initWithAdaptor:

initWithModel:

Getting the adaptor adaptor

Getting the database contexts contexts

Checking connection status hasOpenChannels

Uniquing/snapshotting setUniquesObjects:

uniquesObjects

setKeepsSnapshots:

keepsSnapshots

forgetAllObjects:

uniquesObjects, keepsSnapshots

(EOAdaptor *)adaptor

Returns the EOAdaptor used by the EODatabase for communication with the database server. You interact directly with the EOAdaptor, but should avoid altering its state (for example, by starting a its adaptor contexts).

(NSArray *)contexts

Returns the EODatabaseContexts managing transaction scopes for the EODatabase.

(void)forgetAllObjects

If the EODatabase uniques objects, removes all objects from the receiver's uniquing tables. If the E snapshots, also destroys all snapshots.

uniquesObjects, keepsSnapshots

(void)forgetObject:anEO

If the EODatabase uniques objects, removes anEO from the receiver's uniquing tables, and if the E snapshots also destroys the snapshot recorded under anEO's primary key. An EODatabase invokes an enterprise object uniqued by it is deallocated.

uniquesObjects, keepsSnapshots

(BOOL)hasOpenChannels

Returns YES if the database's adaptor has any open EOAdaptorChannels, NO otherwise.

hasOpenChannels (EOAdaptor), openChannel (EODatabaseChannel)

initWithAdaptor:(EOAdaptor *)anAdaptor

Initializes a newly allocated EODatabase with anAdaptor as its adaptor and returns self. You shoul than one EODatabase with a given adaptor. In general, use initWithModel:, which automatically se is the designated initializer for the EODatabase class.

(BOOL)keepsSnapshots

Returns YES if the EODatabase keeps snapshots for all enterprise objects fetched by its database c
default behavior is to keep snapshots.

If an EODatabase doesn't keep snapshots then none of the uniquing-snapshotting methods affect s
example, recordObject:primaryKey:snapshot: won't record a snapshot (though it may record the ol
EODatabaseContext class specification for information on how snapshots are used.

objectForPrimaryKey:(NSDictionary *)aKey entity:(EOEntity *)anEntity

Returns the unique enterprise object for aKey and anEntity, or nil if one can't be found.

primaryKeyForRow: (EOEntity), uniquesObjects

(void)recordObject:anEO
primaryKey:(NSDictionary *)aKey
snapshot:(NSDictionary *)aSnapshot

If the EODatabase uniques objects, records anEO under aKey so that it becomes the sole instance r
record with the same key is fetched later, its data is applied to the unique instance, which is then re
Raises NSInvalidArgumentException when uniquing if anEO or aKey is nil.

If the EODatabase keeps snapshots, also records aSnapshot under aKey. Raises NSInvalidArgume
making a snapshot if aSnapshot is nil and anEO isn't a fault (see the EOFault class specification fo
.

There may already be a unique instance recorded for anEO's key, so you shouldn't expect that the
this method can be shared. To get the unique enterprise object created for anEO, use objectForPrim

primaryKeyForRow: (EOEntity), uniquesObjects, keepsSnapshots

(void)setKeepsSnapshots:(BOOL)flag

Sets according to flag whether the EODatabase keeps snapshots for fetched enterprise objects. If fl
any existing snapshots. Raises NSInvalidArgumentException if the EODatabase has any open cha

An EODatabase must be set to keep snapshots to allow updates across transactions. See the EODat
specification for information on snapshots and updating.

hasOpenChannels

(void)setUniquesObjects:(BOOL)flag

Sets according to flag whether the EODatabase uniques fetched enterprise objects. If flag is NO, al
enterprise objects. Raises NSInvalidArgumentException if the EODatabase has any open channels

hasOpenChannels, forgetObject:

Returns YES if the EODatabase uniques fetched enterprise objects, NO if not. The default behavio

If an EODatabase doesn't unique objects then none of the uniquing-snapshotting methods affect th
for example, recordObject:primaryKey:snapshot: won't record the object (though it may record a s