

---

# EOSQLQualifier

<b>Inherits From:</b>	EOQualifier : NSObject
<b>Conforms To:</b>	EOQualifierSQLGeneration, NSObject (NSObject)
<b>Declared In:</b>	EOAccess/EOSQLQualifier.h

## Class Description

EOSQLQualifier is a subclass of EOQualifier that contains unstructured text that can be transformed into a SQL expression. EOSQLQualifier is provided for backwards compatibility with pre-2.0 Enterprise Objects Framework releases and to provide a way to create SQL expressions with any arbitrary SQL.

EOSQLQualifier formats are not parsed, they simply perform substitution for keys and format characters. The qualifying information is expressed in the database server's query language (nearly always SQL), and you're responsible for ensuring that the query language statement is valid for your database server.

EOSQLQualifiers can't be evaluated against objects in memory. As a result, you should use EOQualifier whenever possible and only use EOSQLQualifier in cases that absolutely require it.

You create an SQL qualifier using **alloc...** and **initWithEntity:qualifierFormat:**. This method takes as arguments the root entity for the qualifier and a format string like that used with the standard creation method **qualifierWithQualifierFormat:**.

**Note:** Because an SQL qualifier must be rooted to an entity, you can't use **qualifierWithQualifierFormat:** to create EOSQLQualifier objects.

## Formatting an SQL Qualifier

To be sure you create qualifiers that use the proper formats for your database, format attribute names and values using the concrete subclass of EOSQLExpression for your adaptor. The following example uses the EOSQLExpression class method **formatValue:forAttribute:** to ensure that the qualifier string is well-formed for the database.

```
EOAdaptor *adaptor;           // Assume this exists.
EOEntity *employeeEntity;     // Assume this exists.
EOAttribute *lastNameAttribute; // Assume this exists.
Employee *anEmployee;         // Assume this exists.
NSString *lastName;
EOSQLQualifier *qualifier;
EOSQLExpressionClass *expressionClass = [adaptor expressionClass];

lastName = [expressionClass formatValue:[anEmployee lastName]
                                forAttribute:lastNameAttribute];
```

---

```
qualifier = [[EOSQLQualifier alloc]
    initWithEntity:employeeEntity
    qualifierFormat:@"%A = %@", [lastNameAttribute name], lastName];
```

## Adopted Protocols

EOQualifierSQLGeneration      – sqlStringForSQLExpression:  
                                 – schemaBasedQualifierWithRootEntity:

## Class Methods

### **qualifierWithQualifierFormat:**

+ (EOQualifier \*)**qualifierWithQualifierFormat:**(NSString \*)*format*, ...

Overrides EOQualifier's implementation to raise an exception. An EOSQLQualifier must be initialized with an entity, and this method does not provide one. Use **alloc...** and **initWithEntity:qualifierFormat:** to create EOSQLQualifier instances.

## Instance Methods

### **initWithEntity:qualifierFormat:**

– **initWithEntity:**(EOEntity \*)*entity* **qualifierFormat:**(NSString \*)*qualifierFormat*, ...

Initializes a newly allocated EOSQLQualifier rooted in *entity* and built from a format string. *qualifierFormat* is a **printf**()-style format string as presented in the class description. This is the designated initializer for the EOSQLQualifier class. Returns **self** if *qualifierFormat* is successfully parsed, **nil** otherwise.