

第 1 章

全新的系统



就在 2007 年，史蒂夫·乔布斯在 MacWorld 大会上声称运行在 iPhone 上的软件要比其他竞争者领先至少五年。从最初的版本发布到我们编写本书为止，苹果公司每年都在更新操作系统，并且还推出了另外两款支持 iOS 的新设备——iPad 和 Apple TV。由于 iPhone 以外的其他设备也能运行该操作系统，因此它被更名为 iOS。现在，五年时间已经过去了，乔布斯的话果真应验了：iOS 依然是最先进的操作系统。而 iOS 6 加入的新功能——包括一些基于开发者的特性，比如集合视图、自动布局以及改进的数据隐私保护等——使这个系统得到了进一步提升。

这本书主要介绍 iOS 6 编程。本章，我会简单讲解一下 iOS 6 的主要功能，顺便告诉你去哪些章了解详细内容。

1.1 新功能

每个版本的 iOS 都会引入几个关键功能和其他一些小幅改动。iPhone OS 2 的时候开始支持公共的 SDK 开发工具。

iPhone OS 3 将 Mac 上的 Core Data 功能加到了 iPhone 上，此外还有一些其他改进，比如推送通知、ExternalAccessory 工具集、用于应用内购买的 StoreKit.framework、程序内的电子邮件表单、能让开发者在程序内嵌入谷歌地图的 MapKit.framework、对 iPod 库的只读访问机制以及钥匙串数据共享。

小幅更新的 3.1 系统中还添加了视频编辑支持，而 3.2 系统的小幅（而重要的）更新中添加了 Core Text 和手势识别、文件共享以及生成 PDF 文件的支持。3.2 系统还添加了对一个全新产品 iPad 的支持，支持开发可以运行在 iPad 上的应用，而通用的应用可以同时运行在 iPad（3.2）和 iPhone（3.1.3）中。不过，3.2 系统只支持 iPad，无法运行在 iPhone 以及 iPod touch 设备上。

iPhone OS 4（之后更名为 iOS 4）引入了期待已久的多任务支持、本地推送通知、对日历的只读访问机制（EventKit.framework）、代码块（block）和 Grand Central Dispatch、程序内短消息发送以及对 Retina 屏幕的支持。这个版本只支持 iPhone，不支持为 iPad 开发应用。之后更新的 4.2 系统可以同时支持 iPhone 和 iPad。

iOS 5 引入了一些重要的功能，比如 iCloud、自动引用计数、内置 Twitter 框架和其他一些更新。

iOS 6 引入了 Pass 工具集，它是支持提供商将门票、通行证和优惠券直接推送到我们 iPhone 中的一个框架。比方说，iOS 旅馆预定应用可以将登记证明以凭证的形式发送到用户的 iPhone 上。

iOS 6 还向用户提供了全新的地图应用。对程序员来说，可以使用新的 SDK 来显示你的位置。对 MKMapView 类调用 `openMapsWithOptions:launchOptions:` 方法可以打开地图：

```
+ (BOOL)openMapsWithItems: (NSArray *)mapItems launchOptions:  
    (NSDictionary *)launchOptions
```

不过，iOS 6 不支持通过 URL scheme 打开地图程序。因此你需要在 iOS 5 设备上转用谷歌地图，如下代码片段所示：

```
if ([[UIApplication sharedApplication] canOpenURL:  
    [NSURL URLWithString:@"maps"]]) {  
    //特定于 ios 6  
    [MKMapItem openMapsWithItems:<items> launchOptions:<options>]; } else  
{  
    //针对 ios5, 回到谷歌地图方式  
    [[UIApplication sharedApplication] openURL:  
    [NSURL URLWithString:@"http://maps.google.com/<your url>"]]  
}
```

本书着重讲解 iOS 6 中添加的新功能。在接下来的几节中，将介绍 iOS 6 中添加的关键特性，对应用开发加以指点。

1.1.1 集合视图

可以说，iOS 6 中最重要的新功能是用来显示并管理集合视图的全新控制器。集合视图（collection view）提供了一种类似栅格视图的 UI 元素布局方式，就像内置的照片和 iBooks 应用一样。你不再需要依赖第三方的栅格视图框架来创建栅格和使用滚动视图。集合视图还提供了用动画来展示内容的简单方式（类似于 UITableView 的动画），并能对它们执行添加、删除和重新排序操作。（第 7 章将详细介绍集合视图。）

1.1.2 自动布局

自动布局（Auto Layout）是一种新的布局方式，可以决定用户界面上对元素进行布局的新规则。Mac OS X SDK 在去年引入了这个功能，并将其移植到了 iOS 6 中。自动布局比我们以前使用的 Spring 和 Struts 模型框架更加直观，并且内置了对反向语言（从右到左阅读）的镜像 UI 元素的支持。可以说，自动布局带来的最好功能是可以本地化 nib 文件的时候更换文字内容。

1.1.3 自动引用计数

虽然 ARC（Automatic Reference Counting，自动引用计数）是在 iOS 5 中引入的，但是有关 ARC 的信息和 LLVM 编译器的内部工作原理都有了极大的改善，其中包括属性的默认存储类型。本书这一版添加了全新的一章（第 5 章）来讲解 ARC，包括 ARC 的内部工作原理以及如何使用 Convert to Objective-C ARC 工具来转换代码，以使其支持 Objective-C 的 ARC 特性。

ARC 与 Mac OS X 10.5 版本（Leopard）提供的垃圾回收机制不同。垃圾回收是自动内存管理，而 ARC 是自动引用计数。这意味着你作为一名程序员，不需要为每个 retain 语句编写相应的 release 代码，编译器会自动添加这些内容的。

4 第 1 章 全新的系统

ARC 添加了两个新的生命周期修饰符，即 `strong` 和 `weak`，并采用了全新的规则。举个例子，你不需要再为任何对象调用 `release` 和 `retain` 语句。`dealloc` 方法也同样遵循这个规则。当你使用了 ARC，只需要在 `dealloc` 方法中释放资源（文件或端口），而不必释放实例变量。（更多关于 ARC 的内容，请参阅第 5 章。）

1.1.4 应用内购买的托管内容

iOS 6 允许你使用苹果的应用内购买功能提供可下载的内容。举个例子，假设你创建了一个赛车游戏，并且想要通过应用内购买出售跑车和赛道，可以使用苹果公司的服务器托管赛道的贴图以及跑车的模型文件。当购买完成后便启动一次下载，而且你不需要为托管这些文件承担任何成本。（要了解应用内购买的更多知识以及如何下载托管内容，请参阅第 18 章。）

1.1.5 社交框架

iOS 5 将 Twitter 的体验直接集成到了操作系统中。这使得从应用中发送一条推文就像使用程序内置的电子邮件表单发送电子邮件一样简单。这个框架还提供了身份验证功能。iOS 5 上的 Twitter 框架以及 Account 框架一起提供了对账户的验证功能。而在 iOS 6 中，苹果使用 `Social.framework` 替代了 `Twitter.framework`，它更加通用，并且除 Twitter 以外还能处理其他社交网络，比如 Facebook 和新浪微博。对于程序员来说，Facebook 紧密地结合在操作系统中了，发送 Facebook 消息以及账户认证的过程因此变得更加简单，并且不需要再使用 Facebook 的官方 Graph SDK 了。不过，如果你想支持 iOS 5 设备，那么仍然需要 Facebook 的 Graph SDK。`UIKit.framework` 新引入了一个叫作 `UIActivityViewController` 的视图控制器，它和 `Social.framework` 一起将信息发送到社交网络上。

1.1.6 UI 状态保留

在 iOS 3 和更早的版本中，恢复 UI 状态是一件难事。苹果在 iOS 4 中引入了多任务后，应用就能在进入后台的时候自动保留 UI 的状态。但是低内存环境会关闭你在后台的程序进程，并且你的应用不会接收到通知。当程序在后台被关闭后，就很难恢复 UI 的状态了，而恢复 UI 状态的唯一办法是实现各种代码，也就是在 iOS 4 之前程序员用来恢复 UI 的代码。而在 iOS 6 中，通过 `UIKit` 对状态恢复提供的支持，苹果公司令状态保留更加简单了。（参阅第 13 章可以获取更多关于 UI 状态保留的知识。）

1.1.7 其他新功能

iOS 6 还引入了其他一些简单的功能，包括 Pass 工具集、增强的隐私功能、支持与提醒事项程序交互，更重要的改进就是 `UITextField` 可以支持富文本了。

1. Pass 工具集

Pass 工具集是一个新的框架，允许用户在 iOS 设备上存储像优惠券、登机证和门票之类的东西，以代替实物。凭证是由服务器创建的，并被发送到用户的设备上，作为一名 iOS 程序员，你不需要写

代码来控制它们。Pass 工具集在 iOS 6 中是一个重要功能,它更像 APNS(Apple Push Notification Service, 苹果推送通知服务),主要依赖服务器端,而在客户端几乎不用写什么代码。因此,本书不会讨论 Pass 工具集。

2. 提醒事项

iOS 4 引入了 `EventKit.framework` 框架,它可以让第三方程序访问用户的日历。而在 iOS 6 中, `EventKit.framework` 还拥有了与用户的提醒事项程序交互的方法。

3. 隐私

隐私一直是应用很重视的问题。在 iOS 6 之前,一些应用(包括知名度较高的社交网络公司,比如 Path)会不经用户明确同意,就把用户的通讯录信息上传到自己的服务器。而在 iOS 6 中,访问通讯录和照片库(分别通过 `AssetLibrary.framework` 和 `UIImagePickerController` 访问),还有提醒事项程序和日历,都需要用户明确的许可。也就是说,如果你曾经访问过这些信息,那么现在可能会从用户那里获得空数据。你还可以在应用程序的 `Info.plist` 文件中设置这些键的值来说明为什么要访问这些信息: `NSLocationUsageDescription`、`NSPhotoLibraryUsageDescription`、`NSCalendarsUsageDescription`、`NSContactsUsageDescription` 和 `NSRemindersUsageDescription`。你也可使用图 1-1 中展示的选项来利用 UI 设置值。

Key	Type	Value
Information Property List	Dictionary	(15 items)
Localization native development region	String	en
Bundle display name	String	\$(PRODUCT_NAME)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	com.steinlogic.\$(PRODUCT_NAME:rfc1034identifier)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
Main storyboard file base name	String	MainStoryboard
Required device capabilities	Array	(1 item)
Preferences sync exclusion keys	String	(3 items)
Principal class		
Privacy - Calendars Usage Description		
Privacy - Contacts Usage Description		
Privacy - Location Usage Description		
Privacy - Photo Library Usage Description		
Privacy - Reminders Usage Description		
Quick Look needs to be run in main thread		
Quick Look preview height		
Quick Look preview width		

图 1-1 Info.plist 显示了新添加的 Usage Description 键

4. 显示富文本

`UITextView` 和 `UITextField` 都支持 `NSAttributedString`,而你可以通过创建 `NSAttributedString` 来显示富文本。你需要使用 `CoreText.framework` 来创建一个 `NSAttributedString`。(关于 Core Text 的讨论,请参见第 26 章。)

1.2 小结

iOS 的采用率一直都领先于它的竞争者。两年前，iPhone OS 3 发布的时候，因为需要付 10 美元来升级，所以采用率在 iPod touch 上受到了一定的影响。不过，苹果公司很快就让它免费了，而采用率也得到了上升。与之类似，当苹果公司发布 iOS 4 的时候，采用率一开始也是上升缓慢，因为它在旧手机（也就是 iPhone 3G 和第一代 iPhone，当然也包括 iPod touch）上存在性能问题。一些功能（尤其是多任务）也不支持旧设备。虽然如此，最新版本的 iOS 在最初发布的一两个月内通常可以达到 90% 以上的设备采用率。iOS 5 没有受影响，采用率上升速度也比 iOS 4 快，这也许是因为它在旧手机（比如 iPhone 4）上不会遗漏什么功能。此外，旧设备（比如 iPhone 3GS）在 iOS 5 上并不像在 iOS 4 上那样慢。而简洁的通知、iTunes、WiFi 同步和 iMessage 都是加速用户采用 iOS 5 的关键功能。

iOS 5.1 是第一个支持 OTA（Over-The-Air）的升级，采用率甚至比 iOS 5 还要高。到了 iOS 6，我估计会有同样高的采用率。所有一切都意味着，为了让应用更加耀眼，你要尽快使用 iOS 6 的每一个功能。像集合视图以及使用自动布局支持更高的新 iPhone 5 等功能，就足以成为你将应用升级到 iOS 6 版本的理由。废话不多说，是时候开始 iOS 6 之旅了。

1.3 扩展阅读

苹果文档

下面的文档位于 iOS Developer Library（<https://developer.apple.com/library/ios/navigation/index.html>）中，通过 Xcode Documentation and API Reference 也可以找到。

- *What's new in iOS 6*