

Copyright ©1996 by NeXT Software, Inc. All Rights Reserved.

;A_CurrencyConverter_CreateProject.rtf;;→ Previous Section

;C_CurrencyConverter_Design.rtf;;→ Next Section

2. Currency Converter Tutorial

Creating the Currency Converter Interface

When you create an application project, Project Builder puts the *main nib file* in the Interfaces suitcase. A nib file is primarily a description of a user interface (or part of a user interface). The main nib file contains the main menu and any windows and panels you want to appear when your application starts up; at start-up time, each application loads the main nib file.

At the beginning of a project, the main nib file is like a blank canvas, ready for you to craft the interface. Look in the Interfaces suitcase for nib files.

1 Open the main nib file.

Locate CurrencyConverter.nib in the project browser.

Double-click to open.

_PBtoIB_fullscr.eps →

A nib file contains user-interface objects, definitions of custom classes, the connections between objects, and sounds and images that are used in the interface. Besides the main nib file, you can have nib files that you can load whenever you need them. These auxiliary nib files, and the techniques related to using them, are described in the ^aTo Do Tutorial.^o See *OpenStep Development: Tools and Techniques* for an overview of nib files.

By default, the window entitled ^aMy Window^o will appear when the application is launched.

Note: The Interface Builder application is located in */NextDeveloper/Apps*. The icon for the application is this:

IB_Icon.tiff →

2 **Resize the window.**

`_IB_ResizeWin.eps` ↪

Most objects on an interface have attributes that you can set in the Inspector panel's Attributes display.

3 **Set the window's title and attributes.**

Click the window to select it.

Choose Tools *arrow.eps* ↪ Inspector.

Select the Attributes display from the pop-up list.

Enter the window title.

Turn off the resize option.

`_IB_WindowAttr.eps` ↪

Related Concept: ;CurrencyConverterConcepts.rtf;linkMarkername AWindowinOpenStep;, A Window in OpenStep

Put palette objects on the window using the ^adrag and drop^o technique.

4 **Put a text field on the interface and resize and initialize it.**

Select the Views palette.

Drag a text field from the palette onto the window.

`_IB_GetField.eps` ↪

To initialize the text field, double-click ^aText^o and press Delete.

You must get rid of the word ^aText^o in this field; otherwise, that's what the field will show when the nib file is loaded. The text field should be longer so it can hold more digits (you're dealing with millions here):

Lengthen the text field.

_IB_Size2.eps ↵

Currency Converter needs two more text fields, both the same size as the first. You have two options: you can drag another object from the palette and make it the same size; or you can duplicate the first object

5 Duplicate an object.

Select the text field.

Choose Edit *387476_arrow.eps* ↵ Copy.

Choose Edit *515231_arrow.eps* ↵ Paste.

_CC_Duplicate.eps ↵.

Get the third field from the palette and make it the same size as the first field.

6 Make objects the same size.

Drag a text field onto the window.

Delete ^aText^o from the text field.

Select the first text field.

Shift-click to select the new text field.

Choose Format *646672_arrow.eps* ↵ Size *759658_arrow.eps* ↵ Same Size.

_IB_SameSize.eps ↵

You're not done yet with these text fields. The bottom text field displays the result of the computation. It should not be editable and therefore should, by convention, have a gray background.

7 Change the attributes of a text field.

Select the third text field.

Choose Tools *327446_arrow.eps* ↵ Colors

Select the grayscale palette of the Color panel.

Select the gray color that is the same as the window background.
Choose Tools *515678_arrow.eps* → Inspector.
Select the Inspector panel's Attributes display.
Drag the gray color from the Color panel into the Background Color well.
Turn off the Editable and Scrollable options.

`_IB_TextFieldInspector.eps` →:

The Views palette provides a `^Title^` object that you can easily adapt to be a text-field label. (The title object is actually a text field, set to have a gray background and no border, and to be non-editable and non-selectable.) Text in the title object is centered by default, but labels are usually aligned from the right.

8 **Assign labels to the fields.**

Drag a title object onto the window.
Double-click to select the text `^Title^`.
Choose Format *928125_arrow.eps* → Text *384460_arrow.eps* → Align Right to align the text from the right.

`_IB_AlignTitleRight.eps` →

The size of the text is rather large for a label, so change it. You set font family, typeface, and size with the standard OpenStep Font panel.

Make sure the object's text is selected.
Choose Format *213413_arrow.eps* → Font *335927_arrow.eps* → Font Panel.
Set the label text to 16 points.
Make two copies of the label.
Position all labels to the left of their text fields.

`_IB_FontPanel.eps` →

When you cut and paste objects that contain text, like these labels, the object should be selected

and not the text the object contains; if the text is selected, de-select it by clicking outside the text, then click the object again to select it.

Type the text of each label.

`_IB_InitText.eps` ↵

9 Add a button to the interface and initialize it.

Drag the button object from the Views palette and put it on the lower-right corner of the window.

Make the button the same size as a text field.

Change the title of the button to `Convert`.

`_IB_ButtonText.eps` ↵

You can easily give the button the capacity for responding to carriage returns in addition to mouse clicks.

Select the Images display of the nib file window.

Drag the `NSReturnSign` image to the main window and drop it over the button.

`_IB_NSReturnSign.eps` ↵

If you check the attributes of the button in the Inspector panel, you'll notice two things have been added: `NSReturnSign` is now listed as the button's icon, and the Key field contains the escape sequence for a carriage return (`\r`).

You've probably noticed that the final interface for Currency Converter (shown on the first page of this chapter) has a decorative line between the text fields and the button. This line is easy to make.

10 Create a horizontal decorative line.

Drag a box object from the Views palette onto the interface.

Bring up the Attributes display for the box (Command-1), select No Title, and set the Vertical Offset to zero.

Drag the bottom-middle resize handle of the box upward until the horizontal lines meet.

Position the line above the button.

Drag the end points of the line until the line extends across the window.

`_IB_MakeLine.eps` ↪

As you might have noticed, the Currency Converter has a main menu that holds, by default, the commands Info, Hide, and Quit, and the Edit, Services, and Windows menus. The menus contain ready-made sets of commands. The Edit menu includes commands for cutting, copying, and pasting text. The Windows menu lists the titles of open windows as well as common window commands. The Services menu allows your application to communicate with other applications, often with no work on the part of your application. For example, if your application handles text, you can use the Services menu to transfer information to other applications that accept text.

Currency Converter's interface is almost complete. One finishing touch might be to align the text fields and labels in neat rows and columns. Interface Builder gives you several ways to align selected objects on a window.

`SquareBullet.eps` ↪ Dragging objects with the mouse

`389349_SquareBullet.eps` ↪ Pressing arrow keys (with the grid off, the selected objects move one pixel)

`958825_SquareBullet.eps` ↪ Using a reference object to put selected objects in rows and columns

`354228_SquareBullet.eps` ↪ Specifying origin points in the Size display of the Inspector panel

`844511_SquareBullet.eps` ↪ Using a grid (see preceding side bar)

For Currency Converter, use the columns-and-rows technique.

11 Align the text fields and labels in rows and columns.

Select the three text fields and choose `Format 148744_arrow.eps` ↪ `Align 361422_arrow.eps` ↪ `Make Column`.

Select the first text field and its label and choose `Format 700964_arrow.eps` ↪ `Align 82083_arrow.eps` ↪ `Make Row`.

Repeat the last step for the second and third text fields and their labels.

_IB_MakeColRow.eps ↵

The **nextKeyView** variable is an outlet. An outlet is the identifier of an object that another object stores as an instance variable. Outlets enable communication between objects. See the concepts in the "Travel Advisor Tutorial" for more on outlets.

The final step in composing the Currency Converter interface has more to do with behavior than appearance. You want the user to be able to tab from the first editable field to the second, and back again to the first. Many objects on Interface Builder's palettes have an instance variable named **nextKeyView**. This variable identifies the next object to receive keyboard events when the user presses the Tab key (or the previous object if Shift-Tab is pressed). If you want inter-field tabbing you must connect fields through the **nextKeyView** variable.

Related Concept: ;CurrencyConverterConcepts.rtf;linkMarkername AligningonaGrid;, Aligning on a Grid

12 Enable tabbing between text fields.

Select the first text field.

Control-drag a connection line from it to the second text field.

In the Inspector panel (Connections display) select **nextKeyView** and click Connect.

Repeat the same procedure, going from the second to the first field.

_IB_nextText_Comp.eps ↵

When you make a visual connection such as this, Interface Builder brings up the Connections display of the Inspector panel:

_IB_nextKeyViewConnect.eps ↵

Don't connect the **nextKeyView** outlet of the ^aAmount in Other Currency^o field; this field is not supposed to be editable.

The CurrencyConverter interface is now complete. Interface Builder lets you test an interface without having to write one line of code.

13 Test the interface.

Choose Document *656985_arrow.eps* → Save to save Interface to the nib file.

Choose Document *936858_arrow.eps* → Test Interface.

Try various operations in the interface (to see some suggestions, click here

;CurrencyConverterConcepts.rtf;linkMarkername AOpenStepApplicationÑWhatYouGetªForFreeº;,,).

When finished, choose Quit from the main menu.

Note: You can also exit from test mode by double-clicking the Interface Builder icon, which changes to the following image to represent test mode:

simulate_nib.tiff →

Related Concepts: ;CurrencyConverterConcepts.rtf;linkMarkername AOpenStepApplicationÑWhatYouGetªForFreeº;, An OpenStep Application ð What You GetªFor Freeº

;CurrencyConverterConcepts.rtf;linkMarkername AOpenStepApplicationÑThePossibilities;,, An OpenStep Application ð The Possibilities