1

Getting to Know Your Tools

They say that when a craftsman finds a tool that he loves, over time it becomes an extension of him. He learns its idiosyncrasies inside and out, backwards and forwards, and this leads to a relationship that transcends simple use of the tool and instead becomes more involved. The tool becomes an extension of his hand, an extension that he can guide with an almost extrasensory vision.

You are fortunate then, that Apple provides you with an excellent set of free tools for developing software for the iPhone. They are tools that have evolved over the last 15 years of Objective-C development, first on NeXT computers, then on Mac OS X, and now for iPhone.

In this chapter, you will take a brief look at these tools and learn where you can find more information about them. They are incredibly powerful tools that seem to be unique in software development, both for their ability to provide enough power to enable incredibly complex software systems to be developed, and also because they seem to know just when to stay out of your way and simply provide a great text-editing environment for you to write code in.

Unfortunately, a comprehensive description of every last feature of these applications is beyond the scope of this book, and so I won't be delving into them in great detail. My main goal here is simply to introduce you to the tools so that you're familiar with them.

1 In This Chapter

Becoming familiar with Xcode and Interface Builder

Introducing Xcode

The cornerstone of iPhone software development is the Xcode integrated development environment, or IDE. Xcode originated on NeXT Step computers as Project Builder. Over the years, it has gone through many revisions to finally arrive at the version that is available to you today. It uses GCC as its underlying compiler technology and provides many sophisticated features found in modern IDEs today, such as code completion, re-factoring, and sophisticated code navigation. Interestingly, it also has one of the best cross-platform compiling capabilities of any modern IDE. With it, you can compile for Intel, PowerPC, iPhone OS, or even (with third-party tools) Microsoft Windows. You can do all of this simply by configuring targets in the IDE.

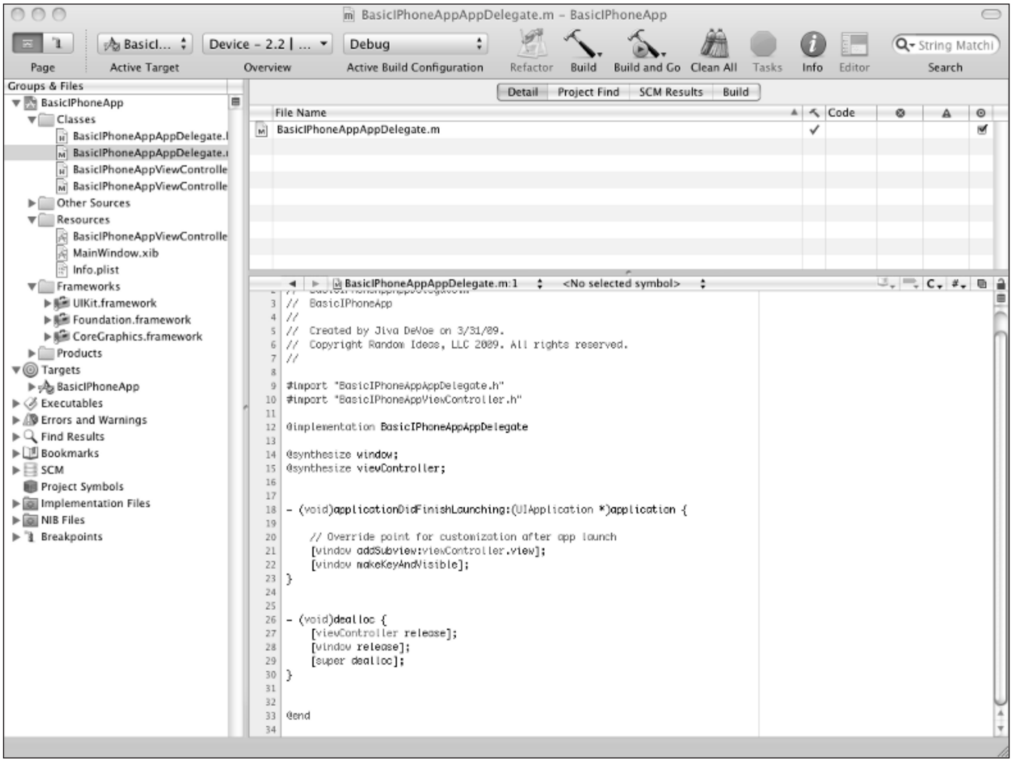
NOTE

To download Xcode, all you need to do is sign up for a free developer account on the Apple Web site. You can do this at `http://developer.apple.com/iphone`.

Figure 1.1 shows the main Xcode window. In it, you can see the left panel, which shows the file organization view. From here, you can drag and drop files into your projects or organize them by groups. It also provides the ability to organize your files by Smart groups, which are built using search queries and can be useful for looking for particular files in your projects.

Figure 1.1

The Xcode interface



```
//  BasicIPhoneApp
//
//  Created by Jiva DeVoe on 3/31/09.
//  Copyright Random Ideas, LLC 2009. All rights reserved.
//

#import "BasicIPhoneAppAppDelegate.h"
#import "BasicIPhoneAppViewController.h"

@implementation BasicIPhoneAppAppDelegate

@synthesize window;
@synthesize viewController;


- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch
    [window addSubview:viewController.view];
    [window makeKeyAndVisible];
}


- (void)dealloc {
    [viewController release];
    [window release];
    [super dealloc];
}


@end
```

On the right side, you can see the main editing window. It is in this window that you will do the majority of your programming work. At the top of the text-editing window, you can see some drop-down menus that enable you to quickly jump to any recently opened files. The second drop-down menu from the left enables you to quickly jump to any method in the current file. Holding down the ⌘ key and using the right and left arrow keys enables you to quickly navigate backwards and forwards through the file history. Additionally, holding down the Option and ⌘ keys together and pressing the up arrow key enables you to quickly swap between the implementation and header files for the currently active compilation unit.

The Xcode editor is quite sophisticated and can be configured with a variety of shortcuts and hot-keys that make your editing much easier and faster. For example, using the Option key and the right and left arrow keys enables you to quickly jump from word to word in your code. Holding down the ⌘ key and using the left and right arrow keys enables you to quickly jump to the front or beginning of the current line.

It's a good idea to learn the intricacies of the Xcode editor inside and out, because it is such a fundamental tool to everything that you will be doing as an iPhone OS developer.

Introducing Interface Builder

The second major component of the Xcode programming environment is the graphical user interface (GUI) builder called Interface Builder, shown in Figure 1.2. It is this application that you will use to draw your GUI for your application and connect your buttons to actions in your code.

Many developers coming to iPhone development from other environments are sometimes confused by Interface Builder because it doesn't generate any code. However, this is an asset and not a liability. IDEs that simply generate code tend to be more difficult to work with over time, as the code that they generate becomes out of sync with the user interface. Interface Builder uses more of a metadata style approach. This means that you tell it that you want to instantiate an object of a given type, and when your nib is loaded, it goes and finds the class for that type, instantiates it, and attaches the outlets and actions that you have configured to the appropriate places. It does not serialize actual instances of your objects, nor does it generate code that is compiled.

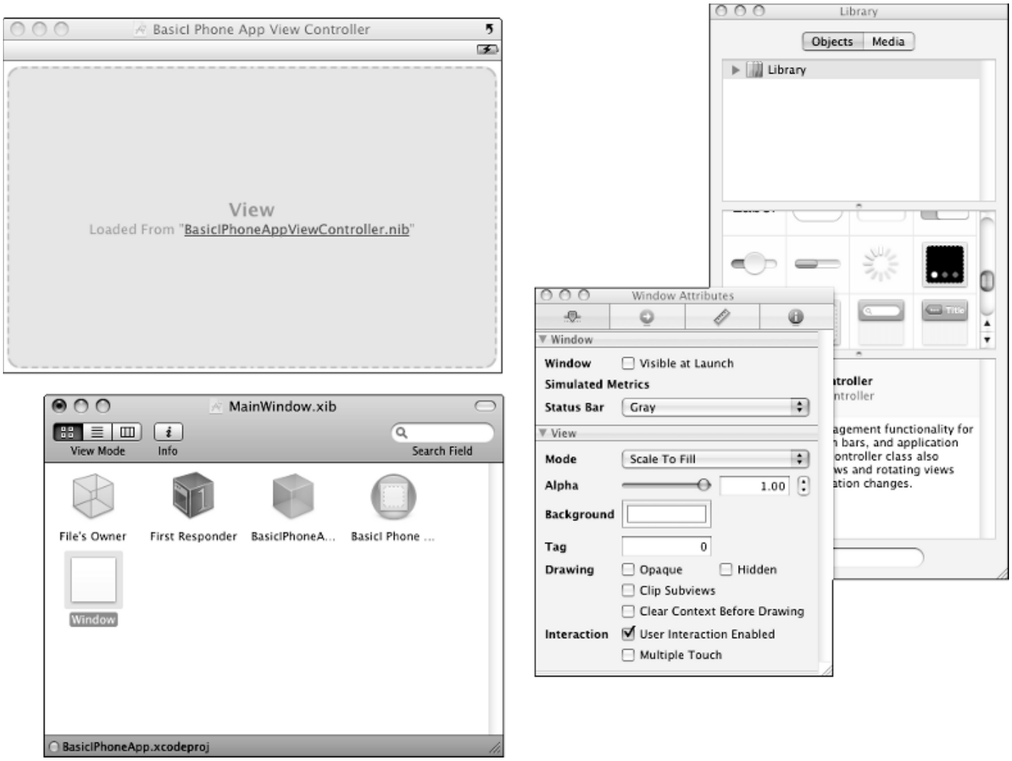
NOTE

The files that Interface Builder saves are referred to as nib files. This stands for NeXT Interface Builder. The file extension of the nib files used with the iPhone is .xib. This is to differentiate them as containing XML versus the older, original NeXT format.

I will talk about Interface Builder in a bit more detail in Part II. For now, the important thing to know is that you can start Interface Builder either separately by simply launching it or by double-clicking any of the `.xib` files in your project.

Figure 1.2

Interface Builder



Window
Window □ Visible at Launch
Simulated Metrics
Status Bar: Gray
View
Mode: Scale To Fill
Alpha: 1.00
Background
Tag: 0
Drawing □ Opaque □ Hidden
□ Clip Subviews
□ Clear Context Before Drawing
Interaction ☑ User Interaction Enabled
□ Multiple Touch

Summary

In this chapter, I introduced you to Xcode and Interface Builder. In the upcoming chapters, you will use these tools in much more depth, but I wanted to simply give you a brief overview of what they look like so that when you use them in the future you will be familiar with them. I encourage you to read through the documentation on them very carefully and try to become as intimately familiar with them as you can.