BARCHART

by Joe Freeman, NeXT Systems Engineer

Overview

This palette demonstrates how to use your custom UI objects with the DBKit. It illustrates in particular the DBCustomAssociation protocol which allows you to retrieve or set values to your database via a custom view object (here the ChartOfMatrix view). Note that instead of a custom UI object, you can also connect the nib's file owner to an attribute of your DBModule, and create a custom association in this way.

Other peculiarities

When compiling this example, you'll notice the compiler warnings about conflicting declarations for the methods **maxValue** and **minValue**. This is a known problem, which will be fixed in a future release of NeXTSTEP.

How to use this example

Demo without DBKit
1)   Load the **palette** and create a **New Application.**
2)   Drag a **plotter** into the **Window.**
3)   Drag off a Slider and create a matrix of **Sliders.**
4)   Connect   the **matrix** to the action method **plotFromMatrix** of the **plotter**.
5)   Go into **Test Interface** mode.
6)   Move the sliders and watch the bars move.
7)   End **Test** mode.
8)   Drag two **TextFields** into the window.
9)   Connect the **minField** and **maxField** outlets of the **plotter** to the **TextFields.**
10) Go back   into **TestInterface** mode.
11) Move the sliders and watch the values displayed in the textfields.

Demo with DBKit using the OracleDemo.dbmodel supplied in /NextDeveloper/Examples/DatabaseKit/Models
1) Load the **palette** and create a **New Application.**
2) Drag a **plotter** into the **Window.**
3) Drag a **DBModule** onto the Objects icon of the NIB file.
4) Drag a **DBTableView** into the **Window.**
5) Select OracleDemo, Employee in the module inspector.
6) Connect the **Employee.name** and **Employee.job** to the **DBTableView.**
7) Connect **Employee.salary** to the **plotter**.
8) Connect the **min** and **max** outlets on the **Plotter** to two **TextFields** .
9) Connect the **Button** target to the DBModule **fetchAllRecords:** method.
10) Connect **Employee.salary** to a **Slider**. Set the **Slider** min to 4000 and max to 6000 via the IB Inspector panel.
11) Connect the **Slider** target to the DBModule **takeValueFrom:** method.
12) Go into **TestInterface** mode.
13) Press the button to fetch all records.
14) Select different rows in the table view and watch the colored bar move.
15) Move the Slider and watch the colored selected bar move.
16) Mouse down on one of the bars in the graph and that row will be selected.

Topics of Interest
Besides showing you how to create a custom palette, this particular example also illustrates the use of the DBCustomAssociation protocol to make your UI object (here the custom view palette) respond to changes in the DBModule and DBFetchGroup. To make your UI object (custom view) respond to this protocol, you'll need to connect it with a DBModule attribute (here Employee.salary).

See File ChartOfMatrix.m for more details:

@interface Object(DBCustomAssociation)
// Called when DBModule/FetchGroup fetches a batch of new values
- associationContentsDidChange:association;

// Called when DBFetchGroup selection changes

//   e.g., when a new record is selected in a TableView
- associationSelectionDidChange:association;

// Called when a property value changes elsewhere in the DBFetchGroup,
//   (usually via editing in the UI)
- association:association setValue:(DBValue *)value;

// Called when the DBFetchGroup needs to get a value,
//   e.g., when the associated UI object calls takeValueFrom:
- association:association getValue:(DBValue *)value;
@end

Last Edited -   Mai Nguyen, 17/ February /1993