adaptor

Creating a new channel	createAdaptorChannel
Finding open channels	hasBusyChannels
Controlling transactions	beginTransaction commitTransaction rollbackTransaction
Notifying of other transactions	transactionDidBegin transactionDidCommit transactionDidRollback
Nesting transactions	canNestTransactions transactionNestingLevel
Setting the delegate	setDelegate: delegate

(EOAdaptor *)adaptor

Returns the EOAdaptorContext's EOAdaptor.

(BOOL)beginTransaction

Returns YES if the database server and the adaptor context can nest transactions, NO otherwise.

(BOOL)commitTransaction

Attempts to commit the last transaction begun, returning YES if successful. commitTransaction wi

·a transaction is not open
·fetches are in progress
·the delegate returns EODelegateRejects in adaptorContextWillCommit
·the database server fails to commit (and may rollback)

In the first three cases, the transaction is not rolled back. In the fourth, it is. There is no way to tel commitTransaction to return NO.

This method invokes the delegate methods adaptorContextWillCommit: and adaptorContextDidCo

transactionDidCommit, beginTransaction, rollbackTransaction

(EOAdaptorChannel *)createAdaptorChannel

Returns a new EOAdaptorChannel, or nil if a new channel cannot be created. The newly created ch context.

delegate

Returns the adaptor context's delegate. A subclass of EOAdaptorContext doesn't need to override

(BOOL)hasBusyChannels

Returns YES if any of the receiver's channels have outstanding operations (that is, have a fetch in otherwise.

isFetchInProgress (EOAdaptorChannel)

(BOOL)rollbackTransaction

Attempts to roll back the last transaction begun, returning YES if successful and NO if not. This m delegate methods adaptorContextWillRollback: and adaptorContextDidRollback:.

transactionDidRollback, beginTransaction, commitTransaction

(void)setDelegate:anObject

Sets the adaptor context's delegate to anObject and retains it. A subclass of EOAdaptorContext do this method.

(void)transactionDidCommit

Informs the receiver that the server has committed a transaction, so that it can update its state to re application should invoke this method whenever it commits a transaction other than by sending a c message (for example, by using EOAdaptorChannel's evaluateExpression:).

This method doesn't send any messages to the delegate you should explicitly notify the delegate if

(void)transactionDidRollback

Informs the receiver that the server has rolled back a transaction, so that it can update its state to re application should invoke this method whenever it rolls back a transaction other than by sending a message (for example, by using EOAdaptorChannel's evaluateExpression:).

This method doesn't send any messages to the delegate you should explicitly notify the delegate if

(unsigned int)transactionNestingLevel

Returns the number of transactions in progress. If the database server and the adaptor support neste number may be greater than 1.

(void)adaptorContextDidBegin:context

Invoked from beginTransaction to tell the delegate that context has begun a transaction. The deleg action it needs based on this information.

(void)adaptorContextDidCommit:context

Invoked from commitTransaction to tell the delegate that context has committed a transaction. The whatever action it needs based on this information.

(void)adaptorContextDidRollback:context

Invoked from rollbackTransaction to tell the delegate that context has rolled back a transaction. Th whatever action it needs based on this information.

(EODelegateResponse)adaptorContextWillBegin:context

(EODelegateResponse)adaptorContextWillRollback:context

Invoked from rollbackTransaction to tell the delegate that context is rolling back a transaction. See
Adaptor Context's Delegateº in the class description for an explanation of the possible return valu