Chapter 2
A Tour Of Interface Builder

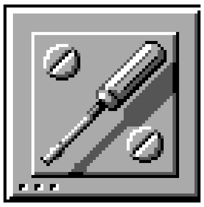
Interface Builder speeds the creation of applications by letting you define an interface (and in some cases, an entire application) graphically rather than by writing Objective-C code. With Interface Builder, you manipulate graphic representations of Application Kit objects just as if you were using a graphics editor to create a drawing.

NeXT System Reference Manual - chapter 8, page 1

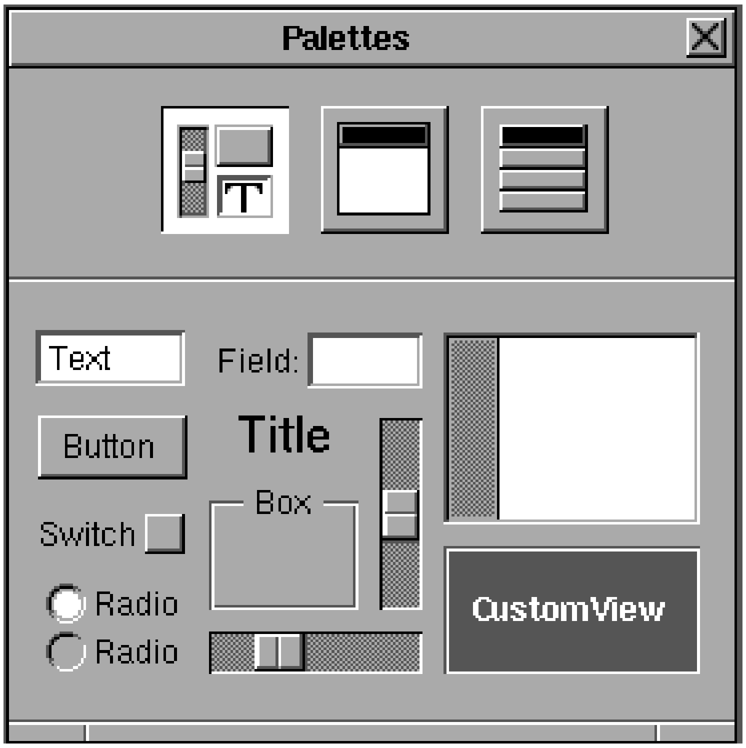
Let's first get a feeling of what it's like to build a user interface using tool-kits. This chapter will give you a short guided tour of the Interface Builder. It assumes you are familiar with a mouse and have had at least a half hour exposure to an icon based computer such as the NeXT or a Macintosh. Keep in mind is that although we will be discussing user interface objects, the ideas of using tool-kits and creating connection based programs are very general and can be applied to many other areas.

The NeXT on-line documentation contains over a hundred page description and tutorial for the Interface Builder. (See /NextLibrary/Documentation/NeXT/SysRefMan/ 08_IntfBuilder.wn) This chapter of this book is meant to complement that information. One of the best ways to learn about the Interface Builder is to start with this chapter and then later go through the on-line documentation for additional details as needed. We will also give you a more general background on encapsulation and inheritance in later chapters, so don't worry if what you are doing when you sub-class an object is not crystal clear.

If you have not done so already, drag the icon form /NextApps/InterfaceBuilder to you application dock. It should look like the following:



Once it is on the dock, double click the mouse button over the icon. The icon will "highlight" itself for less then ten seconds and then the following panel will appear:



This highlighting stage between when you double-click the icon and when the main menu changes is called the "launching" stage of a program. During that time the program is being read off the disk and loaded into memory. If you attempt to do other things during a program's launch stage the system will usually react sluggishly.

The panel above is perhaps the most important single panel you will see in computing in the 1990's. It is much like the painters palette. It contains the elements that you will use in creating your user interface. We will discuss later a t length, how it will affect the way we create programs. In this version of Interface Builder, the top row contains three buttons.

Initially only the first one is high-lighted. This is the "views" panel. It contains all user interface objects that occupy space on the screen.

Try clicking on the object marked "button" or some other objects and then drag them to different parts of the screen. Notice that the selected button will be highlighted but it will always zoom back to the palette. The reason is that the objects have no place to anchor themselves in an application until we create a new application. The new application will be the "canvas" on paint our objects. To do this, we must go to the main Interface Builder menu that has the letters "IB" on the menu title bar. From there we select the "File" menu followed by the "New Application" selection.

After a new application is created, three new panels appear on the screen. The largest is the window titled "My Window" at the top center section of the screen. This is the "canvas" on which we will "paint" our user interface objects. We often refer to this as our "main window" because it is usually the first window to be displayed when a program launches. There are two other panels that also appear. One is a small menu to the left of the screen directly under the main "IB" menu. This is the menu associated with the program we are creating. The last new panel is the one in the lower left corner of the screen. This is usually titled "/PATHNAME/Untitled" where PATHNAME is the path to your home directory. This last panel controls which of the panels on the screen are active. It is the "controller" for the rest of the panels in program and for other Interface Builder tools.

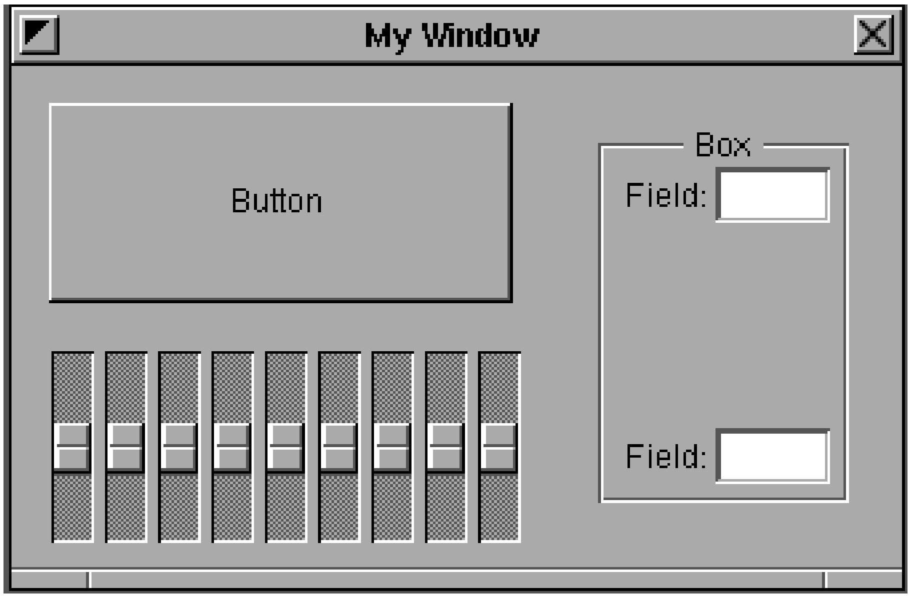
Ignore the menu and controller panel for a moment and go back to our palettes and main window. Add a few objects to our main window by first pointing to the palettes panel and dragging the object labeled "Button" to the upper left corner of the main application window. You will notice that immediately after you let go of the mouse that six "knobs" appear. These knobs are very similar to the knobs used in programs like MacDraw. IB allows you to grab the knobs with you mouse and drag them. This will allow you to re-size the button in any dimension to meet your needs. Try dragging the lower right had corner down and to the right till the button is about two inches wide and an inch high.

Suppose we had sound in our program and we want to create a graphic equalizer to control the bass, mid-range and treble of our sound. The user interface for this might be a row of knobs that slide up and down. We call these objects "sliders". Our left palette (called the View palette) has both vertical and horizontal sliders to choose from. To create a row of vertical sliders we will drag a vertical slider into the main window and place it in the lower left hand corner of the screen. Rather than resizing the slider by dragging the right middle knob, hold the Alternate key down at the same time you drag the knob. You will notice that as the "rubber-band line reaches twice the width of the slider another slider will appear next to the original. As you ALT-DRAG further you can get an entire row of sliders. Create a row that fills about half the screen.

Now let’s say we wanted to create a program that enters names and phone numbers. Lets group them together in a unit so the user will know they are related and so that they can be moved together.

First drag a "Box" object to the upper right corner of the screen. Resize it to fill about one fourth of the main window. Then drag a "form" object, which is a white text box with the label "Field:" to the left of it. While the knobs are still showing on the form, go to the main menu and select "Edit" and then "Copy" (or type COMMAND C). Then select "Paste". You should see two forms on top of each other. Drag the top one to the lower right of the Box object. Then hold down the SHIFT key while you click the initial Field. From the main menu select the Layout, Align, and Make Column options. Your forms should now be directly under each another.

Our last exercise before we test our interface will be to select the edge of the box we created and from the main menu select the "Layout" and "Size to Fit". You should now see you user interface look something like the following with the exception that your interface may have different spacing between the objects:



We have shown that the Interface Builder is a tool that allows you to create user interface objects that are meticulously laid out as fast as you can draw the objects on the screen

But this is only the beginning. The objects that you have placed in the main window are more then just graphics objects. They are software objects. Each one has a specific function One such function would be transferring user-caused events (such as pressing the mouse over a button) to a receiving object which would cause a section of a program to be executed. To show this, lets use the "Test Interface" mode of the Interface Builder. This selection is available from the "File" command of the main menu.

Once you are in the test mode you will find that all of the windows involved in creating the interface have disappeared and the only windows left are the ones that would exist if you were actually running the application. In this case the main menu and the main window.

Try pressing the button. Notice that it will highlight to indicate that it is being pressed. Try to move the sliders in the graphic equalizer. Now try typing text into one of the forms. You should be able to use the COMMAND-C key to copy text from one form into the other. When you are done you can select the quit menu and you will be returned to the build mode of Interface Builder.

Our next step is to make connections between objects. To see the effects of this connection, drag a horizontal slider and a text box into the middle of your main window. Now make a connection from the slider object to the text object. This is done by pointing to the "source object" and CONTROL-DRAGGING a connection to the target object.

After you release the mouse over the destination object an Inspector panel will appear in the lower right corner of the display. This tells you what types of messages the destination object can receive. In this case, select the "takeFloatValue:" message. After you have selected the appropriate item under the label "actions of the destination" you must then press the "Connect" button at the lower right hand corner of the Inspector panel. A small round knob will appear next to the message name to indicate that the connection has been made. Now try testing the interface again using the Test Interface selection of the File menu. When you move the slider the number in the text object should change. Note that the default limits of a slider are zero and one. Select quit when done.

You can change the default values of objects using the inspector panel. By selecting the object that you want to modify and then using the pop-up menu on the inspector and going to the attributes selection you can see the minimum, default and maximum settings on a slider.

Exercises

1. Try selecting other palettes. Add a menu selection and a panel. Have the menu selection send a "orderFront": message to the window.

2. Try changing the text on the button object by double-clicking over the text.

3. Try changing the font of the button object using the "Font" menu. Can you get Greek characters to appear using the symbol font? See the NeXT User Manual for alternate keyboard layouts.

4. Try using all the different layout tools. Note that the "Same Size" will make the second object selected with the SHIFT-click the size of the first object selected.

5*. What other palettes would you like to use? What types of inspectors would they have? What might an object which is a front end to a modem look like? How would you integrate a front end to a SQL database server into IB?

6*. Many people have commented that Interface Builder is as easy to use as a database program with a graphics user interface such as Hypercard(TM) or Supercard(TM). Can you create any arbitrary sets of panels with Interface Builder and have them send messages to any other panels? What are the restrictions? What can you do with databases that you can not do with Interface Builder? What can you do with Interface Builder that you can not do with database programs? What are the basic differences between the **output** of database programs and software development tools like Interface Builder?