EOStoredProcedure

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

Class Description

An EOStoredProcedure represents a stored procedure defined in a database, and associates a name internal to the Framework with an external name—by which the stored procedure is known to the database. If a stored procedure has arguments, its EOStoredProcedure object also maintains a group of EOAttributes which represent the stored procedure's arguments. See the EOAttribute class specification for more information

You usually define stored procedures in your EOModel with the EOModeler application, which is documented in the *Enterprise Objects Framework Developer's Guide*. EOStoredProcedures are primarily used by the Enterprise Objects Framework to map operations for an EOEntity to stored procedures (see the description for EOEntity's **setStoredProcedure:forOperation:** method). You can assign stored procedures to an entity for any of the following scenarios:

- Fetching all the objects for the entity
- Fetching a single object by its primary key
- Inserting a new object
- Deleting an object
- Generating a new primary key

Your code probably won't use EOStoredProcedures unless you're working at the adaptor level.

Like the other major modeling classes, EOStoredProcedure provides a user dictionary for your application to store any application-specific information related to the stored procedure.

Method Types

Creating a new EOStoredProcedure	– initWithName:
Setting the model	– setModel: – model
Setting the name	– setName: – beautifyName – name

Setting the external name	– setExternalName: – externalName
Setting the arguments	– setArguments: – arguments
Setting the user dictionary	– setUserInfo: – userInfo

Instance Methods

arguments

– (NSArray *)**arguments**

Returns the EOAttribute objects that describe the stored procedure's arguments or **nil** if the stored procedure has no arguments.

See also: – **setArguments:**

beautifyName

– (void)**beautifyName**

Renames the receiver's name and its arguments to conform to the Framework's naming conventions. For example, "NAME" is renamed "name" and "FIRST_NAME" is renamed "firstName".

externalName

– (NSString *)**externalName**

Returns the name of the stored procedure as it is defined in the database, or **nil** if the receiver doesn't have an external name.

See also: – **setExternalName:**

initWithName:

– (EOStoredProcedure *)**initWithName:**(NSString *)*name*

The designated initializer for EOStoredProcedure, this method initializes a new EOStoredProcedure object and sets its name to *name*. Returns **self**.

See also: – **setName:**, – **name**

model

– (EOModel *)**model**

Returns the model to which the receiver belongs.

See also: – **setModel:**, – **addStoredProcedure:** (EOModel)

name

– (NSString *)**name**

Returns the name of the receiver.

See also: – **setName:**, – **initWithName:**

setArguments:

– (void)**setArguments:**(NSArray *)*arguments*

Sets *arguments* as the array of EOAttributes that describe the receiver's arguments. The EOAttribute objects in *arguments* must be ordered to match the database stored procedure definition.

See also: – **arguments**

setExternalName:

– (void)**setExternalName:**(NSString *)*name*

Sets the external name of the stored procedure to *name*. *name* should be the name of the stored procedure as it is defined in the database.

See also: – **externalName**

setName:

– (void)**setName:**(NSString *)*name*

Sets the name of the receiver.

See also: – **name**, – **initWithName:**

setUserInfo:

– (void)**setUserInfo:**(NSDictionary *)*dictionary*

Sets the *dictionary* of auxiliary data, which your application can use for whatever it needs. *dictionary* can only contain property list data types (that is, NSDictionaries, NSStrings, NSArrays, and NSDatas).

See also: – **userInfo**

userInfo

– (NSDictionary *)**userInfo**

Returns a dictionary of user data. Your application can use this to store any auxiliary information it needs.

See also: – **setUserInfo:**