openChannel	
	isOpen
	closeChannel
Modifying rows	insertRow:forEntity:
	deleteRowsDescribedByQualifier:
	updateRow:describedByQualifier:
Fetching rows	+ dateForAttribute:year:month:day:hour:minute: second:zone:
	selectAttributes:describedByQualifier:fetchOrder:lock:
	describeResults
	fetchAttributes:withZone:
	isFetchInProgress
	cancelFetch
	dictionaryWithObjects:forAttributes:zone:
Sending SQL to the server	evaluateExpression:
Getting the adaptor context	adaptorContext
Getting schema information	describeModelWithTableNames:
	describeTableNames
Debugging	setDebugEnabled:
	isDebugEnabled
Setting the channel's delegate	setDelegate:
	delegate

(EOAdaptorContext *)adaptorContext

Returns the EOAdaptorChannel's EOAdaptorContext.

adaptor (EOAdaptorContext)

(void)cancelFetch

openChannel, isOpen, transactionNestingLevel (EOAdaptorContext)

delegate

Returns the adaptor channel's delegate. A subclass of EOAdaptorChannel doesn't need to override

(BOOL)deleteRowsDescribedByQualifier:(EOQualifier *)aQualifier

Deletes the rows described by aQualifier. The adaptor channel's context must have a transaction in to work. Returns YES on success, NO on failure. Some possible reasons for failure are:

·No row described by aQualifier exists (so no rows are actually deleted).
·The user logged in to the database doesn't have permission to delete.
·The adaptor channel is in an invalid state (for example, fetching).
·The adaptor context has no transaction in progress.

This method invokes the delegate methods adaptorChannel:willDeleteRowsDescribedByQualifier: didDeleteRowsDescribedByQualifier:.

(NSArray *)describeAttributesForEntity:(EOEntity *)anEntity

Returns the basic attributes for anEntity constructed from metadata in the database server. This me assign the attributes to anEntityÐyour application must do that explicitly.

addEntity: (EOEntity), describeModelWithTableNames:, describeTableNames

(NSArray *)describeEntities

Returns an array of entities constructed from metadata in the database server for example, from the tables. These entities contain no attribute or relationship informationÐyour application must load

addEntity: (EOModel), describeAttributesForEntity:, describeRelationshipsForEntity:, describe , describeTableNames

(EOModel *)describeModelWithTableNames:(NSArray *)tableNames

Constructs a default model out of the database's meta data. It also puts the adaptor name and con the new model. This method is typically used in conjunction with describeTableNames.

describeTableNames

(NSArray *)describeTableNames

Reads and returns an array of table names from the database. This method in conjunction with describeModelWithTableNames: is used to build a default model.

describeModelWithTableNames:

(NSArray *)describeResults

Returns an array object of EOAttributes describing the properties available in the current result set selectAttributes:describedByQualifier:fetchOrder:lock: or a select statement evaluated by evaluateE

(NSMutableDictionary *)dictionaryWithObjects:(id *)objects forAttributes:(NSArray *)attribu
zone

This method is used by AdaptorChannel subclasses to create dictionaries that can be returned from withZone:. Use of this method is optional it was added because it enables performance optimizatio

fetchAttributes:withZone:

(BOOL)evaluateExpression:(NSString *)anExpression

Sends anExpression to the database server for evaluation. Returns YES if no error occurs, NO if an EOAdaptorChannel uses this method to perform all of its database operations.

Evaluation of an arbitrary expression may result in any number of changes in state on the server. A track whether the evaluation of an expression results in a select operation being performed, thereby fetch mode. See the EODatabaseChannel class specification for special considerations in using this EODatabaseChannel's adaptor channel.

This method invokes the delegate methods adaptorChannel:willEvaluateExpression: and adaptorCh didEvaluateExpression:.

isFetchInProgress

(NSMutableDictionary *)fetchAttributes:(NSArray *)attributes withZone:(NSZone *)zone

Fetches the next row from the result set of the last selectAttributes:describedByQualifier:fetchOrde evaluateExpression: message and returns values for the EOAttributes contained in attributes. When rows in the current result set, this method returns nil, and if there are more results sets this method method adaptorChannelDidChangeResultSet:. When there are no more rows or result sets, this met the fetch, and sends adaptorDidFinishFetching: to the delegate. isFetchInProgress returns YES unt or until this method exhausts all result sets and returns nil.

In addition to the delegate methods mentioned above, this method may also invoke the delegate me willFetchAttributes:withZone: and adaptorChannel:didFetchAttributes:withZone:.

cancelFetch

·The adaptor context has no transaction in progress.
·The row fails to satisfy a constraint defined in the database server.

This method invokes the delegate methods adaptorChannel:willInsertRow:forEntity: and adaptorC
forEntity:.

deleteRowsDescribedByQualifier:, selectAttributes:describedByQualfier:fetchOrder:lock:

(BOOL)isDebugEnabled

Returns YES if the adaptor channel logs evaluated SQL and other useful information to the consol
error stream), NO if not.

(BOOL)isFetchInProgress

Returns YES if the adaptor channel is fetching, NO otherwise. An adaptor channel is fetching if it'
selectAttributes:describedByQualifier:fetchOrder:lock: message, or if an expression sent through e
resulted in a select operation being performed. An adaptor channel stops fetching when there are n
or when it's sent a cancelFetch message.

fetchAttributes:WithZone:

(BOOL)isOpen

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

closeChannel

(BOOL)openChannel

Puts the channel and both its context and adaptor into a state where they are ready to perform datab
Returns YES on success, NO on failure or if the channel is already open.

closeChannel, isOpen, hasValidConnectionDictionary (EOAdaptor)

(BOOL)readTypeForAttribute:(EOAttribute *)anAttribute

Loads external type information from the database server into anAttribute. Returns YES on succes

Attributes read with the describeAttributesForEntity: methods already have their type information
use this method to read type information for a manually created attribute, or for an attribute whose
may have changed.

(BOOL)readTypesForEntity:(EOEntity *)anEntity

(BOOL)selectAttributes:(NSArray *)attributes
describedByQualifier:(EOQualifier *)aQualifier
fetchOrder:(NSArray *)fetchOrder
lock:(BOOL)flag

Selects the EOAttributes in attributes for rows matching aQualifier. The selected rows compose on
each row of which will be returned by subsequent fetchAttributes:withZone: messages according t
of EOAttributeOrdering objects). If fetchOrder is nil, rows aren't sorted as they're fetched. If flag i
supports locking, the rows are locked so that no other user can modify them. The adaptor channel's
transaction in effect for this method to work. Returns YES on success, NO on failure.

Some possible reasons for failure are:

·The adaptor channel is in an invalid state (for example, fetching).
·The adaptor context has no transaction in progress.
·Locking was requested but the adaptor doesn't support it.

This method invokes the delegate methods adaptorChannel:willSelectAttributes:describedByQuali
and adaptorChannel:didSelectAttributes:describedByQualifier:fetchOrder:lock:.

isFetchInProgress, cancelFetch

(void)setDebugEnabled:(BOOL)flag

Sets according to flag whether the adaptor channel logs evaluated SQL and other useful debugging
console (or to the standard error stream). The information provided may vary from adaptor to adap
from release to release.

(void)setDelegate:anObject

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

(BOOL)updateRow:(NSDictionary *)newValues describedByQualifier:(EOQualifier *)aQuali

Updates the row described by aQualifier by changing its values to match those in newValues. new
NSDictionary whose keys are attribute names and whose values are the new values for those attrib
need only contain entries for the attributes being changed). The adaptor channel's context must hav
effect for this method to work. Returns YES on success, NO on failure. Some possible reasons for

·No row described by aQualifier exists.
·aQualifier describes more than one row.
·The user logged in to the database doesn't have permission to update.
·The adaptor channel is in an invalid state (for example, fetching).
·The adaptor context has no transaction in progress.
·The new values fail to satisfy a constraint defined in the database server.

This method invokes the delegate methods adaptorChannel:willUpdateRow:describedByQualifier:
didUpdateRow:describedByQualifier:.

isFetchInProgress, beginTransaction (EOAdaptorContext) (void)adaptorChannel:channel didDeleteRowsDescribedByQualifier:(EOQualifier *)aQualifie

Invoked from deleteRowsDescribedByQualifier: to tell the delegate that some rows have been dele
take whatever action it needs based on this information.

(void)adaptorChannel:channel didEvaluateExpression:(NSString *)anExpression

Invoked from evaluateExpression: to tell the delegate that a query language expression has been ev
server. Note that nearly every method that communicates with a database does so by using evaluat
delegate may take whatever action it needs based on this information. Note that the adaptor channe
whether anExpression resulted in a select operation being performed.

(NSMutableDictionary *)adaptorChannel:channel
didFetchAttributes:(NSMutableDictionary *)attributes
withZone:(NSZone *)zone

Invoked from fetchAttributes:withZone: to tell the delegate that a single row has been fetched. The
substitute dictionary that will be used instead of attributes, or return nil to cause fetchAttributes:wi
return nil. Returning nil doesn't cancel the fetch, however.

cancelFetch

(void)adaptorChannelDidFinishFetching:channel

Invoked from fetchAttributes:withZone: to tell the delegate that fetching is finished for the current
method is invoked when a fetch ends in fetchAttributes:withZone: because there are no more resul

(void)adaptorChannel:channel
didInsertRow:(NSDictionary *)aRow
forEntity:(EOEntity *)anEntity

Invoked from insertRow:forEntity: to tell the delegate that a row has been inserted. The delegate m
to update records, redisplay onscreen information, or take whatever other action it needs.

(void)adaptorChannel:channel
didSelectAttributes:(NSArray *)attributes
describedByQualifier:(EOQualifier *)aQualifier
fetchOrder:(NSArray *)fetchOrder
lock:(BOOL)flag

Invoked from selectAttributes:describedByQualifier:fetchOrder:lock: to tell the delegate that some
selected. The delegate may take whatever action it needs based on this information.

(EODelegateResponse)adaptorChannel:channel willDeleteRowsDescribedByQualifier:(EOQu

Invoked from deleteRowsDescribedByQualifier: to tell the delegate that a row is being deleted. Th
aQualifier to affect the result of the delete operation. See ªNotifying the Adaptor Channel's Delega
description for an explanation of the possible return values.

(EODelegateResponse)adaptorChannel:channel willEvaluateExpression:(NSMutableString *)a

Invoked from evaluateExpression: to tell the delegate that anExpression is about to be sent to the d
that nearly every method that communicates with a database does so by using evaluateExpression:
modify anExpression to affect the result. See ªNotifying the Adaptor Channel's Delegateº in the cl
explanation of the possible return values.

(NSMutableDictionary *)adaptorChannel:channel
willFetchAttributes:(NSArray *)attributes withZone:(NSZone *)zone

Invoked from fetchAttributes:withZone: to tell the delegate that a single row will be fetched. If the
nil value, that value will be treated as a dictionary resulting from a fetch operation the adaptor char
fetch, and fetchAttributes:withZone: returns YES. The delegate must ensure that the server skips th
the next fetch message doesn't simply get that same row.

If the delegate returns nil the adaptor channel performs the fetch itself.

(EODelegateResponse)adaptorChannel:channel
willInsertRow:(NSMutableDictionary *)aRow
forEntity:(EOEntity *)anEntity

Invoked from insertRow:forEntity: to tell the delegate that a row is being inserted. The delegate ca
attribute names in aRow, or add or delete attribute-value pairs. See ªNotifying the Adaptor Channe
class description for an explanation of the possible return values.

(EODelegateResponse)adaptorChannel:channel
willSelectAttributes:(NSMutableArray *)attributes
describedByQualifier:(EOQualifier *)aQualifier
fetchOrder:(NSArray *)fetchOrder
lock:(BOOL)flag

Invoked from selectAttributes:describedByQualifier:fetchOrder:lock: to tell the delegate that a sele
performed. The delegate can modify attributes to affect the select operation, but shouldn't modify a
ªNotifying the Adaptor Channel's Delegateº in the class description for an explanation of the poss

(EODelegateResponse)adaptorChannel:channel
willUpdateRow:(NSMutableDictionary *)newValues