Giving Interface Builder Demonstrations

Too often, we tend to go through demonstrations very quickly and not explain to the observers exactly what we are doing. This is natural because we have often given the same demonstration 100 times before. Viewers get the impression that it might be a neat system, but they don't understand exactly what happened. The best way to give a demonstration of a complex system to people that have never seen it before is to use the following three steps:

1) Tell the people what you are about to do.
2) Show them how to do it.
3) Tell them exactly what you just did and the implications of it. Rephrase carefully.

The first step prepares the users for what they will be seeing, and the last step allows them to integrate it with the rest of the demonstration. I call this the TELL, SHOW, TELL system. I have found from the feedback in the class evaluations that although it takes a little longer time, it allows users to digest more overall information and they retain it for a longer period of time.

Script

All actions are in roman font, some sample speech in italic.

Select File/New Application.

Drag to the main window: button, text, form, radio buttons, single horizontal slider and row of vertical sliders by using the <Alternate/Drag> function.

The form would be used for your "check book program" and the row of sliders for a "graphic equalizer".

The IB has all the power of a very powerful graphics editor. It can rapidly create user interfaces with automatic alignment tools.

Make the button about 1 inch square and then add a box overlapping the lower right corner of the button. Select the Draw and select the "send to back". Do a shift click on the button, do a "same size", "make pretty", and an "align/column".

But the Interface builder is a lot more then just a graphics editor, because each of the objects on the screen has "brains". Each object knows how to send and receive messages to other objects.

Select the text object and make its font 64 point. Then do a control click and drag the rubber band line around the screen and finally release over the text object. Explain the inspector panel. Remind them that they are inspecting the internal state variables of the slider but by connecting it to the text object, you are looking at what types of messages the text object can receive. Flip the test switch. Move and click all the objects. Show that the buttons "flash", the sliders move and that the slider connected to the text object will send a message of its current value to the text object. Go back to build and bring up the inspector attributes panel. Change the maximum value to 1000, initial value to be 300 and go back into test.

*Remember that **everything** is an object, every window, every menu, every picture and even sound.*

Add a sub-menu and change the text to be "help...", add a keyboard alternate to the menu and then connect it up to a help panel with some text in it.

Add sound to the button.

Close application.

Script for C to F IB demo

Select File/New Application from main menu

Drag 2 forms and one button to main window. (Note that Text objects will not work). Use the "make pretty", or "align" functions.

Choose the Classes command in the Windows selection of the main menu.

Select the Object (.m icon)

Change the name in the lower left be "Calculator"

Select Outlets

Add "inputForm" and "outputForm".

Select Actions

Add "calc:" including the ":".

Do an "Unparse".

Close the Class Editor.

First Connect the Celsius Form to the Calc button by control clicking on the input form and dragging to the button. Select the perform Click: method.

Choose the NewCustom Object command in the main "Objects" menu.

Control-Click the Calc button to the Calculator Object.

Select the Calculator object from the Objects pallet.

Select the Outlets from the inspector.

Select the inputForm from the Outlets list.

Control-Click the Calculator Object and connect it to the inputForm object.

Select the outputForm from the Inspector.

Control-Click the Calculator Object and connect it to the outputForm object.

Do a Save from the main Window menu.

Select Projects and from the Files go to Class Files and double click on the Calculator.m file.

Add the line "#import <appkit/Form.h" to the top of Calculator.m file.

Add the following to the "calc:" method.

```
float degreesF;
[inputForm selectTextAt:0];
degreesF = ((9.0 * [inputForm floatValueAt:0])/ 5.0) + 32.0;
[outputForm setFloatValue:degreesF at:0];
```

Misc

Examples of code pilishing

Change the title of the main window using the inspector.

Customize the Info panel. Add the by ..., version and Icon

Add the carriage return to the Calc button.

For detailed tutorial tell them to see page 7-21 of NeXT Technical Documentation.