

Introduction to Xcode

Author: Mark Szymczyk

Last Update: January 14, 2010

This article walks you through your first Xcode project. You will learn how to create a project, edit source code, and build your project into a working program.

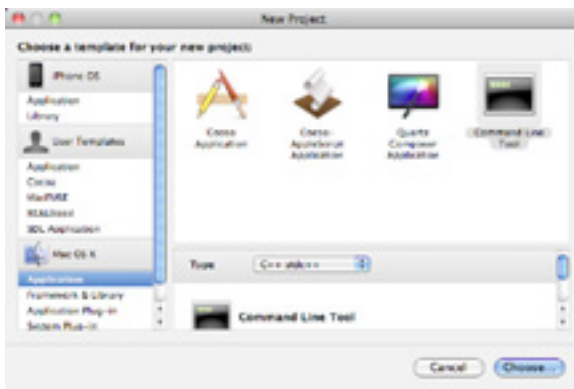
Updates

Updated the article for Xcode 3.2. (January 14, 2010)

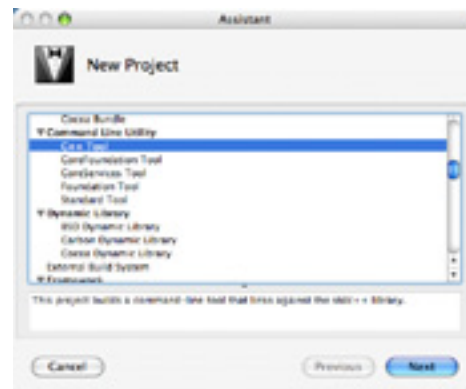
Updated the article for Xcode 3. (March 4, 2008)

Creating the Project

The first step you must take to use Xcode is to create a project. A project contains all the files needed to create a working program. Choose File > New Project to create a new project. A window opens with a list of project types you can create.



Xcode 3.2



Xcode 3.0 and earlier

If you're running Xcode 3.2, select Application under Mac OS X on the left side of the New Project Assistant. Select Command Line Tool. Choose C++ from the Type pop-up menu, and click the Choose button. If you're running an earlier version of Xcode, select a C++ Tool project, which is in the Command Line Utility section, and click the Next button.

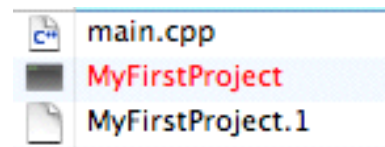
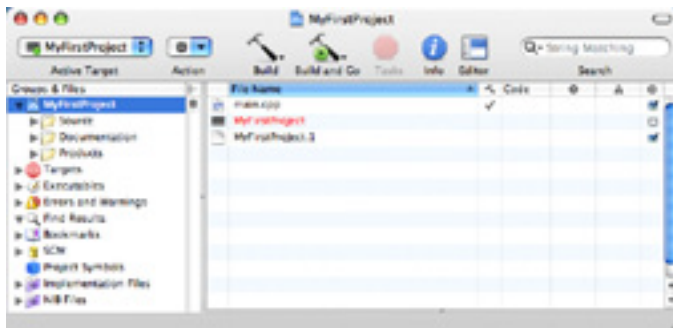
If you're running Xcode 3.1, the New Project Assistant looks similar to the one in Xcode 3.2, but the project choices are similar to the ones in Xcode 3.0. You want a C++ Tool project if you're running Xcode 3.1.

Name your project and pick a location to save it on your hard disk. I used the name MyFirstProject. Click the Save button (or Finish button on some versions of Xcode) to finish creating the project. At this point, you've created your first Xcode project.

Tool projects create programs that do not have a graphical user interface. They are meant to be run from the command line using the Terminal application. Tool projects are simpler to write than regular Mac OS X applications, which is why I'm using a tool project.

Editing Source Code

After creating the project, the project window opens. You should see a list of files in the center of the window. If you don't see a list, select the name of the project from the Groups and Files list on the left side of the window. The project name should be the first entry in the list.



The file `main.cpp` contains the source code for the project. Double-click `main.cpp` to open it in an editor window. The editor window is where you write the source code for your applications. You should see the following line of code:

```
std::cout << "Hello, World!\n";
```

Change this line to the following:

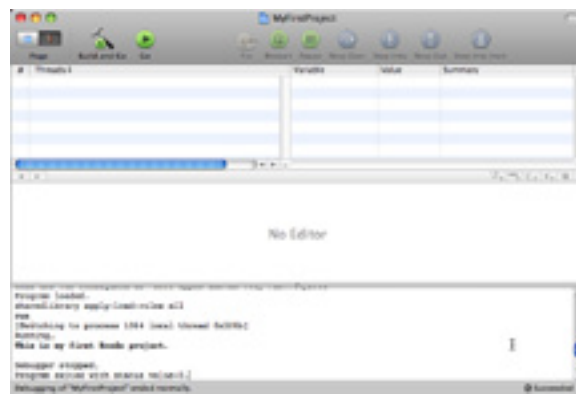
```
std::cout << "This is my first Xcode project.\n";
```

Save the file.

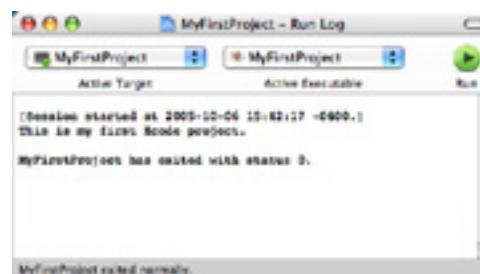
Building the Project

Now it's time to build a working program from the code we wrote. Xcode calls the process building. When Xcode builds your project, it compiles the source code files in your project and links any frameworks and libraries included in the project to create an application. Our project is very simple with no frameworks to link, which makes building the project easy.

Choose Build > Build and Run. Xcode will build the project and run it. If you're running Xcode 3 or later, you must open the debugger's console to see the output of your program. Choose Run > Console to open the console, and you should see the message `This is my first Xcode project`. Choose Run > Run to run your program again.



If you're running an older version of Xcode, Xcode opens a run log when you choose Build > Build and Run. The run log is where you can see the project's output. If you look at the run log, it should say `This is my first Xcode project`. Click the Run button in the run log toolbar to run the program again.



Conclusion

Let's review what you've learned. You learned how to create a project, open an editor window, edit source code, build your project, and run your program. At this stage you're ready to read a programming book and work through the examples in the book. Create a new project for each example.

I used a C++ Tool project for this article, but Xcode provides tool projects for other languages. If you're using Xcode 3.2, use the Type pop-up menu in the New Project Assistant to pick another language: C or Objective-C. Choose Foundation to learn Objective-C. For those of you running older versions of Xcode, pick a Standard Tool project to learn C and a Foundation Tool project to learn Objective-C.