AddressBook

by Mai Nguyen, NeXT Developer Support Team

Overview

In order to run this example, you need to have a SYBASE server with a pubs database installed. The pubs database is a sample database provided with the SYBASE limited server package.

After you are properly connected to the Sybase server, you can perform operations such as Select, Insert, Update or Delete. Select is performed by clicking on a particular item in the scrollview, which will then display the person's record (last name, first name, social security, phone, address, state of residence, zip code). You have to fill in all these fields when adding a record. Text input is not validated since it is beyond the scope of this small example.

Note that this example only uses the DBKit Access Layer, which is useful to know if you need to create your own front-end to a database server. With the DBKit standard UI objects (TableView, DBModule), you could also create a similar example without much programming.

Program Organization

Major Classes in the Application

Controller	A general manager object. A subclass of the Object class. Performs miscellaneous initialzations, adding and deleting records at a high level.
AddressView	A subclass of the ScrollView class with a matrix of textfield cells. It handles lower level operations for adding, updating and deleting records.

Other Peculiarities

The primary key for inserting a new record is the social security number. It is not possible to insert a new record with a

duplicate id (or social security number).
In order to load an Adaptor dynamically, you need to add the **OTHER_LDFLAGS** definition (see the **Makefile.preamble**). You also need to add the **libdbkit_s.a** library into your **PB.project** under **libraries** in order to use the DBKit API.

Topics Of Interest

The purpose of this example is to show you some simple examples of the usage of the DBKit Access Layer:

- How to get connected to the SYBASE server via a default login string
See the method **appDidInit:** in the file Controller.m.

- How to fetch the data from a   database using a DBRecordList object
See the method **loadCellsFrom:** in the file AddressView.m

- **How to view all the values inside a record in a record list using the DBValue object**
See the method **showInfo:** in the file AddressView.m

- **How to add a new record to a database via the DBRecordList object**
See the method **addRecordFrom:at:** in the file AddressView.m

- **How to delete an existing record to a database via the DBRecordList object**
See the method **deleteSelectedRecord:** in the file AddressView.m

- **How to update an existing record to a database via the DBRecordList object**
See the method **updateRecordFrom:at:** in the file AddressView.m

- **How to set up a database delegate to report errors and SQL queries generated by the Sybase adaptor:**
See the file Controller.m

Documentation Update

Please note that the on-line documentation for the method **saveModifications:** for   DBRecordList has some errors. A return

value **0** should mean success. Here are the corrections for DBRecordList.
The possible return values from **saveModifications:** are as follows:

Value	Reason
0	The save operation was successful.
1	The save completed but not all records were saved. This happens if errors are encountered but the delegate requests that the save proceeds anyway.
DB_NoIndex	Either the DBRecordList isn't ready (its status is **DB_NotReady** or **DB_NoRecordKey**), or one or more records in the database have changed since they were fetched and the delegate hasn't forced the modifications to be saved. (See **recordStream:willFailForReason:** (DBRecordStream))

Change History

May 1992	Upgraded for PR1
August 1992	Upgraded to final 3.0 DBKIT API