;DisconnectingObjects.rtfd;;¬   Next Section   ;IdentifyingObjectsInOutlineMode.rtfd;;¬   Previous Section

Enablinginterfieldtabbing;¬Enabling inter-field tabbing

1 **Control-drag between the window and a view object.**

2 **In the Connections display of the Inspector panel, select initialFirstResponder and click Connect.**

3 **Control-drag between two view objects.**

4 **In the Connections display, select nextKeyView and click Connect.**

In OPENSTEP applications, users can navigate between fields and controls on the interface solely through use of the keyboard. Users can change the first responder by pressing the Tab key or Shift-Tab, can navigate through cells in a matrix by pressing the arrow key, and can change the state of a button or select a cell in a matrix by pressing the Spacebar.

You get most of this keyboard navigation feature in your application for free; you don't have to do anything special to allow users to navigate between cells in a matrix or fields in a form. However, you'll want to control what the Tab key does, that is, which view the cursor should go to next when the user presses the Tab key. You do this by connecting NSView objects to each other through the **nextKeyView** outlet.

First, decide which view should respond to keyboard events when the window becomes key, and connect the NSWindow **initialFirstResponder** outlet to that view.

_EnablingInterField1.eps ¬

Next, use NSView's **nextKeyView** outlet to connect view objects to each other.

_EnablingInterField2.eps ¬

Don't connect views that the user cannot select or edit. In the example above, we skip over the gray text field because it exists to show the result of the Convert button's action. The user cannot enter text into this text field, so it does not make sense to make a **nextKeyView** connection to it. You also should be careful not to connect to NSCell objects. For example, you shouldn't connect to an individual cell of a matrix or form; instead hook the preceding object to the entire matrix or form. The NSMatrix and NSForm objects determine the keyboard navigation between their own cells.

_EnablingInterField3.eps ¬

You should also assign key equivalents to buttons. The default button typically has a Return key equivalent, and the Cancel button typically has the Esc key equivalent. See Chapter 3 for more information. ;../03_SettingObjectAttributes/SettingButtonAttributes.rtfd;;¬

If you don't make **nextKeyView** connections, default connections are made at run time. You can use Interface Builder's Test Interface command to see if these connections are satisfactory. See ªTesting the interfaceº in this chapter. ;TestingTheInterface.rtfd;;¬