

---

# Apple Watch 开发

## 快速入门教程 ( Objective-C 版 )

( 内部资料 )



大学霸

[www.daxueba.net](http://www.daxueba.net)



---

# 前言

Apple Watch 是苹果公司推出的一款可穿戴的智能手表。它支持电话、语音回电话、连接汽车，天气、航班信息，地图导航，播放音乐，测量心跳、计步等几十种功能，是一款全方位的健康和运动追踪设备。Watch 搭配 iPhone 手机，将给用户带来全新的体验。

苹果为 Watch 提供全新的开发框架 WatchKit。本教程是国内第一本 Apple Watch 开发教程，全面讲解 WatchKit 的各个方面，帮助读者更为快速的掌握 Watch 开发方式和流程。

## 1.学习所需的系统和软件

- ☐ Mac OS 10.10.3
- ☐ Xcode 6.4

## 2.学习建议

大家学习之前，可以致信到 [xxxxxxx](mailto:xxxxxxx)，获取相关的资料 and 软件。如果大家在学习过程遇到问题，也可以将问题发送到该邮箱。我们尽可能给大家解决。



---

# 目 录

第 1 章	开发环境——Xcode.....	1
1.1	苹果账号 .....	1
1.1.1	苹果账号的成员分类 .....	1
1.1.2	注册免费的苹果账号 .....	1
1.1.3	注册收费的苹果账号 .....	5
1.2	Xcode 的下载和安装.....	7
1.2.1	App Store 中下载和安装 Xcode.....	8
1.2.2	其他网站下载和安装 Xcode.....	10
1.3	苹果账号的绑定 .....	12
1.4	组件的下载和安装 .....	13
1.5	第一次打开 Xcode.....	14
1.6	Xcode 的界面介绍.....	16
1.6.1	导航窗口 .....	17
1.6.2	组件窗口 .....	18
1.6.3	编辑窗口 .....	19
1.6.4	目标窗口 .....	19
第 2 章	编写第一个 iWatch 应用程序.....	21
2.1	iWatch 介绍.....	21
2.2	iWatch 的优点缺点.....	21
2.2.1	iWatch 优点.....	21
2.2.2	iWatch 缺点.....	24
2.3	Apple Watch 1.0.1 新增内容 .....	24
2.4	开发 iWatch 应用程序使用的类.....	25
2.5	编程第一个程序 .....	25
2.5.1	创建项目 .....	25
2.5.2	添加 Watch 应用对象到项目中.....	27
2.5.3	添加 Watch 应用对象时新增内容介绍 .....	29
2.5.4	运行程序 .....	30
2.5.5	编辑界面 .....	32
2.5.6	编写代码 .....	35
2.6	调试程序 .....	38
2.7	使用帮助文档 .....	41
第 3 章	界面设计一 .....	43
3.1	标签 .....	43
3.1.1	添加标签到界面中 .....	43
3.1.2	界面方式设置标签 .....	47

3.1.3	使用代码设置标签 .....	57
3.1.4	为标签加载自定义的字体 .....	70
3.2	图像 .....	76
3.2.1	添加图像到界面中 .....	76
3.2.2	界面方式设置图像对象 .....	77
3.2.3	使用代码设置图像对象 .....	82
3.3	分割线 .....	87
3.3.1	添加分割线到界面中 .....	88
3.3.2	界面方式设置分割线 .....	88
3.3.3	使用代码设置分割线 .....	89
第 4 章	界面设计二 .....	91
4.1	表 .....	91
4.1.1	添加表到界面中 .....	91
4.1.2	界面方式设置表对象 .....	92
4.2	填充表 .....	94
4.2.1	填充文本内容 .....	94
4.2.2	填充图像 .....	98
4.3	对表进行操作 .....	101
4.3.1	响应选择的行 .....	101
4.3.2	添加行 .....	102
4.3.3	删除行 .....	104
4.3.4	获取行数 .....	105
4.3.5	滚动行 .....	106
4.4	组对象 .....	108
4.4.1	添加组对象到界面中 .....	108
4.4.2	界面方式设置 Group 对象 .....	108
4.4.3	使用代码设置 Group 对象 .....	111
第 5 章	用户交互 .....	116
5.1	开关 .....	116
5.1.1	添加开关对象到界面中 .....	116
5.1.2	界面方式设置开关对象 .....	117
5.1.3	使用代码设置开关对象 .....	119
5.1.4	响应开关对象 .....	124
5.2	滑块 .....	129
5.2.1	添加开关对象到界面中 .....	130
5.2.2	界面方式设置滑块对象 .....	131
5.2.3	使用代码设置滑块对象 .....	134
5.2.4	响应滑块对象 .....	137
5.3	按钮 .....	138
5.3.1	添加按钮对象到界面中 .....	139
5.3.2	界面方式设置滑块对象 .....	140

5.3.3	使用代码设置按钮对象 .....	141
5.3.4	响应按钮对象 .....	144
第 6 章	日期时间和计时器 .....	147
6.1	日期时间 .....	147
6.1.1	添加日期时间对象到界面中 .....	147
6.1.2	界面方式设置日期时间对象 .....	148
6.1.3	使用代码设置日期时间对象 .....	151
6.2	计时器 .....	155
6.2.1	添加计时器到界面中 .....	155
6.2.2	界面方式设置计时器对象 .....	156
6.2.3	使用代码设置计时器对象 .....	158
6.3	实现计时功能 .....	160
6.3.1	使用界面方式计时 .....	160
6.3.2	使用代码计时 .....	161
第 7 章	地图与菜单 .....	164
7.1	添加地图到界面中 .....	164
7.2	界面方式设置地图对象 .....	165
7.3	设置地图的显示区域 .....	166
7.4	添加/删除标注 .....	167
7.4.1	添加系统自带的标注 .....	167
7.4.2	添加自定义的标注 .....	169
7.4.3	删除标注 .....	170
7.5	获取当前位置 .....	171
7.6	添加菜单到界面中 .....	178
7.7	设置菜单项 .....	179
7.7.1	界面方式设置菜单项 .....	179
7.7.2	使用代码设置菜单项 .....	182
7.8	添加菜单项 .....	183
7.8.1	界面方式添加菜单项 .....	183
7.8.2	使用代码添加菜单项 .....	185
7.9	响应菜单项 .....	186
7.9.1	界面方式添加菜单项实现的响应 .....	186
7.9.2	使用代码添加菜单项实现的响应 .....	190
7.10	删除菜单项 .....	193
第 8 章	三种界面控制器 .....	195
8.1	界面控制器 .....	195
8.1.1	添加界面控制器 .....	195
8.1.2	界面方式设置界面控制器的界面 .....	197
8.1.3	使用代码设置界面控制器的界面 .....	200
8.2	浏览界面控制器 .....	201
8.2.1	添加浏览界面控制器 .....	202

8.2.2	界面方式设置浏览界面控制器的界面 .....	205
8.3	通知界面控制器 .....	208
8.3.1	添加通知界面控制器 .....	208
8.3.2	界面方式设置通知界面控制器的界面 .....	211
8.4	界面导航 .....	215
8.4.1	使用拖动方式 .....	215
8.4.2	使用代码方式 .....	223
第 9 章	本地化 .....	233
9.1	创建一个具有多种语言的应用程序 .....	233
9.2	本地化资源 .....	242
9.3	区域格式 .....	244
第 10 章	高级应用 .....	249
10.1	九宫格 .....	249
10.2	音频播放器 .....	255
10.3	天气预报 .....	260
10.4	iWatch 控制 iPhone .....	263



# 第 1 章 开发环境——Xcode

开发环境就是为了支持系统软件和应用软件工程化开发和维护的一组软件。它通常简称为 SDE。每一种软件的开发都需要使用一个特定的开发环境，例如 Windows Phone 的开发环境就是 Visual Studio。Xcode 则是开发 iWatch、iPhone 等应用程序的开发工具。本章将讲解苹果账号的注册、Xcode 的下载安装以及使用等内容。

## 1.1 苹果账号

苹果账号是苹果公司专门为苹果开发人员提供的一种可以用来代表身份的账号。苹果账号必须是一个邮箱帐号，并在苹果网站上注册、验证通过。通过苹果账号开发者可以在 App Store 中对所有的软件进行下载。本节将讲解苹果账号的分类以及注册等内容。

### 1.1.1 苹果账号的成员分类

在苹果公司注册苹果账号，就可以成为开发成员。开发成员一共可以分为四种，如表 1-1 所示。

表 1-1 苹果账号的成员

成员类型	成本
在线开发成员	免费
标准iOS开发成员	\$99/年
企业iOS开发成员	\$299/年
大学iOS开发成员	免费

### 1.1.2 注册免费的苹果账号

以下是注册一个免费的苹果账号的具体步骤：

(1) 在 Dock（Dock 一般指的是苹果操作系统中的停靠栏）中，找到浏览器 Safari，如图 1.1 所示。

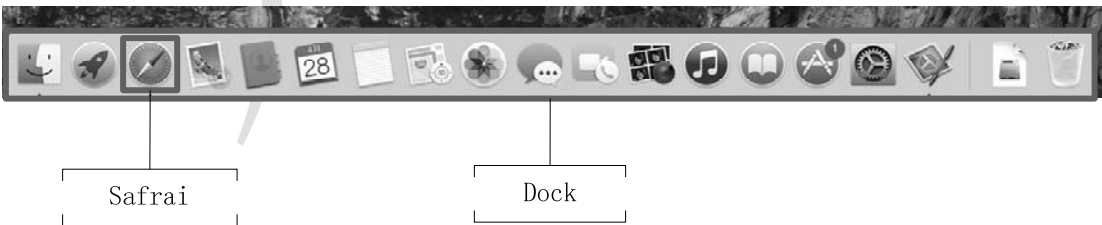


图 1.1 浏览器 Safari

(2) 单击打开 Safari 图标，打开 Safari 浏览器，如图 1.2 所示。



图 1.2 浏览器 Safari

(3) 在地址栏中输入网址（<https://developer.apple.com/devcenter/ios/index.action>），按下回车，进入 iOS Dev Center-App Developer 网页，如图 1.3 所示。

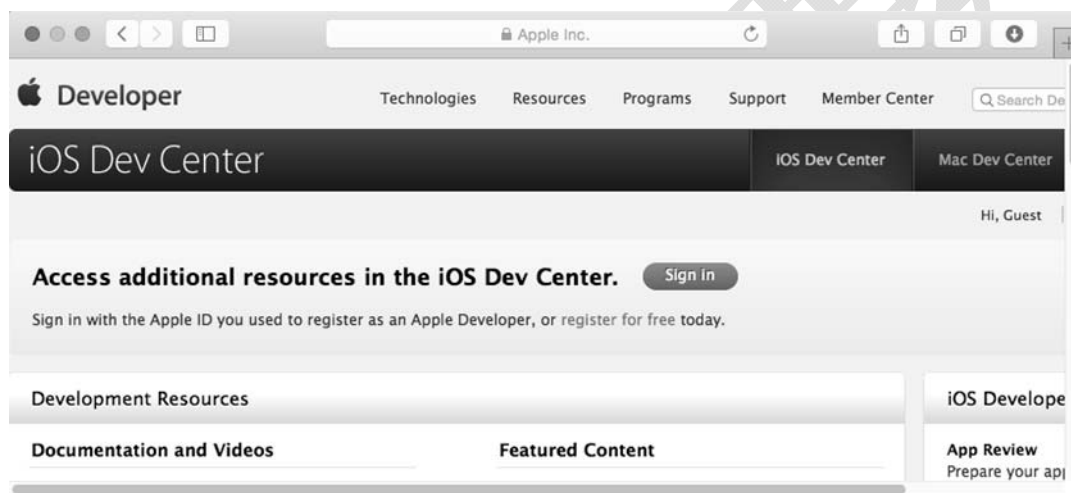


图 1.3 iOS Dev Center-App Developer 网页

(4) 选择 register for free 选项，进入 Apple Developer Registration-Apple Developer 网页，如图 1.4 所示。



图 1.4 Apple Developer Registration-Apple Developer 网页

(5) 单击 Register Now 按钮，进入登录 Apple ID 的网页，如图 1.5 所示。



图 1.5 登录 Apple ID 的网页

(6) 单击 Create Apple ID 按钮，进入 Apple-My Apple ID 网页，如图 1.6 所示。

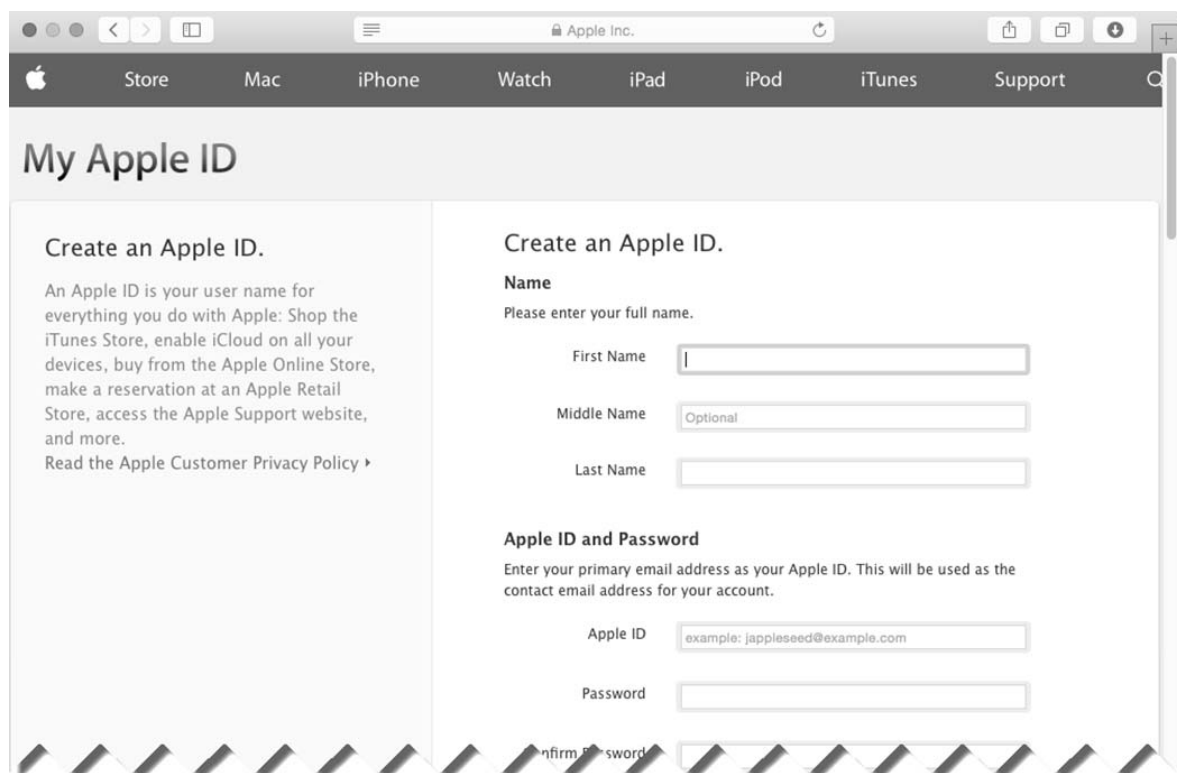


图 1.6 Apple-My Apple ID 网页

(7) 在网页中按照要求输入内容后，单击网页最下方的 **Create Apple ID** 按钮，进入确认邮件地址的网页，如图 1.7 所示。

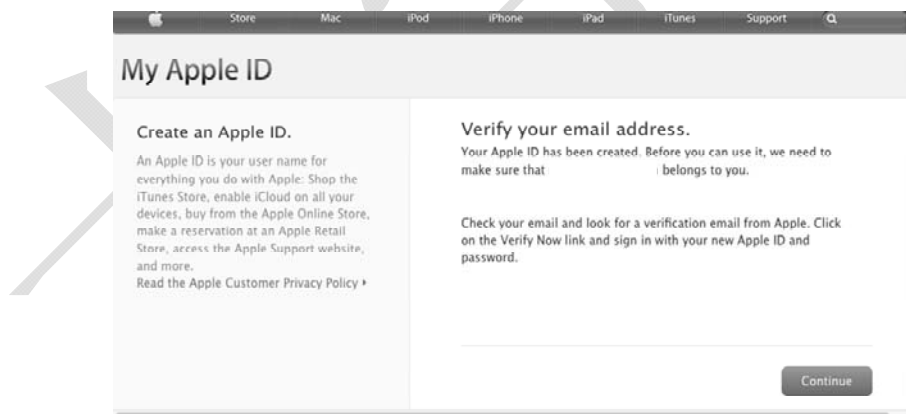


图 1.7 确认邮件地址的网页

(8) 单击 **Continue** 按钮，进入到确定邮件地址的另一个网页。单击此网页中的 **Send Verification Email** 按钮，苹果公司会向作为帐号的邮箱发送一封确认邮件。

(9) 进入帐号所使用的邮箱，就会看到 **Apple** 发来的一封确定邮件地址的邮件。打开该邮件，如图 1.8 所示。

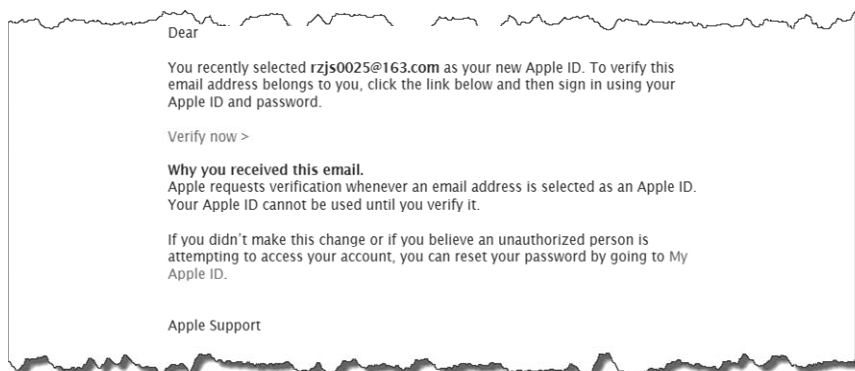


图 1.8 确认邮件

(10) 单击 **Verify now** 链接，进入 **Apple-My Apple ID-Email Verification** 网页，如图 1.9 所示。



图 1.9 Apple-My Apple ID-Email Verification 网页

(11) 输入需要验证的邮箱以及地址，单击 **Verify Address** 按钮，进入到下一个网页，此网页会提示开发者注册的 **Apple ID** 现在已经可以使用了。

### 1.1.3 注册收费的苹果账号

由于免费的苹果账号在开发应用程序时受到了很多的限制，如不可以进行真机测试等。所以需要注册一个收费的苹果账号。以下是注册收费的苹果账号的具体步骤：

(1) 在浏览器 Safari 中输入网址（<https://developer.apple.com/programs/>），按下回车，如图 1.10 所示。

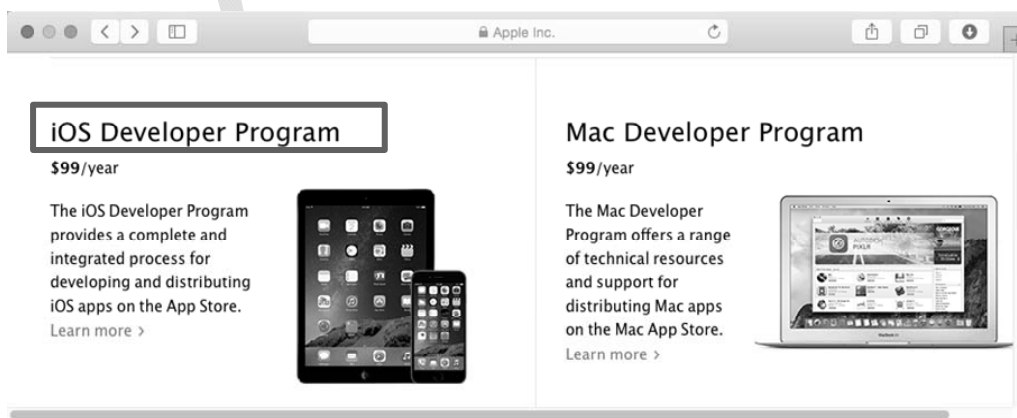


图 1.10 开发者页面

(2) 选择 iOS Developer Program 选项, 进入 iOS Developer Program-Apple Developer 网页, 如图 1.11 所示。

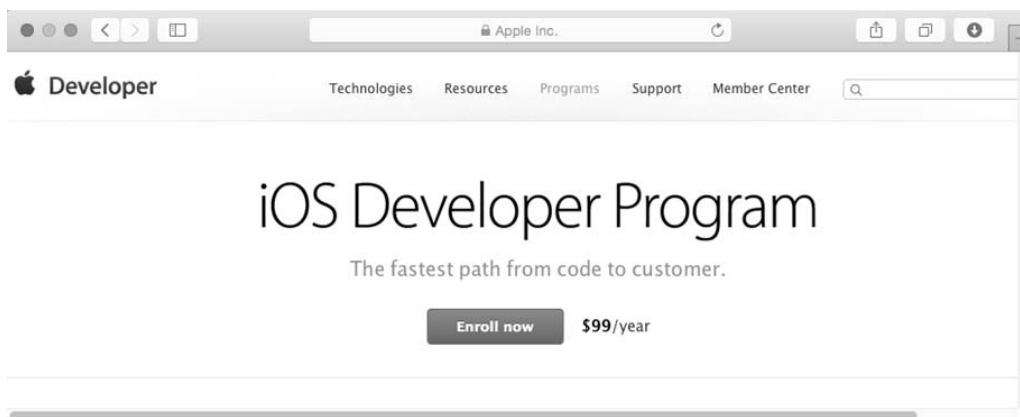


图 1.11 iOS Developer Program-Apple Developer 网页

(3) 选择 EnrollNow 按钮, 进入 Enrolling in Apple Developer Programs-Apple Developer 网页, 如图 1.12 所示。

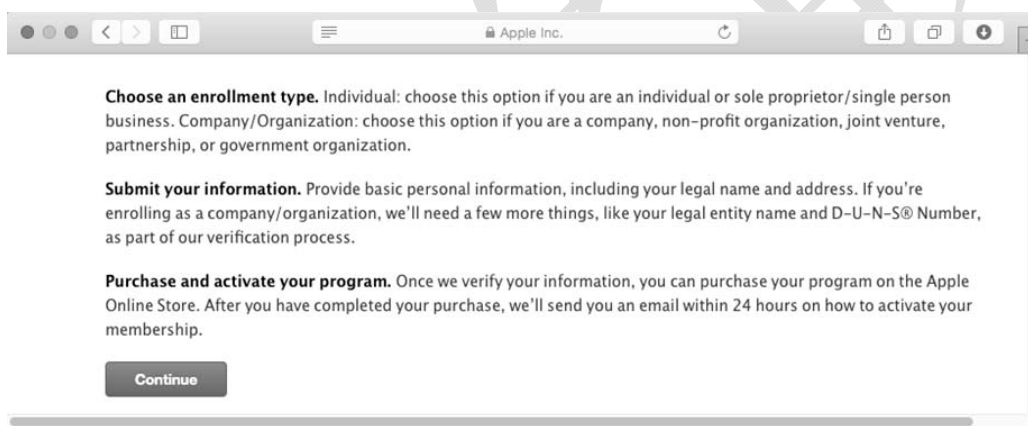


图 1.12 Enrolling in Apple Developer Programs-Apple Developer 网页

(4) 选择 Continue 按钮, 进入 Sign in or create an Apple ID-Apple Developer Program Enrollment 网页, 如图 1.13 所示。

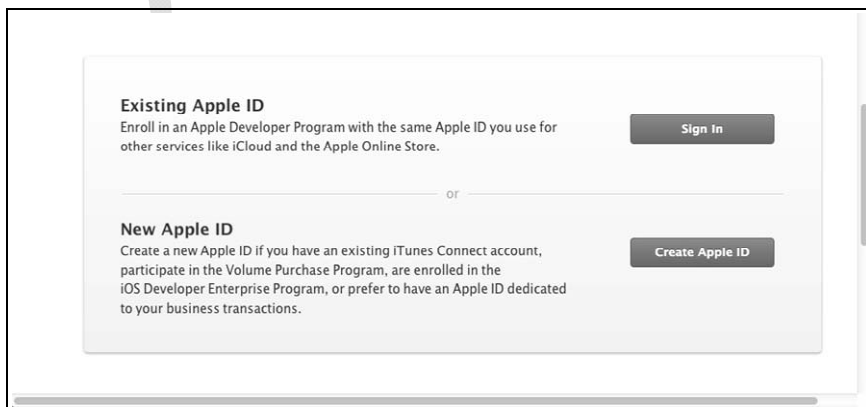


图 1.13 Sign in or create an Apple ID-Apple Developer Program Enrollment 网页

（5）选择 Sign In 按钮，进入 Apple Developer Program Enrollment 网页，如图 1.14 所示。



图 1.14 Apple Developer Program Enrollment 网页

（6）单击 Individual 按钮后，进入 Sign in with your Apple ID-Apple Developer 网页，如图 1.15 所示。



图 1.15 Sign in with your Apple ID-Apple Developer 网页

（7）输入 Apple ID 以及密码后，单击 Sign In 按钮。在 Apple Developer Program Enrollment-Update Information 网页，完善自己的信息，单击 Continue 按钮。以上这几步是申请付费开发者账号的重要步骤，剩下的步骤就需要根据开发者的需求就行填写了。这样就不再做介绍了。

注意：从申请一个付费的开发者账号开始到激活大概需要 3~5 天，这段时间需要开发者留心你的与苹果账号关联的邮箱，苹果公司会为此邮箱发一些邮件。

## 1.2 Xcode 的下载和安装

本节将主要讲解使用两种方式下载和安装 Xcode：一种是在 App Store 中下载和安装 Xcode；另一种是其他网站下载和安装 Xcode。

## 1.2.1 App Store 中下载和安装 Xcode

App Store 中提供了很多的软件，而 Xcode 也在其中。以下就是在 App Store 中下载和安装 Xcode 的具体步骤：

（1）在 Dock 中找到 App Store，如图 1.16 所示。

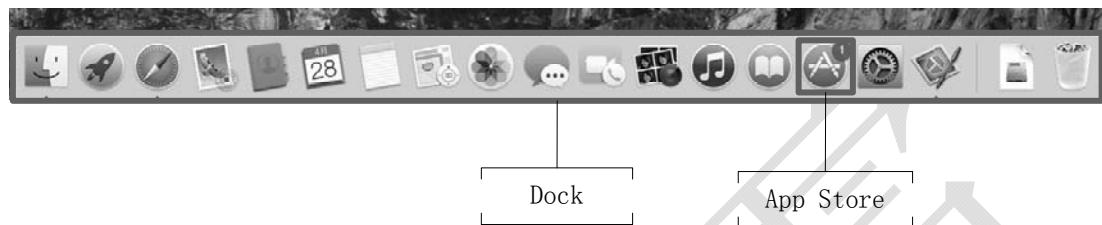


图 1.16 App Store 图标

（2）单击 App Store 图标，打开 App Store 窗口，如图 1.17 所示。



图 1.17 App Store 窗口

（3）在搜索栏中输入要搜索的内容，即 Xcode，按下回车，进行搜索，如图 1.18 所示。

（4）单击 Xcode 右下方的“获取”按钮，此时“获取”按钮变为了“安装 APP”按钮，如图 1.19 所示。





图 1.18 搜索结果



图 1.19 搜索结果

(5) 单击“安装 APP”按钮，弹出“登录 App Store 来下载”对话框，如图 1.20 所示。



图 1.20 “登录 App Store 来下载”对话框

(6) 输入 Apple ID 以及密码后，单击“登录”按钮。此时，“安装 APP”按钮变为了“安装”按钮，如图 1.21 所示。并且 Xcode 会在 Launchpad 中进行下载和安装，如图 1.22 所示。



图 1.21 开始下载



图 1.22 开始下载

(7) 一般在 Launchpad 中下载的软件，都可以在应用程序中找到。选择“前往|应用程序”打开应用程序，如图 1.23 所示。

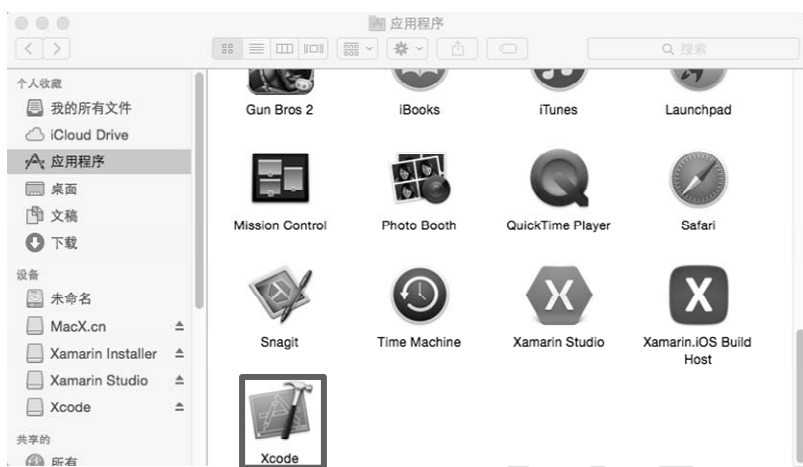


图 1.23 “应用程序”窗口

(8) 双击 Xcode 图标，弹出 Xcode and iOS SDK License Agreement 对话框，如图 1.24 所示。

(9) 单击 Agree 按钮，弹出“Xcode”想要进行更改。键入您的密码以允许执行此操作”，如图 1.25 所示。



图 1.24 Xcode and iOS SDK License Agreement 对话框



图 1.25 输入密码

(10) 输入密码，单击“好”按钮，进行组件的安装。组件安装完成后，就会弹出 Welcome to Xcode 对话框，此时 Xcode 就被启动了。

注意：在 App Store 中只可以下载 Xcode 的正式版，测试版是不可以下载的。

## 1.2.2 其他网站下载和安装 Xcode

有时候，应用商店下载较慢，所以用户也可以选择从其他网站下载 Xcode 安装文件。下面讲解这种 Xcode 的安装步骤：

(1) 双击下载的 Xcode 软件，弹出正在打开此软件的对话框，如图 1.26 所示。

(2) 打开该软件后，就会弹出 Xcode 对话框，如图 1.27 所示。



图 1.26 打开此软件的对话框



图 1.27 Xcode 对话框

(3) 将 Xcode 应用软件拖动到 Applications 文件夹中。此时该软件就会拷贝到应用程序中。

(4) 在菜单栏的“应用程序”窗口中找到安装的 Xcode 图标。双击该图标，弹出 Xcode and iOS SDK License Agreement 对话框，如图 1.28 所示。

(5) 单击 Agree 按钮，弹出“键入您的密码以允许执行此操作”，如图 1.29 所示。

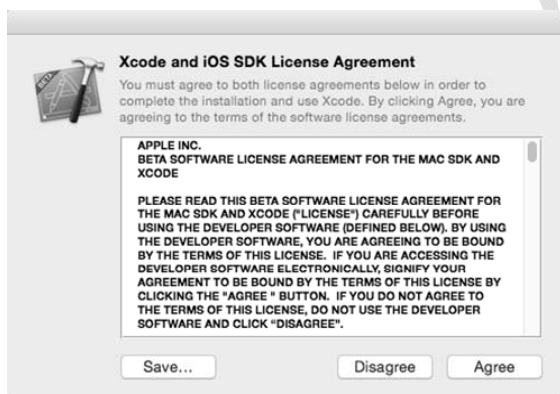


图 1.28 Xcode and iOS SDK License Agreement 对话框



图 1.29 输入密码

(6) 输入密码，单击“好”按钮，进行组件的安装。组件安装完成后，就会弹出 Welcome to Xcode 对话框，此时 Xcode 就被启动了，如图 1.30 所示。

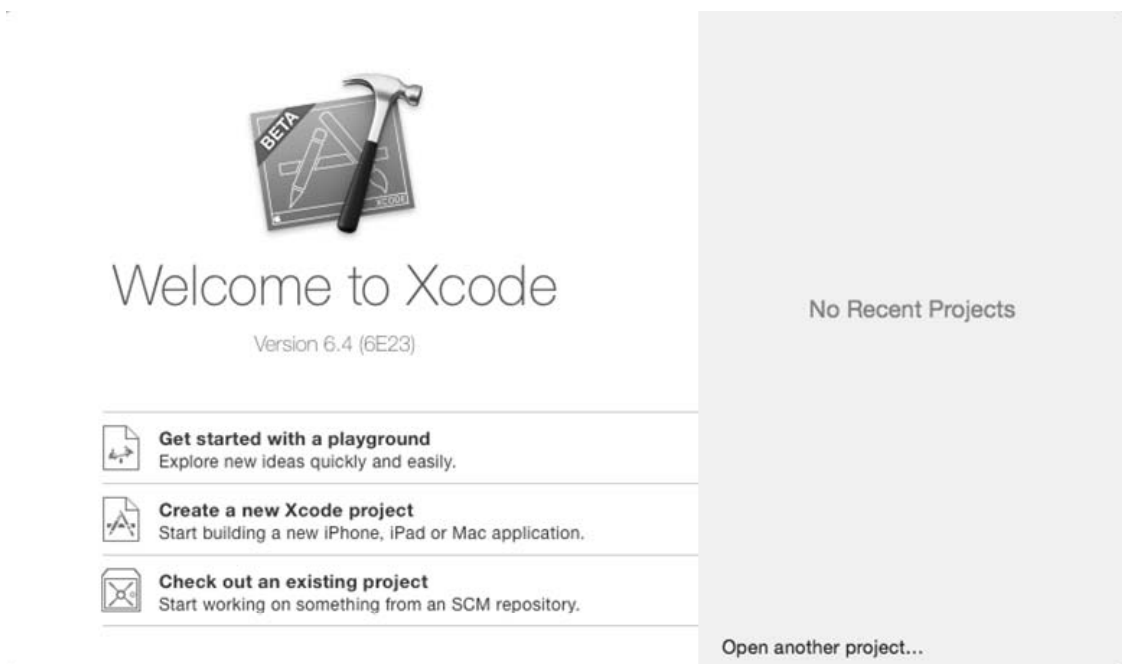


图 1.30 Welcome to Xcode

## 1.3 苹果账号的绑定

有时为了方便 Xcode 中组件以及内容的随时更新，必须要绑定一个苹果账号。以下将讲解了如何绑定一个苹果账号。

- (1) 单击 Xcode 打开，在菜单栏中选择 Xcode|Preferences 命令。
- (2) 在弹出的对话框中选择 Accounts 选项，打开 Accounts 对话框，如图 1.31 所示。
- (3) 选择“+”号，就会出现三个选项分别为 Add Apple ID...、Add Repository...、Add Server...，如图 1.32 所示、

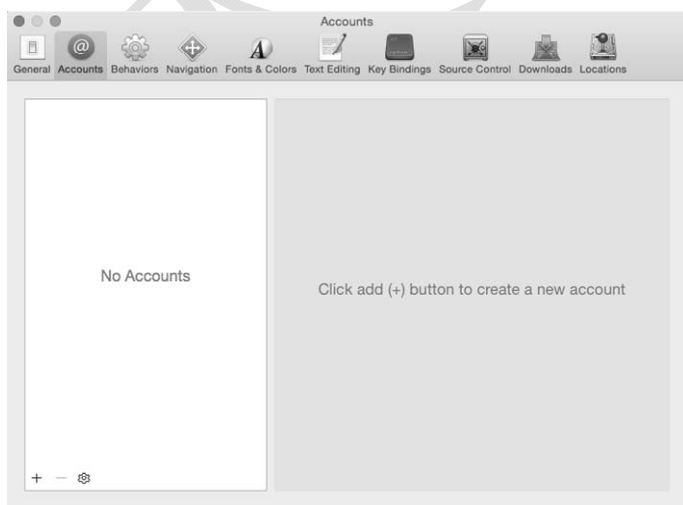


图 1.31 Accounts 对话框

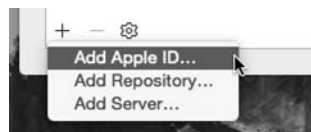


图 1.32 弹出菜单

(4) 单击 Add Apple ID...命令，弹出一个 Enter an Apple ID associated with an Apple Developer Program:对话框，如图 1.33 所示。

(5) 输入苹果账号以及密码后，单击 Add 按钮。此时。苹果账号就被绑定了，并将相关信息显示在 Accounts 对话框中，如图 1.34 所示。



图 1.33 Enter an Apple ID associated with an Apple Developer Program:对话框

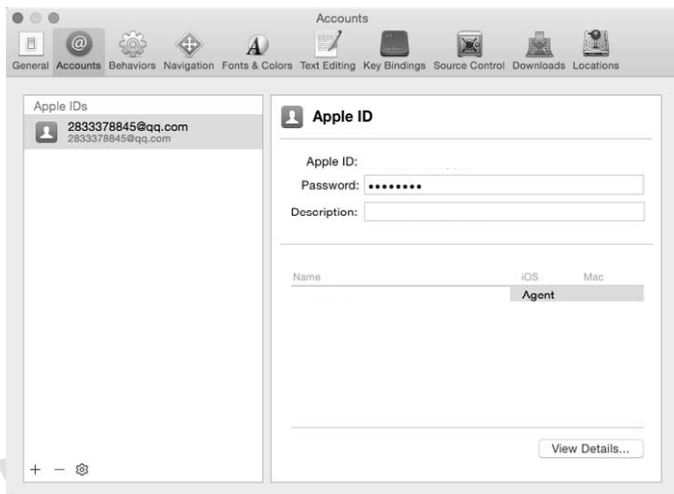


图 1.34 Accounts 对话框

注意：在进行绑定苹果账号的操作时，Xcode 必须处于启动的状态。

## 1.4 组件的下载和安装

Xcode 中组件和文档都是经常更新的。为了获得最新的工具和帮助文档，我们需要定时更新组件和文档。操作方式如下：

(1) 单击 Accounts 对话框中的 Downloads 图标，进入到 Downloads 对话框，如图 1.35 所示。

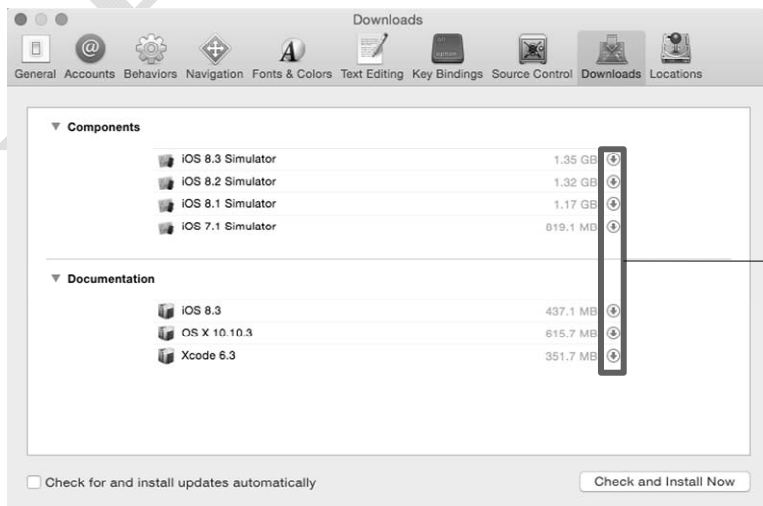


图 1.35 Downloads 对话框

(2) 单击需要进行更新的文档及组件后面的下载安装按钮，进行组件和文档的下载安装，如图 1.36

所示。

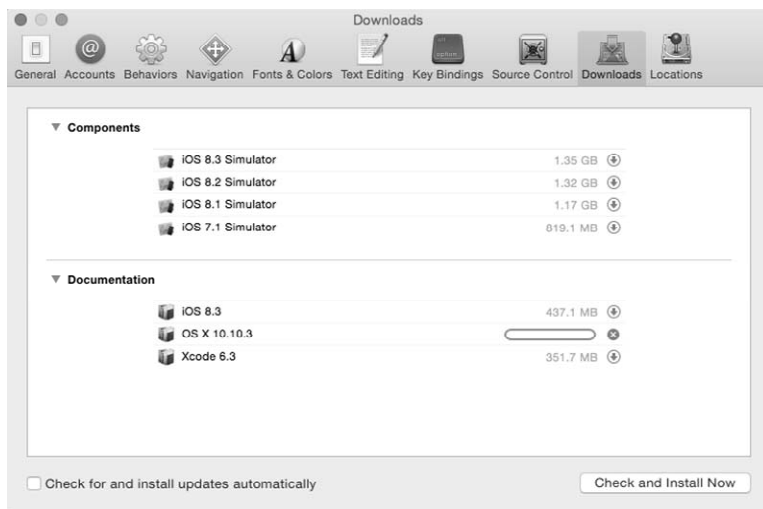


图 1.36 开始更新

在更新组件和文档的操作时需要注意两点：（1）Xcode 必须处于启动的状态。（2）选择 Downloads 对话框对话框中的 Check and Install Now 按钮，可以对在此对话框中所有没有进行下载的组件和文档进行下载和安装

## 1.5 第一次打开 Xcode

对 Xcode 安装好之后，就可以使用了。本节讲解如何使用 Xcode，并实现一个应用程序项目的创建。一个应用的所有文件都在 Xcode 项目下，项目可以帮助用户管理代码文件和资源文件。以下是打开 Xcode 并创建项目的具体操作步骤：

（1）双击 Xcode 启动，进入到 Welcome to Xcode 对话框中，如图 1.37 所示。

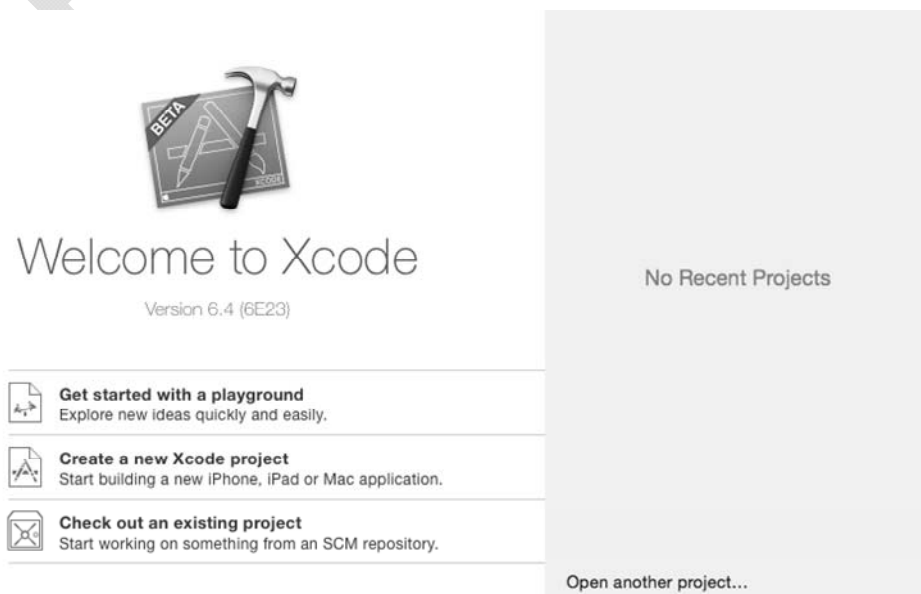


图 1.37 Welcome to Xcode

(2) 选择 Create a new Xcode project 选项，弹出 Choose a template for your new project:对话框，如图 1.38 所示。

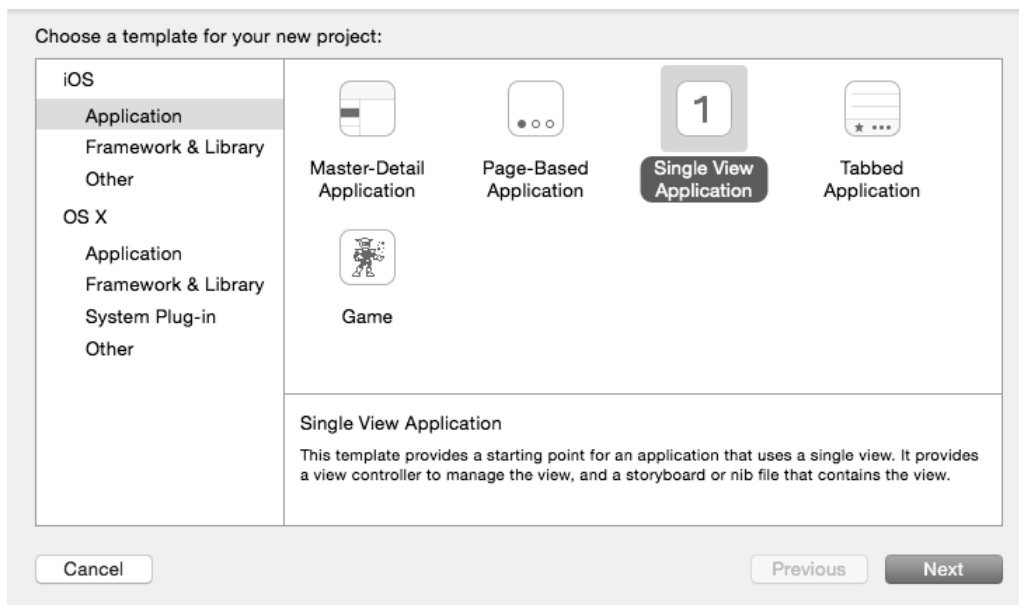


图 1.38 Choose a template for your new project:对话框

(3) 选择 iOS 下的 Application 下的 Single View Application 模板。单击 Next 按钮，弹出 Choose options for your new project:对话框，如图 1.39 所示。

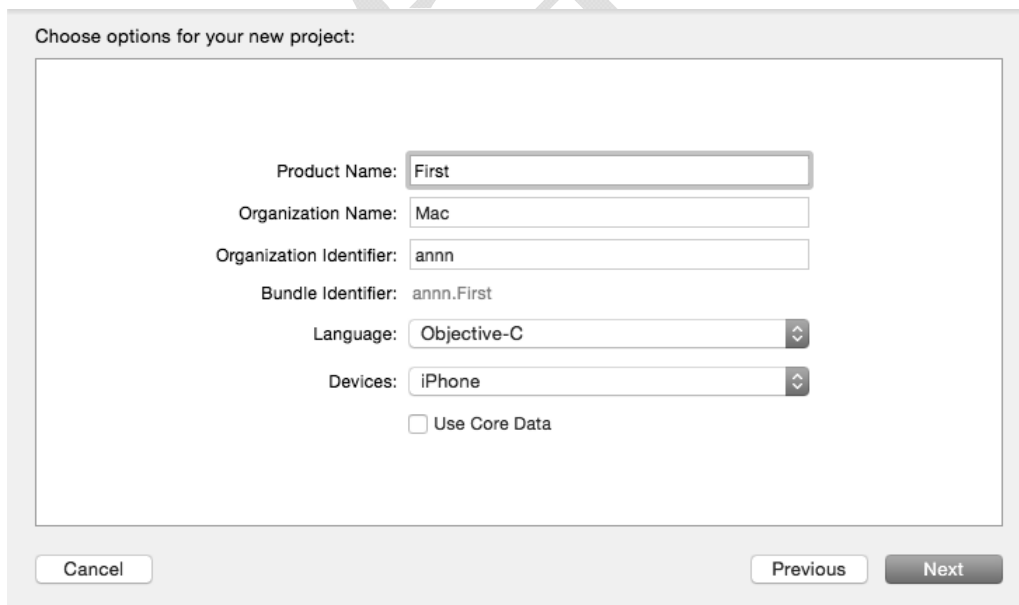


图 1.39 Choose options for your new project:对话框

(4) 填入 Product Name（项目名）、Organization Identifier（标识符）信息以及选择 Language（编程语言）和设备 Devices（设备），如表 1-2 所示。

表 1-2 填写的内容

需要填写的项	填入的内容
Product Name:	First
Company Identifier	annn
Language	Objective-C
Devices	iPhone

注意：Product Name、Organization Identifier 信息是随意的，由开发者决定。由于我们是使用 Objective-C 语言编写程序，所以在 Language 这一项中选择 Objective-C。一般对 Organization Identifier、Language、Devices 设置一次后，在下次创建项目时，到了 Choose options for your new project:对话框中只输入 Product Name（项目名）就可以了。

（4）单击 Next 按钮，弹出保存位置对话框，如图 1.40 所示。

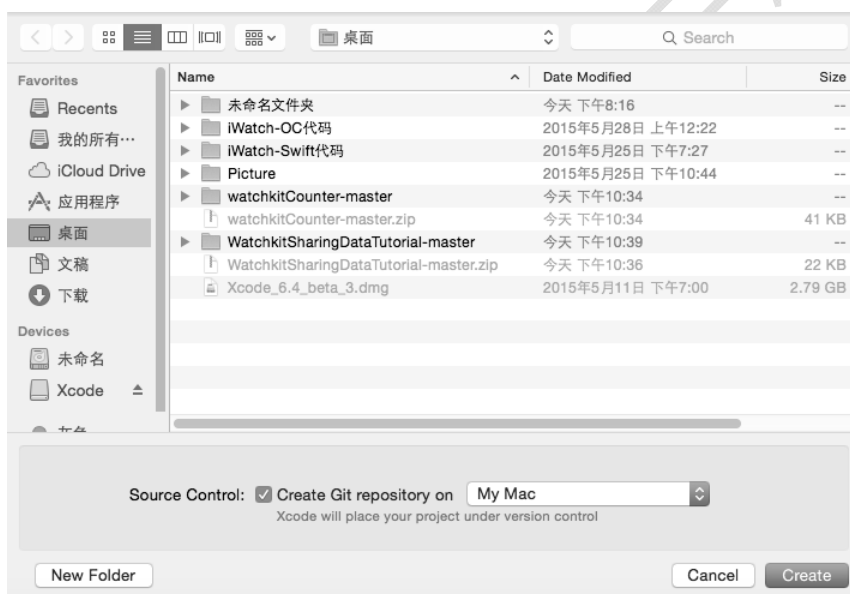


图 1.40 保存位置对话框

（5）选择 Create 按钮，这时一个项目名为 First 的项目就创建好了。

## 1.6 Xcode 的界面介绍

一个 Xcode 项目由很多的文件组成，例如代码文件、资源文件等。Xcode 会帮助开发者对这些文件进行管理。所以，Xcode 的界面也比较复杂，如图 1.41 所示。





图 1.41 Xcode 界面

在图中可以看到 Xcode 的界面大致可以分为 5 大部分。其中，编号为 1 的部分是导航窗口；编号为 2 的部分是代码编辑区域；编号为 3 的部分是组件窗口；编号为 4 的部分是显示程序调试信息的窗口。编号为 5 的部分是工具栏。本节将对几个重要的区域进行讲解。

### 1.6.1 导航窗口

导航窗口的作用是显示整个项目的树状结构。开发者可以根据自己的喜好对其进行大小调整，以及显示和隐藏（View|Navigators|Show/HideNavigator 来实现显示和隐藏，或通过使用 Hide or show the Navigator 按钮来实现显示和隐藏）。导航窗口可以显示 8 类不同的信息，所以又有了 8 个导航器。这 8 个导航器分别为：项目导航器、符号导航器、搜索导航器、问题导航器、测试导航器、调试导航器、断点导航器和日志导航器。可以通过导航窗口顶部的 8 个图标来进行导航之间的切换，如图 1.42 所示。

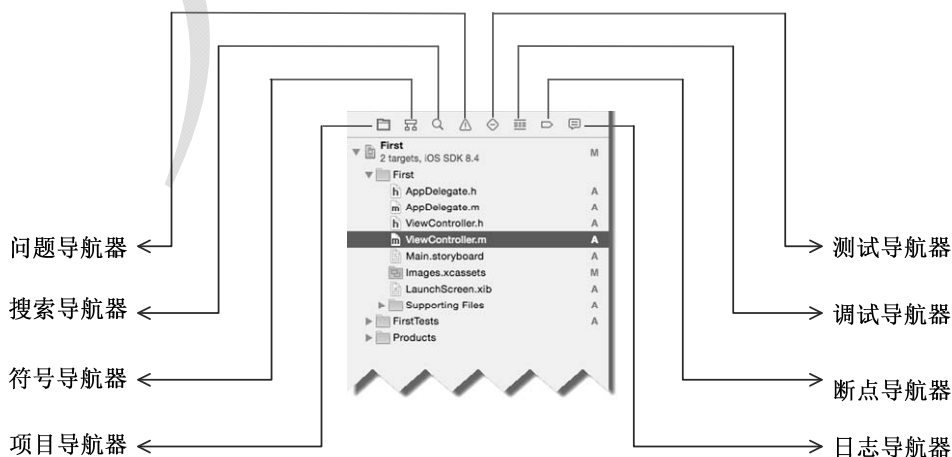


图 1.42 导航窗口

## 1.6.2 组件窗口

组件窗口可以对项目的一些设置信息进行编辑，开发者也是可以进行随时显示和隐藏的（View|Utilities|Show/HideUtilities 来实现显示和隐藏，或通过使用 Hide or show the Utilities 按钮来实现显示和隐藏）。组件窗口可以分为上下两个部分。上半部分显示的内容取决于开发者在编辑器上正在编辑的文件类型，其中文件类型有两种如下：

- ❑ 当编辑器编辑的是代码文件时，组件窗口上半部分显示的内容分为：文件查看器和快速帮助其中一个的内容，如图 1.43 所示。要想实现两个内容的切换，可以通过使用此窗口上半部分在顶部显示的图标来进行切换。
- ❑ 当编辑器编辑的是界面文件时，组件窗口上半部分显示的内容分为：文件查看器、快速帮助、标识符查看器、属性查看器、尺寸查看器和连接查看器其中一个的内容，如图 1.44 所示。要想实现 6 个内容的切换，可以通过使用此窗口上半部分在顶部显示的图标来进行切换。

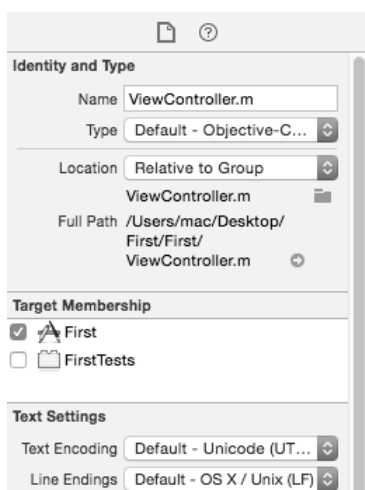


图 1.43 编辑代码显示的内容

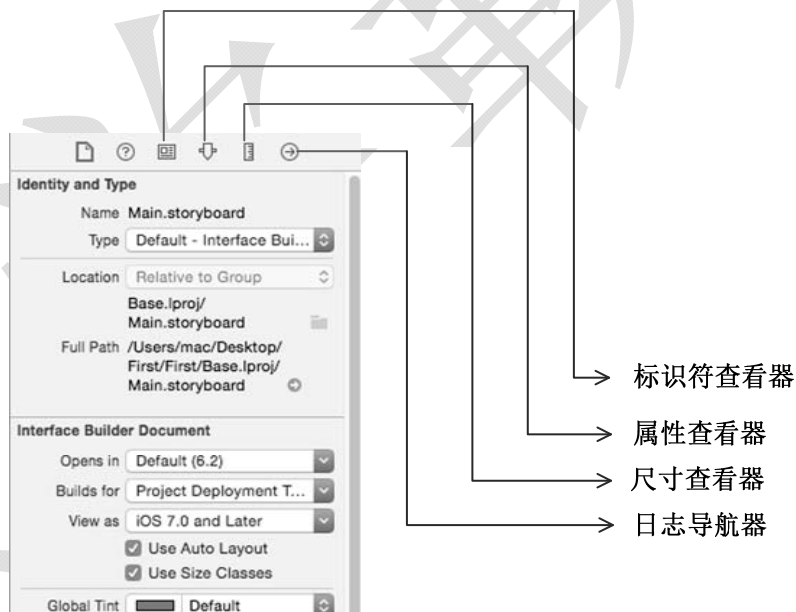


图 1.44 编辑编辑界面显示的内容

上半个组件窗口介绍完后，在来看下半个窗口。下半个组件窗口显示的内容是文件模板库、代码片断库、对象库和媒体库的其中一个内容，如图 1.45 所示。要想实现这 4 个内容的切换，可以通过使用此窗口下半部分在顶部显示的图标来进行切换。

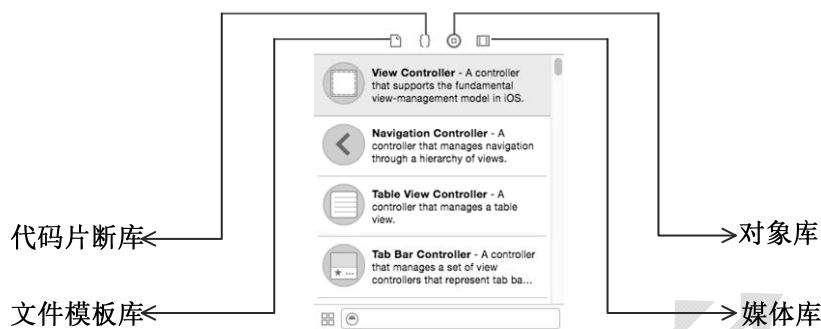


图 1.45 组件窗口的下半部分

### 1.6.3 编辑窗口

编辑窗口可以用来编写代码，或在编辑界面设置用户界面。在顶端，有左右箭头和整个项目的层次显示。编辑窗口可以记住已经编辑的文件名，可以使用顶端的左右箭头来选择曾经编辑过的文件。在一个项目的窗口中至少要包含一个编辑窗口。如果项目要同时打开多个编辑窗口该如何实现呢，以下主要讲解了三个方法。

#### 1.Assistants 方法

单击窗口中的 Show the Assistant editor 按钮后，Xcode 会默认显示两个编辑窗口（后面的窗口叫做 Assistant）。这两个窗口上的内容一般都是关联的。如果需要显示更多的编辑窗口，可以在 Assistant 窗口的顶部选择“+”按钮来实现。

#### 2.Tabs 方法

Tabs 方法是显示各个文件，和 Safari 显示网页的方法一样。通过选择 File|New|Tab 命令或使用快捷键 Windows+T 来启动这个方法。通过单击标签或者使用 Windows+Shift+} 在不同的窗口中进行切换。

#### 3.新的窗口

新的窗口这个方法和 Tab 类似，但是它是用来显示独立的窗口。要创建新的窗口选择 File|New|Window 命令来创建新的窗口。

### 1.6.4 目标窗口

目标窗口中包含了项目的程序和配置，这些配置指定了如何构建程序代码，如图 1.46 所示。在目标窗口的顶部，可以选择 General、Capabilities、Info、Build Settings、Build Phases、Build Rules 中的内容。



图 1.46 目标窗口

## 第 2 章 编写第一个 iWatch 应用程序

iWatch 是苹果公司在 2014 年 9 月 9 日的开发大会上公布的一款可穿戴的智能手表。本章将以编写一个简单的 iWatch 应用程序为线索，为开发者讲解 iWatch 的优缺点、开发 iWatch 使用的类、添加应用对象、调试程序等内容。

### 2.1 iWatch 介绍

iWatch 又名 Apple Watch，它是苹果公司在 2014 年 9 月 9 日的开发大会上公布的一款可穿戴的智能手表。iWatch 具备电话、语音回电话、连接汽车，天气、航班信息，地图导航，播放音乐，测量心跳、计步等几十种功能。它是一款全方位的健康和运动追踪设备。它有普通、运动、定制三个系列。iWatch 有多种颜色可以选择，如图 2.1 所示。



图 2.1 iWatch

### 2.2 iWatch 的优点缺点

本节将主要讲解 iWatch 的优点以及缺点。

#### 2.2.1 iWatch 优点

iWatch 的优点如下：

##### 1. 微妙、免打扰通知

依靠全新触觉反馈系统引擎“Taptic Engine”，苹果智能手表 iWatch 作为 iPhone 手机的一种补充，

最重要的功能之一就是以一种无缝、免打扰模式显示手机上的通知，这样其他人不会注意到你已收到一条新通知。接着，用户可以自行选择是抬起手腕查看通知，或者等会再看。

## 2. 无需输入，快速交流

iWatch 无需编辑文本信息来快速交流，如图 2.2 所示。比如，当你约会迟到时，你只需要点击一个“编码”就可以通知对方，无需编写文字信息。同时，你还可以向你心爱的另一半发送“轻拍”和“图案”来表明你正在想念他/她。



图 2.2 快速交流

## 3. 健康锻炼新途径

苹果正在采用一种与众不同的途径来追踪用户的健康和运动数据，并通过三个圆圈来显示用户的“移动时间”，“锻炼时间”和“站立时间”，这些都是衡量用户身体健康的重要指标，如图 2.3 所示。

## 4. 接听电话和留语言信息

苹果智能手表内置了麦克风。如果我们长时间将手腕在耳朵和嘴巴切换来与对方通话，肯定会感到不适，但是，我们仍然可以进行快速的回复，并留语言回复，如图 2.4 所示。



图 2.3 健康锻炼新途径



图 2.4 接听电话

## 5. 全新的导航方式

应用在 iWatch 中的 Taptic Engine 功能能够在导航指路时，根据用户向左转或是向右转而会发出不同感觉的振动提示。因此，用户就无需死死盯着显示屏看，如图 2.5 所示。

## 6.快速回复信息

iWatch 能够“阅读”信息，并依据某个特定信息给出关联回复内容，而不是直接给出一列预先设定好的快速回复，如图 2.6 所示。



图 2.5 全新的导航方式



图 2.6 快速回复信息

## 7.精准的心率监测

内置在 iWatch 中的心率监测器在准确度方面做了提升。如图 2.7 所示。

## 8.防水

iWatch 是一款具备防水功能的智能手表，因此用户在洗澡时可以继续佩戴它，如图 2.8 所示。



图 2.7 精准的心率监测



图 2.8 防水

## 9.快速分享地理位置

当用户在跑步时，或者尝试给别人提供路线时，iWatch 快速分享信息功能能够为用户节省不少时间，如图 2.9 所示。

## 10.太阳表盘

iWatch 太阳表盘能够让用户看到当前太阳的位置，如图 2.10 所示。



图 2.9 快速分享地理位置



图 2.10 太阳表盘

## 2.2.2 iWatch 缺点

对于任何一款设备来说都不是十全十美的，本节将主要讲解 iWatch 的不足：

### 1.iWatch 几乎不能独立使用

根据开发者文档的描述，开发者需要在现有的 iOS 应用基础上开发 iWatch 应用。

### 2.没有自定义手势

iWatch 中可供使用的手势被严格限制：纵向滑动以滚动浏览内容、横向滑动在页面间切换、点按以选择、重点击（Force Touch，Hard-Pressing）打开菜单，而旋转表冠则以更快的速度滚动浏览页面。此外，从屏幕边缘左划可返回上一页或父目录，从边缘上划则开启 "Glance"视图。

### 3.地图控件不可交互

开发者可使用的地图控件不可与用户进行交互，除了可放置红、绿、紫三色图钉和自定义图片之外基本上就是固定的图片。用户无法直接在空间中拖动查看周边内容，但点击控件后会开启 Apple Maps 应用。

### 4.不能播放视频

应用最多可以缓存 20M 的图片，除此之外的所有内容都必须从 iPhone 处读取/传输获得。不过，开发者可以用一系列现有图片配置成简单动画供应用加载（类似 GIF）。iWatch 并不支持视频的储存和播放，鉴于屏幕分辨率有限，这一设定还是相对合理的。

## 2.3 Apple Watch 1.0.1 新增内容

在网上人们所说的 Apple Watch 1.0.1 就是 Watch OS 1.0.1，其中 Watch OS 为 Apple Watch 的操作系统，并且与 iOS 进行了深度融合。Watch OS 1.0.1 是 2015 年 5 月 20 日发布的，它是自 3 月 24 日发布以来，首次对 Watch OS 进行的更新。Watch OS 1.0.1 主要对性能进行了提升，并修复了此前系统中存在的



bug，具体涉及方面包括：Siri、站立活动的测量、室内单车与划船训练的卡路里计算、户外步行与跑步训练的距离和速度、辅助功能以及第三方应用。

此外，Watch OS 1.0.1 还支持 iOS 8.3 引入的新的 emoji 表情，并增加了对多种语言的支持，比如巴西葡萄牙语、丹麦语、荷兰语、瑞典语、俄语、泰语和土耳其语。

最新的操作系统可通过 iPhone 上专用的 Apple Watch app 进行更新，需进入 General --> Software Update 更新。

注意：在苹果专门开设的 Apple Watch 论坛中有很多开发者想要使用代码对自己的 Apple Watch 操作系统进行查看，这一点是不可以实现的。如果你使用的是真机，可以在 General 中进行查看。

## 2.4 开发 iWatch 应用程序使用的类

WatchKit 可以用来为开发者构建 iWatch 应用程序。WatchKit 是一个专门开发 iWatch 应用程序的框架，在其中包含的类以及功能如表 2-1 所示。

表 2-1 开发 iWatch 应用程序使用的类

类	功能
WKInterfaceController	界面控制器，它是开发 iWatch 应用程序的核心类
WKUserNotificationInterfaceController	通知界面控制器
WKInterfaceDevice	对 iWatch 部分信息的获取
WKInterfaceObject	负责界面的对象
WKInterfaceButton	按钮对象
WKInterfaceDate	日期时间对象
WKInterfaceGroup	组对象
WKInterfaceImage	图像对象
WKInterfaceLabel	标签对象
WKInterfaceMap	地图对象
WKInterfaceSeparator	分割线对象
WKInterfaceSlider	滑块对象
WKInterfaceSwitch	开关对象
WKInterfaceTable	表对象
WKInterfaceTimer	计时器对象

## 2.5 编程第一个程序

本节将通过编写第一个程序，为开发者讲解如何添加 Watch 应用对象、运行程序、界面设计、编写代码等内容。

### 2.5.1 创建项目

在第一章中我们讲过一个应用的所有文件都在 Xcode 项目下，所以在编写应用程序之前，首先需要对项目进行创建，创建项目的具体操作步骤如下：

- (1) 单击 Dock 中的 Xcode，弹出 Welcome to Xcode 对话框，如图 2.11 所示。

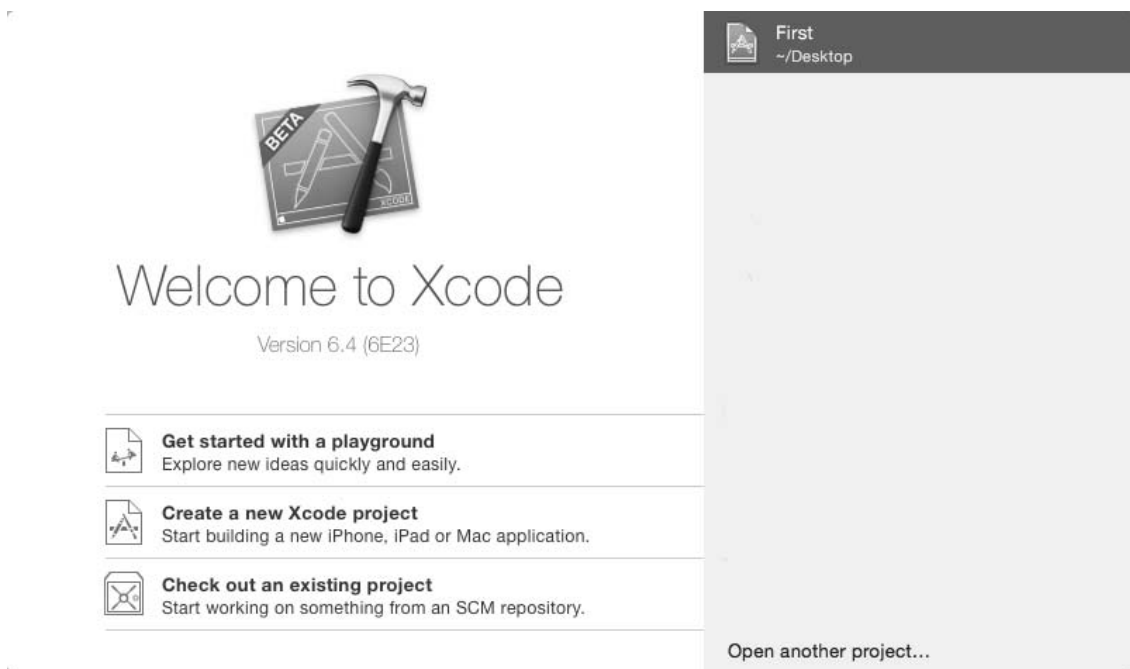


图 2.11 Welcome to Xcode 对话框

(2) 选择其中的 Create a new Xcode project 项目，弹出 Choose a template for your new project:对话框，如图 2.12 所示。

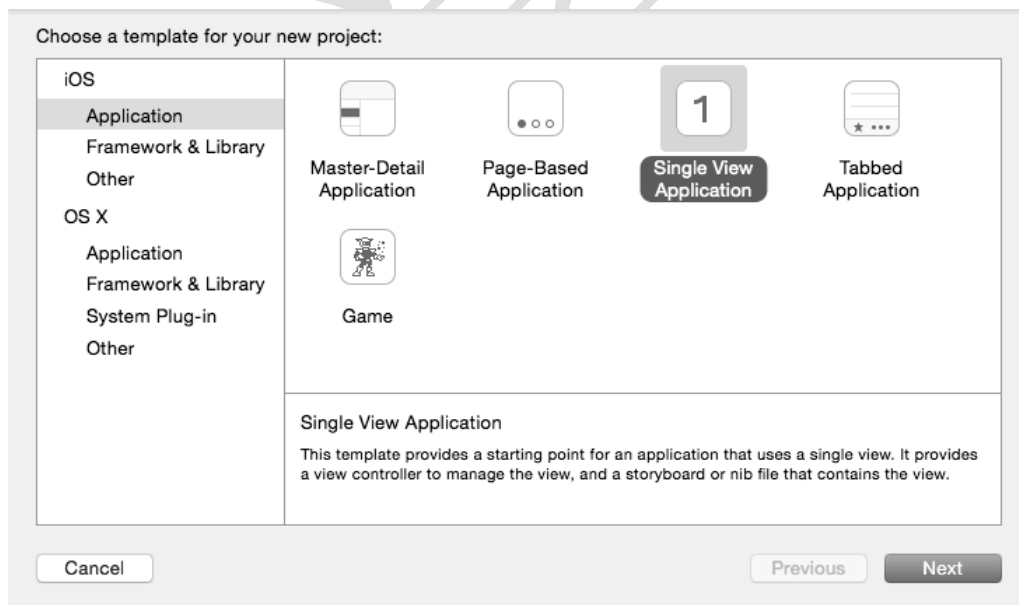


图 2.12 Choose a template for your new project:对话框

(3) 选择 iOS|Application 中的 Single View Controller 模板，单击 Next 按钮后，弹出 Choose options for your new project:对话框，如图 2.13 所示。

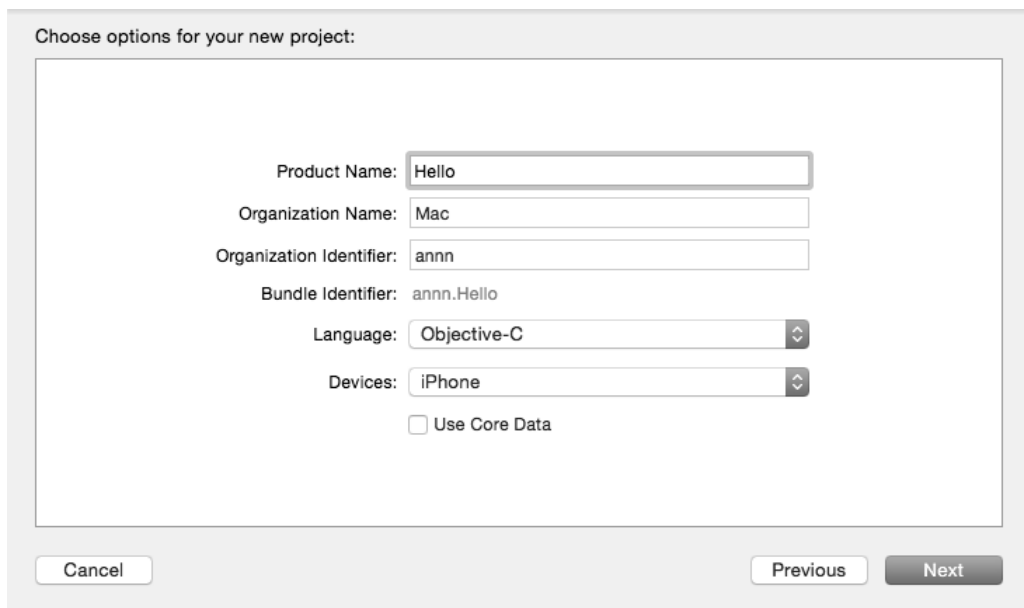


图 2.13 Choose options for your new project:对话框

(4) 在此对话框中输入项目的名称 Hello。单击 Next 按钮，弹出选择项目保存位置的对话框，如图 2.14 所示。

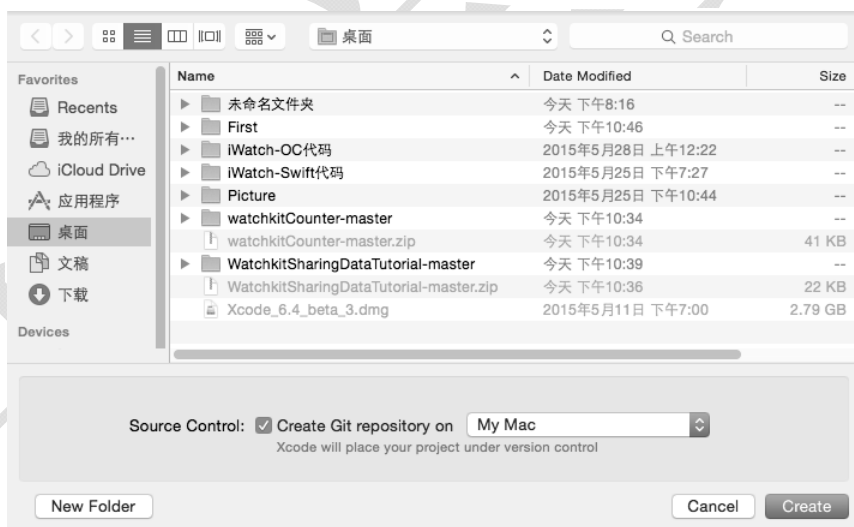


图 2.14 项目保存位置的对话框

(5) 单击 Create 按钮后，一个名为 Hello 的项目就创建好了。

## 2.5.2 添加 Watch 应用对象到项目中

创建好项目之后，我们就可以开发一些 iPhone/iPad 应用程序了。但是如果想要开发 iWatch 应用程序，还需要在创建的项目中添加一个 Watch 应用对象，添加 Watch 应用对象的具体操作步骤如下：

(1) 选择菜单栏中的 File 命令，弹出下拉菜单。如图 2.15 所示。

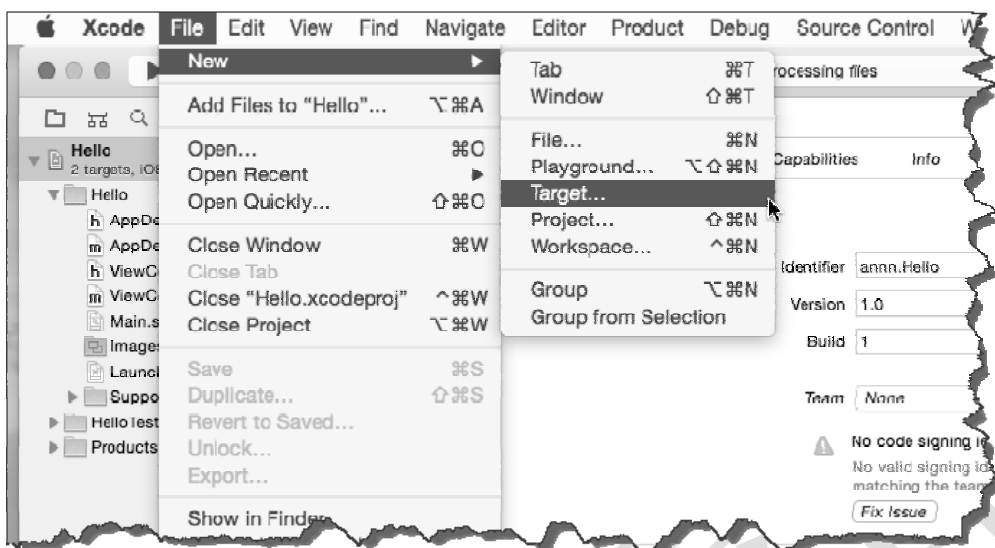


图 2.15 下拉菜单

(2) 选择 New|Target...命令，弹出 Choose a template for your new target:对话框，如图 2.16 所示。

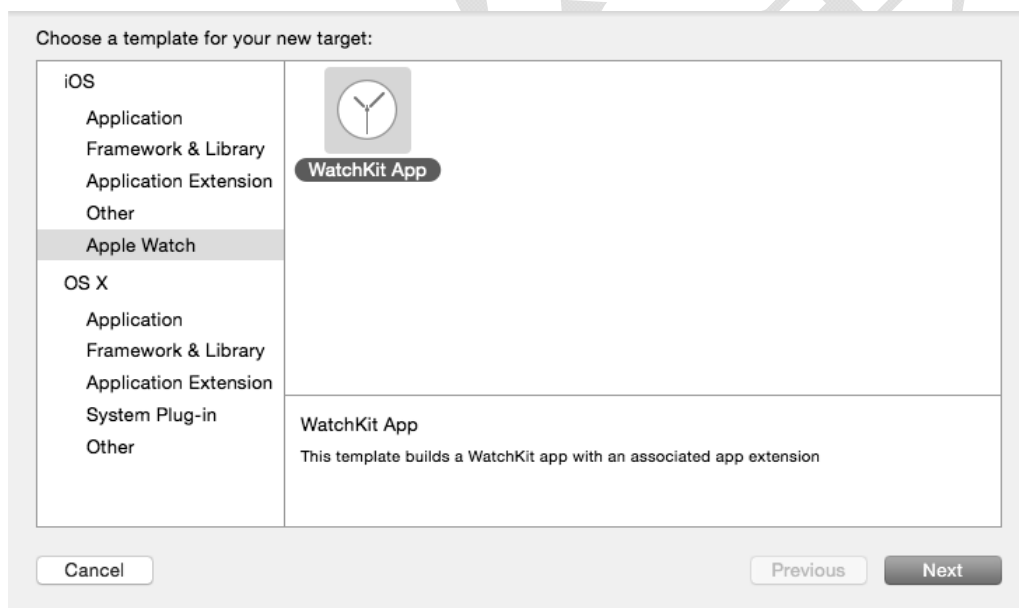


图 2.16 Choose a template for your new target:对话框

(3)选择 iOS|Apple Watch 中的 WatchKit App 模板,单击 Next 按钮,弹出 Choose options for your new target:对话框,如图 2.17 所示。

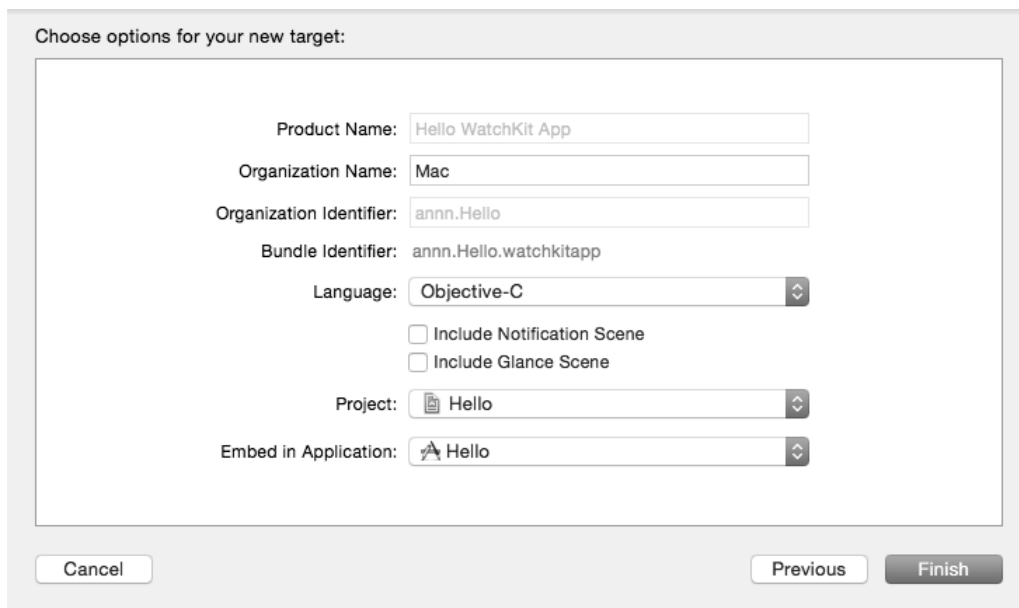


图 2.17 Choose options for your new target:对话框

(4) 取消 Include Notification Scene 的选择，单击 Finish 按钮，此时 Hello WatchKit App 应用对象就添加到了 Hello 项目中。

### 2.5.3 添加 Watch 应用对象时新增内容介绍

Watch 应用对象添加到创建的项目中后，会包含两个部分：Watch App 和 WatchKit Extension，如图 2.18 所示。其中，Watch App 部分位于用户的 iWatch 上，它目前为止只允许包含 Storyboard 文件和 Resources 文件。在我们的项目里，这一部分不包括任何代码。WatchKit Extension 部分位于用户的 iPhone 安装的对应该 App 上，这里包括我们需要实现的代码逻辑和其他资源文件。

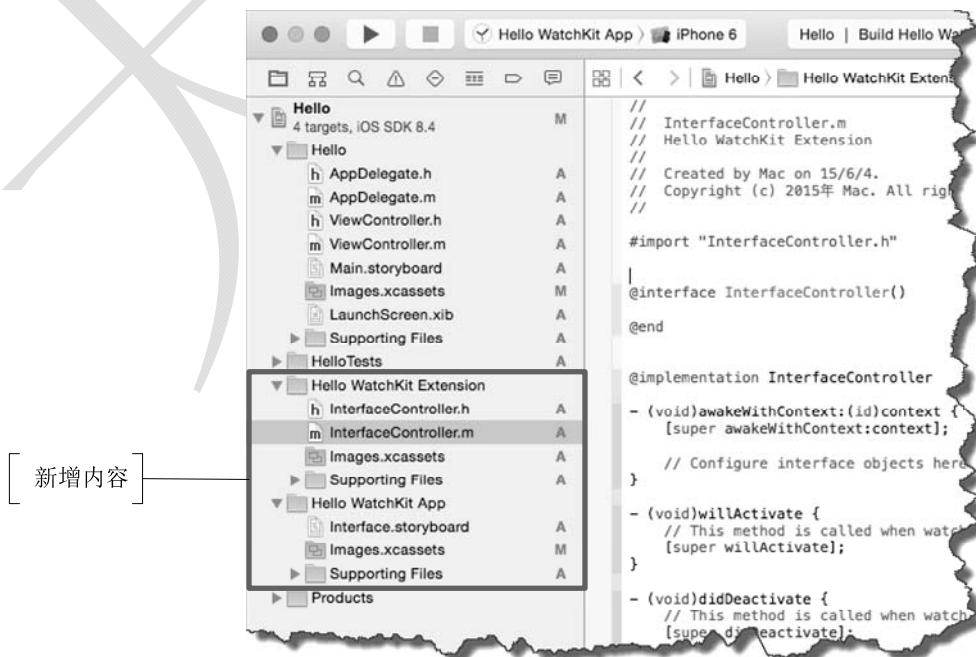


图 2.18 新增部分

这两个部分之间就是通过 WatchKit 进行连接通讯，用户点击 Watch App 后，与 Watch 匹配的 iPhone 会启动 WatchKit extension，然后和 Watch 建立连接，产生通信，如图 2.19 所示。

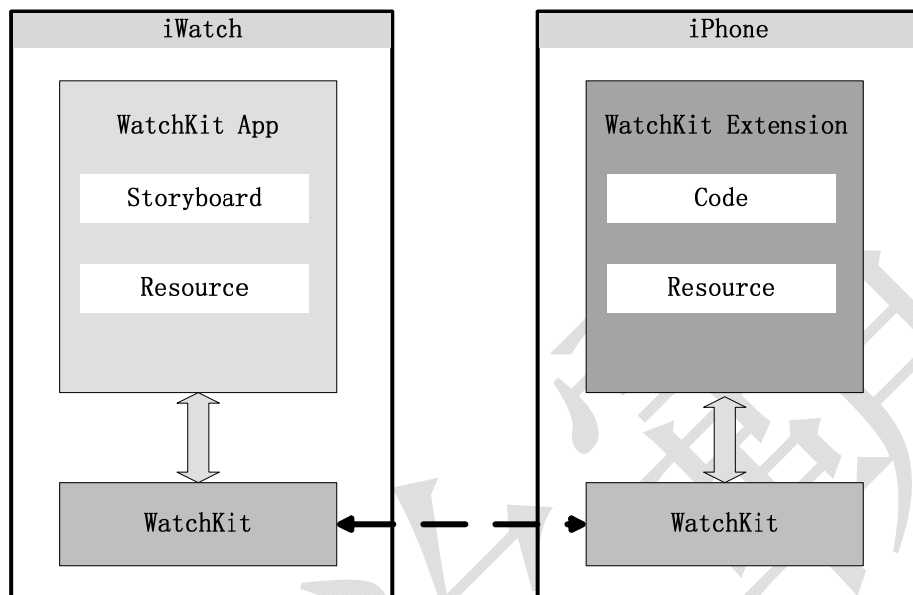


图 2.19 通讯

## 2.5.4 运行程序

将 Watch 应用对象添加到创建的项目中后，就可以运行 iWatch 程序了。运行程序其实很简单，开发者只需要单击运行按钮就可以实现运行了，如图 2.20 所示。在运行程序前首先对程序进行编译，如果程序正确，会出现一个 Build Succeeded 字符串，如图 2.21 所示。如果程序出现错误，那么就会出现一个 Build Failed 字符串，如图 2.22 所示。



图 2.20 运行按钮

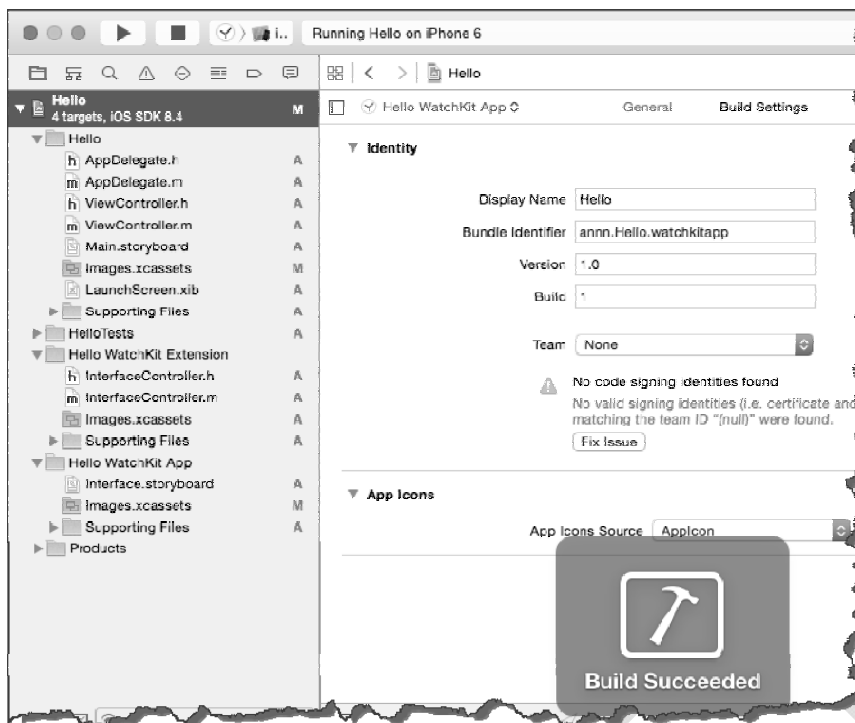


图 2.21 编辑成功



图 2.22 编译失败

在程序编译后，会自动对程序进行连接、运行，运行效果如图 2.23 和 2.24 所示。



图 2.23 运行效果

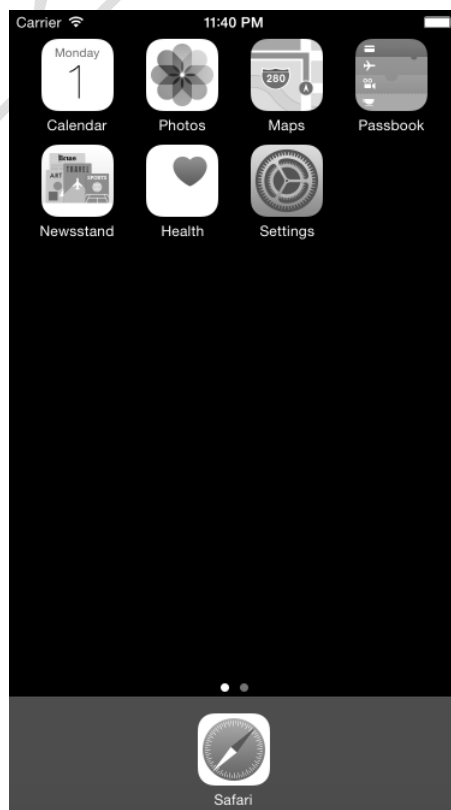


图 2.24 运行效果

注意：图 2.23 为 iWatch 模拟器即 Apple Watch 模拟器，用来对 iWatch 应用程序进行显现的。由于没有对程序进行编写，也没有对编辑界面进行设置，所以这时是不会产生任何效果的。图 2.24 为 iOS 模拟器，它与 iWatch 模拟器进行了关联。通常在运行 iWatch 应用程序时，它都是会出现的。

如果 iWatch 模拟器不出现，这里提供了两种解决方法：

### 1. 设置运行目标

将运行目标（Set the active scheme）设置为 Hello WatchKit App，如图 2.25 所示。

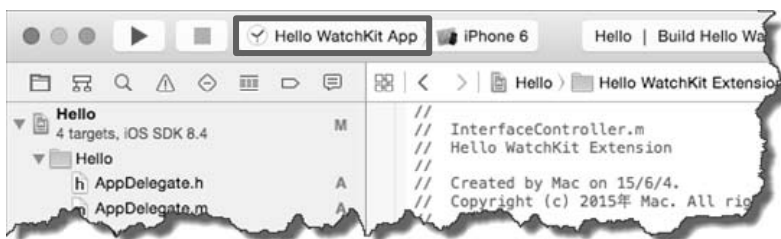


图 2.25 设置运行目标

### 2. 设置硬件

选择 iOS 模拟器，在出现的菜单中选择 Hardware|External Displays|Apple Watch-38mm 或者是 42mm 命令，如图 2.26 所示。

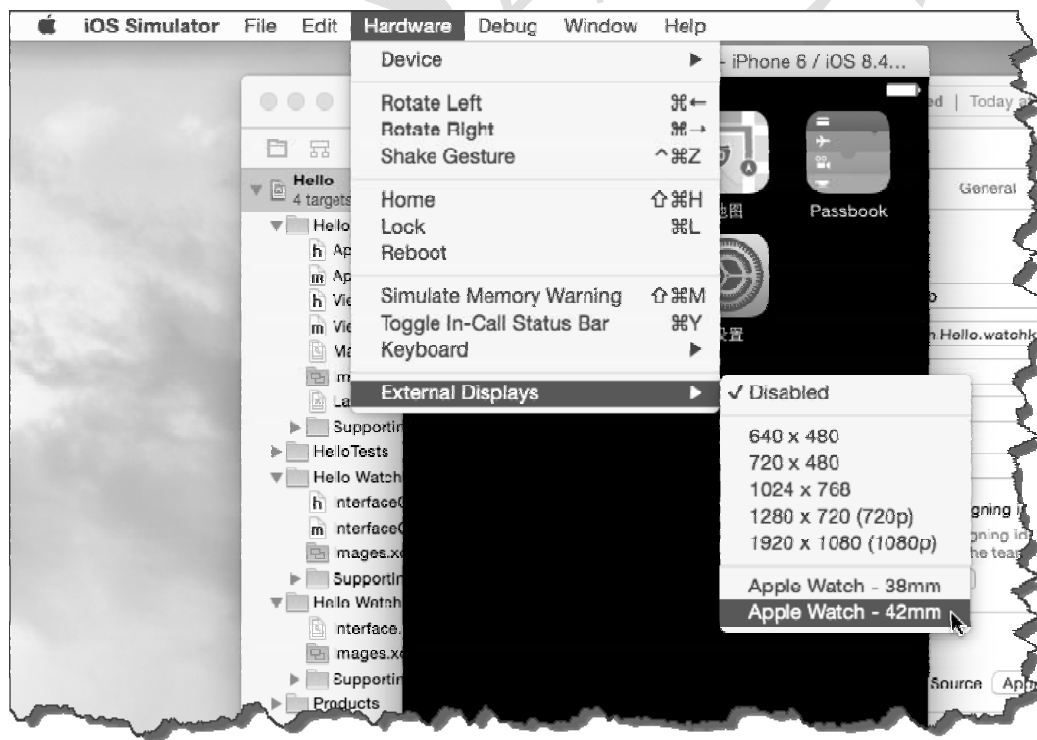


图 2.26 设置硬件

## 2.5.5 编辑界面

在 2.5.4 小节中提到过编辑界面（Interface builder）。编辑界面是用来设计用户界面的，单击打开



Interface.storyboard 文件就打开了编辑界面。在 Xcode 6.4 中，编辑界面直接使用是故事板。本节将对编辑界面进行介绍

## 1. 界面介绍

单击 Interface.storyboard 打开编辑界面后，可以看到编辑界面会有 4 部分组成，如图 2.27 所示。

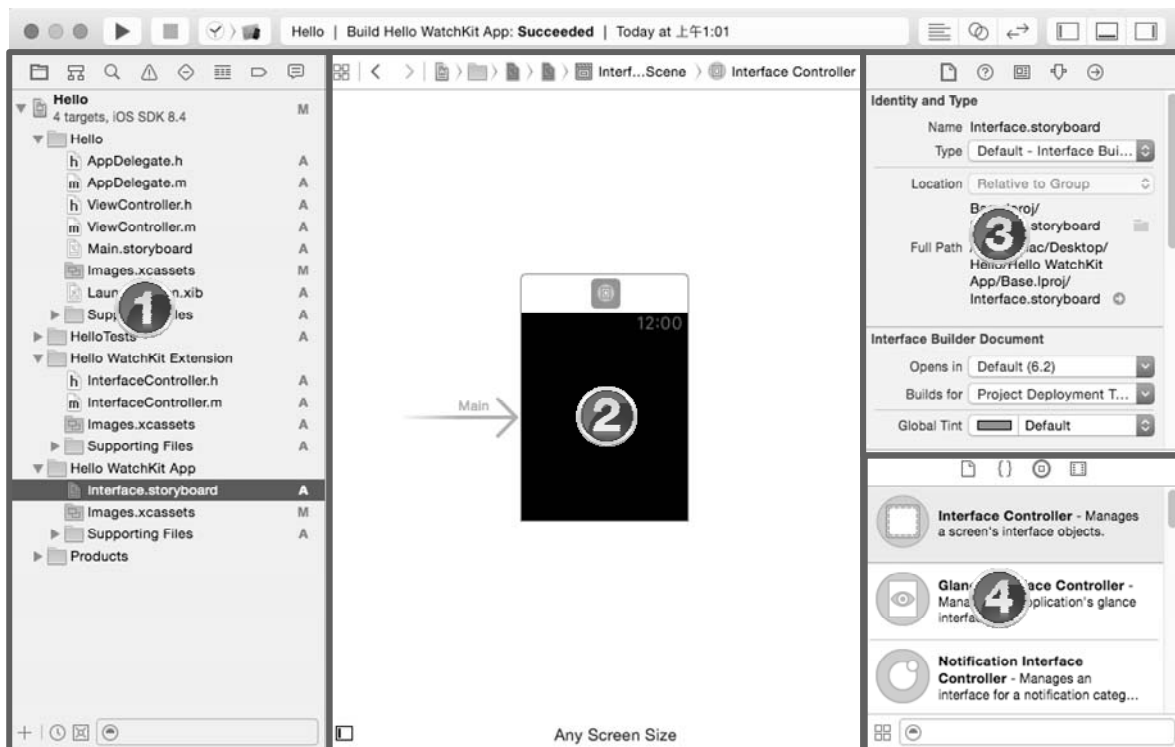


图 2.27 界面的组成

其中，这四部分的说明如下：

- ❑ 编号为 1 的部分为导航窗口。
- ❑ 编号为 2 的部分为画布：用于设计用户界面的地方，在画布中用箭头指向的区域就是界面，在画布中可以有多个界面。
- ❑ 编号为 3 的部分为组件窗口的检查器：用于编辑当前选择的对象的属性。
- ❑ 编号为 4 的部分为组件窗口的库：如果选择的是 Objects，里边存放了很多的对象。在画布的界面上方有一个小的 dock，它是一个文件管理器的缩写版。

注意：dock 展示界面中第一级的控件，每个场景至少有一个 Interface Controller

## 2. 设计界面

在 2.4.4 小节中提到了由于没有对编辑界面进行设计，所以导致在运行的效果中没有任何内容。本小节将会在 iWatch 模拟器上显示一个标签。具体的操作步骤如下：

- (1) 打开 Hello 项目中的 Interface.storyboard 文件。
- (2) 从对象库中找到 Label 即标签对象，将其拖动到界面中，如图 2.28 所示。

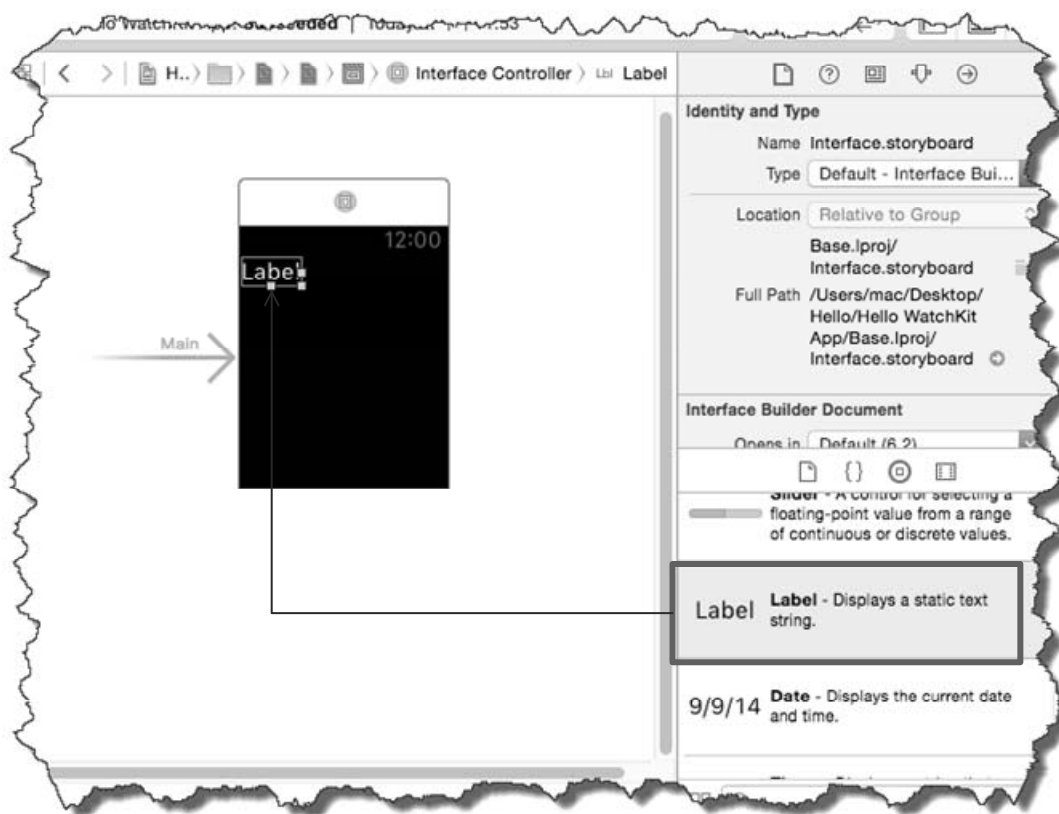


图 2.28 添加标签

此时运行程序，会看到如图 2.29 所示的效果。



图 2.29 运行效果

在设计界面时需要注意两点：

（1）设计界面尺寸

在图 2.26 中我们不难看出 iWatch 模拟器有两种尺寸一个是 38mm 一个是 42mm，对应的编辑界面中的界面也可以将尺寸设置为 38mm 或者 42mm，设置界面尺寸的具体操作步骤如下：

首先，打开 Interface.storyboard 文件，如图 2.30 所示。

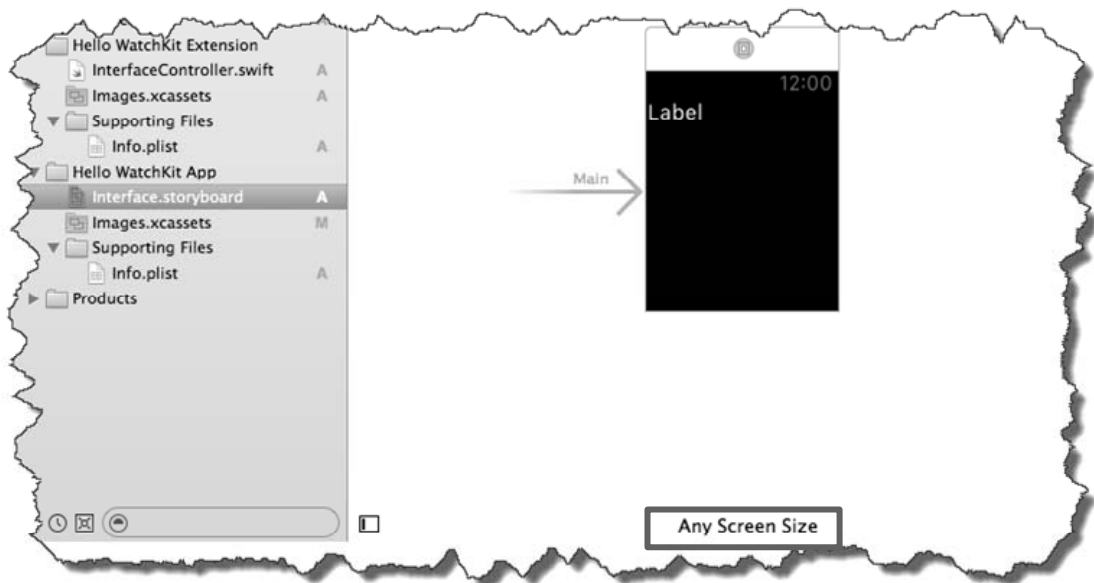


图 2.30 设置尺寸 1

然后，选择 Any Screen Size 选项，此时会出现一个对话框，如图 2.31 所示。



图 2.31 设置尺寸 2

最后选择此对话框中的任意尺寸就实现了界面尺寸的设置。

## （2）设置对象的风格

开发者可以在打开的 Show the Attributes inspector 即属性查看器面板中对标签的颜色，字体等进行设置。

## 2.5.6 编写代码

代码就是用来实现某一特定的功能而用计算机语言编写的命令序列的集合。现在就来实现通过代码在标签对象中显示“Hello,World”字符串，操作步骤如下：

（1）使用设置编辑器的三个视图方式的图标，如图 2.32 所示，将 Xcode 的界面调整为如图 2.33 所示的效果。



图 2.32 编辑器的三个视图方式的图标

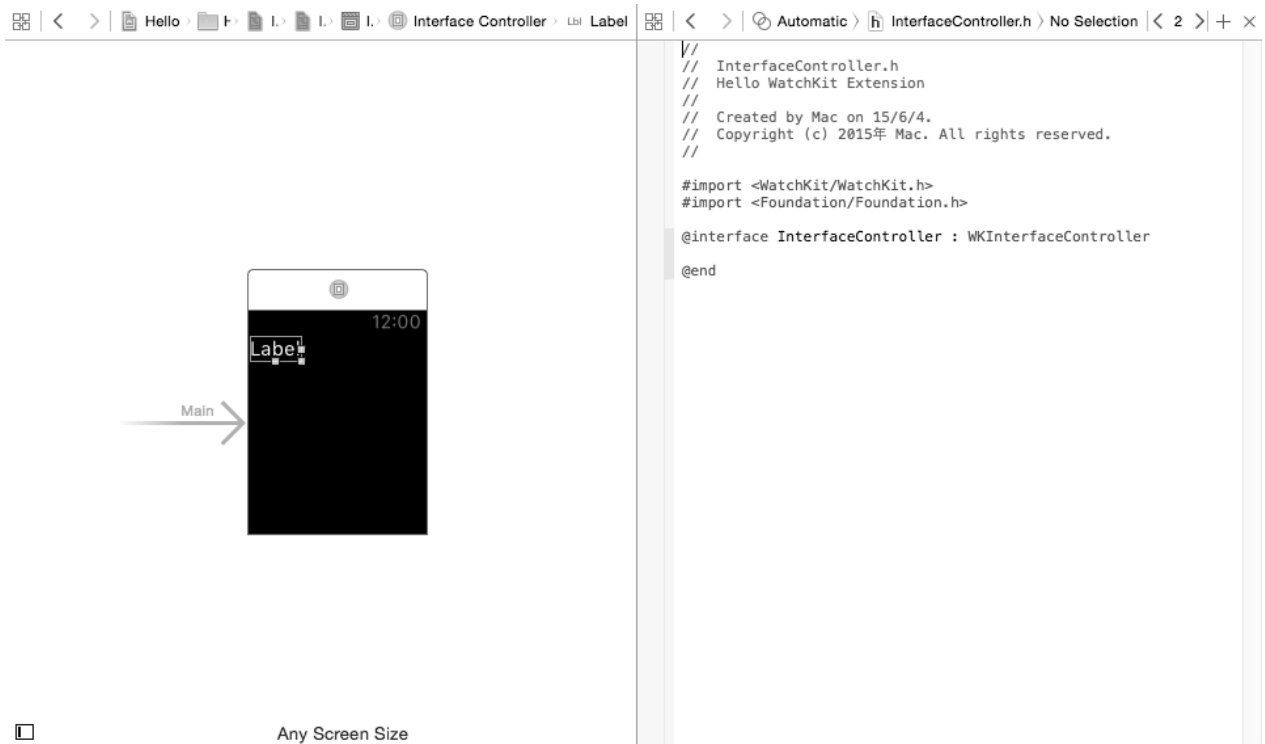


图 2.33 操作步骤 1

(2) 按住 Ctrl 键拖动界面中的标签对象，这时会出现一个蓝色的线条，将这个蓝色的线条拖动到 Hello WatchKit Extension 文件夹下的 InterfaceController.h 文件中，如图 2.34 所示。

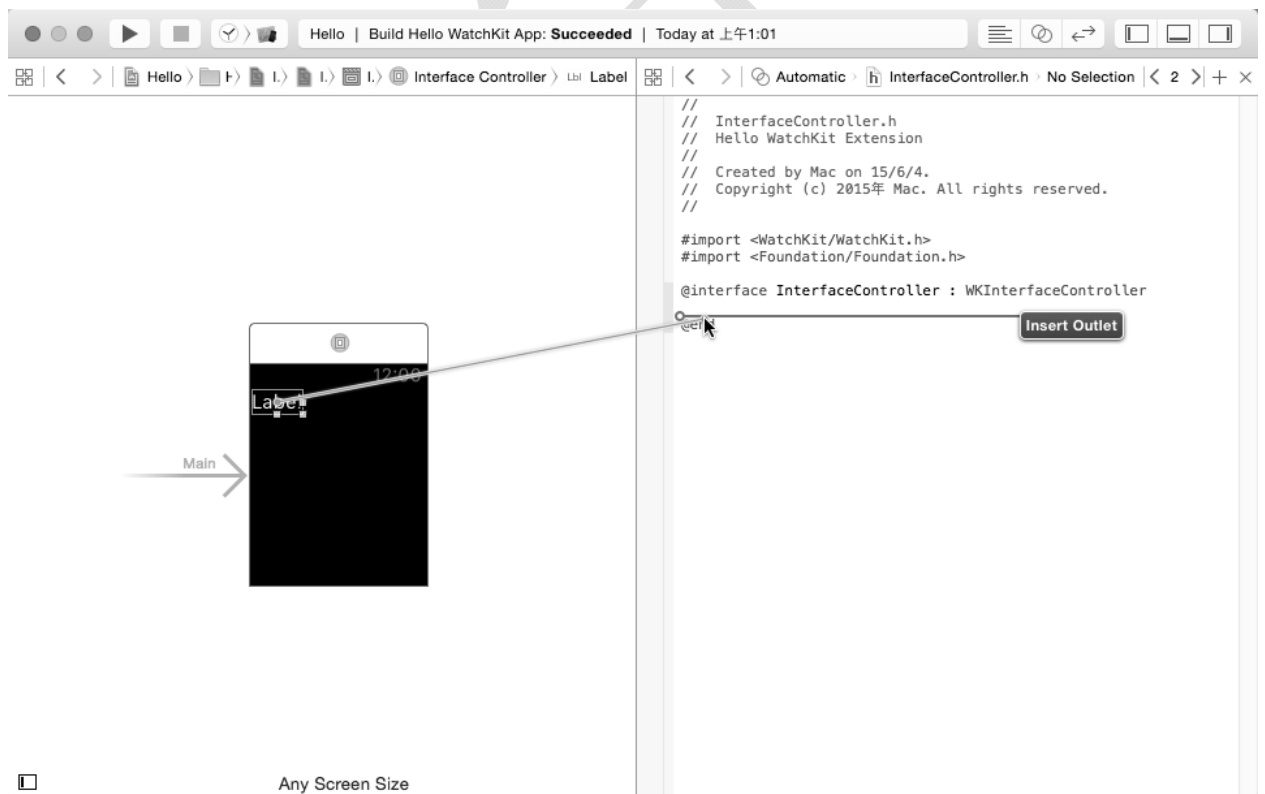


图 2.34 操作步骤 2

(3) 松开鼠标后，会弹出声明和关联插座变量一起进行的对话框，如图 2.35 所示。

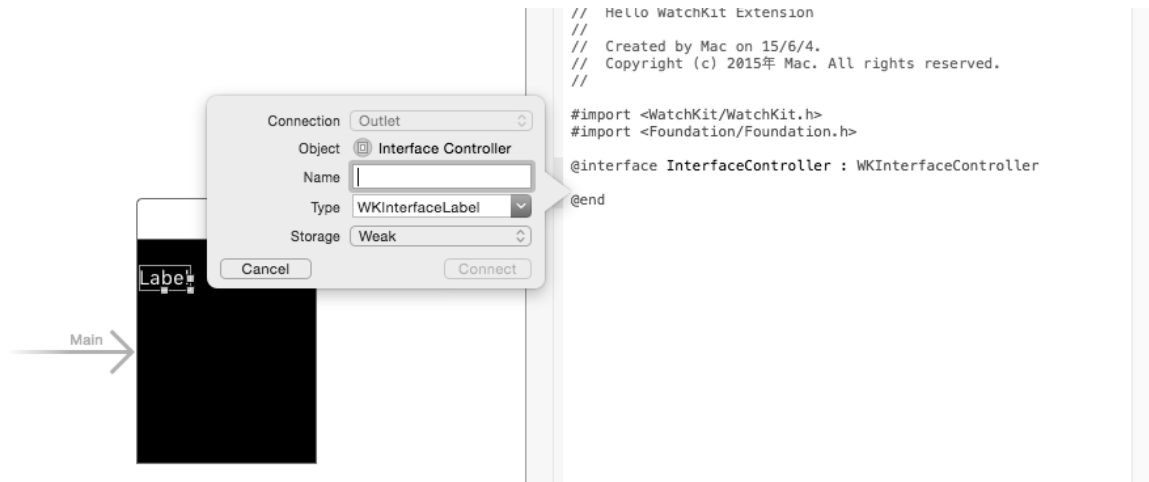


图 2.35 操作步骤 3

(4) 将 Name 设置为 label，如图 2.36 所示。

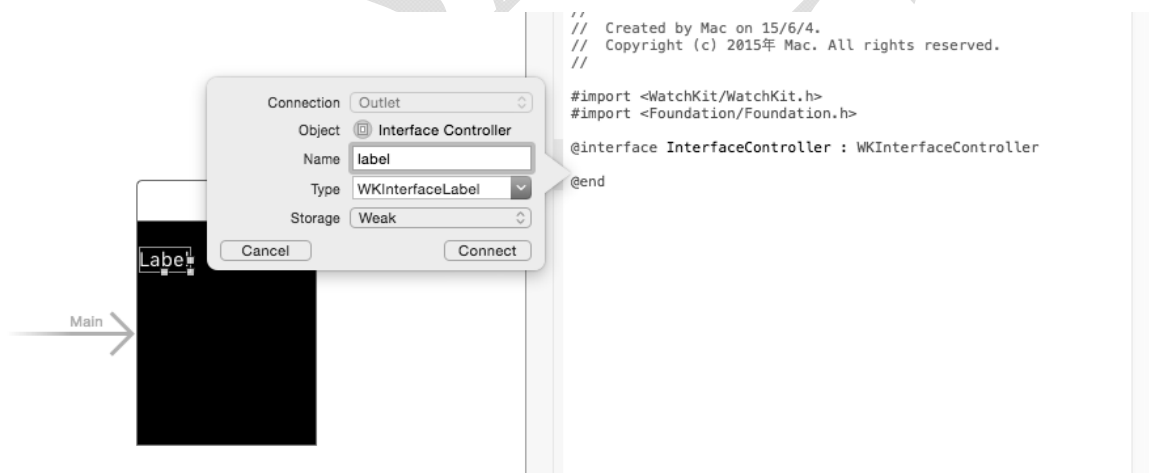


图 2.36 操作步骤 4

注意：这里的 Name 可以是任意的。

(5) 单击 Connect 按钮，会在 InterfaceController.h 文件中看到如图 2.37 所示的代码。

```
//
//  InterfaceController.h
//  Hello WatchKit Extension
//
//  Created by Mac on 15/6/4.
//  Copyright (c) 2015年 Mac. All rights reserved.
//

#import <WatchKit/WatchKit.h>
#import <Foundation/Foundation.h>

@interface InterfaceController : WKInterfaceController

* @property (weak, nonatomic) IBOutlet WKInterfaceLabel *label;

@end
```

插座变量

图 2.37 生成的代码

注意：生成的代码叫做插座变量，插座变量其实就是为关联的对象起了一个别名，开发者就可以对此插座变量进行操作，从而对关联的对象进行操作。

（6）打开 InterfaceController.m 文件，编写代码，此代码实现的功能是将标签中显示的内容变为 Hello,World 字符串。代码如下：

```
#import "InterfaceController.h"
@interface InterfaceController()
@end
@implementation InterfaceController
    @synthesize label;
- (void)awakeWithContext:(id)context {
    [super awakeWithContext:context];
    // Configure interface objects here.
    [label setText:@"Hello,World"]; //设置标签显示的内容
}
..... //这里省略了手表控制器即将可见时调用的方法和不再可见时调用的方法
@end
```

此时运行程序，会看到如图 2.38 所示的效果。

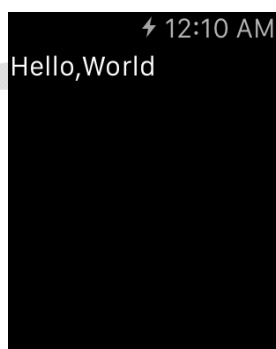


图 2.38 运行效果

## 2.6 调试程序

调试又被称为排错，是发现和减少程序错误的一个过程。在 Xcode 中进行调试的需要实现以下几个

步骤：

### 1.添加断点

在进行程序调试之前，首先需要为程序添加断点，断点是调试器应该停止程序的运行并让开发者可以运来查看成的地方。将光标移到到要添加断点的地方，按住 **Command+**\键或者选择菜单栏中的“**Debug|Breakpoints|Add Breakpoint at Current Line**”命令进行断点的添加，之后会在添加断点代码的最左边看到一个蓝色箭头，这就是一个新断点，如图 2.39 所示。

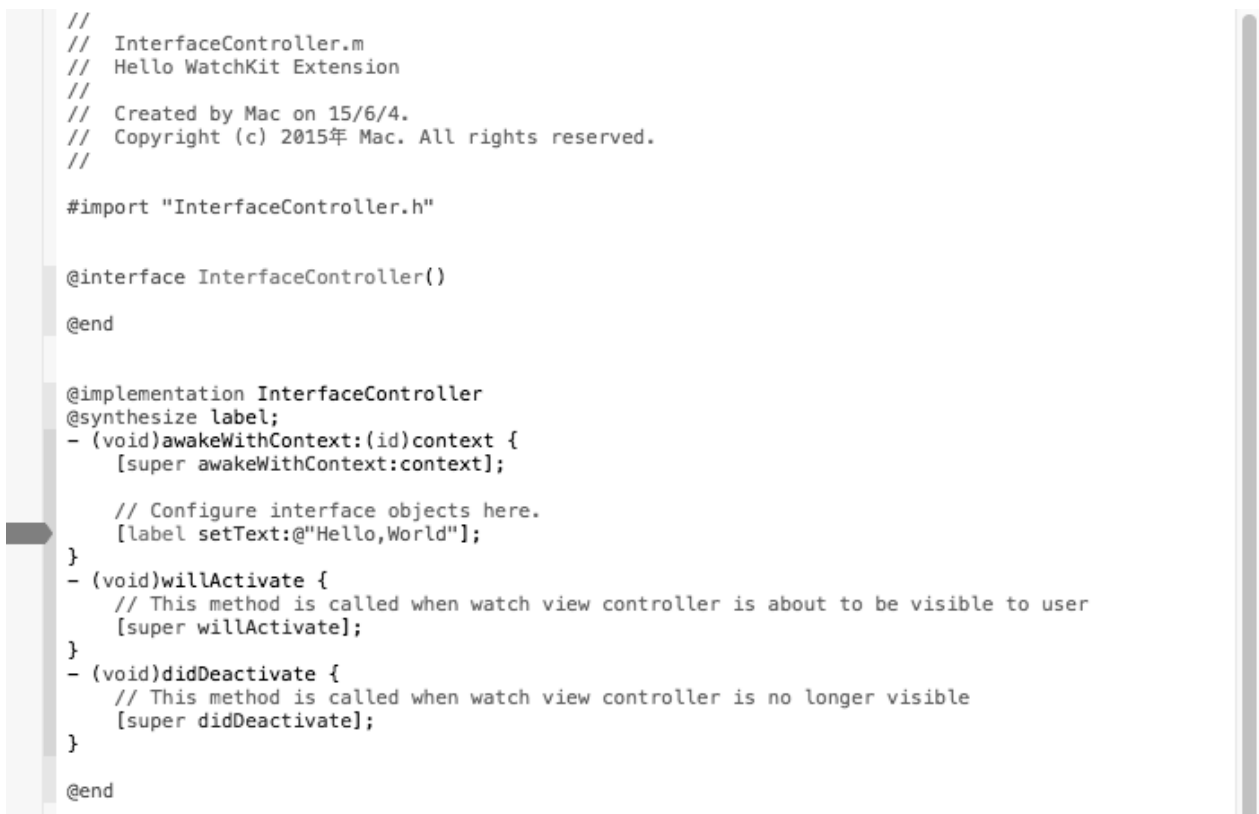


图 2.39 添加断点

### 2.运行程序

单击运行按钮后，程序就会运行。这时运行的程序会停留在断点所在的位置处，并且此代码行会出现绿色的箭头，表示现在程序运行到的位置，如图 2.40 所示。不仅如此，iWatch 模拟器也会显示，但是没有内容。

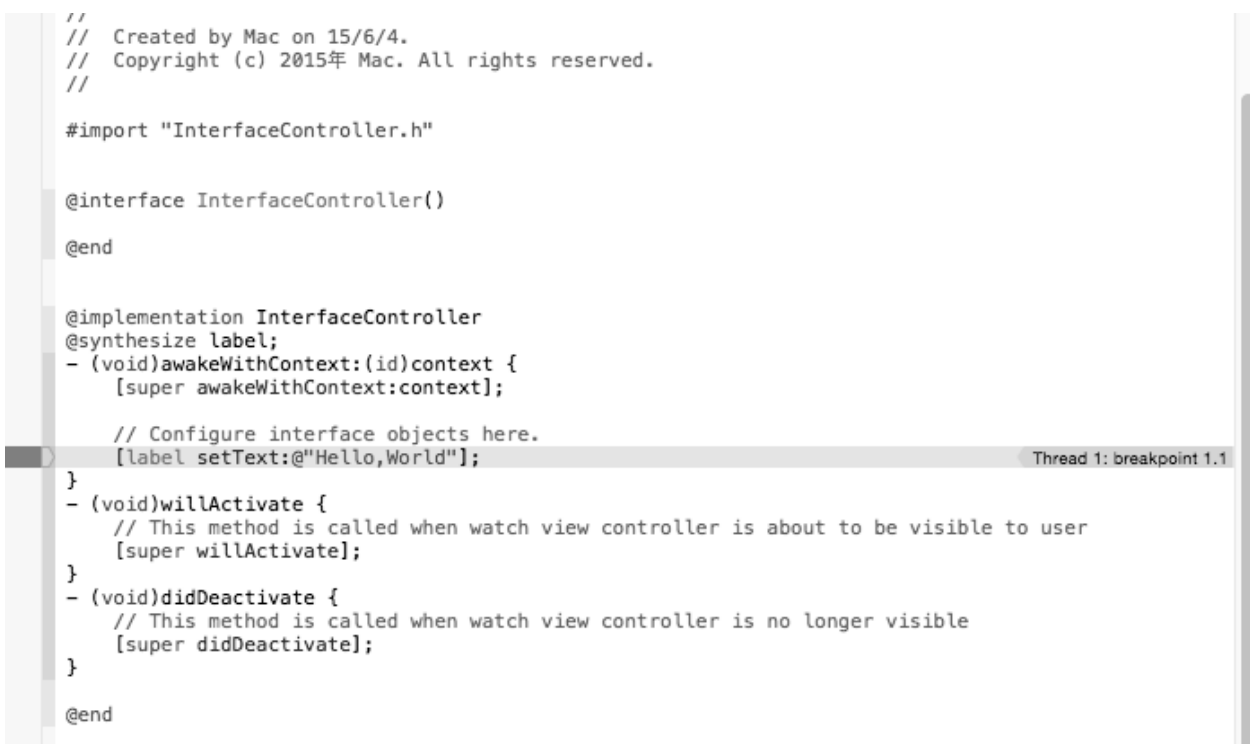


图 2.40 执行断点

### 3.断点导航

在程序停留下来后，程序调试信息窗口就会出现，里面显示了一些调试信息。在程序调试信息窗口顶端，会出现断点导航，如图 2.41 所示。

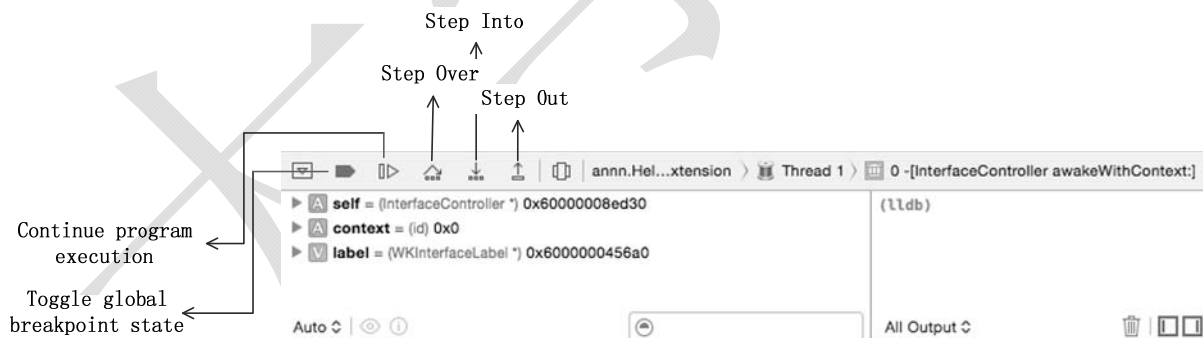


图 2.41 断点导航

- ☐ Toggle global breakpoint state 按钮：选择要查看的线程。
- ☐ Continue program execution 按钮：继续执行当前的代码，如果有下一个断点，就停止在下一个断点上。
- ☐ Step Over 按钮：执行下一个代码。如果当前行是方法调用，则不会进入方法内部。
- ☐ Step Into 按钮：进入方法内容。
- ☐ Step Out 按钮：跳过当前方法，即执行到当前方法的末尾。

这时，单击断点导航中的 Continue program execution 按钮，继续执行当前的代码。如果这时程序出现错误就不会跳到下一断点处；如果程序没有问题就会继续向下执行。现在只有一个断点，单击此按钮，程序会输出最后的结果。



#### 4.删除或废弃断点

如果，程序没有问题，那么，就要将程序中设置的断点进行删除或者废弃。删除断点长用到的方法有三种：

- ❑ 右击设置的断点，在弹出的快捷菜单中选择 Delete Breakpoint 命令。
- ❑ 选中设置断点的行，在 Xcode 的菜单栏中选择 Debug|Breakpoints|Remove Breakpoint at Current Line 命令。
- ❑ 选择断点，将其拖动到别的地方，这时，此断点就进行了删除。

要废弃断点，就是要单击断点，这时，断点就由深蓝色变为了浅蓝色。浅蓝色的断点就说明该断点已被废弃，如图 2.42 所示。

```
// Created by Mac on 15/6/4.
// Copyright (c) 2015年 Mac. All rights reserved.
//

#import "InterfaceController.h"

@interface InterfaceController()
@end

@implementation InterfaceController
@synthesize label;
- (void)awakeWithContext:(id)context {
    [super awakeWithContext:context];

    // Configure interface objects here.
    [label setText:@"Hello,World"];
}
- (void)willActivate {
    // This method is called when watch view controller is about to be visible to user
    [super willActivate];
}
- (void)didDeactivate {
    // This method is called when watch view controller is no longer visible
    [super didDeactivate];
}
@end
```

图 2.42 废弃断点

## 2.7 使用帮助文档

在编写代码的时候，可能会遇到很多的方法。如果开发者对这些方法的功能，以及参数不是很了解，就可以使用帮助文档。那么帮助文本该如何打开以及如何查找相关的内容呢？以下将为你解决这些问题。

### 1.打开帮助文档

要使用帮助文本，必须要对其进行打开。选择 Help|Documentation and API Reference 命令，就可以打开了，如图 2.43 所示。

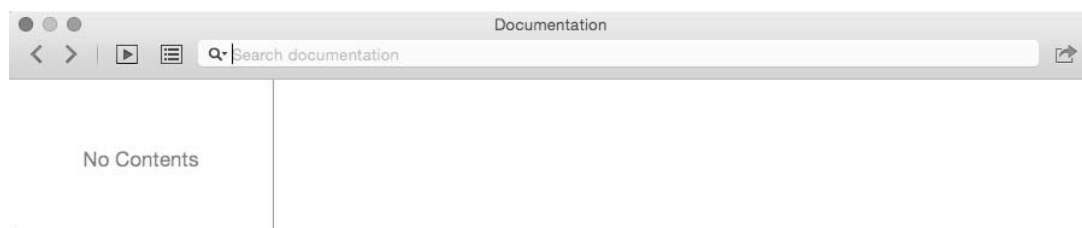


图 2.43 帮助文档

## 2. 查找

如果想要查找一个方法，可以在搜索栏中输入这个方法，如图 2.44 所示。回车后，便可以找到响应的内容。

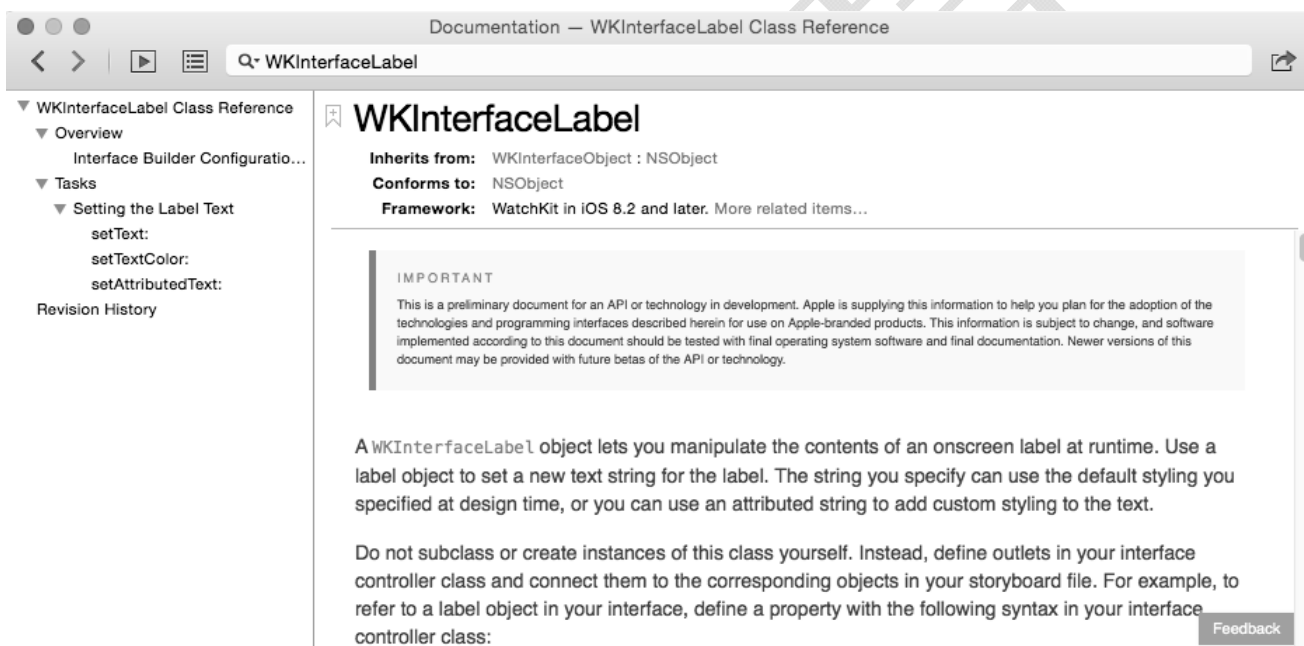


图 2.44 搜索内容