

Defined Types

EODelegateResponse

DECLARED IN eoaccess/EOAdaptorContext.h

SYNOPSIS

```
typedef enum {  
    EODelegateRejects,  
    EODelegateApproves,  
    EODelegateOverrides  
} EODelegateResponse;
```

DESCRIPTION

EODelegateResponse indicates the result a delegate demands on notification that a particular action will be taken. If a delegate returns **EODelegateRejects**, the sender of the delegation message must abort the current operation and return a failure result. If the delegate returns **EODelegateApproves**, the sender should continue the operation and return whatever result is appropriate. If the delegate returns **EODelegateOverrides**, the sender must do nothing, but return a success result. When a delegate overrides an operation it's responsible for ensuring that the result of that operation is achieved.

EOJoinOperator

DECLARED IN eoaccess/EOJoin.h

SYNOPSIS

typedef enum {

```
    EOJoinEqualTo,  
    EOJoinNotEqualTo,  
    EOJoinGreaterThan,  
    EOJoinGreaterThanOrEqualTo,  
    EOJoinLessThan,  
    EOJoinLessThanOrEqualTo  
} EOJoinOperator;
```

DESCRIPTION **EOJoinOperator** defines the possible comparison method to use in building a join. Objects (or records) with attribute values matching by the comparison operator are included in the join; those not fitting are excluded.

EOJoinSemantic

DECLARED IN eoaccess/EOJoin.h

SYNOPSIS

typedef enum {

```
    EOInnerJoin,  
    EOFullOuterJoin,  
    EOLeftOuterJoin,  
    EORightOuterJoin  
} EOJoinSemantic;
```

DESCRIPTION **EOJoinSemantic** specifies the manner in which a join should be made. An inner join produces results only for source records that satisfy the condition of the join. A full outer join produces results for all source records, regardless of the values of the relationships. A left outer join preserves rows in the left (source) table

(keeps them even if there's no corresponding row in the right table), while a right outer join preserves rows in the right (destination) table.

Note: Not all join semantics are supported by all database servers. See your adaptor's documentation for information on the supported join semantics.

EOOrdering

DECLARED IN [eoaccess/EOAttributeOrdering.h](#)

SYNOPSIS

```
EOAnyOrder,  
EOAscendingOrder,  
EODescendingOrder  
} EOOrdering;
```

```
typedef enum {
```

DESCRIPTION	EOOrdering encodes the three possible orders for a set of values. EOAnyOrder indicates no ordering. EOAscendingOrder indicates ordering from least to greatest value. EODescendingOrder indicates ordering from greatest to least value.
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

EOUpdateStrategy

DECLARED IN eoaccess/EODatabaseContext.h

SYNOPSIS

**EOUpdateWithOptimisticLocking,
EOUpdateWithPessimisticLocking,
EOUpdateWithNoLocking,**

```
typedef enum {
```

```
    EONoUpdate,  
} EOUpdateStrategy;
```

DESCRIPTION **EOUpdateStrategy** defines the manner in which an EODatabaseContext performs an update. See the EODatabaseContext class specification for information on update strategies.