PeopleDemo

by Mai Nguyen, NeXT Developer Support Team

Overview

This example shows how to manipulate records using the EOController API and its delegation mechanism. A simple validation scheme is used to validate the user input before an actual insert or update to the database server.

Program Organization

How to build a master-detail view

To build a master-detail view, you need to drag and drop the **People.eomodel** file into IB and select Deparment as the root entity. This will generate an EOController which corresponds to the Department table. You also have to drag and drop another EOController from the EOPalette into the File Window of your NIB file. Then, by connecting the **toEmployee** one-to-many relationship to this second controller, it will become the detail controller which corresponds to the Employee table.

Note that the Department EOController has "Buffer Edits" turned ON. This means that modifications to the record are buffered until you explicitly select the *Modify* button. On the other hand, "Buffer Operations" is OFF, which means that modifying a record is automatically affecting the server. You can see the UPDATE statement in the SQL panel by selecting the menu option "Show SQL...".

User Interface

The User Interface is composed of one window and two panels:

- The main window shows the Department and the Employee tables in a master-detail view.
- A Fetch Options panel builds a new sorting order and a simple qualifier based on the user input.
- A SQL tracing panel shows the SQL statements generated during the insert/update/delete operations.

Important UI Note: To notify a modification in the record panel, you must explicitly hit the *Carriage Return* key. Just tabbing through the fields will not enter this change into the buffer. This is currently a known problem in EOF Beta Release 1.0. See the Supplemental Release Notes for EOF.

Major Classes in the Application

MainController	A subclass of Object. It coordinates the fetch, insert, and delete operations. It also manages validation of the record before an insert or update actually happens.
Department	The Department object corresponds to the Department table in the People database. This example of an enterprise object shows how to manipulate additional data not contained in the model file (the averageSalary).

Other Peculiarities

To load the adaptors dynamically, you need to add the option:

```
OTHER_LDFLAGS = -all_load
```

in your Makefile.preamble.

This application includes the following libraries in the Project Builder's libraries suitcase: libEOInterface_s.a, libEOAccess_s.a, and libFoundation_s.a.

To take advantage of the AutoRelease Pool, the Application class of this example is defined as EOApplication. To do so, select Attributes in Project Builder, and enter *EOApplication* in the TextField labeled App. Class. You also need to change the File Owner's Application class to *EOApplication.* Finally, you need to edit your PeopleDemo_main.m file to change the import line to #import <eointerface/EOApplication.h>.

Topics Of Interest

How to validate the user input before an Update
This is done in the controller delegate method - `(NSDictionary*)controller: (EOController *)controller willSave: (NSDictionary *)edits object:object`

How to properly delete a master record
When deleting a master record, the associated detail records are not deleted. However, since the Employee detail record refers to the master record with the DeptId foreign key, those keys must be properly nulled out in the associated detail records.

Debugging/Tracing
There are two types of debugging and tracing methods:
1) In the file MainController.m, the adaptor channel method **setDebugEnabled:** will trace all

operations that go through the adaptor channel.
2) In the file Department.m, the method **respondsTo:** will show what methods your enterprise object can respond to.
To turn the verbose mode on, you can add the following option in your Makefile.preamble:

```
OTHER_CFLAGS = -DDEBUG
```

Tips for moving from a Sybase to an Oracle database

As you may have noticed, the two examples PeopleDemo_oracle and PeopleDemo_sybase have identical source codes and nib files. When moving from a Sybase to an Oracle based example, you'll need to create a new **.eomodel** file from the new server. However, if you keep the internal names of the database as well as the external names identical, you won't have to redo all the connections inside Interface Builder. Naturally, you should also test your new Oracle application to make sure that it behaves exactly like the version on Sybase.

SQL scripts and Model Files

installPEOPLE.sqlsybase
installPEOPLE.sqloracle Used to create and restore the databases for the Sybase and Oracle

sybasePeople.eoModel
oraclePeople.eoModel Model files to use respectively for Sybase or Oracle.

Valid for NEXTSTEP Release 3.2 installed with the Early Access Version of EOF (June 1994)

June 1994 Created example