGenericRecord is used.

className (EOEntity)

(EORelationship *)databaseChannel:channel
    relationshipForRow:(NSDictionary *)aRow
    relationship:(EORelationship *)aRelationship

Invoked when a relationship is instantiated on a newly fetched object. aRow contains the values fo
fetched, and aRelationship is the relationship about to be set up. The delegate can use the informat
determine which entity the destination of the relationship should be associated with, and return a su
aRelationship as needed.

This method allows you to reuse the same column in a table for different kinds of relationships bas
the object. For example, suppose you have one table containing two kinds of

(void)databaseChannel:channel
willFetchObjectOfClass:(Class)class
withZone:(NSZone *)zone

Invoked before channel fetches an object.

databaseChannel:channel willInsertObject:anEO

Invoked before channel inserts anEO. The delegate may return a substitute object to insert or nil to

(BOOL)databaseChannel:channel willLockObject:anEO

Invoked before channel locks anEO. The delegate may return YES to allow anEO to be locked or N

(NSDictionary *)databaseChannel:channel
willRefetchConflictingObject:anEO
withShapshot:(NSDictionary *)snapshot

Invoked when channel has fetched new values for anEO, but anEO's current values differ from its
happens when anEO is updated during a transaction that hasn't been committed. In such a situation
potential states, and the database channel can't determine which is valid this usually indicates a pro
requires intervention by the delegate. See ªNotifying the Database Channel's Delegateº in the clas
information.

If the delegate doesn't implement this method the database channel raises NSInternalInconsistency

(NSDictionary *)databaseChannel:channel
willRefetchObject:anEO
fromSnapshot:(NSDictionary *)aSnapshot

Invoked before channel refreshes anEO with the values in aSnapshot just fetched from the database
return aSnapshot, a substitute snapshot, or nil to disallow the operation.

(BOOL)databaseChannel:channel willSelectObjectsDescribedByQualifier:(EOQualifier *)aQu
(NSArray *)fetchOrder

Invoked before channel performs a select operation. The delegate shouldn't modify aQualifier or fe
delegate returns YES the select proceeds if the delegate returns NO the select operation is aborted.

databaseChannel:channel willUpdateObject:anEO