
EOAdaptorOperation

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	EOAccess/EODatabaseOperation.h

Class Description

An EOADaptorOperation object represents a primitive operation in a database server—lock, insert, update, or delete a row; or execute a stored procedure—and all the necessary information required by the operation. An EOADaptorOperation is processed by an EOADaptorChannel object in the method **performAdaptorOperation:**. You don't ordinarily create instances of EOADaptorOperation; rather, the Framework automatically creates an EOADaptorOperation object and sends it to an adaptor channel when your application needs the database server to perform an operation. You generally interact with EOADaptorOperation objects only if you need to specify the order in which a set of operations are carried out (see the description for the EODatabaseContext delegate method **databaseContext:willOrderAdaptorOperationsFromDatabaseOperations:**).

An EOADaptorOperation has an entity and an operator (the type of operation the object represents). An adaptor operation's operator (EOAdaptorLockOperator, EOADaptorInsertOperator, EOADaptorUpdateOperator, EOADaptorDeleteOperator, or EOADaptorStoredProcedureOperator) determines additional, operator-dependent information used by the EOADaptorOperation object. For example, only a stored procedure operation has an EOStoredProcedure object. The operator-dependent information is accessible using the methods described below.

Method Types

Creating a new EOADaptorOperation	– initWithEntity:
Accessing the entity	– entity
Setting the operator	– setAdaptorOperator: – adaptorOperator
Setting the qualifier	– setQualifier: – qualifier
Setting locking attributes	– setAttributes: – attributes

Setting operation values	<ul style="list-style-type: none"> – setChangedValues: – changedValues
Setting a stored procedure	<ul style="list-style-type: none"> – setQualifier: – storedProcedure
Handling errors during the operation	<ul style="list-style-type: none"> – setException: – exception
Comparing operations	<ul style="list-style-type: none"> – compareAdaptorOperation:

Class Methods

adaptorOperator

– (EOAdaptorOperator)**adaptorOperator**

Returns the receiver's adaptor operator. The operator indicates which of the other adaptor operation attributes are valid. For example, an adaptor operation whose operator is EOAdaptorInsertOperator uses **changedValues**, but not **attributes**, **qualifier**, or **storedProcedure**.

See also: – setAdaptorOperator:

attributes

– (NSArray *)**attributes**

Returns the array of attributes to select when locking the row. If attributes have not been assigned to the receiver, the primary key attributes are selected. Only valid for adaptor operations with the EOAdaptorLockOperator.

See also: – setAttributes:

changedValues

– (NSMutableDictionary *)**changedValues**

Returns the dictionary of values that need to be updated, inserted, or compared for locking purposes.

See also: – setChangedValues:

compareAdaptorOperation:

– (NSComparisonResult)**compareAdaptorOperation:**(EOAdaptorOperation *)*operation*

Orders adaptor operations alphabetically by entity name and by adaptor operator within the same entity. The EOAdaptorOperators are ordered as follows:

- EOAdaptorLockOperator
- EOAdaptorInsertOperator
- EOAdaptorUpdateOperator
- EOAdaptorDeleteOperator
- EOAdaptorStoredProcedureOperator

EOAdaptorLockOperator precedes EOAdaptorInsertOperator, EOAdaptorInsertOperator precedes EOAdaptorUpdateOperator, and so on.

An EODatabaseContext uses **compareAdaptorOperation:** to order adaptor operations before invoking EOAdaptorChannel's **performAdaptorOperations:** method.

entity

– (EOEntity *)**entity**

Returns the entity to which the operation will be applied.

See also: – **initWithEntity:**

exception

– (NSException *)**exception**

Returns the exception raised when an adaptor channel attempted to process the receiver.

See also: – **setException:**

initWithEntity:

– **initWithEntity:**(EOEntity *)*entity*

The designated initializer, initializes a new EOAdaptorOperation instance, and sets the entity to which the operation will be applied to *entity*. Returns **self**.

See also: – **entity**

qualifier

– (EOQualifier *)**qualifier**

Returns the qualifier that identifies the specific row to which the operation applies. Not valid with adaptor operations with the operators EOAdaptorInsertOperator and EOAdaptorStoredProcedureOperator.

See also: – **setQualifier:**

setAdaptorOperator:

– (void)**setAdaptorOperator:**(EOAdaptorOperator)*adaptorOperator*

Sets the receiver's operator to *adaptorOperator*, which is one of the following:

- EOAdaptorLockOperator
- EOAdaptorInsertOperator
- EOAdaptorUpdateOperator
- EOAdaptorDeleteOperator
- EOAdaptorStoredProcedureOperator

For more information, see the discussion on adaptor operators in the class description above.

See also: – **adaptorOperator**

setAttributes:

– (void)**setAttributes:**(NSArray *)*attributes*

Sets the array of attributes to select when locking the row. The selected values are compared in memory to the corresponding snapshot values to determine if a row has changed since the application last fetched it. *attributes* is an array of EOAttribute objects that can't be compared in a qualifier (generally BLOB types); it should not be **nil** or empty. Generally, an adaptor operation's qualifier contains all the comparisons needed to verify that a row hasn't changed since the applicaiton last fetched, inserted, or updated it. In this case (if there aren't any attributes that can't be compared in a qualifier), *attributes* should contain primary key attributes. This method is only valid for adaptor operations with the EOAdaptorLockOperator.

See also: – **attributes**, – **entity**

setChangedValues:

– (void)**setChangedValues:**(NSDictionary *)*changedValues*

Sets the dictionary of values that need to be updated, inserted, or compared for locking purposes. *changedValues* is a dictionary object whose keys are attribute names and whose values are the values for

those attributes. As summarized in the following table, the contents of *changedValues* depends on the receiver's operator:

Operator	Contents of changedValues Dictionary
EOAdaptorLockOperator	snapshot values used to verify that the database row hasn't changed since this application last fetched it
EOAdaptorInsertOperator	the values to insert
EOAdaptorUpdateOperator	the new values for the columns to update
EOAdaptorDeleteOperator	snapshot values (changedValues is only valid for <i>EOAdaptorDeleteOperation</i> if the receiver's entity uses a stored procedure to perform delete operations.)

EOAdaptorStoredProcedureOperators**snapshot values**

See also: – **changedValues**

setException:

– (void)**setException:**(NSException *)*exception*

Sets the receiver's exception to *exception*. This method is typically invoked from EOAdaptorChannel's **performAdaptorOperations:** method. If a database error occurs while processing an adaptor operation, the adaptor channel creates an exception and assigns it to the adaptor operation.

See also: – **exception**

setQualifier:

– (void)**setQualifier:**(EOQualifier *)*qualifier*

Sets the qualifier that identifies the row to which the adaptor operation is to be applied to *qualifier*.

See also: – **qualifier**

setStoredProcedure:

– (void)**setStoredProcedure:**(EOStoredProcedure *)*storedProcedure*

Sets the receiver's stored procedure to *storedProcedure*.

See also: – **storedProcedure**

storedProcedure

– (EOStoredProcedure *)**storedProcedure**

Returns the receiver's stored procedure. Only valid with adaptor operations with the EOAdaptorStoredProcedureOperation.

See also: – **setQualifier:**