

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

;E\_CurrencyConverter\_Implement.rtf;linkMarkername ;→ Previous Section

## 2. Currency Converter Tutorial

# Building the Currency Converter Project

The Build process in Project Builder compiles and links the application guided by the information stored in the project's makefiles. You must begin builds from the Project Build panel.

### 1 Build the project.

Save source code files and any changes to the project.

Click the Build button on the main window (icon below).

Click the Build button on the Project Build panel (same icon).

build.tiff →

When you click the Build button on the main window, the Project Build panel is displayed.

\_PB\_BuildSuccess.eps –

You don't have to maintain makefiles in Project Builder. It updates **Makefile** according to the variables specified through its user interface. You can customize the build process by modifying the **Makefile.preamble** and **Makefile.postamble** files. For more information on customizing these files, see *OpenStep Development: Tools and Techniques* (which doubles as Help for Interface Builder and Project Builder).

When you click the Build button on the Project Build panel, the build process begins; Project Builder logs the build's progress in the lower split view. When Project Builder finishes and encounters no errors along the way, it displays "Build succeeded."

**Related Concept:** [CurrencyConverterConcepts.rtf](#); linkMarkername

WhatHappensWhenYouBuildanApplication;, What Happens When You Build an Application

Of course, rare is the project that is flawless from the start. Project Builder is likely to catch some errors when you first build your project. To see the error-checking features of Project Builder, introduce a mistake into the code.

## 2 Build the project after correcting errors.

Delete a semicolon in the code, creating an error.

Click the Build button on the Project Build panel.

Click the error-notification line that appears in the build-error browser (upper split view).

Fix the error in the code.

Rebuild.

\_PB\_BuildError.eps ↗

**Related Concept:** ;CurrencyConverterConcepts.rtf;linkMarkername  
WhereToGoForHelp;, Where To Go For Help

# Run Currency Converter

Congratulations. You've just created your first OpenStep application. Find **CurrencyConverter.app** in the Workspace, launch it, and try it out. Enter some rates and dollar amounts and click Convert. Also, select the text in a field and choose the Services menu; this menu now lists the other applications that can do something with the selected text.

You can use Project Builder's graphical debugger or **[gdb](#)** to track bugs down. See **[TravelAdvisorConcepts.rtf](#)** for an overview of the graphical debugger.

Of course, the more complex an application is, the more thoroughly you will need to test it. You might discover errors or shortcomings that necessitate a change in overall design, in the interface, in a custom class definition, or in the implementation of methods and functions.

Although it's a simple application, Currency Converter still introduced you to many of the concepts, tools, and skills you'll need to develop OpenStep applications. Let's review what you've learned:

- SquareBullet.eps ↪ Composing a graphical user interface (GUI) with Interface Builder
- 70256\_SquareBullet.eps ↪ Testing the interface
- 388343\_SquareBullet.eps ↪ Designing an application using the Model-View-Controller paradigm
- 608989\_SquareBullet.eps ↪ Specifying a class's outlets and actions
- 972176\_SquareBullet.eps ↪ Connecting the class instance to the interface via its outlets and actions
- 295562\_SquareBullet.eps ↪ Class implementation basics
- 674888\_SquareBullet.eps ↪ Building an application and error resolution

***618700\_TableRule.eps ↪ Optional Exercise***

**Nesting Messages:** You can nest message expressions; in other words, you can use the value returned by a message as the receiver of another message or as a message argument. It is thus possible to rewrite the first three messages of the ConverterController's **convert:** method as one statement:

```
total = [converter convertAmount:[dollarsField floatValue]  
        byRate:[rateField floatValue]];
```

It is possible to go even further. Try to incorporate the fourth message (**[totalField setFloatValue:total]**) of the **convert:** method into the above statement.

951340\_TableRule.eps ⇐