

2. Currency Converter Tutorial

Creating the Currency Converter Project

Every OpenStep application starts out as a *project*. A project is a repository for all the elements that go into the application, such as source code files, makefiles, frameworks, libraries, the application's user interface, sounds, and images. You use the Project Builder application to create and manage projects.

1 Launch Project Builder.

In File Viewer navigate to the **/NextDeveloper/Apps** directory.
Select **ProjectBuilder.app** and double-click its icon.

PBappicon.tiff ↪

When Project Builder starts up, only its main menu appears on the screen. You must create or open a project to get Project Builder's main window. The New Project panel allows you to specify a new project's name and location.

2 Make a new project.

Choose New from the Project menu (Project *arrow.eps* ↪ New).
In the New Project panel, select the project location.
Enter "CurrencyConverter" as the project name.
Click OK to create the project.

_PB_NewProjectPanel.eps ↪

Project Builder creates a project directory named after the project—in this case CurrencyConverter—and populates this directory with an assortment of ready-made files and directories. It then displays its main

window.

Note: Here's a variation on project creation: Create a project directory using File Viewer and then, in the New Project panel, navigate to that directory, type `^PB.project^` in the Name field, and click OK.

_PB_Startup.eps ↪

Go ahead and click an item in the left column of the project browser (a grouping of project resources sometimes called a `^suite^`); see what some of these suites contain already:

Other Sources: This suite contains **CurrencyConverter_main.m**, the **main()** routine that loads the initial set of resources and runs the application. (Do not modify this file!)

Interfaces: This suite contains **CurrencyConverter.nib**, the file that contains the application's user interface. More on this file in the next step.

Supporting Files: This suite contains the project's default makefiles and template source-code files. You can modify the preamble and postamble makefiles, but leave **Makefile** unchanged.

*A makefile specifies file dependency relations and compiler and linker instructions for building the project. See [OpenStep Development: Tips and Techniques](#) for common changes to **Makefile.preamble** and **Makefile.postamble**.*

Related Concept: ;CurrencyConverterConcepts.rtf;linkMarkername ProjectIndexing;, Project Indexing