
EOKeyComparisonQualifier

Inherits From:	EOQualifier : NSObject
Conforms To:	EOQualifierEvaluation EOQualifierSQLGeneration
Declared In:	EOControl/EOQualifier.h

Class Description

EOKeyComparisonQualifier is a subclass of EOQualifier that compares a named property of an object with a named value of another object. For example, to return all of the employees whose salaries are greater than those of their managers, you might use an expression such as “salary > manager.salary”, where “salary” is the *left key* and “manager.salary” is the *right key*. The “left key” is the property of the first object that’s being compared to a property in a second object; the property in the second object is the “right key”.

EOKeyComparisonQualifier adopts the EOQualifierEvaluation protocol, which defines the method **evaluateWithObject:** for in-memory evaluation. When an EOKeyComparisonQualifier object receives an **evaluateWithObject:** message, it evaluates the given object to determine if it satisfies the qualifier criteria.

In addition to performing in-memory filtering, EOKeyComparisonQualifier can be used to generate SQL. When it’s used for this purpose, the key should be a valid property name of the root entity for the qualifier (or a valid key path).

Adopted Protocols

EOQualifierEvaluation	– evaluateWithObject:
EOQualifierSQLGeneration	– sqlStringForSQLExpression: – schemaBasedQualifierWithRootEntity:

Method Types

Initializing an EOKeyComparisonQualifier	– initWithLeftKey:operatorSelector:rightKey:
--	--

Returning information about an EOKeyComparisonQualifier

- leftKey
- rightKey
- selector

Instance Methods

evaluateWithObject:

@protocol EOQualifierEvaluation
– (BOOL)evaluateWithObject:*object*

Returns YES if the object *object* satisfies the qualifier, NO otherwise. When an EOKeyComparisonQualifier object receives an **evaluateWithObject:** message, it evaluates *object* to determine if it meets the qualifier criteria. This method can raise one of several possible exceptions if an error occurs. If your application allows users to construct arbitrary qualifiers (such as through a user interface), you may want to put exception handlers around this method to properly respond to errors (for example, by displaying a panel saying that the user typed a poorly formed qualifier).

initWithLeftKey:operatorSelector:rightKey:

– **initWithLeftKey:**(NSString *)*leftKey* **operatorSelector:**(SEL)*selector* **rightKey:**(id)*rightKey*

Initializes the receiver to compare the properties named by *leftKey* and *rightKey*, using the operator selector *selector*.

EOQualifierOperatorEqual
EOQualifierOperatorNotEqual
EOQualifierOperatorLessThan
EOQualifierOperatorGreaterThan
EOQualifierOperatorLessThanOrEqualTo
EOQualifierOperatorGreaterThanEqualTo
EOQualifierOperatorContains
EOQualifierOperatorLike

Enterprise Objects Framework supports SQL generation for these selectors only.

For example, the following excerpt creates an EOKeyComparisonQualifier **qual** that has the left key "lastName", the operator selector EOQualifierOperatorEqual, and the right key "member.lastName". Once constructed, the qualifier **qual** is used to filter an in-memory array. The code excerpt returns an array of Guest objects whose **lastName** properties have the same value as the **lastName** property of the guest's sponsoring member (this example is based on the Rentals sample database).

```
NSArray *guests;      /* Assume this exists. */  
EOQualifier *qual = [[EOKeyComparisonQualifier alloc]
```

```
initWithLeftKey:@"lastName"  
operatorSelector:EOQualifierOperatorEqual  
rightKey:@"member.lastName"];  
  
return [guests filteredArrayUsingQualifier:qual];
```

leftKey

– (NSString *)**leftKey**

Returns the receiver's left key.

rightKey

– (NSString *)**rightKey**

Returns the receiver's right key.

selector

– (SEL)**selector**

Returns the receiver's selector.