iPhone Human Interface Guidelines

User Experience

iPhone人机界面导引（交互设计指南）

版本说明

本英文文本来源于apple开发者官方网站

中文翻译由浙江科技学院交互设计与用户研究实验室师生共同完成

鉴于译者水平，有不妥之处欢迎指正

联系电邮：pemiamos@gmail.com

twitter：@pemiamos

版本修订时间：2010年8月

2010年8月3日Apple又更新了新的的HIG版本，本文档更新并添加了多线程和iAd部分，声音设置与自定义图标部分新版更新较多，本翻译稿因为时间关系未全面更新。将在下一版翻译稿中完成这些内容。

参与翻译工作的有：

华梅立 / 郑一帆 / 邵东君 / 骆航兵 / 陈俊 / 李静静 / 吴佳 / 俞俊琛 / 费妍 / 叶晨 / 汤超 / 包德福 / 施越锋 / 陈珊

编辑校对：华梅立

总体规划：卢艺舟

鸣谢：

SURECCO™ STUDIOS

Contents 内容

Chapter 10 System-Provided Buttons and Icons
系统预制的按钮与图标

Chapter 11 Creating Custom Icons and Images
定制图标与画面

Introduction 简介

iPhone and iPod touch are sophisticated devices that combine the revolutionary Multi-Touch interface with powerful features, such as email and instant-messaging capability, a full-featured web browser, iPod, and, in iPhone, a mobile phone. iPhone OS is the system software that runs on iPhone and iPod touch. With the advent of the iPhone SDK, these powerful features are extended to include significant developer opportunities. In addition to creating web content for use on iPhone OS–based devices, developers can use the iPhone SDK to create native applications people can store and use on their devices.

iPhone及iPod touch是融合了革命性的多点触摸界面及强大特性的尖端科技设备，其包含了诸如电子邮件、即时通讯功能、全性能网络浏览器、iPod音乐播放器、以及iPhone本身——一个移动电话。iPhone OS作为系统软件运行于iPhone及iPod touch。借助于iPhone SDK的到来，那些强大特性的扩展带来影响深远的开发者机遇。除了在iPhone OS设备上为用户创造网页内容，开发者还可以使用iPhone SDK来创造本地应用程序，是人们能在自己的设备上存储与使用。

Read this document to learn about the range of application types you can develop for iPhone OS and the human interface design principles that form the foundation of great iPhone applications. In this document you learn how to follow those principles as you design a superlative user interface and user experience for your iPhone application. Whether you’ re an experienced computer application developer, an experienced mobile-device application developer, or a newcomer to the field, the guidelines in this document will help you produce iPhone applications users want.

阅读本文档你可以到，能为iPhone 开发何种类型的应用程序，来自以优秀iPhone应用程序为基础的人机界面设计原则。在这个文档中还可以学到在为iPhone设计精彩的人机界面与用户体验时，如何通过遵循这些设计原则。无论你是有经验的桌面电脑软件开发者，还是有经验的移动设备应用程序开发者，抑或完全是个新手，本设计导引将帮助你制造用户想要的iPhone应用程序。

> Note: This document briefly summarizes web-based development for iPhone OS–based devices. For more in-depth information specific to designing web content for these devices, see *iPhone Human Interface Guidelines* for Safari Reference Library.
> 注：本文概述了用于基于 iPhone OS 设备的 Web 的开发。关于专门为这些设备设计 Web 内容的深入信息，参见*iPhone Human Interface Guidelines* for Safari Reference Library.。

Organization of This Document 文章结构

iPhone Human Interface Guidelines is divided into two parts, each of which contains several chapters:

“iPhone 人机界面导引”分为两个部分，各包含若干章节：

The first part, “Planning Your iPhone Software Product” describes the iPhone OS environment and the types of software you can develop for it. It also covers fundamental human interface design principles and describes how to apply these principles to the design of your iPhone application.
第一部分：“规划你的iPhone软件产品”。讲述了 iPhone OS 开发环境以及你可以进行开发的软件类型，同时也包含了基础的人机界面设计原则以及如何应用这些原则来设计iPhone 应用程序。

The second part, “Designing the User Interface of Your iPhone Application,” delves into the components you use to create the user interface of your iPhone application. It describes the various views and controls that are available to you and provides guidance on how to use them effectively.
第二部分：“设计你iPhone应用程序的用户界面”。深入讨论了那些用于创建 iPhone 应用程序用户界面的组件，并描述了可用的各种视图和控件以及如何有效利用它们的准则。

See Also

To learn how to code your iPhone application, read:
■ iPhone Application Programming Guide

To learn about designing a web application for iPhone OS–based devices, read:
■iPhone Human Interface Guidelines for Web Applications

Part 1 :

Planning Your iPhone Software Product

规划你的iPhone软件产品

This part of iPhone Human Interface Guidelines describes ways to think about designing and developing software for iPhone OS. Read the chapters in Part I to learn about the different types of software you can develop for iPhone OS and the design principles you can use to inform your work. You' ll also learn how to apply those principles to specific aspects and tasks in your application, so you can create a superlative product that provides an intuitive and compelling user interface.

此部分iPhone人机界面设计导引描述了iPhone OS下软件设计与开发的方法。阅读第一部分的这些章节可以了解到iPhone OS下可以开发不同软件类型及其如何在项目中应用这些设计原则。你还可以学习如何在不同的细节中应用这些设计原则，来帮助你创建具有易用且精彩用户界面的极致产品。

CHAPTER 1

The iPhone OS Platform: Rich with Possibilities

iPhone OS平台：丰富的可能性

iPhone OS supports numerous types of software, ranging from webpages that users view in Safari on iPhone to iPhone applications that run natively on iPhone OS–based devices. This chapter outlines the different types of software solutions you can create for iPhone OS–based devices.

iPhone OS支持很多种类的软件，从用户在iPhone上通过safari浏览器查看的网页到可在iPhone OS设备上运行自如的iPhone应用程序。这一章节概括了不同类型的软件解决方案，你可以用来设计基于iPhone OS系统的设备

If you’ re new to the platform, be sure to begin with the summary of differences between iPhone OS–based devices and computers given in the first section, “Platform Differences to Keep in Mind.” Although the information in that section is not comprehensive, it touches on the issues you need to be aware of as you design an iPhone application.

如果你是刚刚接触这个平台，首先需要明确以iPhone OS为基础的设备和前面提到过的计算机设备之间的差异性。尽管“需慎记平台特性”那个章节提到信息不是很全面，但是它所涉及的问题，是你在设计iPhone应用程序时必须注意的。

Then, to help you plan an iPhone application, this chapter describes ways to think about different application styles and the characteristics that define them. This chapter also describes how some of the bundled Mac OS X applications were transformed into versions appropriate for iPhone OS. If you have an existing computer application you’ d like to refashion for iPhone OS, understanding this process is key.

为了帮助你设计一个iPhone应用程序，这一章节介绍了一些方法来了解不同风格的程序和它们的特征。这一章节也介绍了如何将与Mac OS X系统捆绑的应用程序转换成适用于iPhone OS系统的版本。如果你想要将一个已有的电脑程序重新适用于iPhone OS系统，理解这个过程是非常关键的。

Device Characteristics to Keep in Mind 需谨记平台特性

An iPhone OS–based device is not a desktop or laptop computer, and an iPhone application is not the same as a desktop application. Although these seem merely common-sense statements, it is nonetheless paramount to keep them in mind as you embark on developing software for these devices.

一个基于iPhone OS的设备不是一个桌面或者一台笔记本电脑。IPhone 应用程序不等同于一个桌面应用程序。 尽管这些属于常识性道理，但是重要的是当你着手为这些设备设计软件的时候，你要将它们牢记在心里。

Designing software for iPhone OS–based devices requires a state of mind that may or may not be second nature to you. In particular, if the bulk of your experience lies in developing desktop applications, you should be aware of the significant differences between designing software for a mobile platform and for a computer.
为基于iPhone OS系统的设备设计软件需要一种思维状态，这种思维状态可能不能你的自然思维，尤其是你之前都是从事桌面系统的开发，你应该意识到为移动设备平台和为电脑设计软件是不一样的。

This section summarizes the concrete differences that have the highest potential impact on your design decisions. For detailed information on how to handle these and other issues in your iPhone application development process, see iPhone Application Programming Guide.
这一章节概括的一些具体要点，有很大的可能性会影响到你的设计决策。关于在你的iPhone应用程序开发过程中如何处理这些问题的细节，请查阅iPhone Application Programming Guide。

Screen Size is Compact 屏幕尺寸是受限的

The small, high-resolution screens of iPhone OS–based devices make them powerful display devices that fit into users’ pockets. But that very advantage to users may be challenging to you, the developer, because it means that you must design a user interface that may be very different from those you’re accustomed to designing.
小巧、高分辨率屏幕的iPhone OS设备使它们成为了可以放进使用者口袋的功能强大的显示设备。但是这一点对一个软件开发者来说可能是个挑战，因为它意味着你必须设计一个与你之前的设计完全不一样的用户界面。

Keep in mind the screen size of 480 x 320 pixels and use that as a motivation to focus the user interface on the essentials. You don’t have the room to include design elements that aren’t absolutely necessary, and crowding user interface elements makes your application unattractive and difficult to use.
记住屏幕的尺寸为480 x 320 像素点，使其作为专注于用户界面设计的驱动力。你将没有空间去放置那些不必要的设计元素，使用户界面因为过于拥挤变得不美观和不易操作。

Memory is limited 内存是有限的

Memory is a critical resource in iPhone OS, so managing memory in your application is crucial. Because the iPhone OS virtual memory model does not include disk swap space, you must take

care to avoid allocating more memory than is available on the device. When low-memory conditions occur, iPhone OS warns the running application and may terminate the application if the problem persists. Be sure your application is responsive to memory usage warnings and cleans up memory in a timely manner.
内存是iPhone OS里一个极端重要的资源。所以内存管理在你的程序设计里非常重要，因为iPhone OS的虚拟内存模型不包括磁盘交换空间。你必须注意避免内存使用超标。当内存不足时，iPhone OS会对正在运行的程序提出提醒，如果问题没有得到解决，它会终止该程序。确保你的程序回应内存使用提醒，并有及时的清理内存的机制。

As you design your application, strive to reduce the application's memory footprint by, for example, eliminating memory leaks, making resource files as small as possible, and loading resources lazily. See iPhone Application Programming Guide for extensive information about how to design iPhone applications that handle memory appropriately.
当你设计你的程序的时候，尽量减少应用程序的内存机体。例如，消除内存溢出，减少资源文件大小，延后装载资源。更多关于iPhone程序设计上的内存处理信息请参考iPhone Application Programming Guide。

People See One Screen at a Time 任一时间用户只有一个屏幕

One of the biggest differences between the iPhone OS environment and the computer environment is the window paradigm. With the exceptions of some modal views, users see a single application screen at a time on an iPhone OS–based device. iPhone applications can contain as many different screens as necessary, but users access and see them sequentially, never simultaneously.
iPhone OS环境和计算机环境最大的一个区别在于窗口范式。不同于其它的视图模式，iPhone OS上用户每次只能看到一个应用程序窗口。iPhone应用程序在必要时候可以容纳很多不同的窗口，但是使用者只能顺序进入并查看到每一个，而不能同时查看。

If the desktop version of your application requires users to see several windows simultaneously, you need to decide if there's a different way users can accomplish the same task in a single screen or a sequence of screens. If not, you should focus your iPhone application on a single subtask of your computer application, instead of trying to replicate a wider feature set.
如果应用程序的桌面版本需要使用者同时查看不同的窗口，你需要决定是否有不同的方式可以使使用者在单个窗口或者一个连续的窗口下完成相同的任务。如果不可以，你应该将你的iPhone应用程序作为你计算机程序的一个单独的子任务，而不是将它做成一个多种功能重复的集合体。

People Interact with One Application at a Time 一次只和一个程序交互

Only one application is visible in the foreground at a time. When people switch from one application to another, the previous application quits and its user interface goes away. Prior to iPhone OS 4.0, this meant that the quitting application was immediately removed from memory. In iPhone OS 4.0 and later, the quitting application transitions to the background, where it may or may not continue running. This feature, called multitasking, allows applications to remain in the background until they are launched again or until they are terminated.
每次只有一个应用程序显示在前台。当用户从一个应用程序切换到另一个应用程序时，之前的应用程序会退出，用户界面也会消失。在iPhone OS 4.0之前，退出应用程序意味着程序缓存立即从内存中清除。而从iPhone OS 4.0及之后的版本，退出的应用程序将过渡到后台，在那里暂时保存起来。这个特性，被称作“多线程”，其允许应用程序留驻后台，直到它们再次被调用或被完全终结。

> Note: Multitasking is available on certain devices running iPhone OS 4.0.
> 注意：运行iPhone OS 4.0的设备上提供了已经多线程。

Most applications enter a suspended state when they transition to the background. When people restart a suspended application, it can instantly resume running from the point where it quit, without having to reload its user interface.
当转入后台时，大多数应用程序会被暂停。当用户重新启动一个被暂停的应用程序时，该程序将立刻转入被暂停时的状态点，并且不需要重载用户界面。

Some applications might need to continue running in the background while users run another application in the foreground. For example, users might want an application that plays audio to continue playing even while they’ re using a different application to check their calendar or handle email.
当用户在前台使用另外的应用程序时，一些在后台的应用程序可能会继续运行。例如，用户可能会希望当他们使用不同的程序查看日历或处理邮件时，应用程序能继续播放音频。

To learn about how multitasking can impact your application’s behavior, see “Accommodating Multitasking” (page 46).
了解多线程对你的应用程序行为的影响，见 *“Accommodating Multitasking”*

Onscreen User Help is Minimal 屏幕上的用户帮助最小化

Mobile users don’ t have the time to read through a lot of help content before they can use your application. What’ s more, you don’ t want to give up valuable space to display or store it. A hallmark of the design of iPhone OS–based devices is ease of use, so it’ s crucial that you meet users’ expectations and make the use of your application immediately obvious. There are a few things you can do to achieve this:

移动用户在使用一个程序之前不会花很多时间去阅读一大堆的帮助说明文件。而且你也不希望花很多宝贵的空间去储存或显示这些内容。iPhone OS设备的设计核心是好用。所以满足用户需求，使你的程序使用起来更加直观，是很重要的。以下是一些方法提示：

■ Use standard controls correctly. Users are familiar with the standard controls they see in the built-in applications, so they already know how to use them in your application.
正确使用标准控件。用户对于一些标准控件十分熟悉。所以他们已经知道如何使用它们在你的程序里。

■ Be sure the path through the information you present is logical and easy for users to predict. In addition, be sure to provide markers, such as back buttons, that users can use to find out where they are and how to retrace their steps.
确保你设定的信息路径对于使用者来说是可以合理预测的。另外，提供一些标记，例如后退按钮，用户可以使用它来找到他们的所在位置并且退回到上一步。

What Are Your Options? 你的选择是什么

Before you decide how to present your product to iPhone OS users, you need to understand the range of options you have. Depending on the implementation details of your proposed product and its intended audience, some types of software may be better suited to your needs than others.
在你决定如何向iPhone OS用户介绍你的产品之前，你必须了解你所有的一系列选择。根据你的目标产品的详细定义和它的潜在客户，一些软件类型可能会更好的满足你的需求。

This section divides software for iPhone OS–based devices into three broad categories, primarily based on implementation method. Roughly speaking, you can create:
这一部分将iPhone OS设备软件分为三个明确类型，主要取决于实现方式。粗略地说，就是你可以自己创建。

■ An iPhone application, which is an application you develop using the iPhone SDK to run natively on iPhone OS–based devices.
iPhone应用软件，你可以使用iPhone 的软件开发工具包开发并在iPhone OS设备上自如运行的应用软件。

■ Web-only content, including web applications, which are websites that behave like built-in iPhone applications.
只适用于网络内容，包括WEB应用程序，一种就好像植入于iPhone应用软件的网络应用。

■ A hybrid application, which is an iPhone application that provides access to web content primarily through a web-content viewing area, but includes some iPhone OS user interface elements.
混合应用软件，本身是包含iPhone OS用户界面元素的iPhone应用程序，但可以通过网页视图接入查看网络信息。

iPhone Applications iPhone应用软件

---

iPhone applications resemble the built-in applications on iPhone OS–based devices in that they reside on the device itself and take advantage of features of the iPhone OS environment. Users install iPhone applications on their devices and use them just as they use built-in applications, such as Stocks, Maps, Calculator, and Mail.
iPhone 应用程序看起来就好像是基于iPhone OS的设备的内置程序一样，它们存在于设备本身并且能利用iPhone OS下的功能优势。用户安装iPhone应用程序在他们的设备上而且使用他们就像在用系统内置的程序，如股票，地图，计算器和邮箱一样。
An iPhone application is quick to launch and easy to use. Whether the application enables a task like sending email or provides entertainment to users, it is characterized by responsiveness, simplicity, and a beautiful, streamlined user interface.
一个iPhone的应用程序是能快速启动并且容易使用的。无论是诸如发送邮件还是提供娱乐，这个应用程序的特质应该是快速响应，简洁以及用户界面的优美与流畅性。

Web-only Content 只适用于网络内容

You have a few different options when it comes to providing web-only content to iPhone OS users:
你有几个不同的选项，当开发仅提供网络内容的iPhone应用程序。

■ **Web application 网页应用程序**
Webpages that provide a focused solution to a task and conform to certain display guidelines are known as web applications, because they behave similarly to the built-in iPhone OS applications. A web application, like all web-only content, runs in Safari on iPhone; users do not install it on their devices, instead they go to the web application’ s URL.
网页，其使用类似于其它iPhone OS的内置程序，是聚焦于一种任务的解决方案，并与网络程序的显示规范保持一致。一个网页应用程序，其内容都在iPhone上的Safari上运行，用户不需要把他们安装到设备上，只需输入网页应用程序的网址。

■ **Optimized webpages 优化的网页**

Webpages that are optimized for Safari on iPhone display and operate as designed (with the exception of any elements that rely on unsupported technologies, such as plug-ins, Flash, and Java). In addition, an optimized webpage correctly scales content for the device screen and is often designed to detect when it is being viewed on iPhone OS–based devices, so that it can adjust the content it provides accordingly.
网页被优化以显示并运行于iPhone的safari上（一些例外的元素除外，一般是不被支持的技术，像插件，Flash和Java）。此外，被优化的网页可以缩放尺寸应该符合设备的屏幕，并且被设计为可以检测到（通过检测User-Agent(UA)，并自动调整内容。）当它在iPhone OS设备上被观看时，能提供适合的相应内容。

■ **Compatible webpages　兼容网页**

Webpages that are compatible with Safari on iPhone display and operate as designed (with the exception of any elements that rely on unsupported technologies, such as plug-ins, Flash, and Java). A compatible webpage does not tend to take extra steps to optimize the viewing experience on iPhone OS–based devices, but the device usually displays the page successfully.
iPhone的safari上的兼容网页的显示和运作就像被设计过一样（除了一些例外的元素如不能被支持的技术，象插件，flash和java）。一个兼容的网页并不倾向于采取额外的步骤去优化基于iPhone OS的设备的视觉体验，但是iPhone设备通常显示页面成功。

If you have an existing website or web application, first ensure that it works well on iPhone OS–based devices. Also, you should consider creating a custom icon users can put on their Home screens using the Web Clip feature. In effect, this allows users to keep on their Home Screens a bookmark to your website that looks like a native application icon. To learn more about creating a custom icon and how to make web content look great on iPhone OS–based devices, see iPhone Human Interface Guidelines for Web Applications.
如果你有一个已存在的网站或者网页应用程序，首先确保他能在以iPhone OS的设备上运作良好。此外，你应该考虑创建一个自定义图标，使用户能将该网页的剪辑变成图标，放在主屏幕上。实际上，这允许用户在他们的主屏幕上保留网站的书签，而看起来就像是本机应用程序的图标。要了解更多关于自定义图标的创建和如何使网页内容在设备上看起来好看 ,请看*iPhone Human Interface Guidelines for Web Application*。

Hybrid Applications　混合应用程序

With iPhone OS, you can create an application that combines features of native applications and webpages. A hybrid application is a native iPhone application that provides most of its structure and functionality through a web viewing area, but also tends to contain standard iPhone OS user interface elements.

在iPhone OS上，你能创建一个结合本机应用程序和网页功能的应用程序。混合应用程序是一个iPhone本机的应用程序，其通过网页视图区域提供大多数的本地程序的结构和功能，而且包含标准的iPhone OS的用户界面元素。

A hybrid application gives users access to web content with an element called a web view (described in "Web Views" (page 104)). Precisely how you use a web view in your application is up to you, but it's important to avoid giving users the impression that your application is merely a mini web browser. A hybrid application should behave and appear like a native iPhone application; it should not draw attention to the fact that it depends upon web sources.
混合应用程序给予用户一个访问网页内容的入口，叫做网络视图的元素（ "Web Views"中有描述）。如何在应用程序上运用网页视图完全取决于设计者，但重要的是避免给用户一个该应用程序只是个迷你浏览器的印象。混合应用程序应该在操控行为和外观上于iPhone本地应用程序一样；并且不应该将注意力集中在依赖于网络资源上。

Three Application Styles 三个类型的应用程序

This document identifies three application styles, based on visual and behavioral characteristics, data model, and user experience. Before you read further, it's important to emphasize that these varieties are named and described to help you clarify some of your design decisions, not to imply that there is a rigid classification scheme that all iPhone software must follow. Instead, these styles are described to help you see how different approaches can be suitable for different types of information and functionality.
这段文档定义了3种应用程序的风格，是基于视觉和行为特征，数据模型，和用户体验。在进一步阅读之前，要强调的是这么多的命名和描述，是帮助你厘清设计决定；并不意味着这里有一个所有iPhone软件都必须遵循的一个僵化的分类方案。相反，这些样式描述可以帮助你了解不同的方法可以适用于不同类型的信息和功能。

> Note: Bear in mind that application style does not dictate implementation method. This document focuses on designing native iPhone applications, but the application styles explored here can be implemented in web or hybrid applications for iPhone OS–based devices.
> 注：请记住，应用程序的样式不决定实施的方法，以上这段文档比较注重于设计iPhone本地应用程序，但是应用程序的风格探索也可以在网页或者混合应用程序里面实施。

As you read about these three application styles, think about how the characteristics of each might enhance your proposed feature set and the overall user experience you plan to deliver in your iPhone application. To help you discover the combination of characteristics that best suit

your application, keep the following questions in mind as you learn about different design styles for iPhone applications:
当你在阅读这三个应用程序的样式，应思考如何通过每一个特性来提升你设想的性能，以及你期望在该iPhone应用程序中所能感受到的用户体验。为了帮助你发现最适合于你应用程序的特性，请记住下面的问题，当你学习关于iPhone应用程序的不同设计样式时：

- What do you expect to be the user’s motivation for using the application?
你希望用户使用该应用程序的动机是什么？
- What do you intend to be the user’s experience while using the application?
你想要给用户什么样的体验当他使用该应用程序时？
- What is the goal or focus of your application?
该应用程序设计的目标或者关注点是什么？
- How does your application organize and display the information people care about? Is there a natural organization associated with the main task of the application?
你的应用程序是如何组织和显示人们所关心的信息？是否有一个自然的组织与应用程序的主要任务相关联？

Productivity Applications 生产力辅助程序

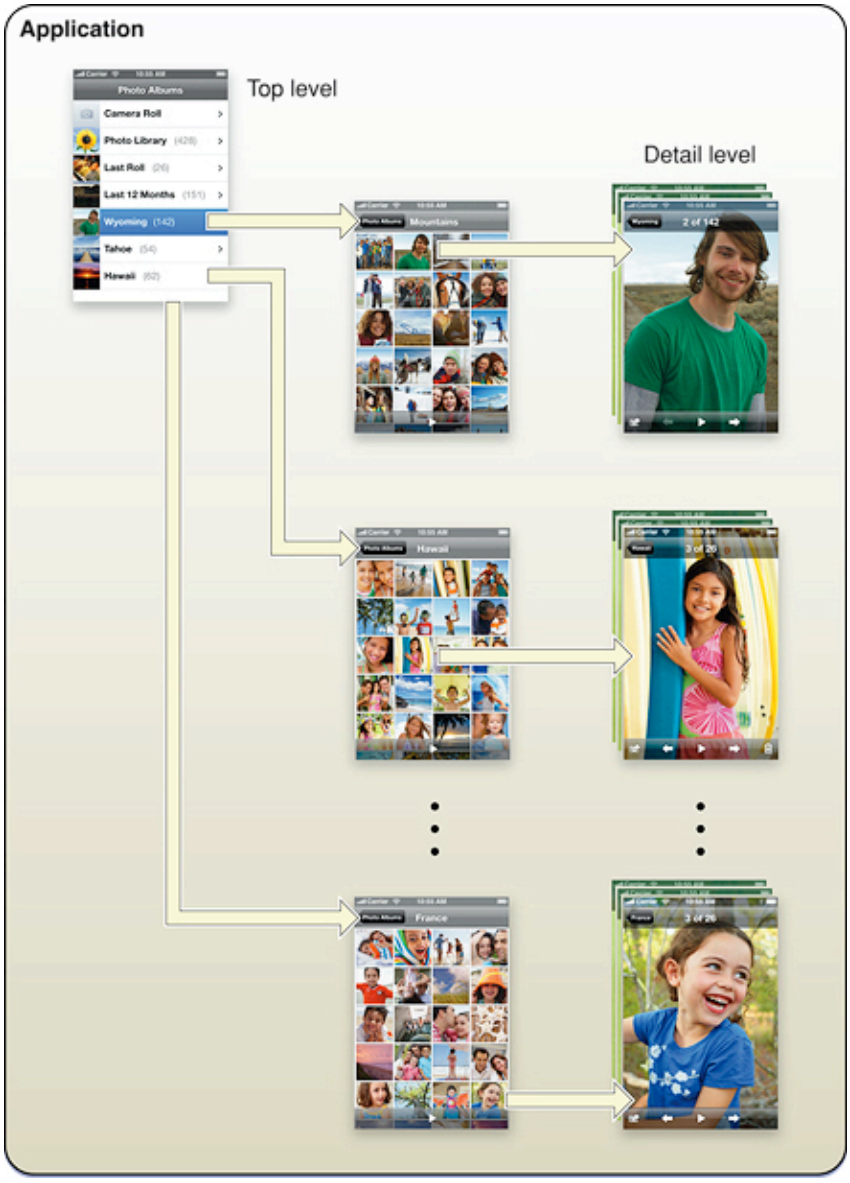
A productivity application enables tasks that are based on the organization and manipulation of detailed information. People use productivity applications to accomplish important tasks. Mail is a good example of a productivity application.
生产力辅助程序用来完成对具体信息进行组织和操作的任务。人们使用生产力辅助程序完成重要的任务。邮件是一个关于生产力辅助程序的很好的例子。

Seriousness of purpose does not mean that productivity applications should attempt to appear serious by providing a dry, uninspiring user experience, but it does mean that users appreciate a streamlined approach that does not hinder them. To this end, successful productivity applications keep the user experience focused on the task, so people can quickly find what they need, easily perform the necessary actions, complete the task, and move on to something else.
严肃的目的并不意味着生产力辅助程序应出现严肃的表达，一个枯燥，缺乏启发性的用户体验。相反，它意味着用户欣赏不妨碍他们的流畅的解决之道。为此，成功的生产力辅助程序应保持将用户体验聚焦到任务上，以便人们可以迅速找到他们需要什么，容易执行必要的行动，完成任务，并且转移到别的事物上。

Productivity applications often organize user data hierarchically. In this way, people can find information by making progressively more specific choices until they arrive at the desired level of detail. iPhone OS provides table elements that make this process extremely efficient on

iPhone OS devices (see “Table Views” (page 89) for more information about these user interface elements). Figure 1-1 shows an example of this type of data organization.
生产力辅助程序经常分层次地组织用户数据。这样，人们可以通过逐步制定更为具体的选择来查找信息，直到到达他们所期望的详细程度。iPhone OS提供的表格元素，使这一进程的效率极高（更多关于用户界面元素的信息，见“Table Views”）。图1-1显示了这种数据组织类型的例子。

Figure 1-1 Productivity applications tend to organize information hierarchically
生产力辅助程序分层次信息组织框架



Typically, the user interaction model in a productivity application consists of:
典型生产力辅助程序的用户交互模型包含：

- Organizing the list 列表组织
- Adding to and subtracting from the list列表的增减

■ Drilling down through successive levels of detail until the desired level is reached, then performing tasks with the information on that level逐步细化信息直到所期望的详细程度，然后执行对应的级别的任务

Productivity applications tend to use multiple views, usually displaying one level of the hierarchy per view. The user interface tends to be simple, uncluttered, and composed of standard views and controls. Productivity applications do not tend to customize the interface much, because the focus is on the information and the task, and not as much on the environment or the experience.
生产力辅助应用程序倾向于使用多个视图，通常会在每个视图显示一个级别的信息。用户界面倾向于简洁，并由标准的视图和控件组成。生产力辅助应用程序不太倾向于自定义界面，因为其的专注于信息和任务，而不是环境或体验。

Among all types of iPhone applications, a productivity application is the most likely to supply preferences, or settings, the user can specify in the Settings application. This is because productivity applications work with lots of information and, potentially, many ways to access and manage it. It’ s important to emphasize, however, that the user should seldom need to change these settings, so the settings should not target simple configuration changes that could be handled in the main user interface.
在所有的 iPhone 应用程序中，生产力辅助应用程序是最有可能在“设置应用程序”中提供具体的用户选项或者设置功能。这是因为生产力辅助应用程序处理着大量的信息，因而也就有许多方法来访问和管理它们。需要强调的是，用户须很少需要更改这些设置，所以，这些设置不应该放在绑定程序主界面的简单配置内。

Utility Applications实用工具程序

A utility application performs a simple task that requires a minimum of user input. People open a utility application to see a quick summary of information or to perform a simple task on a limited number of objects. The Weather application (shown in Figure 1-2) is a good example of a utility application because it displays a narrowly focused amount of information in an easy-to-scan summary.
实用工具程序用于执行简单的，很少需要用户输入的任务。人们打开一个实用工具程序是为了查看一段快速的信息摘要，或者对有限的几个对象执行一个简单的操作。这个天气预报程序（图1-2）是一个很好的例子，因为它将一系列信息以一种很方便阅览的方式展示出来。

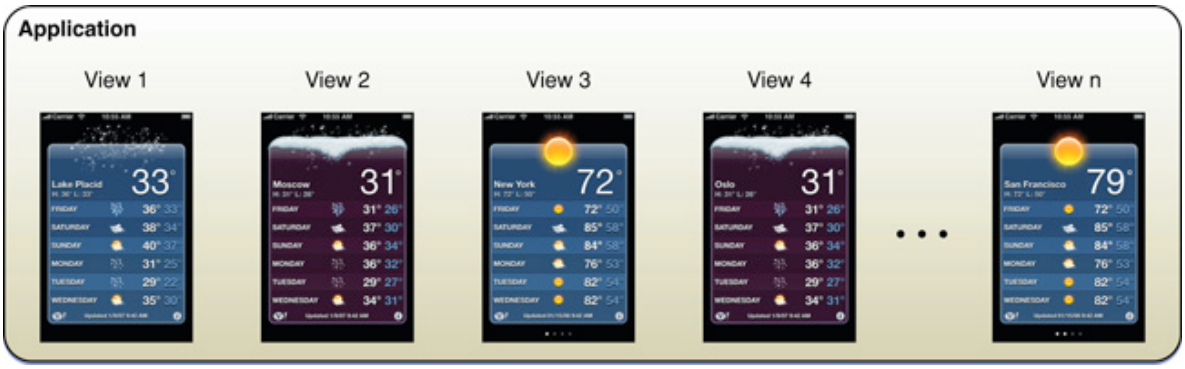
Figure 1-2 Weather is an example of a utility application 气象预报，实用工具程序案例

Utility applications are visually attractive, but in a way that enhances the information they display without overshadowing it. People use utility applications to check the status of something or to look something up, so they want to be able to spot the information they’ re interested in quickly and easily. To facilitate this, a utility application’ s user interface is uncluttered and provides simple, often standard, views and controls.
实用工具程序看上去比较美观。但是在某种程度上也强调了所呈现的信息。人们使用实用工具程序来查看状态或者查询一些东西。所以他们希望能够快速便捷的查找到他们想要的信息。为了实现这一点，一个实用工具程序的用户界面必须简洁，在标准，视图，控件上保持统一。

A utility application tends to organize information into a flattened list of items; users do not usually need to drill down through a hierarchy of information. Typically, each view in a utility application provides the same organization of data and depth of detail, but can be served by a different source. In this way, users can open a single utility application to see similar treatments of multiple subjects. Some utility applications indicate the number of open views; users can navigate through them sequentially, selecting one view after another. Figure 1-3 shows an example of this type of data organization.
一个实用工具程序往往会将一些信息组织成一个扁平项目列表的架构形式。用户不需要通过一个信息的层次结构来深入。通常，在实用工具程序中每一个信息，虽然来源不同，但是都提供了一个相同的构架来放置数据和深入细节。用户可以打开一个单个的实用工具程序来查看不同事物之间的相似处。一些实用工具程序显示视图窗口数目，用户可以依次查看。图1-3列举了这类数据架构的案例。

Figure 1-3　Utility applications tend to present data in a flattened list

实用工具程序以扁平列表方式显示数据

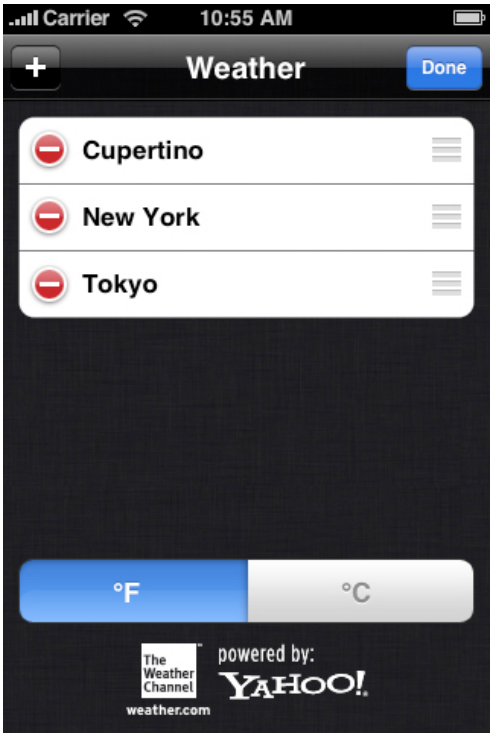


The user interaction model for a utility application is very simple: Users open the application to scan a summary of information and, optionally, change the configuration or source of that information. Utility applications may need to support frequent changes to configuration or information source, so they often provide a small set of such options on the back of the main view. Users tap the familiar Info button in the lower-right corner of the main view to see the back. After making adjustments, users tap the Done button to return to the front of the main view. In a utility application, the options on the back of the main view are part of the functioning of the application, not a group of preference-style settings users access once and then rarely, if ever, again. For this reason, utility applications should not supply application-specific settings in the Settings application. Figure 1-4 shows how the Weather application provides configuration options on the back of the main view.

实用工具程序用户交互模型非常简单：用户打开程序浏览信息概要，有选择的更改配置或者信息来源。实用工具程序可能需要支持配置和信息来源的不断更变。所以它们设置了一系列很小的按钮在主菜单的背后。使用者点击主菜单右下角的信息按钮进入背后视图。当做好调整，用户按下“完成”按钮回到正面视图。在实用工具程序上，主菜单背面的选项是程序运行的一部分，而不是一组参数设置。因此，实用工具程序不应该提供一些特定程序设置在“系统设置”程序里。图1-4显示了天气预报系统主视图背面的配置选项。

Figure 1-4 Users can make adjustments on the back of Weather

用户可以在天气应用程序后面调整设置



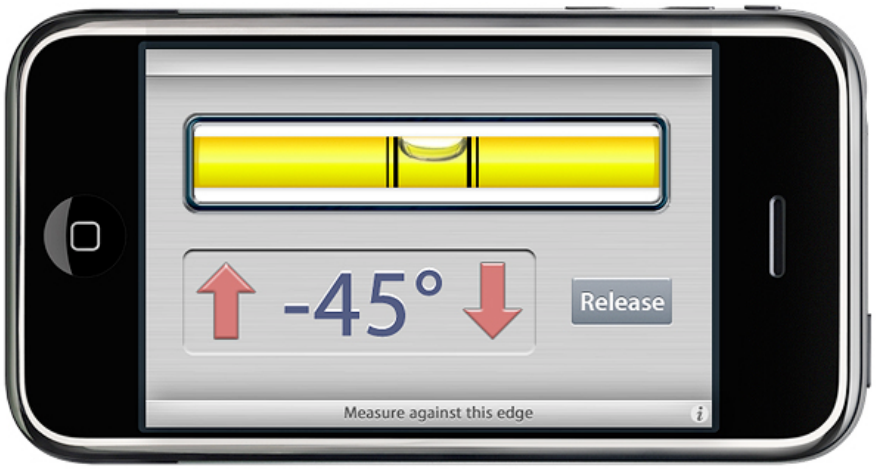
Immersive Applications 沉浸式应用程序

An immersive application offers a full-screen, visually rich environment that’ s focused on the content and the user’ s experience with that content. People often use immersive applications to have fun, whether playing a game, viewing media-rich content, or performing a simple task.
一个沉浸式应用程序提供了一个全屏，视觉丰富的环境，并专注于内容和用户对于内容的体验。人们使用沉浸式应用程序来娱乐，玩游戏，看富媒体内容或者执行一个简单任务。

It’ s easy to see how games fit this style of iPhone application, but you can also imagine how characteristics of immersive applications can enhance other types of tasks. Tasks that present a unique environment, don’ t display large amounts of text-based information, and reward users for their attention are good candidates for the immersive approach. For example, an application that replicates the experience of using a bubble level works well in a graphics-rich, full-screen environment, even though it doesn’ t fit the definition of a game. In such an application, as in a game, the user’ s focus is on the visual content and the experience, not on the data behind the experience. Figure 1-5 shows an example of an immersive application that replicates an actual experience and enables a simple task.
很容易发现如何在游戏中运用这种类型的iPhone程序，但是你也可以想象一下这种沉浸式程序如何用于其他类型的任务的提升。沉浸式程序最大的特点是给任务提供一个特定的场景，而不是显示一大堆的文本信息，以及可以强烈地吸引住使用者。例如，一个显示水平气泡的程序，可以在一个图形丰富，全屏的环境下运行良好，即使它不适合被定义为一个游戏。像这样一个程序，就如一个游戏，使用者的注意

力集中在视觉内容和视觉体验上，而不是背后的数据。图1-5例子显示的就是如何用一个真实的体验来完成一个简单的任务。

Figure 1-5 An immersive application doesn’t have to be a game
沉浸式应用程序不一定要是一个游戏



Note: Although applications that launch in landscape orientation should launch so that the Home button is on the right, the Bubble Level application shown above in Figure 1-5 launches in the opposite orientation. This ensures that the physical buttons on the edge of the device don’t interfere with the measurement. See “Starting” (page 45) for more launch guidelines.
注:虽然启动横屏程序时主按钮是须在右边，但是这个水平气泡程序的主页按钮是在左边。这个确保了位于边界的物理按钮不会影响尺寸。更多启动设计导引见“Starting”章节。

An immersive application tends to hide much of the device’s user interface, replacing it with a custom user interface that strengthens the user’s sense of entering the world of the application. Users expect seeking and discovery to be part of the experience of an immersive application, so the use of nonstandard controls is often appropriate.
一个沉浸式程序一般都会隐藏大部分的设备用户界面，用一个标准界面代替来增强使用者进入程序的感觉。使用者期待的探索发现过程成为沉浸式程序体验的一部分，所以使用一些非标准的控件是恰当的。

Immersive applications may work with large amounts of data, but they do not usually organize and expose it so that users can view it sequentially or drill down through it. Instead, immersive applications present information in the context of the game-play, story, or experience. Also for this reason, immersive applications often present custom navigational methods that complement the environment, rather than the standard, data-driven methods used in utility or productivity applications.

沉浸式程序工作时需要运行大量的数据，但是它们通常不会编组及展现，以使用户去查看或研究。相反，沉浸式程序是以一种玩游戏，讲故事的体验方式来呈现信息。因此，沉浸式程序经常呈现为设计者自定义地引导模式，并成为程序的场景补充，而不是像在实用工具或生产力辅助工具程序那样以数据驱动展开。

The user interaction model for an immersive application is determined by the experience the application provides. Although it's not likely that a game would need to offer application-specific settings in Settings, other types of immersive applications might. Immersive applications might also furnish configuration options on the back of the main view.
沉浸式应用程序的用户交互模型是由应用程序提供的体验所决定的。尽管游戏那样的该类程序不需要在“设置”应用程序中里提供特定设置，但其他类型的沉浸式应用程序可能需要。沉浸式应用程序也可以在主视图背后提供配置选项。

Choosing an Application Style
选择一个应用程序样式

After reading about productivity, utility, and immersive application styles, think about the type of information your application displays and the task it enables. In theory, the type of application you should create is obvious to you and you're ready to get started; in practice, it's not always that simple. Here is a hypothetical scenario to consider as you make your decision.
在阅读完生产力辅助、实用工具、沉浸式程序之后，考虑一下你程序所显示的信息类型和能完成的任务。理论上来说，你设计的程序类型应该是你很了解的且很容易上手的。但是实际上并没有那么简单。这里有一个假想的场景供你参考。

If you have a subject you'd like to explore, think about the objects and tasks related to it. Imagine the different perceptions people have of that subject. For example, consider the subject of baseball. Baseball brings to mind, among other things, teams, games, statistics, history, and players. Baseball is probably too extensive a subject for a single application, so consider just the players. Now imagine how you might create an application that relates to players—for example, using their likenesses on baseball cards.
如果你有一个需要探索的项目，考虑一下和它相关的事物，想象一下人们对于这个项目的不同看法。例如，一个有关棒球的课题，棒球给你的联想包括队伍，比赛，统计数据，历史，和球员。对于一个程序来说，棒球是一个很宽泛的主题，如果只考虑球员，想象一下你可能设计的一个与球员相关的程序，例如可以利用一下他们对于棒球卡的喜爱。

You could develop a productivity application that helps serious collectors manage their baseball card collections. Using list-based formats, you could display cards in a hierarchy of teams, then players, then seasons. In the most detailed view, you could give users the ability to note where they acquired the card, how much they paid for it, its current market value, and how

many copies they have. Because the focus of this application is on the data that defines the collection, the user interface streamlines the tasks of seeking and adding information.
你可以开发一个生产力辅助程序来帮助那些收藏发烧友来管理他们的棒球卡。使用列表格式，你可以按队伍，球员，赛季这样的顺序来排列卡片。更细致一点，你可以让用户知道他们是在哪里获得的卡片，花了多少钱，它目前的市场价，有多少个拷贝。因为这个程序的重点是对于收藏数据的定义，所以用户界面须使这个搜寻与添加信息的工作更加流畅。

You could also develop a utility application that displays the current market value of particular baseball cards. Each view could look like a baseball card with its current value added to the picture, and the back of the view could allow users to select specific cards to track and display. The focus of this application is on individual cards, so the user interface emphasizes the look of the cards and provides a simple control or two that allows users to look for new cards.
你可以开发一个用于显示棒球卡市场价值的实用工具程序。每个视图都是一个附有当前值的棒球卡。视图的背面可以允许用户选择一个特别的卡片来跟踪与显示。这个程序的重点在于卡片个体，所以用户界面需增强卡片的外观，并且提供一些简单的控件以允许用户搜寻新的卡片。

Or, of course, you could develop a game. Perhaps the game would focus on the user’ s knowledge of certain statistics on individual baseball cards or ability to recognize famous cards. Or perhaps it would simply use baseball cards as icons in another type of game, such as a sliding puzzle. In each of these cases, the focus of the application is on the images on the baseball cards and the game play. The user interface complements this by displaying a few baseball-themed controls and hiding the iPhone OS user interface.
或者你还可以开发一个游戏，这个游戏重点在于考察使用者对于棒球卡的某些统计数据的了解或者是识别一些名贵卡的的能力。或者它只是将棒球卡作为一个其他类型游戏（滑动拼图）的图标，在这些例子中，程序的重点在于对卡片的图形和游戏的设想。用户界面可以显示一些带有棒球主题的控制，并且隐藏iPhone OS的用户界面。

It’ s important to reiterate that you’ re not restricted to a single application style. You may find that your application idea is best served by a combination of characteristics from different application styles.
很重要的一点是你不可以只局限于一个单一的程序类型。你可以发现你的很多点子都来源于不同风格程序的综合。

When in doubt, make it simple. Pare the feature list to the minimum and create an application that does one simple thing (see “Create a Product Definition Statement” (page 35) for advice on how to focus your application). When you see how people use and respond to the application, you might choose to create another version of the application with a slightly shifted focus or altered presentation. Or, you might discover a need for a more (or less) detail-oriented version of the same concept.

当有疑问，就简化它。将功能列表削减到最小，让程序去完成一件简单的工作（参考“Create a Product Definition Statement” 的建议）。当你了解到人们如何使用这个程序，你可能会创建新的版本进行一些程序侧重点或内容上的更改或者削减，或者你会因此发掘出一个概念相同但有不同细节的版本。

When You Have an Existing Computer Application
当你已开发了一个桌面应用程序

If you have an existing computer application, don’t just port it to iPhone OS. People use iPhone OS–based devices very differently than they use desktop and laptop computers, and they have different expectations for the user experience.
如果你已经开发了一个桌面应用程序，不要马上硬塞到iPhone OS系统中。人们对于iPhone OS设备的使用不同于使用电脑桌面，他们对于用户体验也存在不一样的期望值。

Remember that people use iPhone OS–based devices while on the go, and often in environments filled with distractions. This generally means that they want to open your application, use it briefly, and move on to something else. If your application relies on the user’s undivided attention for long stretches of time, you need to rethink its structure and goals if you want to bring it to iPhone OS.
因为人们对于iPhone OS设备的使用环境是不固定的，充满各种干扰。这个意味着他们在打开你的程序，很快的使用一下，然后就转向其他事物。如果你的程序需要使用者很长时间的注意力集中，你可能需要重新思考一下他的结构与目标，再决定是否应该将它移植到iPhone OS系统中。

If your desktop application enables a complex task or set of tasks, examine how people use it in order to find a couple of subtasks they might appreciate being able to accomplish while they’re mobile. For example, a business-oriented application that supports project scheduling, billing, and expense reporting could spawn an iPhone utility application that shows progress summaries for a project, or an iPhone productivity application that allows mobile users to keep track of their business-related expenses.
如果你的桌面应用程序有一个或一串复杂的任务，观察人们是如何使用它的，以便于找到当用于移动设备上人们可能会喜欢的子项目。例如一个商业性质的程序（支持时刻表，付账，费用报告等服务）可以利用iPhone实用程序来显示一个项目的进度状况，或者可以用iPhone生产力辅助程序来追踪移动用户的商业开销。

As you think about how to bring ideas from your desktop application to an iPhone application, apply the 80-20 rule to the design of your application. Estimate that the largest percentage of users (at least 80 percent) will use a very limited number of features in an application, while only a small percentage (no more than 20 percent) will use all the features. Then, consider carefully whether you want to load your iPhone application with the power features that only a small

percentage of users want. Be aware that a desktop computer application might be the better environment in which to offer those features, and that it' s usually a good idea to focus your iPhone application on the features that meet the needs of the greatest number of people.
当你考虑如何将桌面系统的程序转移到iPhone系统中时，在你的设计中遵循80-20的规则，绝大多数的使用者（至少80%）只会使用部分功能，只有很少一部分人会使用所有的功能。仔细考虑一下是否你需要安装那些只有少数人需要的所谓强大的功能。你的电脑桌面系统已经提供了这些功能，所以在你的iPhone系统中最好将重点放在那些满足最广大人群需求的功能上。

Case Studies: Bringing a Desktop Application to iPhone OS
案例研讨：将桌面程序移植到iPhone系统

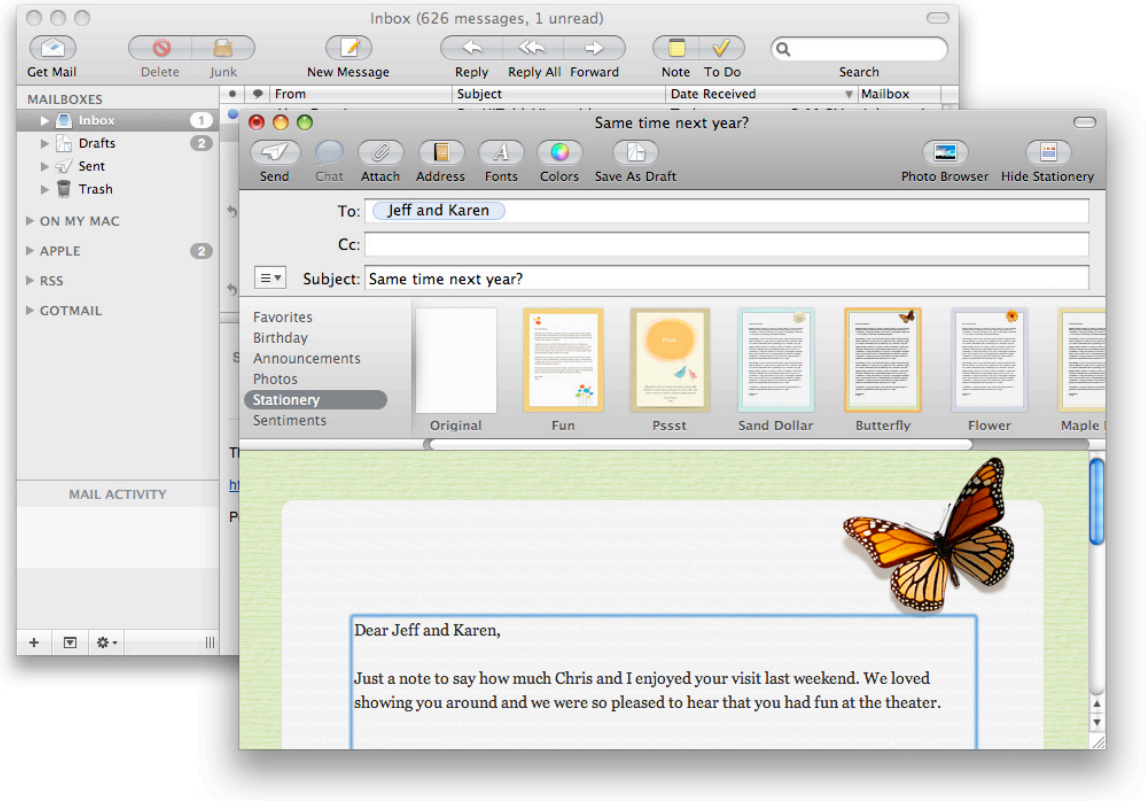
To help you visualize ways you can create an iPhone OS version of a desktop computer application, this section describes some of the design differences between familiar Mac OS X applications and their iPhone OS counterparts. As you learn about which features and functions in each application were adapted for its iPhone OS version, you will gain insight into the types of design decisions you need to make for your own iPhone application.
为了帮助你更好的理解将桌面系统转为iPhone OS系统，这一章节介绍了Mac OS X系统应用程序和iPhone OS系统之间的差异。当你了解到哪些功能适用于iPhone OS版本，你将会更深入的了解你自己的iPhone系统需要什么样的设计决定。

Mail 电邮

Mail is one of the most highly visible, well-used, and appreciated applications in Mac OS X. It is also a very powerful program, one that allows users to create, receive, prioritize, and store email, track action items and events, and create notes and invitations. Mail provides most of this functionality in a single multipane window. This is convenient for people using a desktop computer, because they can leave a Mail window on the display screen (or minimized to the Dock) all the time and switch to it whenever they choose. Figure 1-6 illustrates many of the features available in the Mail message-viewing and compose windows on the desktop.
电邮是Mac OS X系统里可视性强，好用并且比较受好评的程序之一，这也是一个功能很强大的程序。它允许使用者创建，接收，排序，储存邮件，追踪项目与事件，写笔记和邀请。电邮的绝大多数功能在一个单一的多平面窗口显示。这项功能方便了那些习惯使用电脑桌面的人们，因为他们可以在显示屏上不显示或者最小化电邮窗口，但又可以随时打开。图 1 - 6 说明了邮件信息窗口和桌面撰写窗口里的很多项功能。

Figure 1-6 Mail on the desktop offers a wide range of powerful features in a couple of windows

桌面邮件在一对窗口中提供了一系列强大的功能



But when people are mobile, their needs for an email application are simpler, and they want access to core functionality quickly. For this reason, Mail on iPhone OS–based devices focuses on the most important things people do with their email: receive, create, send, and organize messages. To do this, it displays a pared-down user interface that makes the organization of the user’ s accounts and mailboxes clear and centers the user’ s attention on the messages.
但是当人们处于移动状态时，他们对于邮件程序的需求就降低了，他们需要一个快速便捷的核心功能。因此，iPhone OS设备上的邮件程序设计重点放在人们如何接收，创建，发送，组织信息。为了实现这点，应该简化用户界面，使用户账号，信箱架构更加明晰，用户注意力放在信息本身。

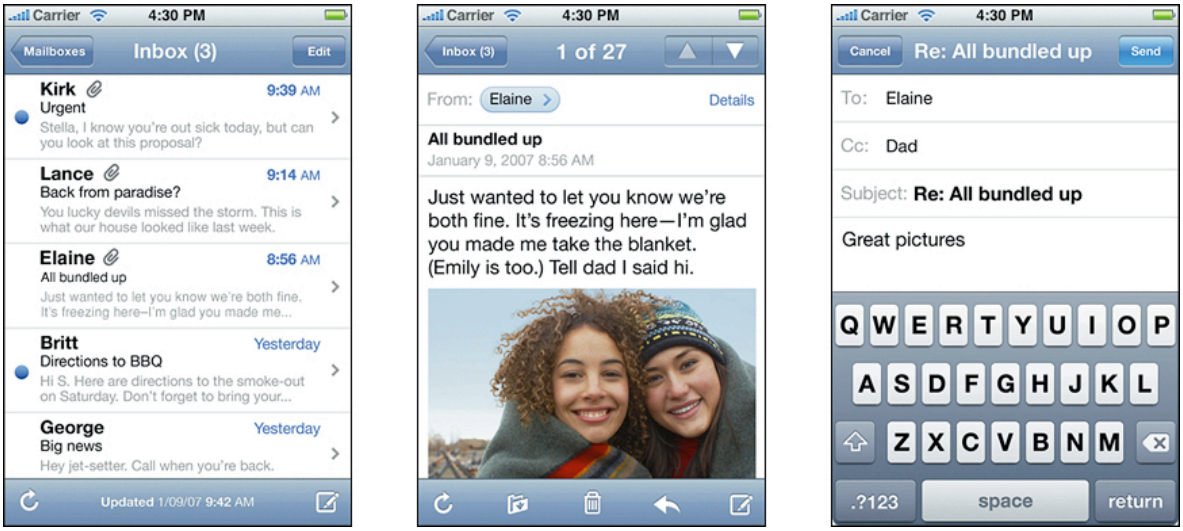
Mail in iPhone OS is a perfect example of a productivity style application: To ease navigation through the content, Mail in iPhone OS takes advantage of the naturally hierarchical organization of people’ s email and displays on successive pages accounts, mailboxes, message lists, and individual messages. Users drill down from the general (the list of accounts) to the specific (a message) by selecting an item in a list and viewing the things associated with that item. To learn more about the productivity style of iPhone applications, see “Productivity Applications” (page 19).
iPhone OS系统上的邮件程序就是一个很好的生产力辅助程序的例子。为了简化导航内容，iPhone OS根据人们发邮件的习惯动作，将帐户信息，邮箱，信息列表，个人信息都呈现在连续的自然分级页面上。使用者可以通过在一堆列表项目中选择其中一项来深入查看细节内容和与其相关的其他信息。更多关于生产力辅助类型的程序，请看 “Productivity Applications”

In addition, Mail in iPhone OS enables actions, such as create and send, by displaying a handful of familiar controls that are easy to tap. Figure 1-7 shows how Mail makes it simple to view and send email in iPhone OS. Note how elements at the top of each screen make it easy for users to know both their current and previous location in the application.

而且，iPhone OS里的邮件创建，发送都采用比较容易操作的控制。图 1 - 7 显示了iPhone中如何使收发邮件变得简单好用，每个显示屏上部的都有一些图标元素使用户可以同时了解他们目前和之前所处的位置。

Figure 1-7 Mail in iPhone OS makes it easy to view and send email

iPhone OS 系统里的邮件更加容易查看且发送



iPhoto 照片

Another instructive example of a Mac OS X application that was reimagined for iPhone OS is iPhoto. On the desktop, iPhoto supports comprehensive searching and organization, powerful editing capabilities, and creative printing options. When people use iPhoto on their desktop or laptop computers, they appreciate being able to see and organize their entire collection, make adjustments to photos, and manipulate them in various ways. Although the main focus of iPhoto is on the user’ s content, the application also offers extensive functionality in its window. Figure 1-8 shows the iPhoto user interface on the desktop.

另一个对iPhone OS有启发的是来自MAC操作系统的应用程序iPhoto。在桌面上，iPhoto支持全面的搜索和组织，强大的编辑功能，以及创造性的打印选项。在桌面电脑上，使用者能看到他们的整个收藏并且进行整理，对照片进行调整，并以各种方式操纵他们。虽然iPhoto的焦点是用户的内容，但在其窗口上还提供了很多的功能。图1-8显示了桌面上的iPhoto的用户界面。

Figure 1-8 The iphoto user interface

iPhoto的用户界面

But when they’re mobile, people don’t have time to edit their photos (and they don’t expect to print them); instead, they want to be able to quickly see and share their photos.
但是，在移动状态下，人们没有时间去编辑他们的照片（他们不希望打印出来）;相反，他们希望能够尽快看到并分享他们的照片

To meet this need on iPhone OS–based devices, Apple has provided the Photos application, which focuses on viewing photos and sharing them with others. The Photos user interface revolves around photos; so much so, in fact, that even parts of the device user interface can be hidden. When users choose to view a slideshow of their photos, the Photos application hides the navigation bar, toolbar, and even status bar, and displays translucent versions of these elements when users need to see them.
为了满足设备的这一需求。苹果提供了Photos程序，重点解决查看照片和与他人分享的功能。Photos的用户界面上排布了很多照片，事实上，部分用户界面可以被隐藏。当用户选择幻灯播放照片时，照片程序将会隐藏导航条，工具条，甚至状态条；而这些界面构件又会以透明方式呈现出来，当用户需要它们时。

Photos makes it easy for users to organize and find their photos by using a hierarchical arrangement: Users select an album, which contains a collection of photos, and then they select a single photo from the collection. In this way, Photos is an example of an application that combines features of the productivity style and the immersive style (to learn more about these

styles, see “Three Application Styles” (page 18)). Figure 1-9 shows how users can view photos in the Photos application.

Photos程序的分层管理架构使得用户很方便地整理、查找他们的照片：使用者可以选择一个相册，然后从中选择一张照片。通过这种方式，Photos程序很好的将生产力辅助程序风格和沉浸式风格相结合（延伸阅读请查看“Three Application Styles”）。图1-9显示了使用者如何在Photos程序中查看照片。

Figure 1-9 Three screens in the Photos application

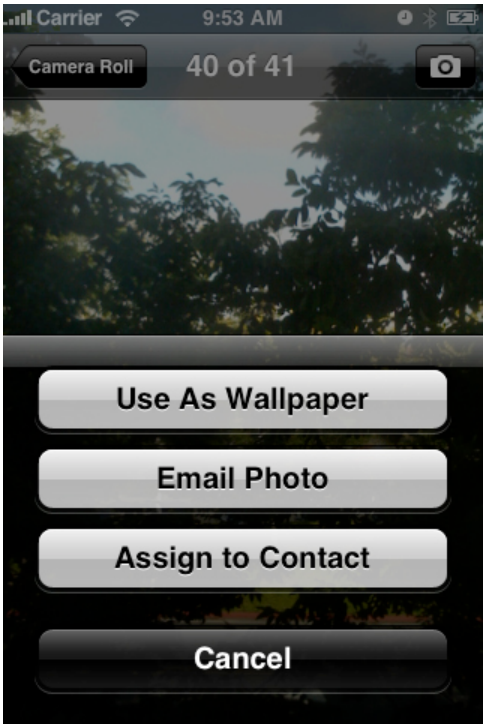
photos程序上的三个显示窗口

In addition, Photos uses a transient view, called an action sheet (described in “Alerts, Action Sheets, and Modal Views”), to give users additional functionality without taking them out of the photo-viewing experience. Figure 1-10 shows how Photos provides options for using an individual photo.

此外，Photos程序采用了一个快速查看的功能，“动作表单”，使用户可以不跳出浏览地照片来完成一些其他操作。图1-10显示Photos程序如何提供单张照片的设置选项。

Figure 1-10 Photos gives users options in an action sheet

照片管理程序在一个操作列表中给出的用户选项


Carrier
9:53 AM
Camera Roll
40 of 41
Use As Wallpaper
Email Photo
Assign to Contact
Cancel

chapter 2

Human Interface Principles: Creating a Great User Interface

人机界面原则：创建一个卓越的用户界面

A great user interface follows human interface design principles that are based on the way people—users—think and work, not on the capabilities of the device. A user interface that is unattractive, convoluted, or illogical can make even a great application seem like a chore to use. But a beautiful, intuitive, compelling user interface enhances an application's functionality and inspires a positive emotional attachment in users.

卓越的用户界面所需遵循的人机界面设计原则往往基于普通人——用户——思维和工作方式，而非设备本身的性能。一个毫无吸引力，令人费解甚至不符合逻辑的用户界面会使（即使是）优秀地应用程序使用困难。反之一个友好的、直观的、令人信服的用户界面往往能增强程序的功能性并且能激发用户正面的情感肯定。

You should read this chapter even if you are already familiar with these fundamental principles, because it focuses on how to apply them to iPhone applications.

你应该阅读这个章节即使你已经对这些基础原则很熟悉了，因为在这一章节将着重介绍如何在 iPhone 软件上运用这些原则。

Metaphors 隐喻

When possible, model your application's objects and actions on objects and actions in the real world. This technique especially helps novice users quickly grasp how your application works. Folders are a classic software metaphor. People file things in folders in the real world, so they immediately understand the idea of putting data into folders on a computer.

如果可能，请为你的程序中的对象、基于对象的动作、动作本身在现实生活中找一个参照物。这项技术有助于新手快速领会你的应用程序是如何工作的。文件夹是一种典型的软件隐喻。人们在现实生活中往往将文件归类放置于文件夹，这样一来，我们就可以轻而易举地理解计算机中将资料数据归整放入文件夹的概念。

Metaphors in iPhone OS include iPod playback controls, tapping controls to make things happen, sliding on-off switches, and flicking through the data shown on picker wheels

在 iPhone OS 的隐喻包括 iPod 的回放控制，点击控制以触发事件，轻划动作以实现开启与关闭的切换，轻击滑轮按键以控制数据。

Although metaphors suggest a use for objects and actions in the iPhone OS interface, that use does not limit the software implementation of the metaphor. To return to the folder example, a folder object implemented in software has a capacity that's completely unrelated to the physical capacity of its real-world counterpart.

虽然隐喻暗示了 iPhone OS 界面上的对象和动作的使用含义，这种隐喻的使用并不限制软件的执行能力。回到文件夹的例子，文件夹在应用程序中履行作为一个容器的功能，但其容量与现实生活中文件夹的容量并无关联。

As you design your application, be aware of the metaphors that exist in iPhone OS and don't redefine them. At the same time, examine the task your application performs to see if there are natural metaphors you can use. Bear in mind, though, that it's better to use standard controls and actions than to stretch a real-world object or action just to fit your application's user interface. Unless the metaphors you choose are likely to be recognized by most of your users, including them will increase confusion instead of decrease it.
当设计应用程序时，必须要意识到 iPhone OS 中已存在的隐喻，不要重新定义它们。同时，检查你应用程序的执行任务检查是否有一些自然的隐喻你可以使用。请记住，使用标准的操控和动作比去尝试引用现实世界的对象和动作更能适应你程序的用户界面。除非你的隐喻能够被大多数人认可，包括那些会使人更迷惑的也不行。

Direct Manipulation 直接操控

Direct manipulation means that people feel they are controlling something tangible, not abstract. The benefit of following the principle of direct manipulation is that users more readily understand the results of their actions when they can directly manipulate the objects involved.
直接操纵意味着人们可以感受到他们操控的东西是有形的而非抽象的。这种做法的好处是用户在操作过程中可以更直观直接地感受到他们的操控结果。

iPhone OS users enjoy a heightened sense of direct manipulation because of the Multi-Touch interface. Using gestures, people feel a greater affinity for, and sense of control over, the objects they see on screen, because they do not use any intermediate device (such as a mouse) to manipulate them.
iPhone OS 用户能喜爱直接操控缘于多点触碰界面。通过手势，人们能感受到屏幕上图标与自己手指之间的亲密关系，因为他们并没有使用任何中间设备（例如鼠标）而是直接操控他们。

To enhance the sense of direct manipulation in your iPhone application, make sure that:
为了提高你 iPhone 应用程序中的直接操控感，请确保：

- Objects on the screen remain visible while the user performs actions on them
  在用户操控时保持相屏幕上相应的对象保持可见
- The result of the user's action is immediately apparent
  用户操作结果能够即时显现于屏幕

See and Point 即看即点

An iPhone application is better than a person at remembering lists of options, commands, data, and so on. Take advantage of this by presenting choices or options in list form, so users can easily scan them and make a choice. Keeping text input to a minimum frees users from having to spend a lot of time typing and frees your application from having to perform a lot of error checking.
一个 iPhone 应用程序在记录选项清单，命令，数据等方面要强于人脑。通过利用这个优点呈现选项清单继而用户可以轻松浏览并且做出选择；支持文本输入最低程度（自动联想与匹配）帮助用户节省文本输入时间与免去拼写检查的麻烦。

Presenting choices to the user, instead of asking for more open-ended input, also allows them to concentrate on accomplishing tasks with your application, instead of remembering how to operate it.
给用户以选择，不要使用开放式的输入（无引导），引导用户将精力更集中于完成任务而不是操作过程。

Feedback 反馈

In addition to seeing the results of their actions, users need immediate feedback when they operate controls and status reports during lengthy operations. Your application should respond to every user action with some visible change. For example, make sure list items highlight briefly when users tap them. Audible feedback also helps, but it can’t be the primary or sole feedback mechanism because people may use iPhone OS based devices in places where they can’t hear or where they must turn off the sound. In addition, you don’t want to compete with the iPhone OS system sounds users already associate with system alerts.
除了将操控结果可视化，用户在冗长的操作过程中需要即时反馈。你的应用程序必须对每一个操作行为做出明显变化的反馈。例如，请确保项目列单被点击时会短暂闪亮。声音反馈也有所帮助，但是这种方式不能是主要或者唯一的反馈机制，因为人们有时会在喧闹环境或必须静音的环境下操作 iPhone OS。除此之外，不要使用 iPhone OS 系统声音，因为用户会将这些声音与警报系统声音相联系。

iPhone OS automatically provides feedback when it’s temporarily busy by displaying the activity indicator. During operations that last more than a few seconds, your application should show elapsing progress and, if appropriate, display an explanatory message.
iPhone OS 会自动提供反馈当设备暂时正忙于实时活动消息。在操作过程持续的几秒钟里，你的应用程序应该显示出操作过程并且做出必要的解释信息。

Animation is a great way to provide feedback to users, as long as it’s both subtle and meaningful. Animation pervades iPhone OS, even in nonimmersive applications. As a means of providing feedback, however, it is used to enhance the user’s experience, not as the focus of the user’s experience.

动画是一种很好的用户反馈方式取决于它的微妙性和丰富含义。动画贯穿于 iPhone 的操作系统，即使在非沉浸式的应用程序中也是如此。总之，作为一种反馈方法，它的作用是提高用户体验，而不是用户体验的焦点。

User Control 用户操控

Allow users, not your application, to initiate and control actions. Keep actions simple and straightforward so users can easily understand and remember them. Whenever possible, use standard controls and behaviors that users are already familiar with.
让你的用户，而不是应用程序，去了解和操控各种动作。保持操作过程中动作的简单明了，以方便用户理解并记忆。只要有可能，使用绝大多数用户已经熟识的各种标准操控方式和行为。

Provide ample opportunity to cancel operations before they begin, and be sure to get confirmation when the user initiates a potentially destructive action. Whenever possible, allow users to gracefully stop an operation that’ s underway.
提供多种机会使用户能够取消某项操作，并且当用户开始某个潜在破坏性的操作能够得到确认。只要有可能，允许用户能够正常停止任一操作过程。

Aesthetic Integrity 审美的完整性（既美观与实用间的平衡）

Although the ultimate purpose of an application is to enable a task, even if that task is playing a game, the importance of an application’ s appearance should not be underestimated. This is because appearance has a strong impact on functionality: An application that appears cluttered or illogical is hard to understand and use.
虽然一项应用程序的最终目的是使任务得到完成，即使这项任务是游戏，该应用程序的外观重要性同样不可被低估。因为外观对功能发挥有着巨大影响：看上去零乱或不合逻辑的应用程序很难让人理解和使用。

Aesthetic integrity is not a measure of how beautiful your application is. It’ s a measure of how well the appearance of your application integrates with its function. For example, a productivity application should keep decorative elements subtle and in the background, while giving prominence to the task by providing standard controls and behaviors.
审美的完整性并不是指你的应用程序有多光鲜，它是指应用程序的样子与它本身的功能是否能相互结合。例如，一个生产力辅助的应用程序应该附着细小的装饰元素，同时要通过提供标准的操控与行为指示来凸现任务本身。

An immersive application is at the other end of the spectrum, and users expect a beautiful appearance that promises fun and encourages discovery. Although an immersive application

tends to be focused on providing diversion, its appearance still needs to integrate with the task. Be sure you design the user interface elements of such an application carefully, so that they provide an internally consistent experience.

一个沉浸式应用程序则是完全不同一种，用户期望一个美丽的外观其中包含乐趣并且使人乐意探索。虽然一个沉浸式应用程序往往是侧重于提供娱乐性，它的功能与外观仍就需要整合。需谨慎地为这类应用程序设计界面元素，使之可提供一个内部连贯的使用体验。

Chapter 3

Designing an iPhone Application: From Product Definition to Branding

设计一个 iPhone 应用程序：从产品定义到品牌推广

As you develop an iPhone application you need to learn how iPhone OS and various aspects of the mobile environment impact your design decisions. This chapter covers a range of guidelines for application design issues, from product definition to branding, and describes how to address them in an iPhone application.

当你开发一个 iPhone 应用程序，您需要学习 iPhone OS 和各种移动环境如何影响着你的设计决策。本章涵盖的范围为应用程序设计指南的问题，从产品定义到品牌的推广，并介绍了在 iPhone 应用程序如何解决他们。

Create a Product Definition Statement

创建一个产品定义说明

Before you begin designing your application, it’ s essential to define precisely what your application does. A good way to do this is to craft a product definition statement—a concise declaration of your application’ s main purpose and its intended audience. Creating a product definition statement isn’ t merely an exercise. On the contrary, it’ s one of the best ways to turn a list of features into a coherent product.

在你开始设计你的应用程序之前，你必须弄清楚你的应用程序到底用来干什么。有一个很好的方法就是指定一个产品定义说明——一个简明扼要说明关于其主要目的和目标受众。创建一个产品定义说明不仅仅是一个练习。相反，它是将要点说明变成一合理的产品的最好方法之一。

To begin with, spend some time defining your user audience: Are they experienced or novice, serious or casual, looking for help with a specific task or looking for entertainment? Knowing these things about your users helps you customize the user experience and user interface to their particular needs and wants.

首先，花一些时间界定你的用户群：他们是经验达人还是菜鸟，他们严谨还是随意，是为了寻找帮助去做事还是为了娱乐休闲？知道这些能够帮助你为满足他们的特定需求而设计良好的用户界面和帮助其获得用户体验。

Because you’ re designing an iPhone application, you already know a lot about your users. For example:

因为你正在设计一个 IPhone 应用程序，你已经了解了很多关于你用户的资料。例如：

- They're mobile.

  他们是移动状态
- They want to be able to open your application quickly and see useful content immediately .

他们希望能快速打开你的应用程序并且迅速查看有效信息

- They need to be able to accomplish things in your application with just a few taps.
  他们需要在极少操作过程的前提下达成目标。

Now ask yourself what traits might set your users apart from all other iPhone OS users. Are they business people, teenagers, or retirees? Will they use your application at the end of every day, every time they check their email, or whenever they have a few extra moments? The more accurately you define your audience, the more accurate are your decisions about the look, feel, and functionality of your user interface.
现在问问你自己有什么特征可以将你的用户与其他 iPhone OS 用户区分开。他们是商务人士，青少年还是退休人员？他们会在每天结束前用你的应用程序，查看电子邮件时，还是在任意时间段？对你的用户群体定义得越准确，你用户界面的外视图觉和功能就越完美。

For example, if your application helps business people keep track of their expenses, your user interface should focus on providing the right categories and making it easy to enter costs, without asking for a lot of details that aren’ t central to the task. In addition, you might choose a subtle color palette that appears professional and is pleasant to look at several times a day.
例如，如果您的应用程序是帮助商务人士跟踪他们的业务开支的，你的用户界面应该着重于提供正确的分类使之更容易输入各类开销，同时避开询问各种无关细节。此外你还可以选择设计一个微妙的颜色专业模板让人能够一天打开多次而不厌烦。

Or, if your application is a game for a target audience of teenagers, you might instead want a user interface that is exciting, language that imparts a feeling of exclusivity, and a color palette that evokes current fashions.
或者，如果您应用程序的目标受众是一款青少年游戏的，你可以将你的用户界面设计及成令人兴奋，言语透露个性，颜色能引领潮流的感觉。

Finally, examine the set of features you intend to deliver. With the image of your user audience in mind, try to distill the list of features into a single statement, a product definition statement, that describes the solution your product offers and who your users are. For example, the desktop iPhoto application allows users to, among other things, organize, edit, share, print, and view photos. But a good product definition statement doesn’ t just focus on features, it also describes the intended audience. Therefore a sound product definition statement for iPhoto could be “An easy-to-use photo management application for amateur photographers.” Notice how important it is to include a definition of your user audience in the product definition statement: Imagine how different an application iPhoto would be if it was designed to be “an easy-to-use photo management application for professional photographers.”
最后，检查你打算提交的要点设置。带着脑海中的用户形象，将各项要点提炼成一份简单的说明文件，一份产品定义说明，其中包含了产品提供的解决方案和目标用户的定义。例如在桌面上的 iPhoto 应用程序允许用户使用其功能，编辑，共享，打印和查看相片。但是，一个好的产品定义说明不仅仅侧重于产

品特点，它同时也会描述潜在用户。因此 iPhoto 的产品定义说明就有可能是“一个为业余摄影师管理照片提供方便的应用程序”。注意包含你的用户群定义在该说明中是十分重要的。想象一下它如果被设计成一个“为专业摄影师管理照片提供方便的应用程序”那将会有多大的不同。

A good product definition statement is a tool you should use throughout the development process to determine the suitability of features, tools, and terminology. It’ s especially important to eliminate those elements that don’ t support the product definition statement, because iPhone applications have no room to spare for functionality that isn’ t focused on the main task.
一个好的产品的定义说明是一个工具，贯穿于整个开发过程中，帮助确定下特征要点，工具及术语的适合度。这对于剔除那些有悖于产品定义说明的设计元素尤其重要，因为 IPhone 应用程序没有多余的空间浪费在非主要目的的任务上。

Imagine, for example, that you’ re thinking of developing an iPhone application people can use when they shop for groceries. In the planning stage, you might consider including a wide range of activities users might like to perform, such as:
试想，你正在开发一个人们用于食品杂货店的 IPhone 应用程序。在规划阶段，你可以尽可能多的去考虑用户的可能性行为，诸如：

Getting nutritional information about specific foods
获取有关特定食品的营养信息
Finding coupons and special offers
寻找优惠券和优惠信息
Creating and using shopping lists
创建并使用购物清单
Locating stores
定位商店
Looking up recipes
查找食谱
Comparing prices
对比价格
Keeping a running total of prices
掌握总体价格

However, you believe that your users are most concerned with remembering everything they need to buy, that they would like to save money if possible, and that they’ re probably in a hurry to get home with their purchases. Using this audience definition, you craft a product definition statement for your application, such as “A shopping list creation and coupon-finding tool for people in a hurry. ” Filtering your list of potential features through this product definition

statement, you decide to focus primarily on making shopping lists easy to create, store, and use. You also offer users the ability to find coupons for the items on their list. Even though the other features are useful (and might become primary features of other applications), they don't fit the product definition statement for this application.
但是，你相信你的客户能记住要购买的物品，同时也会尽可能省钱，他们可能会很匆忙地带着购置物回家。使用对象确认能帮助应用程序确定产品定义说明，例如“一个短时间购物清单和寻找优惠券的工具”。通过这个产品定义说明过滤清单中潜在的特点，您决定把精力主要集中于购物清单创建，存储和使用，并让客户能在清单上找到优惠券。别的特性即使非常有用（甚至有可能成为其他应用程序的特点），但它极可能并不适用于该应用程序。

When you've settled on a solid product definition statement and you've started to use it as a filter for your proposed features, you might also want to use it to make sure your initial decision on application type is still the right one. If you began your development process with a specific application type in mind, you might find that the process of defining a product definition statement has changed the landscape. (See "Three Application Styles" (page 18) for more on different types of applications you can develop.)
当你完成了实物产品的定义说明并且你已经开始用它来过滤产品的一些特性，你也许仍想要用它来确定最开始的决定是正确的。如果你开始在脑海中想象研发一个特定应用程序的过程，你可能 发现，在确定一个产品的定义声明的过程中已经改变了它的面貌。（详见“"Three Application Styles”以获得更多不同类型可开发的应用程序。）

Incorporate Characteristics of Great iPhone Applications
优秀 iPhone 应用程序的具体特性

Great iPhone applications do precisely what users need while providing the experience users want. To help you achieve this balance in your application, this section examines some of the characteristics of great iPhone applications and provides advice on how to build them into your product.
好的 iPhone 应用程序往往能准确地定位用户的需求并提供相应的使用体验。为了更好的帮助你达到这个平衡，本章例举一些优秀 iPhone 程序的特性并且提供如何将它们运用到产品中的建议。

Build in Simplicity and Ease of Use
增强简洁性与易用性

Simplicity and ease of use are fundamental principles for all types of software, but in iPhone applications they are critical. iPhone OS users are probably doing other things while they simultaneously use your application. If users can't quickly figure out how to use your

application, they’re likely to move on to a competitor’s application and not come back.
简单和易用是适用于所有类型软件的的基本原则，但在 iPhone 应用程序他们是至关重要的。iPhone 用户在使用你的应用软件同时可能在做其他事情，如果用户不能迅速了解如何使用你的程序，他们很可能会转移到竞争对手产品中去不再回来。

As you design the flow of your application and its user interface, follow these guidelines to build in simplicity and ease of use:
当您设计您的应用程序和用户界面流程时，请遵循这些准则，确保简洁性和易用性

- Make it obvious how to use your application.
  确保你应用程序的操作方法直观明了。
- Concentrate frequently used, high-level information near the top of the screen.
  确保常用功能和常用信息置于屏幕顶上方
- Minimize text input
  文本输入量最小化.
- Express essential information succinctly.
  尽量简洁地表达必要的信息
- Provide a fingertip-size target area for all tappable elements.
  为所有点触元素提供一个合适大小的触碰区域。

The following sections explain each guideline for simplicity and ease of use in more detail.
以下各节分别说明简洁性和易用性的细节导引。

Make It Obvious
使之显而易见

You can’t assume that users have the time (or can spare the attention) to figure out how your application works. Therefore, you should strive to make your application instantly understandable to users.
你不能假设用户有时间（或能够腾出时间）去研究应用程序如何使用。因此，你应该努力使您的应用程序能即刻被用户所理解。

The main function of your application should be immediately apparent. You can make it so by minimizing the number of controls from which users have to choose and labeling them clearly so users understand exactly what they do. For example, in the built-in Stopwatch function (part of the Clock application), shown in Figure 3-1, users can see at a glance which button stops and starts the stopwatch and which button records lap times.
你应用程序的主要用途应该是一目了然的，你可以最大程度地减少控件以便用户能够清楚地知道自己在干什么。例如，在内置的秒表功能（时钟应用程序的一部分），详见图 3-1，用户可以一眼看出哪一个按钮停止秒表，哪一个按钮启动秒表或是哪一个按钮记录每一次的时间。

Figure 3-1 The built-in Stopwatch function makes its usage obvious

内置的秒表功能的使用方法显而易见

Think Top Down 自上而下

People can tap the screen of an iPhone OS based device with their fingers or their thumbs. When they use a finger, people tend to hold the device in their nondominant hand (or lay it on a surface) and tap with a finger of the dominant hand. When they use thumbs, people either hold the device in one hand and tap with that thumb, or hold the device between their hands and tap with both thumbs. Whichever method people use, the top of the screen is most visible to them.

人们可以用他们的手指或拇指轻点 iPhone OS 设备与屏幕。在使用设备时，人们往往用非习惯性手扶住设备（或将它平放于水平面），而习惯性手的手指用于触碰界面。当人们使用拇指操作，用户要么单手持握，要么双手持握。无论是哪种方法，屏幕的顶端都是最显而易见的区域。

Because of these usage patterns, you should design your application’ s user interface so that the most frequently used (usually higher level) information is near the top, where it is most visible and accessible. As the user scans the screen from top to bottom, the information displayed should progress from general to specific and from high level to low level.

由于这些使用范式，应用程序的用户界面应该将最常使用（通常较高的水平）的信息放置于屏幕顶部附近，那是最明显和最方便的方位。用户往往会从上到下扫描整个屏幕，而信息的分布位置必须是从一般到具体的，从高层次到低层次。

Minimize Required Input
最大限度地减少所需的输入

Inputting information takes users' time and attention, whether they tap controls or use the keyboard. If your application requires a lot of user input before anything useful happens, it slows users down and can discourage them from persevering with it.
用户输入的信息需要的时间和注意力，需要触点控制、键盘控制。如果您的应用程序需要用户在得到有用的反馈前输入很多，它会减慢用户的使用速度并不利于他们乐意去使用。

Of course, you often need some information from users, but you should balance this with what you offer them in return. In other words, strive to provide as much information or functionality as possible for each piece of information users provide. That way, users feel they are making progress and are not being delayed as they move through your application.
当然，你经常会需要一些来自用户的信息，但你同时应该注意平衡你提供的，与反馈的信息。换句话说，对于用户的输入信息，需提供尽可能多的反馈信息及相关功能。这样，用户会觉得他们通过该程序正在取得工作进展，而并没有拖延。

When you ask for input from users, consider using a type of table view (or a picker) instead of text fields. It's usually easier for users to select an item from a list than to type words. For details on table views and pickers, see "Table Views" (page 89) and "Pickers" (page 114), respectively.
当你要求用户输入时，可以考虑使用一个表格视图（或拨选器），而不是文本字段。它通常比输入单词使用户更容易选择。有关表的意见和细节，请分别看"Table View"和"Picker"。

Express Information Succinctly
简明地表达有效信息

When your user interface text is short and direct, users can absorb it quickly and easily. Therefore, identify the most important information, express it concisely, and display it prominently so users don't have to read too many words to find what they're looking for or to figure out what to do next.
当你的用户界面文本是简短并且直接的，用户可以迅速掌握并理解。因此，确定最重要的信息，简洁地解释它，并且突出它的显示，以便用户不必阅读太多的文字就能找出下一步该怎么做。

To help you do this, think like a newspaper editor and strive to convey information in a condensed, headline style. Give controls short labels (or use well-understood symbols) so that users understand how to use them at a glance.

为了帮助您做到这一点，试着把自己想象成一个报纸的编辑，努力传达一个简明，标题式风格。使控制标签尽量简短（或用好理解的符号），让用户能够一目了然。

Provide Fingertip-Size Targets

提供合适大小的触点区域

If your layout places controls too close together, users must spend extra time and attention being careful where they tap, and they are more likely to tap the wrong element. A simple, easy-to-use user interface spaces controls and other user-interaction elements so that users can tap accurately with a minimum of effort.

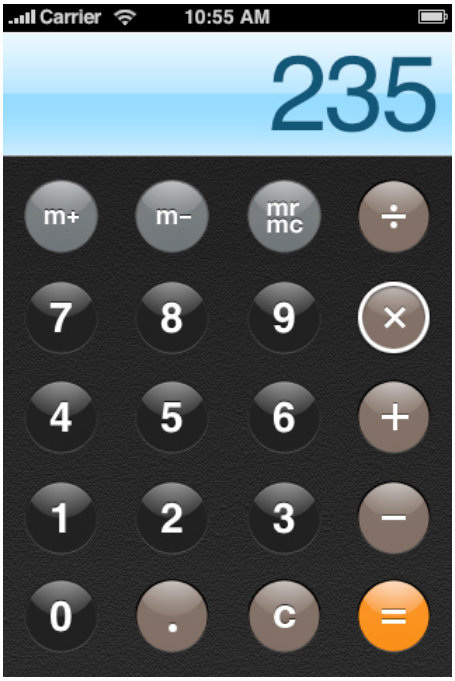
如果你的控件布局相互太接近，用户自然会花费额外的时间和注意力去避免按错键，必然地，他们犯错的几率也就更大。一个简单易操作的用户界面往往会将控制键和其它用户交互要素区分，使用户不会轻易出错。

For example, the built-in Calculator application displays large, easy-to-tap controls that each has a target area of about 44 x 44 pixels. Figure 3-2 shows the Calculator application.

例如 ,内置计算应用程序的按键显示非常易于操控。它的每一个按键都拥有自己的专属区域 ,大约 44x44 像素。图 3-2 为该计算器。

Figure3-2 The built-in Calculator application displays fingertip-size controls

内置的计算器应用程序显示尺寸控制



Focus on the Primary Task

An iPhone application that establishes and maintains focus on its primary functionality is satisfying and enjoyable to use. As you design your application, therefore, stay focused on your product definition statement and make sure every feature and user interface element supports it. See “Create a Product Definition Statement” (page 35) for some advice on how to create a product definition statement.

一个 iPhone 应用程序，被创建并关注于其的首要功能将满足用户需求并使之使用愉快。当您设计应用程序时，始终将焦点要放在产品定义说明上，并确保所有功能和用户界面元素都支持它。请参阅“Create a Product Definition Statement “（一些关于如何创建一个产品定义说明）的意见。

A good way to achieve focus is to determine what’s most important in each context. As you decide what to display in each screen always ask yourself. Is this critical information or functionality users need right now? Or, to think of it in more concrete terms. Is this information or functionality the user needs while shopping in a store or while walking between meetings? If not, decide if the information or functionality is critical in a different context or if it’s not that important after all. For example, an application that helps users keep track of car mileage loses focus on this functionality if it also keeps track of car dealer locations.

一个好的方法来聚焦重点是在任何情景下都要确定什么是最重要的。在确定每一屏上显示什么时，问问自己：这个重要的信息或功能是用户急需的吗？或者更深入地思考具体环境，对于该信息或功能用户在购物或是赶去开会期间会需要用到吗？如果不是，再确定在其他具体的环境下它是否是必须的信息或功能；或者它根本就不是重要的。例如，一个帮助用户记录汽车行驶里数的程序是否也需要记录原始位置？

When you follow the guidelines for making your application simple and easy to use, you help make your solution focused. In particular, you want to make the use of your application obvious and minimize user input. This makes it easier for users to arrive quickly at the most important parts of your application, which tightens the focus on your solution (for specifics on these guidelines, see “Build in Simplicity and Ease of Use” (page 36))

当你按照本导引从而尽力使你的应用程序简约易用，将有利于你集中解决问题。特别是，你想让你的应用程序直观并且最大程度地减少用户的输入过程。你所专注的解决方案将大大方便你的用户达到自己的操作目的。（详情参见具体章节“Build in Simplicity and Ease of Use”）

For example, the built-in Calendar application (shown in Figure 3-3) is focused on days and the events that occur on them. Users can use the clearly labeled buttons to highlight the current day, select a viewing option, and add events. The most important information, that is, the days and the events associated with them, is the most prominent. User input is simplified by allowing users to choose from lists of event times, repetition intervals, and alert options, instead of requiring keyboard entry for all input.

例如，内置的日历应用程序（如图 3-3 中所示）主要关注日期和该天的重要事件。用户可以用清晰的标签强调当前日期，并且添加事件注释。最重要的是，强调日期和事件的关联。事件列表选取，重复间隔，

提醒选项等都将帮助用户更简单的选取输入信息，而取代总出现文本输入。

Figure 3-3

The built-in Calendar application is focused on days and events

内置的日历程序主要着重于日期和事件

Communicate Effectively 高效沟通

Communication and feedback are as important in iPhone applications as they are in desktop computer applications. Users need to know whether their requests are being processed and when their actions might result in data loss or other problems. That said, it’ s also important to avoid overdoing communication by, for example, alerting the user to conditions that aren’ t really serious or asking for confirmation too often.

沟通和反馈，在 iPhone 的应用程序中显得尤为重要，如同桌面电脑上的应用程序。用户必须知道他们的请求是否正在被处理或者他们的行为是否正造成数据的丢失或其他事故。同时，也要避免出现矫枉过正的沟通行为，例如，在不必要的情况下警告用户或者过多的让用户确认自己的操作行为。

Animation is a great way to communicate effectively, as long as it doesn’ t get in the way of users’ tasks or slow them down. Subtle and appropriate animation can communicate status, provide useful feedback, and help users visualize the results of their actions. Excessive or gratuitous animation can obstruct the flow of your application, decrease its performance, and annoy users.

动画是一种极佳地，积极有效地沟通方式，前提是不会过度影响用户的操作过程。微妙恰当的动画可以起到沟通，提供反馈，并帮助用户操作结果视觉化。反之过多或无意义的动画则会阻碍您应用程序的运行，降低其性能，并惹恼用户。

In all your text-based communication with users, be sure to use user-centric terminology; in particular, avoid technical jargon in the user interface. Use what you know about your users to determine whether the words and phrases you plan to use are appropriate. For example, the Wi-Fi Networks preferences screen uses clear, nontechnical language to describe how the device connects to networks, as shown in Figure 3-4.

在所有文本为基础的沟通中，一定要使用以用户为中心的术语，避免技术术语在用户界面中出现。在使用你所知道的用户用语时，确定该单词和短语是否适当。例如，“Wi-Fi 设置”非常清晰，是在用非技术性用语来描述如何连接到网络的设备。如图 3-4 所示。

Figure 3-4 Use user-centric terminology in your application’ s user interface

在你的界面中使用以用户为中心的术语

Support Gestures Appropriately

恰当地手势支持

People use their fingers to operate the unique Multi-Touch interface of iPhone OS-based devices, tapping, flicking, and pinching to select, navigate, and read web content and use applications. There are real advantages to using fingers to operate a device: They are always available, they are capable of many different movements, and they give users a sense of immediacy and connection to the device that’ s impossible to achieve with an external input device, such as a mouse.

人们用他们的手指来操作独特的多触点的 iPhone OS 设备界面。轻点，轻划和轻按用以选择，导航，阅读网页内容和使用应用程序。用手操控设备有它显而易见的优势：它们总是可实现的，它们可以胜任不同的行为动作，它们能带给用户直接和与设备联系在一起的感觉。这对于外部输入设备来说是不可能实现的，比如鼠标。

However, fingers have one major disadvantage: They are much bigger than a mouse pointer, regardless of their size, their shape, or the dexterity of their owner. In the context of a display screen, fingers can never be as precise as a mouse pointer.
不过，手指有一个主要缺点：它们始终比一个鼠标指针要大，更不要说手指的尺寸，形状甚至本身的灵巧度。在显示屏方面，手指永远比不上鼠标指针来的精准。

Fortunately, you can meet the challenges of a finger-based input system by having a good user interface design. For the most part, this means making sure your layout accommodates the average size of a fingertip. It also means responding to finger movements with the actions users expect. Users perform specific movements, called gestures, to get particular results. For example, users tap a button to select it and flick or drag to scroll a long list. iPhone users understand these gestures because the built-in applications use them consistently. To benefit from users' familiarity, therefore, and to avoid confusing them, you should use these gestures appropriately in your application.
幸运的是，在一个优秀的交互界面设计中可以满足手触输入的挑战。在大多数情况下，这意味着确保你的布局可容纳一个指尖的平均尺寸，也意味着需要及时响应用户手指移动的操作行为；用户执行特定的动作，即手势。例如，用户点击一个按钮用以选择，轻划或拖移用以拨动一列清单。iPhone 用户能理解这些手势因为 iPhone 内置应用程序会与这些手势保持一致性。受益于用户的熟悉，同时避免混淆他们，你应该在应用程序中恰当地使用他们。

The more complex gestures, such as swipe or pinch open, are also used consistently in the built-in applications, but they are less common. In general these gestures are used as shortcuts to expedite a task, not as the only way to perform a task. When viewing a list of messages in Mail, for example, users delete a message by revealing and then tapping the Delete button in the preview row for the message. Users can reveal the Delete button in two different ways:
更复杂的手势，例如重划或者双指拉开，在内置应用程序中也具有一致性，但是他们不太常见。一般来说，这些手势被用作为快捷方式，而不是唯一的方式来执行任务。例如，当你查看邮件信息清单时，用户通过展开邮件然后轻点删除按钮来删除消息。用户可以通过两种不同的方式来展开“删除键”

- Tap the Edit button in the navigation bar, which reveals a delete control in each preview row. Then, tap the delete control in a specific preview row to reveal the Delete button for that message.
点击导航条的编辑按钮，然后预览栏会展开删除控制键，接着，点击在特定预览栏里该信息的删除控制键。
- Make the swipe gesture across a specific preview row to reveal the Delete button for that

message.
在那条需要删除的信息预览框内使用重划手势，以显示删除按钮。

The first method takes an extra step, but is easily discoverable because it requires only the tap and begins with the clearly labeled Edit button. The second method is faster, but it requires the user to learn and remember the more specialized swipe gesture.
第一种方法需要一个额外的步骤，但它更容易被理解和发现。因为它只需要轻击提示简洁明确编辑按钮即可。第二种方法更快捷，但它需要用户掌握并记忆住额外特定的手势。

To ensure that your application is discoverable and easy to use, therefore, try to limit the gestures you require to the most familiar, that is, tap and drag. You should also avoid making one of the less common gestures, such as swipe or pinch open, the only way to perform an action. There should always be a simple, straightforward way to perform an action, even if it means an extra tap or two.
为确保您的应用程序更直观并且易于使用，请设法减少你的手势要求。要求（人们）使用最熟悉的手势：点击和拖拽。你同时也要尽力避免那些鲜用手势例如重划或者双指拉开成为某种执行操作的唯一方式。尽量使用精简直观的方法去操控程序，即使需要多击对象一次或两次。

In most applications, it’ s equally important to avoid defining new gestures, especially if these gestures perform actions users already associate with the standard gestures. The primary exception to this recommendation is an immersive application, in which custom gestures can be appropriate. For example, a productivity application that requires users to make a circular gesture to reveal the Delete button in a table row would be confusing and difficult to use. On the other hand, a game might reasonably require users to make a circular gesture to spin a game piece.
在大多数应用程序中，避免创造新的手势行为也显得尤为重要，尤其是当用户对该控制行为已经有了先入为主的标准手势印象。如果有例外那可能就是在沉浸式的应用程序中，自定义的手势也可适用。例如，一种生产力辅助程序要求用户做一个圈形手势实现删除行为，这会带来很大困扰并且操作困难。相反，一个游戏程序可能会要求用户做一个圈形手势去旋转画面就显得合理多了。

Table 3-1 lists the standard gestures users can perform. Be sure to avoid redefining the meaning of these gestures; conversely, if you support these actions in your application, be sure to respond appropriately to the gestures that correspond to them. For more information on how to handle events created by gestures, see iPhone Application Programming Guide.
表 3-1 列出了用户可以执行的标准手势。一定要避免重新定义它们的手势含义。相反，如果在你的应用程序中选择支持它们，将会得到良好的反馈。想要了解更多有关如何处理手势创建活动的信息，请参阅 iPhone Application Programming Guide。

Table 3-1 Gestures users make to interact with iPhone OS-based devices
用户用于与 iPhone 设备互动的手势

Gesture	Action
Tap 点击	To press or select a control or item (analogous to a single mouse click). 按下或选择一个控件或条目（类似于普通的鼠标点击）
Drag 拖移	To scroll or pan. 滚动或平移
Flick 轻划	To scroll or pan quickly. 快速的滚动或平移
Swipe 重划	In a table-view row, to reveal the Delete button. 在一个表格视图的行中，显示删除按键
Double tap 双击	To zoom in and center a block of content or an image. 放大并将内容或图片置于中央 To zoom out (if already zoomed in). 缩小（如果已经放大）
Pinch open 双指拉开	To zoom in. 放大
Pinch close 双指拉近	To zoom out. 缩小
Touch and hold 长按	In editable text, to display a magnified view for cursor positioning. 在可编辑文本中，用来显示放大视图中的光标定位

Incorporate Branding Elements Cautiously

谨慎表达品牌推广元素

Branding is most effective when it is subtle and understated. People use your iPhone application to get things done or to be entertained; they don’t want to feel as if they’re being forced to watch an advertisement. Therefore, you should strive to incorporate your brand’s colors or images in a refined, unobtrusive way. For example, you might use a custom color scheme in views and controls.

不经意时，品牌往往是最具影响的。人们用你的 iPhone 应用程序开始某件事情或者娱乐休闲，他们不想感觉好像自己被强迫观看广告。因此，你应该努力致力于将你品牌的颜色或图像自然的融入其中。例如，你可以使用一个自定义的视图和配色方案在视图区或控件中。

The exception to this is your application icon, which should be focused on your brand. (The application icon is the icon users can see on their Home screens after they install your application.) Because users see your application icon frequently, it’s important to spend some

time balancing eye-appeal with brand recognition. For some guidelines on creating an application icon, see “Application Icons”

对于品牌效应来说，你的应用程序图标必须强化你的品牌。（应用程序图标是指安装应用程序之后用户可以在主界面看到的图标）由于你的用户能够经常看见你的应用图标，花时间在品牌辨识度上博取眼球显得十分重要。详情请参见“创建应用程序图标”导航，详见“Application Icons”。

CHAPTER 4

Handling Common Tasks
一般任务处理

iPhone applications handle many common tasks in ways that may seem different to you, if your experience is with desktop or laptop computer applications. This section describes these tasks from the human interface perspective; for the technical details you need to implement these guidelines in code, see iPhone Application Programming Guide.

如果你的经验是台式机或笔记本电脑运用的话，iPhone的应用程序处理方式似乎是很不同的。本节从用户界面的角度描述了这些任务;如需具体技术细节，则需代码来编译实现这些设计导引，详见iPhone Application Programming Guide。

Starting 启动

iPhone applications should start instantly so users can begin using them without delay. When starting, iPhone applications should:

iPhone应用程序应立即迅速地启动，这样用户没有延时即开始使用。当启动时，iPhone程序应该：

- Specify the appropriate status bar style (see “The Status Bar” (page 67) for information about the available styles).

指定适当的状态条样式（见“The Status Bar”有关可用的样式信息）。

- Display a launch image that closely resembles the first screen of the application. This decreases the perceived launch time of your application. For more information, see “Launch Images” (page 131).

呈现非常类似于应用程序初始画面的启动截图。这减少了应用程序的启动时用户所感受到的时间。欲了解更多信息，请参阅“Launch Images”。

- Avoid displaying an About window, a splash screen, or providing any other type of startup experience that prevents people from using your application immediately.

避免显示“相关信息”窗口，效果画面，或任何其它样式，这些都将使用户不能立即使用程序。

- By default, launch in portrait orientation. If you intend your application to be used only in landscape orientation, launch in landscape regardless of the current device orientation. Allow users to rotate the device to landscape orientation if necessary.

默认情况下，启动画面是纵向的。如果你打算你的应用程序只能用于横向，可以忽视当前设备的方向而直接横向发送。应允许用户在需要时通过旋转设备至横向模式。

A landscape-only application should support both landscape orientations—that is, with the Home button on the right or on the left. If the device is already physically in a landscape orientation, a landscape-only application should launch in that orientation. Otherwise, a landscape-only application should launch in the orientation with the Home button on the right by default.
一个只有横向模式的应用程序应该支持双横向，也就是说，主界面按钮可以在右侧或左侧。如果设备已经处在一个横向模式，那只有横向模式的应用程序就应在这个方向上启动。其它，只有横向模式的应用程序的主界面按钮启动时默认位于右侧。

■ Restore state from the last time your application ran. People should not have to remember the steps they took to reach their previous location in your application.
恢复应用程序上一次关闭时的使用状态。人们可以不需要记住他们上一次在你程序中所到达的位置。

Important: Don’ t tell users to reboot or restart their devices after installing your application. If your application has memory-usage or other issues that make it difficult to run unless the system has just booted, you need to address those issues. For example, see “Using Memory Efficiently” in iPhone Application Programming Guide for some guidance on developing a well-tuned application.
注意：不要让用户在安装了应用程序后重新启动设备。如果应用程序有内存使用或其它难以运行的问题，只能在系统刚刚启动时加载，则你需要解决这些问题。例如，借鉴iPhone Application Programming Guide中“Using Memory Efficiently”关于开发调整好的程序之相关指导。

Stopping 停止

People quit an iPhone application by opening a different application. In particular, note that people don’ t tap an application close button or choose Quit from a menu. In iPhone OS 4.0 and later, and on certain devices, the quitting application moves to a suspended state in the background. All iPhone applications should:
用户通过打开另一个应用程序来关闭前一个应用程序。特别注意地是，用户不会去点击程序地关闭按钮或者从菜单中选择退出。在装有iPhone 4.0及以后版本的设备中，退出的应用程序将在后台保存起来。所有的iPhone应用程序应该：

■ Be prepared to receive an exit or terminate notification at any time. Therefore, save user data as soon as possible and as often as reasonable.
准备好在任何时候都能接收退出操作或给出提醒提示。因此，应尽可能快且实时保存用户数据。
■ Save the current state when stopping, at the finest level of detail possible. For example, if your application displays scrolling data, save the current scroll position.

当程序停止时，应尽可能详细地保存当前状态。例如，如果您的应用程序显示滚动数据，保存当前滚动位置。

iPhone applications should never quit programmatically because doing so looks like a crash to the user. There may be times, however, when external circumstances prevent your application from functioning as intended. The best way to handle this is to display an attractive screen that describes the problem and suggests how users can correct it. This helps users in two ways:
iPhone应用程序应避免由于编程原因突然退出，因为这样做会让用户觉得是崩溃了。然而也许有些时候会发生这种情况，即当外部环境按预定好的阻止您的应用程序。最好的方法来处理这种情况就是呈现一个有吸引力的显示屏幕，它描述有出现的问题，并建议用户如何纠正它。这在两个方面有助于用户：

■ It provides feedback that reassures users that there’ s nothing wrong with your application
它提供反馈，告诉用户应用程序本身并没有问题。

■ It puts users in control, letting them decide whether they want to take corrective action and continue using your application or press the Home button and open a different application.
使用户处于主导地位，让他们决定是否要采取纠正措施并继续使用该应用程序，或按主界面按钮，打开另一个应用程序。

If certain circumstances prevent only some of your application's features from working, you can display either a screen or an alert when users activate the feature. Although an alert doesn't allow much flexibility in design, it can be a good choice if you can:
如果某些情况下只是应用程序的某些功能无法工作，当用户激活该功能时可以显示屏幕或提醒。虽然提醒在设计上并不灵活和柔性，但它可以是一个很好的选择，如果可以：

■ Describe the situation very succinctly
非常简洁地描述情况
■ Supply a button that performs a corrective action
提供一个纠正错误的按钮
■ Display the alert only when users try to access the feature that isn’ t functioning
仅当用户尝试访问不能运作的功能时显示提醒

As with all alerts, the less users see them, the more effective they are. See “Using Alerts” (page 80) for more information about creating alerts.
对于所有的提醒，用户越少看到它们，它们越有效。见“Using Alerts”以获取更多有关创建提醒的信息。

Accommodating Multitasking 适应多线程

Thriving in a multitasking environment hinges on achieving a harmonious coexistence with other applications on the device. At a high level, this means that all applications should:
多线程环境的兴盛，取决于设备上的应用程序能实现和谐共处。在高级别上，这意味着所有应用程序应该：

■ Handle interruptions or audio from other applications gracefully
优雅地处理其它应用程序的中断和音频

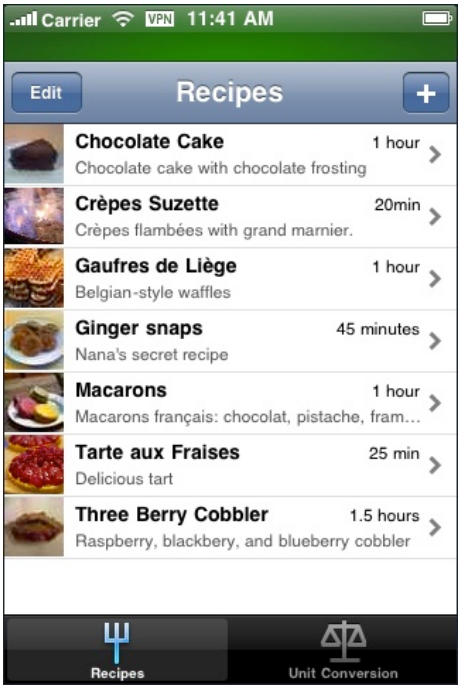
■ Stop and restart (that is, transition to and from the background) quickly and smoothly
快速流畅地停止和重起（应用程序既转进或转出后台）

■ Behave responsibly when not in the foreground
当不在前台运行时行为响应

The following specific guidelines help your application succeed in the multitasking environment introduced in iPhone OS 4.0:
以下具体的指导会帮助你的应用程序在iPhone OS 4.0的多线程环境中取得成功：

■ **Be prepared for interruptions, and be ready to resume.** Multitasking increases the probability that a background application will interrupt your application. Other features, such as the presence of ads and faster application-switching, can also cause more frequent interruptions. The more quickly and precisely you can save the current state of your application, the faster people can relaunch it and continue from where they left off.
准备好被打断与恢复。多线程增加了一个在后台的程序转入前台并打断你的应用程序的可能性。iOS的其他特性如广告的出现，快捷程序开关，都将导致更频繁地打断。越快越精确地保存你应用程序的当前状态，就能越快重启程序并恢复到前一次打断的状态。

■ **Make sure your UI can handle the double-high status bar.** The double-high status bar appears during events such as in-progress phone calls, audio recording, and tethering. In unprepared applications the extra height of this bar can cause layout problems. For example, the UI can become pushed down or covered. In a multitasking environment, it’ s especially important to be able to handle the double-high status bar properly because there are likely to be more applications that can cause it to appear. You can trigger the double-high status bar during testing to help you find and correct any views that don’ t handle it well. (To learn how to do this using the Simulator, see “Manipulating the Hardware” in iPhone Development Guide.)
确保你的UI可处理双高状态栏。当程序运行时出现电话呼叫，音频录音及tethering时，双高状态栏就会出现。应用程序如果为状态栏的高度做好调整准备将引起布局问题。例如，UI将会被推下或覆盖。在多线程环境中，处理双高状态栏的准备是重要的，因为越来越多的应用程序会导致该状况的出现。你可以在测试时触发双高状态栏，以发现并修正任何可能没有处理好的视图。（了解如何利用模拟器测试，见iPhone Development Guide中的“Manipulating the Hardware”）



■ **Be ready to pause activities that require people's attention or active participation.** For example, if your application is a game or a media-viewing application, make sure your users don't miss any content or events when they switch away from your application. When people switch back to a game or media viewer, they want to continue the experience as if they'd never left it.

准备好暂停活动以求人们的关注与积极参与。例如，你的应用程序是一个游戏或者媒体观看程序，确保你的用户在切换出你的应用程序时不丢失任何内容与事件。当人们切换回来游戏或媒体视图时，他们可以继续游戏或观看的体验，就像从未离开过一样。

■ **Ensure that your audio behaves appropriately.** Multitasking makes it more likely that other media activity is occurring while your application is running. It also makes it more likely that your audio will have to pause and resume to handle interruptions. For specific guidelines that help you make sure your audio meets people's expectations and coexists properly with other audio on the device, see "Using Sound" (page 59).

确保你音频的恰当行为。多线程使其它媒体活动侵占你正在运行的应用程序的可能性更多。同时也使你的音频不得不暂停与恢复以处理打断的情况更多。为了确保你的音频符合人们的期望并与其它音频和谐共存于设备上，请参考"using Sound"的详细指导。

■ **Use local notifications sparingly.** An application can arrange for local notifications to be sent at specific times, whether the application is suspended, running in the background, or not running at all. For the best user experience, avoid pestering people with too many notifications, and follow the guidelines for creating notification content, described in "Enabling Local and Push Notifications" (page 53).

有节制地使用本地通知。应用程序会在特定时间发送本地通知，无论应用程序在暂停中，在后台中还是根本不在运行。为了最佳的用户体验，避免使用过多通知来纠缠用户，并且遵循“Enabling Local and Push Notifications”中的创建通知内容的指导建议，

■ **When appropriate, finish user-initiated tasks in the background.** When people initiate a task, they usually expect it to finish even if they switch away from your application. If your application is in the middle of performing a user-initiated task that does not require additional user interaction, you should complete it in the background before suspending.
在合适时，在后台完成用户创建的任务。当用户创建了一个任务，他们一般会希望任务即使在程序切换入后台时也能继续完成。如果你的应用程序正在执行一项用户创建的任务，那么不要再附加交互活动，直接在后台中自行完成它。

Hosting Ads 广告托管

In iPhone OS 4.0 and later, you can allow advertisements to display within your application and you can receive revenue when users see or interact with them. It’ s essential that you plan when and how to integrate ads with your UI so that people are motivated to view them without being distracted from your application.
在iPhone OS 4.0及以后的系统中，可以允许将广告置入并显示在你开发的应用程序上，并且在用户观看并与这些广告互动时，你还可以为此得到收入。计划何时及如何整合这些广告到你的用户界面中以使人们愿意点击广告而不影响你的应用程序的正常运行是十分重要的。

You can host an iAd, which contains the ad content, in a specific view in your UI. When people tap an ad in this view (called a banner view), the iAd performs a preprogrammed action, such as playing a movie, displaying interactive ad content, or launching Safari to open a webpage. The action can display content that covers your UI or it might cause your application to transition to the background.
你可以代理一支iAd，其包含广告内容，显示在你用户界面的一个特别视图中。当人们点击视图（被称作“横幅视图”）中的广告，iAd就呈现一系列程序的动作，诸如播放一段影片，一个互动的广告内容，或者启动Safari以打开一个网页。这个动作可以将显示的内容覆盖在你的用户界面之上，或者将你的应用程序转入到后台中去。

The dimensions of the banner view are:
这个横幅视图的大小是：

■ In portrait, 320 x 50 points 在纵向视图中为320 x 50 points
■ In landscape, 480 x 32 points 在横向视图中为480 x 32 points

To ensure seamless integration with banner ads and to provide the best user experience, follow these guidelines:

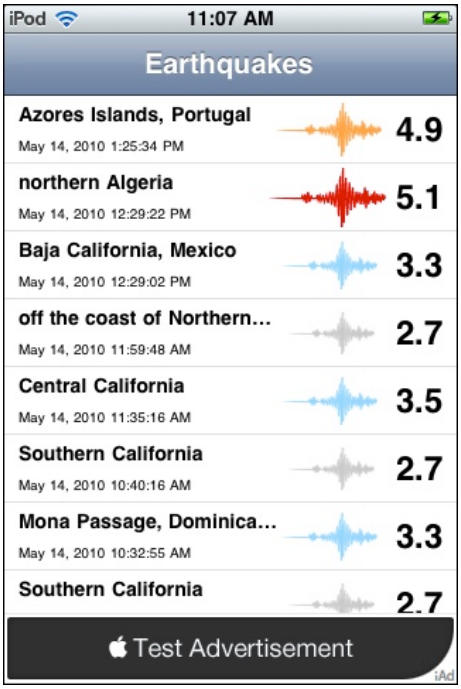
为了确保将横幅视图无缝整合并且提供最佳的用户体验，请遵循这些指导：

Place the banner view at or near the bottom of the screen. This placement differs slightly, depending on the bars that can be in the screen:

请将横幅视图放置在屏幕底部或附近。这个位置可能稍有不同，这取决于屏幕上的各种栏目条：

■ If there are no bars at the bottom of the screen, put the banner view at the bottom edge of the screen.

如果没有其它的栏目条在屏幕底部，横幅视图就放置在屏幕底部边缘。

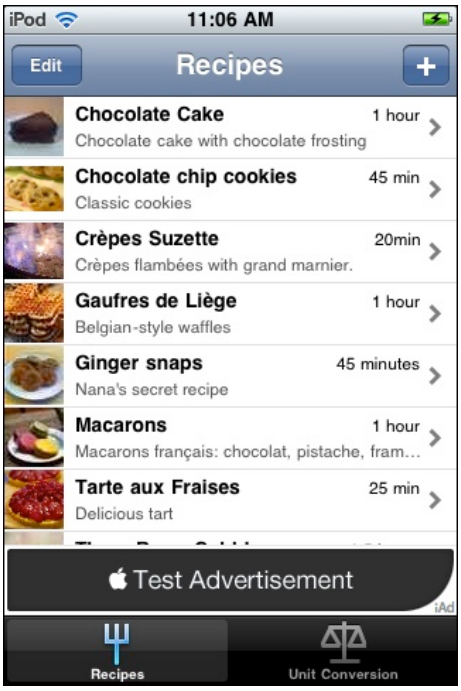


■ If there are no bars at all, put the banner view at the bottom edge of the screen.

如果根本没有栏目条，把横幅视图放置在屏幕底部边缘。

■ If there is a toolbar or tab bar, put the banner view directly above the toolbar or tab bar.

如果有工具栏或标签栏在那，直接放置横幅视图在它们的上部。



Ensure that banner views appear when it makes sense in your application. Although it’ s recommended that a banner view be at the bottom of a screen, you choose which screens should contain banner views. For example, you might want to choose a context that functions as a sort of interlude in the main task of your application. People are more likely to enter an iAd experience when they don’ t feel like they’ re interrupting their workflow to do so. This is especially important for immersive applications such as games: You don’ t want to place banner views where they will conflict with gameplay.

确保横幅视图的出现在你应用程序的视图中是有道理的。尽管横幅视图被建议放置在屏幕底部，但是你可以选择在哪个视图中出现横幅视图。例如，你可能会选择一个上下文关系，将横幅视图做为你应用程序重要任务面板的小插曲。当用户没有感到自己的工作流程被打断时，人们更愿意接受的iAd体验。这对诸如游戏这样的沉浸式应用程序尤为重要：不要在人们打游戏的地方放置横幅视图。

As much as possible, display banner ads in both orientations. It’ s best when users don’ t have to change the orientation of the device to switch between using your application and viewing an ad. Also, supporting both orientations allows you to accept a wider range of advertisements. To learn how to make sure a banner view responds to orientation changes, see iAd Programming Guide.
尽可能在两个方向上显示横幅视图。最好使用户不需改变持握方向以在你的应用程序与广告欣赏间切换。而且，支持双方向使你可以接受广泛范围的广告。学习如何确保横幅视图在方向转变中响应，见iAd Programming Guide。

While people view or interact with ads, pause application activities that require their attention or interaction. When people choose to view an ad, they don’ t want to feel that they’ re missing events in your application, and they don’ t want your application to interrupt them. A good rule of thumb is to pause the same activities you would pause when your application transitions to the background.
当用户观看并与广告互动时，暂停应用程序要求用户注意与互动的所有活动。当人们选择观看一支广告，他们不想感到会错过你应用程序中的重要事件，而且也不想你的应用程序去打断他们。一个好的经验是，当你的应用程序转入后台时，暂停你需要暂停的相同活动。

Don’ t stop an ad, except in rare circumstances. In general, your application continues running and receiving events while users view and interact with ads, so it’ s possible that an event will occur that urgently requires their immediate attention. However, there are very few scenarios that warrant the dismissal of an in-progress ad. One possibility is with an application that provides Voice over Internet Protocol (VoIP) service. In such an application, it probably makes sense to cancel a running ad when an incoming call arrives.
不要终止一支广告，除非在极少数情况下。一般情况下，当用户在观看广告并与之互动时，你的应用程序可以继续运行并接收事件，除非这个事件很紧急，需要用户立刻予以关注。然而极少有这样的情景是必须终止一支正在进行的广告的。一个可能的情况是，这个应用程序提供了 VoIP 服务。在这样的程序中当有呼叫接入时，终止一支正在进行的广告是有道理的。

Note: Canceling an ad might adversely impact the kinds of advertisements your application can receive and the revenue you can collect.
注意：取消广告可能会带来不利影响，毕竟应用程序中的广告能给你带来收入。

Managing Settings or Configuration Options 管理设置或配置选项

iPhone applications can offer settings that define preferred application behaviors or configuration options users can set to change some functionality of the application.
iPhone应用程序可以提供界定应用程序的设置，或提供配置选项使用户可以自己来改变一些应用程序的功能。

Settings should represent information, such as an account name, that users set once and rarely (if ever) change.
设置应该能阐明信息，如用户名，用户只需设置一次或很少有变化（如果有的话）。

Users view application specific settings in the built-in Settings application. Configuration options are values that users might want to change frequently, such as category types displayed in a list; configuration options should be available within the application itself.
用户检视特定应用设置是在内置的“设置”应用程序中。用户可能要经常改变配置选项，如列表中显示的类别;配置选项应在应用程序本身的中提供。

You should consider settings and options to be mutually exclusive. That is, you should not offer both settings and configuration options in your application.
你应该考虑设置和选项是相互排斥的。也就是说，你不能在应用程序中同时提供“设置”和“配置”这两个选项。

It’s best when iPhone applications do not ask users to specify any settings at all. Users can begin to use these applications right away without being asked to supply set-up information. To achieve this in your application, there are a few design decisions you can make:
最好是iPhone应用程序用不着让用户进行任何设置。用户不用被要求提供设置信息就可以马上开始使用这些程序。为了在应用程序中实现这个目标，你可以考虑下面一些设计决策：

■ Focus your solution on the needs of 80 percent of your users.
把解决问题的重心放在你80%的用户身上。
When you do this, the majority of users do not need to supply settings because your application is already set up to behave the way most users expect. If there is functionality that only a handful of users might want, or that most users might want only once, leave it out.
当你这样做时，由于应用程序已经按照大部分用户期望的那样设置好了，所以大多数用户不必再进行设置了。如果有些功能是只有一小部分用户想拥有，或者大部分用户仅仅想经历一次，那忽略这些功能。
■ Get as much information as possible from other sources. If you can use any of the information users supply in built-in application or device settings, query the system for these values; don’t ask users to enter them again.
从其他资源上尽可能多地获取相关信息。如果你可以从内置的应用程序或设备的设置选项中找到并使用用户提供过的任何信息，那就向系统查询这些值，别要求用户再输入一遍。

■ If you must ask for set-up information, prompt users to enter it within your application.
如果你必须询问设置的信息，就在应用程序中提示用户输入相关信息。

Then, as soon as possible, store this information in your application’ s settings. This way, users aren’t forced to quit your application and open Settings before they begin to benefit from your application. If users need to make changes to this information later, they can go to your application’ s settings at any time.
然后尽快将这个信息存储到应用程序的设置中。这样，用户不用在受益于你的应用程序之前被迫退出应用程序，去打开“设置”选项进行设置。如果之后用户需要改变这一信息，他们可以在任何时候到你的应用程序中进行设置。

It’ s not possible for users to open the Settings application without first quitting your application, and you should not encourage them to take this action. There is no system-provided icon or control that supports this action, and it’ s recommended that you avoid creating a custom icon or control that does. If you decide you must provide settings in your iPhone application, see “The Settings Bundle” in iPhone Application Programming Guide to learn how to support them in your code.
用户不可能在先退出你的应用程序之前打开“设置”应用程序，而且你不应该鼓励他们采取这一行动。系统不会提供任何图标或控件来支持这一行为，并建议你不要创建这样一个自定义图标或控件。如果您决定一定要在您的iPhone应用程序提供这种设置，请参阅iPhone Application Programming Guide中的“The Settings Bundle”，以了解如何用您的代码来实现。

Note: Application-specific settings should not include user help content.
特定的应用程序设置不应该包括用户帮助项。

Unlike settings, configuration options are likely to be changed frequently as users choose to see information from new sources or in different arrangements. You can react dynamically to changes users make to these options, because users do not leave your application to access them. You can offer configuration options in the main user interface or on the back of a screen.
不同于设置，布局选项有可能经常被改变，因为用户会选择看一些新资源中的信息或进行不同的安排。您可以对用户对这些选项所作的改变作出动态反应，因为用户不必离开您的应用程序去访问它们。你可以在用户主界面或在返回界面时提供布局选项。

To decide which technique makes sense, determine if the options represent primary functionality and how often users might want to set them. For example, Calendar allows users to view their schedules by day, week, or month. These options could have been offered on the back of the screen, but viewing different parts of a calendar is primary functionality and users are likely to change their focus frequently.

为了确定哪种技术比较合理，应该先确定这些选项是否代表了基本功能，以及用户可能希望过多久才设置它们。例如，“日历”允许用户查看一天、一周或一个月的时间表。程序后台提供有这些选项，但查看日历的不同部分是基本功能，用户可能会经常改变他们关注的资讯。

On the other hand, the primary functionality of Weather is to display a city’s current conditions and 6-day forecast. Although it’s important to be able to choose whether temperatures are displayed in Celsius or Fahrenheit, users are not likely to change this option very often, so it would not make sense to put it in the main user interface. Offering the temperature-scale option on the back of the Weather screen makes it conveniently available, but not obtrusive.
另一方面，”天气”的主要功能是显示一个城市的现况和6天内的天气预报。虽然选择气温是用摄氏或华氏来显示是很重要的，但用户不太可能经常改变这个选项，所以把它放在用户主界面是没有意义的。 把“天气“项放在后背页面能让人们使用更加方便，而不会显得碍眼。

Supporting Copy and Paste 支持复制与粘贴

iPhone OS provides an edit (or pasteboard) menu that supports Cut, Copy, Paste, Select, and Select All operations in text views, web views, and image views. One of the ways users can reveal the menu is to first touch and hold to display the magnified view (which allows them to move the insertion or selection point to the correct location) and then release.
iPhone OS提供一个编辑（或剪贴板）菜单，支持剪切，复制，粘贴，选择，全选文本，网页预览，以及图片预览。用户展现菜单的方法之一是：首先触摸以显示放大的视图（这使他们能够移动插入点或选择点至正确的位置），然后释放。

If the menu is supported in the current context, it appears when the finger is lifted. Choosing Select in the menu selects the word or application-defined item in the view.
如果菜单是在当前情况下被支持，那当手指从屏幕提起后它就会出现。选择菜单中的“选择”项来选择单词或应用程序给出的项目。

Users can extend their selection by dragging the handles on the currently selected area. After a selection is made, the menu can display Cut, Copy, or Paste, as appropriate.
用户可以在当前选择区域通过拖动来延伸选择区域。当作出一项选择后，菜单就可以相应地显示“剪切”，“复制”或“粘贴”。

You can adjust some of the behaviors of the edit menu to suit your application. (For information on how to implement these behaviors in code, see “Copy and Paste Operations” in iPhone Application Programming Guide.) For example, you can specify the subset of commands the menu displays and you can influence where the menu appears. You have no control over the color or shape of the menu itself.

您可以调整编辑菜单的一些动作来满足您的应用程序。(关于如何用代码来实现这些行为，请参阅iPhone Application Programming Guide中的“Copy and Paste Operations”。) 例如，您可以指定菜单显示的命令的子集，你可以控制菜单出现的位置。但是你不能控制菜单本身的颜色或形状。

The commands visible in the edit menu make sense in the current context. For example, if nothing is selected, the menu does not contain Copy or Cut because these commands act on a selection. Similarly, if something is selected, the menu does not contain Select. If you support an edit menu in a custom view, you’ re responsible for making sure that the commands the menu displays are appropriate for the current context. Note that you cannot specify custom commands to display in the menu
这些在编辑菜单中可见的命令在当前情况下是有意义的。举例来说，如果没有内容被选中，那菜单不会包含“复制”或“剪切”选项，因为这些命令要在有内容被选择的情况下才可用。同样的，如果有内容被选中，菜单就不包含“选择”选项。如果程序支持在自定义视图中编辑菜单，那应确保菜单中的命令在当前情况下显示是合适的。请注意您的程序不能指定自定义命令显示在菜单中。

UIKit displays the edit menu above or below the insertion point or selection, depending on available space, and places the menu pointer so that users can see how the menu commands relate to the content. You can programmatically determine the position of the menu before it appears, so you can prevent important parts of your application’ s user interface from being obscured, if necessary.
UIKit根据可用空间在插入点或被选择点的上方或者下方显示编辑菜单，并且安放菜单指针，以便用户可以看到该菜单命令涉及到的内容。如果必要的话，您可以通过编程确定菜单在出现之前所在的位置，这样就可以防止用户应用程序界面的重要部分被遮住。

Note that although the touch and hold gesture is the primary way users reveal the edit menu, they can also double-tap a word in a text view to select it and reveal the menu at the same time. If you support the menu in a custom view, you should respond to both gestures. In addition, you can define the object that is selected by default when the user double taps.
请注意，虽然触摸和手势是用户在编辑菜单上进行操作的主要方式，但他们也可以通过双击文本中的单词来选择它，并同时显示菜单。如果程序支持自定义视图的菜单，那应该对这些手势都设置好反馈。另外当用户双击时，程序可以事先定义默认的被选对象。

Avoid creating a button that performs a command that’ s available in the edit menu. For example, it’ s better to allow users to perform a copy operation using the edit menu than to provide a Copy button, because users will wonder why there are two ways to do the same thing in your application.
避免创建一个编辑菜单中已有的命令按钮。例如，当用户执行复制操作时，直接在编辑菜单中操作比提供一个“复制”按钮要好得多，因为用户会想为什么在同一程序中要有两种方法做同一件事。

You can enable selection of static text, but only when the static text represents content that's useful to the user. For example, a user might want to copy the caption of an image, but they're not likely to want to copy the label of a tab item or a screen title, such as Accounts. In a text view, selection by word should be the default.
程序可以启用静态文本，但要在静态文本的内容对用户是有用的前提下。例如，用户可能会想要复制某一图像的标题，但不太可能要复制一个标签或者一个屏幕上的标题，例如“帐户”。在一个文本视图中，应该默认以文字来选择。

Button titles should not be selectable because it would be difficult for users to reveal the edit menu without activating the button. In general, elements that behave as buttons don't need to be selectable. If you support the Cut, Copy, and Paste commands in your application, you should also support undo and redo (described in “Supporting Undo and Redo” (page 48)). This is because the edit menu does not require confirmation before the actions are performed and users often expect to be able to undo recent operations if they change their minds.
按钮的标题应不能被选择，因为用户必须先激活按钮才能显示编辑菜单。一般来说，像按钮这类元素是不需要被选择的。如果程序支持“剪切”，“复制”和“粘贴”命令，那也应该支持“撤销”和“重做”（如“Supporting Undo and Redo”所述）。这是因为在命令被执行之前，编辑菜单并不能确认用户会执行什么动作，而当用户改变主意时，往往希望能够撤销刚执行过的动作。

Supporting Undo and Redo 支持撤销和重做

iPhone OS gives users the ability to undo and redo their typing in text views. Users initiate an undo by shaking the device, which displays an alert that allows users to undo what they just typed, redo previously undone typing, or cancel the undo.
iPhone操作系统使用户在文本输入时撤销和重做成为可能。用户通过晃动设备来实现撤销命令，此时会显示一个提醒，使用户可以撤销刚刚打的字、重做刚刚打的字或是取消撤销。

UIKit allows you to support undo in a more general way in your application (for information on how to implement this behavior in code, see Undo Architecture). You can specify:
UIKit程序允许你以一个更简单的方式执行撤销（就如何通过编码落实这一行为，请参阅Undo Architecture。可以指定：

■ The actions users can undo or redo
可以撤销或重做的动作
■ When your application should interpret a shake event as the shake to undo gesture
程序什么时候应识别出作为撤销的晃动手势
■ How many levels of undo to support
能够支持多少层次的撤销

To provide a great user experience for the undo and redo capability in your application, you should:
为了能在应用程序中提供一个出色的撤销和重做的用户体验，应该：

■ Supply brief descriptive phrases that tell users precisely what they’ re undoing or redoing. UIKit automatically supplies the strings “Undo ” and “Redo ” for the undo alert button titles, but you need to provide a word or two that describes the action users can undo or redo. (Note that the Cancel button cannot be changed.) For example, you might supply the text “Delete Name” or “Address Change,” to create buttons titles such as “Undo Delete Name” or “Redo Address Change.”
提供简短的描述性用语告诉用户想要执行的撤销或重做命令是什么。 UIKit 自动提供了“撤销”和“重做”作为撤销按钮的标题，但程序需要提供一两个词来描述用户可以执行的撤销或重做命令。（请注意，“取消”按钮是无法被更改的。）例如，程序可以提供“删除名称”或“改变位置”等文本，以创建诸如“撤销删除名称”或“重做改变位置“等按钮的标题。

Be sure to avoid supplying text that is too long: A button title that is too long is truncated and is difficult for users to decipher. Also, because this text is in a button title, use title-style capitalization and do not add punctuation. (Briefly, title-style capitalization means to capitalize every word except articles, coordinating conjunctions, and prepositions of four or fewer letters.)
一定要避免提供的文本太长：一个标题太长的按钮会被截断并让用户很难理解。此外，因为这是在一个按钮上的标题，最好使用标题字母大写规则，而且不要加标点。（简单地说，标题字母大写规则就是指大写除正文以外的每个单词，协调连词和四个或四个以下的介词。）

■ Avoid overloading the shake gesture. Even though you can programmatically set when your application interprets a shake event as shake to undo, you run the risk of confusing users if they also use shake to perform a different action.
避免超载的晃动手势。虽然可以通过编程来设置何时激活应用程序的晃动手势事件，如晃动表示撤销，但如果他们也用晃动来执行不同的操作，那你就有使用户迷惑的风险了。

The shake gesture is the primary way users expect to initiate undo and redo, but you can also include the system-provided Undo and Redo buttons in a navigation bar, if appropriate. You might do this if it’ s essential that you display an explicit, dedicated button to perform these functions within the context of your application, but this is unusual.
晃动的手势是用户期望实现撤销和重做的主要方式，但如果适当的话，你也可以将系统提供的撤销和重做按钮放在导航条。如果是必要的话，程序可以显示一个明确的、专用的按钮来执行应用程序里的这些功能，但注意只有必要时才这样。

■ Consider the context of the actions you allow to be undone or redone. In general, users expect their changes and actions to take effect immediately. As much as possible, the undo and

redo capability should be clearly related to the user’ s immediate context, and not to an earlier context.
考虑程序允许的被撤销或重做的动作的范围。一般来说，用户希望他们的变动和行为能立即生效。撤销和重做功能应尽可能明确地与用户当前内容相关，而不是与较早的内容相关。

Enabling Push Notifications 启用推送通告

When you register your application with Apple Push Notification Service, you can arrange to alert users when new data arrives, even if your application isn’t running. When a device receives a message for an application that isn’ t running, it can notify the user by:
当注册时程序带有推送通告服务，那么新数据到达时，程序就可以安排通知提醒用户，即使程序不运行也能提醒用户。当设备接收到某一信息而该应用程序没有运行时，它就可以通知用户：

■ Updating a badge on the application’ s Home screen icon
在主界面应用程序的图标处更新标记
■ Playing an alert sound
播放提醒声
■ Displaying an alert message
显示一个提醒消息

or some combination of these. The user might respond by starting the application to manage the new data or merely acknowledging its arrival. (To learn how to handle push notifications in code, see Apple Push Notification Service Programming Guide.)
或这些的组合。用户可能会作出反应，启动应用程序来管理新的数据，或者仅仅是知道了。（要了解如何在代码中处理推送通告，见Apple Push Notification Service Programming Guide）

Note: Delivery of push notifications is not guaranteed. Also, users can refuse to receive notifications system-wide. Push notifications are intended to notify users of new data, not to deliver critical data to your application.
注：推送通告的发送是没有保证的。此外，用户可以拒绝接收全系统通告。推送通告的目的是通知用户新数据，而不是向你的应用程序提供关键数据。

The Notifications section in the built-in Settings application hosts push notification settings for every application that registers for the Apple Push Notification Service. For each application, iPhone OS provides settings for users to allow or disallow badging, sounds, and alert messages.
内置的设置” 应用程序“中的通告部分承担着为注册了Apple Push Notification Service的用户的每个应用程序推送通告的任务。对于每个应用程序， iPhone操作系统为用户提供了允许或禁止标记、声音和提醒信息的设置。

Take some time to think about the types of events for which users would appreciate notification. A notification should give users useful, actionable information they want to receive, even when they’ re not using your application.
花一些时间来思考用户希望看到哪种类型的通告。通知中应给用户一个有用的、可操作的、他们想要的信息，即便他们没有使用你的应用程序。

After you’ ve identified the events your users are likely to care about, you should also allow users to decide which events should generate what type of notification, if any. If you don’ t allow users to customize the push notification experience in your application, you’ re liable to pester users with notifications they’ re not interested in.Users can choose which types of notifications they want to receive, so you should support all three types:
在确认了您的用户很可能会关心的事件后，你应该让用户来决定哪些事件应以何种类型来被通知，如果有的话。如果不允许用户自定义应用程序中的推送通知方式，那你就要为因为他们不感兴趣的通告而与他们纠缠不清负责了。用户可以选择他们想要接收的通告类型，所以你应该支持所有三种类型：

■ Badge. Badging is the least intrusive way to tell users that there is new content they might be interested in. The badge is a small red oval that appears over the upper-right corner of your Home screen icon. You do not have any control over the appearance of the badge and it contains only numbers, not letters or punctuation.
标徽。标徽是用来告诉用户有他们可能感兴趣的新内容而最不侵扰的方式了。标徽是一个出现在您的主界面应用程序图标右上角的红色小椭圆。您不能改变标徽的外观，它只包含数字，而不是字母或标点符号。

Badging works well when you want to tell users how many items are waiting for their attention. For example, the number in a badge might indicate unread messages, newly assigned tasks, or how many remote players are currently playing a game.
当你想告诉用户有多少项目等待他们处理时，标徽是很有效的。例如，标徽中显示的数可能暗示了未读信息、新分配的任务、或者远程有多少玩家目前在玩游戏。

■ Sound. You can supply a custom alert sound, or you can use a built-in alert sound. If you create a custom sound, be sure it is short, distinctive, and professionally produced. (To learn about the technical requirements for this sound, see “Preparing Custom Alert Sounds” in Apple Push Notification Service Programming Guide.) Note that you cannot force the device to vibrate when a notification is delivered; the user has control over whether alerts are accompanied by vibration.
声音。你可以提供一个自定义提醒声，或者你可以使用内置的提醒声。如果要创建一个自定义的声音，请确保它是短暂的、独特的、专业的。（要了解有关提示音的技术要求，请参阅Apple Push Notification Service Programming Guide中的“Preparing Custom Alert Sounds”）请注意，你不能在通告被发送时强行振动设备;用户自己控制接收提醒时是否附有振动。

An easily recognized sound is ideal for situations in which the notification arrival itself provides enough information for users to act. For example, a collaborative task management system might use a unique sound to accompany the completion of a member’ s assignment. Merely hearing this sound tells the user that the assignment has been completed.
一个容易辨别的声音是最理想的，最好是在通知到达时其本身就能为用户提供足够的信息以采取行动。例如，一个协同任务管理系统可能会使用一个独特的声音来伴随一项任务完成。只要听到这个声音就可以告诉该用户任务已完成。

■ Alert. An alert is the most intrusive way to notify users of new content. An alert displays your application name at the top, your message below that, and one or two buttons at the bottom. If you specify two buttons, the alert displays the Close button on the left and the View button on the right (users can tap the View button to simultaneously dismiss the alert and launch your application). If you specify one button, the alert displays an OK button. The Close button and the OK button both dismiss the alert without opening your application.
提醒。提醒是通知用户新内容的最具侵入性的方式。提醒中您的应用程序名称呈现在顶部，消息处于其下方，并有一两个按钮在底部。如果你指定两个按钮，则左侧按钮显示“关闭”，右侧按钮显示“查看”（用户可以点击“查看”按钮，同时关闭提醒，并启动您的应用程序）。如果你指定一个按钮，则提醒显示“确定”按钮。“关闭”按钮和“确定”按钮都会不打开应用程序就关闭该提醒。

Alerts interrupt the user’ s workflow, so they’ re best used sparingly to deliver a short, important message about an event. In particular, be sure to avoid including any advertising content in your alert message.
提醒会中断用户的工作流程，因此提醒应最好被用来偶尔发送简短且重要的事件信息。特别注意，一定要避免任何广告内容出现在提醒信息中。

Making Your Application Accessible 使程序具无障碍性

An application is accessible when users with disabilities can use it successfully, perhaps with the help of an assistive application or device. iPhone OS–based devices include many features that make it easier for all users, including disabled users, to use the device, such as visual voicemail, zoom, and voice control. You do not have to take any steps in your application to ensure that your users can benefit from these features.
当使用者包括残疾人都能借助一些辅助应用程序或设备，成功使用该应用程序时，程序将是易于使用的。基于iPhone操作系统的装置有很多使所有用户更简单地进行操作的特征。包括残疾人，使用者可以使用像可视语音信箱、变焦和声控的设备。你不必在应用程序中采取任何步骤来确保用户可以受益于这些功能。

With VoiceOver, the story is a little different. VoiceOver is Apple’ s innovative screenreading technology, which gives users control over their devices without requiring them to see the

screen. To make sure VoiceOver users can use your application to its fullest, you might need to provide some custom information about the views and controls in the user interface.
有了VoiceOver，故事就有点不一样了。 VoiceOver 是苹果首创的屏幕阅读技术，使用户们不用看屏幕也能控制他们的设备。为了确保VoiceOver的用户能充分地使用应用程序，你可能需要在用户界面提供一些关于视图和控件的自定义信息。

Fortunately, UIKit controls and views are accessible by default, so when you use standard elements in a completely standard way, you have little (if any) additional work to do. The more custom your user interface is, the more custom information you need to provide, so that VoiceOver can properly describe your application to users with visual impairments.
幸运的是，UIKit 视图和控件在默认下是被接受的，所以当你以一个完全标准的方式使用标准元素时，你不用做额外的工作（如果有的话也很少）。你的用户界面自定义项越多，你需要提供的自定义信息就越多，以便VoiceOver 向有视觉障碍的用户合理地描述应用程序。

Important: The job of making your application accessible consists of giving VoiceOver the information it needs to help people use your application. The job does not include changing the visual design of the user interface to accommodate VoiceOver.
注意：使程序无障碍性的工作包括提供VoiceOver所需要的信息，这样它可以帮助人们使用应用程序。这个工作不包括改变用户界面的视觉设计以适应VoiceOver。

Making your iPhone application accessible to VoiceOver users is the right thing to do. It can also increase your user base and it might help you address accessibility guidelines created by various governing bodies.
使iPhone程序对VoiceOver的用户提供无障碍支持是需要做的事。它也可以增加你的用户群，同时能帮助你应付由各种各样理事机构制定的无障碍指导方针。

Providing Search and Displaying Search Results
提供搜索并显示结果

UIKit provides the search bar control you can use to display a consistent interface to initiate searching, but you are responsible for implementing search in your application. (To learn more about the search bar, see “Search Bars” (page 116); to learn more about handling search results in code, see UISearchDisplayController Class Reference.) To ensure that search is a useful and convenient experience users appreciate, take some time to consider how to implement the process and how to display the results. In general, you should:
UIKit提供了一个搜索条控件，你可以用这个控件来显示连续的界面并开始搜索，但是您要对在您的应用程序里实施的搜索负责。（若要了解更多关于搜索条的信息，请参阅“Search Bars”；了解更多关于如何

用代码处理搜索结果，请参阅UISearchDisplayController Class Reference）。若想使搜索有效而方便并为用户所欣赏，请花一些时间思考如何实施搜索过程和如何展现搜索结果。总之，你应该：

- Build indexes of your data so you are always prepared for search.

为你的数据建立索引，以便可以随时搜索。

- Live-filter local data so you can display results as soon as users begin to type, and narrow the results as users continue typing.

即时过滤本地数据，以便当用户一开始输入时你可以立刻将结果呈现给他们，当用户继续输入时可以缩小结果的范围。

- When possible, also filter remote data as users type, but be sure to get the user's permission if the response time is likely to delay the results by more than a second or two.

如果可能，你还要过滤掉用户输入时关联不大的信息，但是要确定得到用户的许可，这也许会使回应的时间超过好几秒而延迟结果。

- Display a search bar above a list or the index in a list.

呈现的搜索条要建立在列表或是列表中索引的基础上。

- Avoid using a tab for search unless it is a primary function in your application that should be featured as a distinct mode.

避免使用标签搜索，除非它是你应用程序中的具有独特模式的主要功能。

Although live-filtering data usually produces a superior user experience, it's not always practical. When this is the case, you can begin the search process after the user taps the Search button in the keyboard. If you do this, be sure to provide feedback on the search's progress so users know that the process has not stalled. One way to do this is to display textual results as soon as possible and display placeholder content for data that might take longer to retrieve.

虽然现场过滤数据通常能产生一种优越的用户体验，但它并不总是实用。比如，当用户按下键盘上的“搜索”按钮后你才能开始搜索。如果你这样做，请确定在搜索过程中提供一个反馈信息，以使用户了解到搜索并没有停止。操作这个的其中一个方法是尽快显示文字结果，及可能要更长时间来显示的数据占位符内容。

In YouTube, for example, users initiate a search for videos by tapping the Search button. If the network connection is slow, YouTube first displays the Loading... message along with a spinning activity indicator so users know that search is proceeding. Then, YouTube displays a results list in which each row is populated with textual results, such as video title and viewer rating, and a custom image of a box with a dotted outline. As users scan the list of video titles, the video thumbnails replace the dotted boxes as they are downloaded. Displaying partial search results while additional data is still downloading gives users useful information promptly.

比如在Youtube上，用户通过按下“搜索”键开始搜索视频，如果网速很慢，Youtube先提供“正在下载...”这一信息并旋转活动指示灯来让用户知道搜索正在进行。然后Youtube呈现一个搜索结果列表，其中每一列都由文字结果填充，例如视频名称和观看率，还有一围绕虚线框的自定义图像。当用户浏览

视频名称列表开始下载时，视频的缩略图会代替虚线框。当更多的数据正在下载时，显示部分搜索结果，给用户提供及时有用的信息。

If you handle data that sorts naturally into different categories, you can provide a scope bar. A scope bar contains up to four scope buttons, each representing a category. For example, Mail provides a scope bar that allows users to focus their search on the From, To, or Subject fields of messages, or broaden the search to include all fields. Consider providing a scope bar if it helps users focus their search or if it significantly reduces the number of results. (To learn how to implement a scope bar in your code, see UISearchBar Class Reference.)
如果你要处理的数据本身就是属于多个分类的，你可以提供一个分类条。这个分类条可包括最多四个的分类，每一个代表一种分类。比如，Mail下的搜索就提供了一个分类条，使用户的搜索专注于“发件人”，“收件人”，或者邮件主题，或者扩大搜索使之能包含所有领域。如果分类条能有助于用户专注于他们的搜索或者它很有效地减少了结果的数量，那么便考虑提供一个分类条。（要了解如何在代码中实现分类条，见UISearchBar Class Reference）

Using the User’s Location 利用用户地理坐标

Users appreciate application features that allow them to automatically tag content with their physical location, or to find friends that are currently nearby. Users also appreciate being able to disable features like these when they don’t want to share their location with others. Users can grant (or deny) systemwide access to their physical location with the Location Services setting in Settings > General.
用户偏好应用程序提供自动标记他们地理位置的功能，或帮助找到正在附近朋友的功能。同样也偏好能够在他们不想与他人分享地理坐标时禁用这一功能。用户可以在“设定” / “通用”中设定”定位服务“授予（或拒绝）全系统调用他们的地理坐标。

If users turn off Location Services, and later use an application feature that requires their location, they see an alert that tells them they must change their preference before they can use the feature. The alert does not allow them to make this change within the application; instead, they must go to Settings and change their preference. This ensures that users are fully aware that they are granting system-wide permission to use their location information.
如果用户关闭”定位服务“，但后来某应用程序需获取需要他们的位置，那么将看到一个提醒，告诉使用者必须打开“定位服务”，才可以使用该功能。提醒不允许他们在此应用程序内做改变，相反，他们必须去”设置“程序以更改。这确保了用户能充分认识到，他们给予全系统许可才能调用他们的位置信息。

To help users understand why they might need to turn Location Services on, it’s best if they see the alert only when they attempt to use a feature that clearly needs to know their current

location. For example, people can use Maps when Location Services is off, but they see the alert when they access the feature that finds and tracks their current location.
为了帮助用户了解为什么他们可能需要打开”定位服务“，最好只在他们显然需要知道他们的位置而尝试来使用时才让他们看到提醒。例如，当”定位服务“是关闭时，人们可以使用地图，但当他们访问定位功能时，他们将看到提醒。

If Location Services is turned off, iPhone OS displays the alert the first time your application tries to access location information. The Core Location framework provides a way for you to get the user’s preference so that you can avoid triggering this alert unnecessarily or inappropriately. (See Core Location Framework Reference to learn more about this programming interface.)
如果定位服务是关闭的，iPhone OS会显示你的应用程序首次试图访问的定位服务信息。Core Location framework（核心地理坐标韧体）提供了一种方法来获取用户的偏好，避免引发不必要的或不适当的提醒。（想学习该界面代码的更多信息，见Core Location Framework Reference。）

With knowledge of the user’s preference, you can trigger the alert as closely as possible to the feature that requires location information, or perhaps avoid it altogether.
获悉用户的偏好后，你可以触发需要位置信息功能的密切提醒，或者完全避免提醒。

■ If your application cannot perform its primary function without this information, it’s best if users see the alert as soon as they start your application. Users will not be bothered by this, because they understand that the main function of your application depends on knowing their location.
假如你的应用程序没有定位服务就无法执行主要程序功能，那在程序启动时，尽快让用户看到提醒。用户将不会感到困惑，因为他们明白应用程序的主要功能依赖“定位服务”的信息。

■ If the user’s location is not part of the essential function of your application, you might choose to simply restrict the feature that uses it. For example, when Location Services is turned off, Camera automatically turns off the feature that adds the user’s location to the photos they take. It does not prevent users from taking photos unless they change their preference, because adding location information to photos is appreciated, but not essential.
如果用户的地理坐标不是应用程序基本功能的一部分，您可以选择限制它的使用。例如，当位置服务是关闭的，相机会自动关闭把用户的地理坐标添加到照片。它不会阻止用户拍照，除非使用者改变了设置，因为照片加入位置信息是附加的，但是并非必要。

■ If a feature needs location information to function, be sure to avoid making any programmatic calls that trigger the alert before the user actually selects the feature. (The call that gets the user’s preference does not trigger the alert.) This way, you avoid causing users to wonder why your application wants their location information when they’re doing something that doesn’t appear to need it.

如果一个功能需要先获取位置信息，则一定要避免在使用者实际选择该功能前产生任何预定请求来触发提醒。（获得用户偏爱的请求不会触发提醒。）这样，可避免用户产生疑惑，为什么你的应用程序想得到他们的位置信息，而他们正在做的一些事情，似乎并没有这个必要。

Handling Orientation Changes 处理方向的变化

Users can rotate iPhone OS–based devices at any time, and they expect the content they're viewing to respond appropriately. In your iPhone application, be sure to:
用户可能在任何时候旋转iPhone 操作系统的设备，并期待浏览对象能够相应变化调整。在iPhone应用程序中，请务必注意：

■ Be aware of accelerometer values (for more information on the accelerometer and references to accelerometer programming interfaces, see iPhone Application Programming Guide). If appropriate, your application should respond to all changes in device orientation.
注意加速度的读数值（欲知更多的关于加速度和加速度计的界面编程信息，见iPhone Application Programming Guide）。如果使用恰当，你的应用程序会根据相应的设备位置变化而作出回应。

■ If there's a part of your application's user interface that displays in one orientation only, it's appropriate for that area to appear in that orientation and not respond to changes in device orientation. For example, when a user selects an iPod video to view, the video displays in landscape orientation, regardless of the current device orientation. This signals the user to physically rotate the device to view the video. The important point about this example is that iPod does not provide a "rotate now" button; instead, the user knows to rotate the device because the video appears in landscape orientation.
如果应用程序的用户界面有一部分只显示一个方向，这就是它所适合显示的方向，而不会根据设备方向的改变而发生变化。例如，当用户选择了浏览一个iPod视频，视频是横向显示，而不管当前设备的方向。这就暗示使用者应该旋转设备的物理方向来看视频。这个例子的重点是iPod不提供旋转按钮，而使用者知道根据横向模式的出现来旋转设备。

Allow users to physically rotate the device to correctly view the parts of your application's user interface that require a specific orientation. Avoid creating a control or defining a gesture that tells users to rotate the device.
允许用户通过旋转改变设备的物理方向，来看到应用程序中需要特定方向的用户界面。避免创建一个控件或定义一个手势动作来告诉用户旋转设备。

■ Take advantage of the one-step orientation-change process to perform smoother, often faster rotations. However, if your screen layout is very complicated, you might choose instead

to perform a cross-fade transition when an orientation-change occurs. To learn how to support the one-step process in your code, see UIViewController Class Reference.
充分利用一步旋转操作来改变方向状态，使之更加顺畅，快速。不过，如果你的屏幕布局是非常复杂的，那么当需要改变方向时用选择，而不是用默认的淡入淡出效果。要了解一步操作的代码支持，见UIViewController Class Reference。

■ Users often rotate their devices to landscape orientation because they want to "see more." If you respond by merely scaling up your content, you fail to meet users' expectations. Instead, you should respond by rewrapping lines of text and, if necessary, rearranging the layout of the user interface so that more content fits on the screen.
用户经常旋转他们的设备至横向模式，因为他们想"看到更多"。如果你只是扩大您的内容，是不能满足用户的期望的。相反，你应该安排好文字，如果有必要，重新安排用户界面以便内容布局更适合屏幕。

Using Sound 使用音效

Users expect great sound from iPhone OS–based devices, whether they're hearing system sounds, such as ringtones and alert sounds, or application sounds, such as media playback, ambient sounds, and soundtracks. In addition, users expect sounds from their devices to obey both their preferences and their intentions.
用户期望基于iPhone 操作系统的设备有非常棒的音效，无论他们听到的是系统音，如铃声和提醒声，或应用程序声音，如媒体播放的声音，情境声音，以及音轨。此外，用户希望他们的设备音效都服从于自己的喜好和意图。

Users decide how loud sounds should be and whether they want to hear them at all. Sometimes, however, users expect to hear certain sounds even when their current settings indicate that they prefer silence. For example, users always expect to hear alarms that they have set. Essentially, users want to hear sounds they ask for, but avoid hearing sounds they don't ask for.
用户决定声音应该有多大及他们是否要听到它们。但是，有时甚至当他们的当前设置是静音时，用户还是希望听到某些声音。例如，用户总是希望听到他们已设置的闹铃声。从本质上讲，用户想听到他们想要的声音，但要避免听到他们不想听到的声音。

To help you accommodate this, iPhone OS provides programming interfaces you can use to:
为了帮助程序满足这一点，iPhone操作系统提供了编程界面，你可以用来：

■ Describe how your application's sounds should fit in with other sounds on the device
描述你的应用程序的声音应该如何与设备上的其他声音协调
■ Ensure that your application's sounds play according to users' expectations
确保应用程序的声音根据用户的期望而播放

Before you decide how to handle sound in your application, you need to understand how users expect applications and the device to behave when they adjust device controls and use external devices, such as headphones and headsets.
在你决定如何处理应用程序的声音前，你需要了解当用户调节设备和使用外接设备，如耳麦和耳机时，他们希望怎样应用程序和设备怎样来运作。

The Ring/Silent Switch—What Users Expect
响铃还是静音--用户期望什么

Users use the Ring/Silent switch to silence their devices when they want to:
用户使用”响铃/静音“切换开关时，他们希望：

- Avoid being interrupted by unexpected sounds, such as Phone ringtones and incoming message sounds.
避免被意想不到的声音，如电话铃声和消息提示音打断。
- Avoid hearing sounds that are the byproducts of user actions, such as keyboard or other feedback sounds, incidental sounds, or application startup sounds.
避免听到用户操作时附带的声音，如键盘或其他反馈的声音，偶然声音，或应用程序启动的声音。
- Avoid hearing game sounds, including incidental sounds and soundtracks, that are not essential to using the game.
避免听到游戏声音，包括附带的声音和配乐，这在使用游戏时是没有必要的。

For example, in a theater users switch their devices to silent to avoid bothering other people in the theater. In this situation, users still want to be able to use applications on their devices, but they don’ t want to be surprised by sounds they don’ t expect or explicitly request, such as ringtones or new message sounds.
例如，在一个剧场时，用户切换设备至静音，以避免打扰剧场的其他人。在这种情况下，用户仍然希望能够使用他们设备的应用程序，但他们并不想被他们不期望或奇怪的声音，如铃声或新信息提示音干扰。

However, the Ring/Silent switch does not silence sounds that result from user actions that are solely and explicitly intended to produce sound. For example:
然而，”响铃/静音“开关不能避免由用户明确意图的行动所产生的声音。例如：

- Media playback in a media-only application is not silenced by the Ring/Silent switch because the media playback was explicitly requested by the user.
在媒体播放应用程序中不能使用”响铃/静音“开关，因为媒体播放是明确的用户请求。
- A Clock alarm is not silenced by the Ring/Silent switch because the alarm was explicitly set by the user.
一个闹钟铃声不能使用”响铃/静音“开关，因为它是由用户明确设置的。

■ A sound clip in a language-learning application is not silenced by the Ring/Silent switch because the user took explicit action to hear it.
语音学习的应用程序不能使用”响铃/静音“开关，因为用户采取明确的行动以听到这种声音。
■ Conversation in an audio chat application is not silenced by the Ring/Silent switch because the user started such an application for the sole purpose of having an audio chat.
在一个语音聊天应用程序中的对话不能被”响铃/静音“开关设置，因为用户开始这个应用程序，音频聊天是唯一目的。

This behavior follows the principle of user control because it is up to the user, not the device, to decide whether it's appropriate to hear sounds the user explicitly requests.
这种行为遵循用户控制原则，因为它是由用户，而不是设备，来决定是否在适当时听到用户明确要求的声音。

Volume Buttons—What Users Expect
音量调节键--用户期望什么

Users use the device’s volume buttons to adjust the volume of all sounds their devices can play, including songs, application sounds, and device sounds. This means that users can always use the volume buttons to quiet any sound, regardless of the position of the Ring/Silent switch.
用户使用设备的音量调节键来调整他们设备可以播放的声音，包括歌曲、应用程序声效和设备的声效。这意味着用户可以随时使用音量调节键来使任何声音静音，而不用“响铃/静音”开关。

In some cases, it might be appropriate for an application to give users volume-setting capability in the application’s user interface. For example, YouTube displays a volume slider users can use to adjust the volume of the video they’re watching. While YouTube is running, users can use the slider and the volume buttons interchangeably to affect the video’s volume. This is because the slider acts as a proxy for the volume buttons while the application is running: The slider affects both the application’s volume and overall system volume, with the exception of the ringer volume.
在某些情况下，在应用程序的用户界面中也可以安置音量调节开关。例如，YouTube呈现一个音量滑杆使用户可以调整他们正在观看的视频音量。尽管YouTube正在播放，用户可以使用音量调节键和滑杆交替着控制视频的音量。这是因为滑杆在应用程序运行时充当音量调节键： 除了特殊的响铃音量，滑杆影响应用程序的音量和整个系统的音量。

If you need to display a volume slider, be sure to use the system-provided volume slider available when you use the MPVolumeView class. Note that when the currently active audio output device does not support volume control (such as an A2DP device), the volume slider is replaced by the appropriate device name.

如果你需要显示一个音量滑杆，一定要使用系统提供的MPVolumeView类库中的音量滑杆。请注意，如目前活跃的音频输出设备不支持音量控制（如A2DP设备），音量滑杆由相应的设备名称取代。

Using the volume buttons to adjust an application’ s currently playing audio also adjusts the overall system volume, with the exception of the ringer volume. (Using the volume buttons when no audio is currently playing adjusts the ringer volume.)
使用音量调节键来调整应用程序当前播放的音频也调整了整个系统的音量，响铃音量例外。（如当前没有音频播放时，则音量调节键可以调整响铃音量。）

This behavior follows the principle of user control because the user always decides how loud sounds from their device should be.
这种行为遵循用户控制原则，因为永远由用户决定他们设备的声音应该多响。

Sometimes, an application might need to adjust relative, independent volume levels to produce the best mix in its audio output. But the volume of the final audio output should always be governed by the system volume, whether it’ s adjusted by the volume buttons or a volume slider. This means that control over the application’ s audio output remains in users’ hands, where it belongs.
有时，应用程序可能需要调整相对的、独立的音量使其音频输出达到最佳混合。但是，最后的音频输出量应该永远是受系统音量限制，无论是由音量调节键或音量滑杆调整。这意味着，应用程序的音频输出仍然受用户控制。

Headsets and Headphones—What Users Expect
耳机与耳麦--用户期望什么

Users plug in headsets and headphones to hear sounds privately and to free their hands. Users have different expectations for application behavior, depending on whether they’ re plugging in or
unplugging these accessories.
用户插入耳机和耳麦可以独享声音并且释放双手。用户对应用程序行为有不同的期盼，这些都取决于他们是否插入这些配件。

When users plug in a headset or headphones, they intend to continue listening to the current audio, but privately. For this reason, they expect an application that is currently playing audio to continue playing without pause.
当用户插入耳机或耳麦，他们打算继续并独自享受当前音频。出于这个原因，他们希望一个不用停顿地播放当前音频的应用程序。

When users unplug a headset or headphones, they don’ t want to automatically share what they’ ve been listening to with others. For this reason, they expect an application that is currently playing audio to pause, allowing them to explicitly restart playback when they’ re ready.
当用户拔出耳机或耳麦时，他们不希望应用程序继续播放并与其他人分享正在听的内容。出于这个原因，他们希望有一个可以暂停当前播放的应用程序，允许他们在明确准备好时可以继续开始。

Wireless Audio—What Users Expect
无线音频--用户期望什么

Users appreciate the convenience of wireless headsets, such as Bluetooth A2DP devices.People use wireless headsets and headphones for the same reasons they use wired headsets and headphones: they want to hear sounds privately and they want to free their hands.
用户欣赏无线耳机设备的便利性，如蓝牙A2DP设备。人们使用无线耳机和耳麦与他们使用有线的理由是一样的：他们想获得私人声音体验，他们希望解放双手。

Users also have very similar expectations for the user experience of wireless headsets:
用户对无线耳麦的用户体验也有非常类似的期望：

■ When users connect to a wireless audio device, they intend to continue listening to the current audio, but privately. In this situation, they expect the audio to continue playing without pause.
当用户连接到一个无线音频设备，他们打算继续听当前音频，但只有其一人享用。在这种情况下，他们希望音频继续播放而不用暂停。

■ When users disconnect from a wireless device (or the device goes out of range or turns off), they don’ t want to automatically share what they’ ve been listening to with others. In this situation, they expect currently playing audio to pause, allowing them to explicitly restart playback when they’ re ready.
当用户断开无线设备（或设备超出范围或关闭），他们不希望自动与他人分享他们在听的音频。在这种情况下，他们期望暂停正在播放的音频，当他们准备好后再明确地重新开始播放。

Even though people don’ t physically plug in or unplug a wireless audio device, they still expect to be able to choose a different audio route. To handle this, iPhone OS automatically displays a control that allows users to pick an output audio route. Because choosing a different audio route is a user-initiated action, users expect currently playing audio to continue without pause.

即使用户不会实际插入或拔出无线音频设备，他们仍然希望能够选择不同的音频通道。为了解决这个问题，iPhone操作系统会自动显示一个允许用户选择音频输出路线的控制。由于选择不同的音频通道是用户发起的动作，用户期望正在播放的声音不会暂停，能继续播放。

Defining the Audio Behavior of Your Application
定义你程序的音频行为

If sound enhances or is essential to the user experience or functionality of your application, you need to decide how your audio should fit in with the audio environment of the device and how it should respond to user actions. For example, you need to decide whether:
如果声音增强了或对于用户体验或应用程序功能的运作是必不可少的，那么你需要决定你的音频应如何配合设备的音频环境，以及应如何响应用户操作。例如，你需要决定是否：

- Your audio should continue playing when the device locks or is switched to silent
当设备锁住或切换到无声时，音频应继续播放
- Your audio should mix with other audio that is currently playing (such as a song in iPod)
音频应该与其他目前播放的音频混合（如iPod的歌曲）
- Your application needs to handle both audio input and output, either sequentially or simultaneously
应用程序需要按顺序还是同时处理音频输入和输出
- Your audio should automatically resume playing after an interruption
音频中断后能自动恢复播放

To influence how your application’s audio should behave in situations such as these, use Audio Session Services or the AVAudioSession class. These programming interfaces do not produce sound; instead, they help you express how your audio should interact with audio on the device and respond to interruptions and changes in device configuration. Audio Session Services governs sound produced using technologies such as the AV Foundation framework, Audio Queue Services, OpenAL, and the I/O audio unit.
Audio Session Services或者 the AVAudioSession类库可以帮助设定应用程序的音频在以上情况下如何作出反应。编写这些界面代码不产生声音，而是帮助你表达你的音频应如何与设备上的音频交互，以及响应中断和设备配置的变化。Audio Session Services管制使用技术产生的声音，如AV Foundation framework、Audio Queue Services、OpenAL, and the I/O audio unit.

Note: If your application produces only user-interface sound effects that are incidental to its functionality, you can use System Sound Services. System Sound Services is the iPhone OS technology that produces alerts and user-interface sounds and invokes vibration; it is unsuitable for any other purpose and the sounds it produces are not governed by Audio Session Services. See the SysSound sample project for an example of using this technology.

注：如果您的应用程序只产生与其功能相匹配的用户界面音效，你可以使用System Sound Services。System Sound Services是产生提醒和用户界面音效和振动调用的iPhone 操作系统技术，它不适用于任何其他目的，它产生的声音不由Audio Session Services管辖。见SysSound示例项目中使用这项技术的例子。

Important: No matter what technology you use to produce audio or how you define its behavior, the phone can always interrupt the currently running application. This is because no application should prevent users from receiving an incoming call.
重点：无论你使用何种技术来制作音频或你如何定义它的行为，来电话可以随时中断当前正在运行的应用程序。这是因为任何应用程序不应该妨碍用户接收到来电。

The audio session is an intermediary for audio between your application and the system. From a user-experience perspective, the most important facet of the audio session is the category that defines the audio behavior of your application.
audio session是应用程序和系统的一个音频中介。从用户体验角度看，audio session最重要的方面是定义应用程序的音频性能。

To provide a great audio user experience, select the category that best describes the audio in your application. Be sure to make your choice based on the semantic meaning of a category, not its precise set of behaviors. This ensures that your application behaves according to users' expectations. In addition, it gives your application the best chance of working properly if the exact set of behaviors is refined in the future.
为了提供一个良好的音频用户体验，选择最能描述应用程序中的音频的类型。一定要在类别语义的基础上做出选择，而不是基于它行为的精确设定。这将确保应用程序的行为迎合用户的期望。此外，如果精确的行为在以后被重新定义，那么它将是使应用程序保持正常工作的最好机会。

In rare cases, you might need to enhance or refine a category' s standard behavior by adding a property to the audio session. For example, you can add the kAudioSessionProperty_OtherMixableAudioShouldDuck property to make sure your application' s audio is louder than all other audio (except phone audio). You might do this if it' s important for the user to hear the audio from your application while other audio is playing. However, you should be aware that a category' s standard behavior represents what most users expect, so you should consider carefully before you add a property to refine that behavior. To learn more about audio session properties, see "Fine-Tuning the Category" in Audio Session Programming Guide.
在极少数情况下，您可能需要通过给audio session增加属性提高或改进类别的标准行为。例如可以增kAudioSessionProperty_OtherMixableAudioShouldDuck属性，以确保您的应用程序的音频是比所有其他音频（电话响铃除外）音频大声。如果让用户听到来自应用程序的音频很必要，而与此同时其他音

频仍要播放，你可以这么做。但是，你应该知道，一个类别的标准行为代表了大多数用户的期望，所以你应该仔细考虑，然后再添加一个属性，以完善这一行为。要了解audio session属性，请参阅Audio Session Programming Guide中的Fine-Tuning the Category。

You can base your category selection on the current audio environment of the device. You might want to do this if, for example, users can use your application while listening to other audio instead of the soundtrack you provide. If this makes sense for your application, be sure to avoid forcing users to stop listening to their music or make an explicit soundtrack choice when your application starts. See Scenario 2 in “Putting it All Together” (page 57) to learn how to do this.
您可以根据当前设备的音频环境选择分类。举个例子，你可能想这样做，用户可以边使用应用程序，边听不是你提供的其他音乐。如果这对您的应用程序很合理，那一定要避免当您的应用程序启动时强迫用户停止听他们的音乐或让他们作出明确的选择。见“Putting it All Together”的“情景2” ，学习如何做到这一点。

It’ s also possible to change categories while your application is running, although this is seldom necessary. The primary reason for doing this is when an application needs to support recording and playback at different times. In such an application, it can be better to switch between the Record category and the Playback category as needed, than to choose the Play and Record category. This is because choosing the Record category ensures that no alerts (such as an incoming text message alert) will sound while the recording is in progress.
当你的应用程序在运行时，它也有可能发生改变，虽然这很少有必要。对于这样做的主要原因是，应用程序需要在不同的时间支持记录和回放。在这样的应用程序中，最好在Record类和Playback类之间按需要切换，而不是选择Play and Record类。这是因为选择Record类将确保没有警示会在记录正在进行时发出声音（如息提示音）。

Table 4-1 lists the audio session categories you can use. iPhone OS assigns the Solo Ambient category to an audio session by default.
表4-1列出了你可以使用的音频会话类别。 iPhone操作系统为默认的audio session分配了Solo Ambient类。
Table 4-1 Audio session categories you can use to define sound behavior in your application
你可以用来定义应用程序中的声音行为的Audio session类

Category 类	Meaning 含义	Silenced by Ring/Silent switch and locking 在静音和锁定状态下静音	Mixes with other audio 和其它音频混合
SoloAmbient	Sounds enhance application functionality, and should silence other audio 增强应用程序功能的声音，并在播放时将其它声音都转为静音	Yes	No
Ambient	Sounds enhance application functionality, but should not silence other audio 增强应用程序功能的声音，在播放时不会将其它声音转为静音	Yes	Yes
Playback	Sounds are essential to application functionality, and might mix with other audio 对于应用程序功能至关重要，并且可以与其它音频混合	No	No (default) 否（默认） Yes (when the MixWithOthers property is added) 是（当添加了 MixWithOthers 属性时）
Record	Audio is user-recorded 用户录音	No	No
PlayAndRecord	Sounds represent audio input and output, sequentially or simultaneously 顺序地或者是同时地音频输入和输出	No	No (default) 否（默认） Yes (when the MixWithOthers property is added) 是（当添加了 MixWithOthers 属性时）
AudioProcessing	Application performs hardware-assisted audio encoding (it does not play or record) 执行硬件辅助的音频编码（不录也不播音）的应用程序	-	No

Note: In the interest of space, Table 4-1 displays only the last part of each category name. The actual symbol name of each category begins with AVAudioSessionCategory. The actual symbol name of the
MixWithOthers property is kAudioSessionProperty_OverrideCategoryMixWithOthers.
注：为了节省空间，表4-1只显示了每个类名称的最后一部分。每个类的实际符号的名称由AVAudioSessionCategory开头。至于实际符号名MixWithOthers
是 kAudioSessionProperty_OverrideCategoryMixWithOthers。

Putting it All Together把所有的都放一起？

Here are some scenarios that illustrate how to choose the audio session category that provides an audio experience users appreciate。
这里有一些场景可以说明怎样选择audio session类来提供一个令人满意的音频用户体验。

Scenario 1. Imagine that you’ re developing an educational application that helps people learn a new language.
情景1 ：假设你正在开发一款教育应用程序来帮助人们学习一种新语言。
You provide feedback sounds that play when users tap specific controls and recordings of words and phrases that play when users want to hear examples of correct pronunciation.
当用户点击特定控件时，需提供用户声音反馈；当用户想听到正确的发音例子时，要提供录制好的单词或词组发音。

In this application, sound is essential to the primary functionality. People use this application to hear words and phrases in the language they’ re learning, so the sound should play even when the Ring/ Silent switch is set to silent or the device locks. Because users need to hear the sounds clearly, they expect other audio they might be playing to be silenced.
在此应用程序中，声音是必不可少的主要功能。人们使用此应用程序来听他们学习的语言中的单词和短语，所以即使当“响铃/静音”开关设置为静音或锁定时，声音应仍然能发挥。由于用户需要听到清楚的声音，他们希望其他可能打开的音频被静音。

To produce the audio experience users expect for this application, you would use the Playback category. Although this category can be refined to allow mixing with other audio (as described in Table 4-1 (page 57)), this application should use the default behavior to ensure that other audio does not compete with the educational content the user has explicitly chosen to hear.
为了产生用户期望的这种应用程序的音频体验，你可以使用Playback类。虽然这个类可以加以重定义，以便允许与其他音频混合（如表4-1介绍），这个应用程序应该使用默认的行为，以确保其他音频没有与用户已明确选择要听到的教育的音频内容冲突。

Scenario 2. Imagine that you’ re developing a game that allows users to guide an onscreen character through many different tasks. You provide various gameplay sound effects and a musical soundtrack.
情景2：假设你正在开发一个游戏，让用户引导游戏角色完成许多不同的任务。您提供各种游戏音效和音乐原声。

In this application, sound greatly enhances the user experience, but is not essential to the main task. Also, users are likely to appreciate being able to play the game silently or while listening to songs in their music library instead of to the game soundtrack.
在此应用程序中，声音大大提高了用户体验，但对于完成主要任务来说不是必需的。此外，用户很可能想要安静地玩游戏，或想边玩边听他们歌曲库的音乐，而不是游戏配乐。

The best strategy is to find out if users are listening to other audio when your application starts. Don’ t ask users to choose whether they want to listen to other audio or listen to your soundtrack. Instead, use the Audio Session Services function AudioSessionGetProperty to query the state of the kAudioSessionProperty_OtherAudioIsPlaying property. Based on the answer to this query, you can choose either the Ambient or Solo Ambient categories (both categories allow users to play the game silently):
最好的策略是查明当应用程序启动时用户是否在听其他音乐。不要询问用户是否选择听其他的音频还是听游戏配乐。相反，使用“Audio Session Services”构架的“AudioSessionGetProperty”来查询“kAudioSessionProperty_OtherAudioIsPlaying”的属性状态 。基于对这个疑问的回答，你可以选择其中的“Ambient”或“Solo Ambient”类（这两类允许用户安静地玩游戏）：

■ If users are listening to other audio, you should assume that they’ d like to continue listening and would not appreciate being forced to listen to the game soundtrack instead. In this situation, you would choose the Ambient category.
如果用户正在收听其他音频，你应该假设他们想继续听，不喜欢被强迫听游戏原声。在这种情况下，您会选择“Ambient”类别。
■ If users are not listening to any other audio when your application starts, choose the Solo Ambient category.
如果应用程序启动时用户没有听任何其他音频，选择“Solo Ambient” 类别。

Scenario 3. Imagine that you’ re developing an application that provides precise, real-time navigation instructions to the user’ s chosen destination. You provide spoken directions for every step of the journey and a few feedback sounds. In addition, you think people would appreciate being able to listen to their own audio while they use your application.
情景3：假设你正在为用户选择目的地而开发一个能提供准确、实时导航指示的应用程序。为旅程的每一步提供语音指示方向并反馈一些声音。此外，应考虑当用户在使用你的应用程序时可能希望能听他们自己的音频。

In this application, the spoken navigation instructions represent the primary task. For this reason, you would use the Playback category, which allows your audio to play when the device is locked or the Ring/Silent switch is set to silent.
在此应用程序中，语音导航指示代表了主要任务。基于这个原因，可以使用Playback类，它允许你的音频在设备被锁定或“响铃/静音” 开关设置为静音时播放。

To allow people to listen to other audio while they use your application, you can add the kAudioSessionProperty_OverrideCategoryMixWithOthers property. However, you also want to make sure that users can hear the spoken instructions above the audio they’ re currently playing. To do this,
you can apply the kAudioSessionProperty_OtherMixableAudioShouldDuck property to the audio session. This ensures that your audio is louder than all currently playing audio (except phone audio).
为允许人们在使用你的应用程序时能听到其他音频，你可以添加“kAudioSessionProperty_OverrideCategoryMixWithOthers property”属性。但是，您还需要确保用户在他们目前正在播放的音频上还可以听到以上的语音指示。要做到这一点， 可以应用“kAudioSessionProperty_OtherMixableAudioShouldDuck”属性到audio session。这可以确保您的音频比目前播放的所有音频（除了电话响铃）大声。

Scenario 4. Imagine that you’ re developing a blogging application that allows users to upload their text and graphics to a central website. You might have a short startup sound file, various short sound effects that accompany user actions (such as a sound that plays when a post has been uploaded), and an alert sound that plays when a posting fails.
方案4：假设你正在开发一个博客应用程序，允许用户向中央网站上传他们的文字和图片。你可能有一个短暂的启动声音，各种伴随用户动作的短音效（如当上传完帖子时发出的声音），和帖子上传失败时的提醒声。

In this application, sound enhances the user experience, but it is incidental. The main task has nothing to do with audio and users do not need to hear any sounds to successfully use the application. In this scenario, you would use System Sound Services to produce sound. This is because the audio context of all sound in the application conforms to the intended purpose of this technology, which is to produce user interface sound effects and alert sounds that obey device locking and the Ring/Silent switch as users expect.
在此应用程序中，声音提高了用户体验，但它是次要的。主要任务与音频无关，而且用户不需要听到任何声音也能成功使用该应用程序。在这种情况下， 你可以使用System Sound Services来产生音效。这是因为应用程序中所有声音的音频内容都符合该技术的预期目的，是要产生符合用户期望的装置锁定和“响铃/静音”转换的用户界面音效和提醒音效。

Providing Choices 提供选择

iPhone OS includes a few elements that help users make selections. When you need to offer choices in your application, you should use these selection methods because users are already familiar with their behavior.
iPhone操作系统包括帮助用户作选择的一些元素。当您需要在程序中提供选择时，你应该使用这些选择方法，因为用户已经熟悉它们的行为。

In general, you should not try to replicate the appearance and behavior of selection controls you might see in a desktop computer application, such as an application menu or a set of radio buttons.
在一般情况下，你不应该尝试复制在桌面计算机应用程序下选择的外观和行为控件，如应用程序菜单或单选按钮设置。

iPhone OS provides the following elements you can use to offer choices to users:
iPhone 操作系统由以下内容，来给用户提供选择：

■ Lists (that is, table views). Users tap a row in a list to select an item. Lists are suitable for displaying almost any number of choices. For details on the ways you can use table views in your application, see “Table Views” (page 89).
列表（即表格视图），用户点击列表中的一行以选择一个项目。列表几乎适用于呈现任何数量的选择。有关详情，见“Table Views”。

■ Pickers, including date and time pickers. Users spin the wheels in a picker until each wheel displays the desired part of a multipart value, such as a calendar date that comprises year, month, and day. For more information about using pickers in your iPhone application, see “Date and Time Pickers” (page 108) and “Pickers” (page 114).
拨选器，包括日期和时间选择。用户旋转在拨选器上的滚轮，直到每个轮子显示如由年、月、日组成的所需部分。了解更多见“Date and Time Pickers”和“Pickers”。

■ Switch controls. Users slide a switch control from one side to the other, revealing one of two values. A switch control is intended to offer a simple choice within a list. For more information about switch controls, see “Switch Controls” (page 99).
开关控件，用户从一侧到另一侧滑动一个开关控制，揭示了两个值之一。一个开关控制的目的是提供一个在列表内的简单的选择。更多信息，见 “Switch Controls” 。

Providing a License Agreement or a Disclaimer
提供一个许可协议或免责声明

If you provide an end-user license agreement (or EULA) with your iPhone application, be aware that the App Store displays it, so that people can read it before they get your application.
如果在您的iPhone应用程序中提供一个最终用户许可协议（或EULA），通常包含描述责任限制和控制使用的各种术语和条件。果你的iPhone应用程序有许可协议，向App Store提供这个协议，使人们可以阅读它，然后才使用应用程序。

If possible, try to avoid requiring users to indicate their agreement to your EULA when they first start your application. This allows users to enjoy your application without delay. However, even though this is the preferred user experience, it might not be feasible in all cases. If you must display a license agreement within your application, try to do so in a way that harmonizes with your user interface and causes the least inconvenience to users.
如果可能，在第一次启动应用程序时尽量避免要求用户同意你的最终用户许可协议。这使得用户能够立即享用您的应用程序。不过，即使这是首选的用户体验，它可能无法在所有情况下都是可行的。如果您必须在您的应用程序显示一个许可协议，尝试一个与您的用户界面相协调的方式，将带给用户的不便减到最少。

Similarly, if you need to provide a disclaimer, be sure to balance your business needs with maintaining a great user experience. If you can, provide your disclaimer within your application description or EULA, so that it is available in the App Store.
同样，如果你需要提供一个声明，你一定要平衡业务需求与维持良好的用户体验。如果可以的话，在您的应用程序提供说明或最终用户许可协议，以便它在App Store可以获得。

Part 2：

Designing the User Interface of Your iPhone Application

设计你 iPhone 应用程序的用户界面

User interface elements in iPhone OS include views and controls. Views provide content regions with well-defined sets of functionality. Controls are graphic objects that cause instant actions or visible results. Although all an application’ s views and controls are contained in the application’ s single window, users see and interact with them in screens, which roughly correspond to different visual states in the application.

iPhone OS 系统的用户界面元素包括视图与控件。视图为内容提供了定义良好的功能集合。控件是引起即时动作二视觉结果的图形对象。应用程序的所有视图与控件都被封装在应用程序的单一视窗内，用户需在屏幕上查看并与之交互，这些屏幕大致与应用程序的不同视觉状态向对应。

iPhone OS defines the standard appearance of these user interface elements, and delivers consistent behaviors that users expect. Read the chapters in Part II to learn about the types of user interface elements available and how to use them to build the user interface of your application.

iPhone OS 系统定义这些用户界面元素的标准外观，并提供与用户期望相一致的行为。阅读第二部分的章节可以学习可用的用户界面元素类型，及如何使用它们来构建你的应用程序的用户界面

CHAPTER 5

A Brief Tour of the Application User Interface

程序用户界面的摘要回顾

Before you delve into the details about specific views and controls, it’ s helpful to gain a high-level understanding of the way these elements can work together and how users expect them to behave. This chapter introduces the views that comprise the building blocks of most applications, describing where they belong and touching on how they’ re used.

在你深入研究关于具体的视图和控件的详细资料之前，对这些元素如何共同工作以及用户期望它们怎样工作有一个深入的理解是很有帮助的。本章介绍了组成大多数应用程序模块的视图，描述了它们的归属，以及如何使用。

To learn more about the appearance, behavior, and usage guidelines of individual user interface elements,be sure to read the chapters following this one. Understanding how each user interface element is designed to be used helps you use it correctly in your application and, if appropriate, customize it to meet your needs.

了解更多关于各个用户界面元素的外观、行为、及使用指南，请务必阅读以下这一章节。理解每个用户界面元素的设计是如何使用的，能帮助你在自己应用程序中正确地使用它，并根据你的需要做适当定制。

Application Screens and Their Contents

应用程序屏幕及内容

Every application, regardless of type, has an application window. Programmatically, the window provides the background on which you present all your application’ s information. But users are not aware of this window; instead, they experience your application as a collection of screens through which they navigate.

每个应用程序，无论类型，都有一个应用程序视窗。通过编程，视窗为你处理所有应用程序的信息提供了背景舞台。但是用户并没有意识到这个视窗，反而他们将应用程序当做一系列由自己导航与切换的屏幕来体验。

Although it’ s not a programmatic construct, you can think of a screen as corresponding to a distinct visual state or mode in your application. Users can see individual screens when they navigate through an information hierarchy, tap different tabs in a tab bar, or tap an Info button to view flip-side configuration options.

虽然它不是程序化的构造，但你可以认为每一屏幕当成应用程序中一独立的视觉状态或模式。用户自上而下地浏览每一屏信息时，在标签条点击不同的标签时，或者点击一个信息按钮查看另一面配置选项时，都会看到一个个独立的屏幕。

Depending on the style of your application, you might have a large number of screens or just a few. For example, Mail can display an accounts screen, screens that list the mailboxes in each account, screens that list the contents of each mailbox, and a screen for each message, in addition to a message composition screen. On the other hand, the Stocks application displays two screens: One screen displays a list of companies and a stock-performance graph and the second screen displays application configuration information.
根据应用程序种类的不同，可以看到很多屏内容，也有可能很少。例如，邮件会有显示一个账户的屏幕，显示每个账户邮箱列表的屏幕，显示每个邮箱内容列表的屏幕，以及显示每条信息的屏幕，此外还有撰写邮件的屏幕。另一方面，股票应用程序会只有两个屏幕：一个屏幕显示公司名单和股票行情图，第二个屏幕显示应用程序的配置信息。

For the most part, users think of an application screen and the device screen as identical. However, an application screen’ s content can extend beyond the bounds of the device screen, requiring users to scroll. For example, the Contacts screen is a single screen in the Phone application, even though it’ s likely to list enough names to fill the device screen several times over.
在大多数情况下，用户认为应用程序的屏幕和设备的屏幕是相同的。然而，通过用户的操作，应用程序屏幕的内容可以超出设备屏幕的范围。例如，尽管联系人这一屏可能会列出足够填满该设备屏幕好几次的姓名，但联系人屏幕也只是电话应用程序内的一个屏幕而已。

Each application screen can contain various combinations of views and controls. Some views include specific controls that do not belong anywhere else, and some controls can be used in a variety of views.
每一个应用程序屏幕内都可能包含各种视图和控件。有些视图中包括了一些特有的具体的控件，有些控件可以在许多视图中应用。

Alerts, action sheets, and modal views are distinct types of views that do not exist in an application screen like most other views; instead, they float above application screens and their views. See “Alerts, Action Sheets, and Modal Views” (page 79) for more information about these views.
快讯，动作表单及模态视图是截然不同的视图，它们不像其他大多数视图一样存在于应用程序屏幕，而是在应用程序屏幕和它们的视图上方浮动。更多关于这些视图的信息见“Alerts, Action Sheets, and Modal Views” 。

Four types of views have special status in the user interface of an application, although they do not need to be included or always be visible in every application. These are:
四种类型的视图在应用程序的用户界面中有着特殊状态，虽然它们不需要包含在每个应用程序中，或者总是可见。它们是：

■ The status bar. 状态条。

This is a unique view that isn’t technically part of the application window, although an application can customize the appearance of the status bar to some extent. See “The Status Bar” (page 67) for more information.

状态条是一个独特的视图，它不是应用程序视窗技术上的一部分，尽管应用程序在一定程度上可以自定义状态条的外观。获取更多信息请见 “The Status Bar” 。

■ The navigation bar. 导航条。

This optional view appears just beneath the status bar and can include titles, buttons, and segmented controls. See “Navigation Bars” (page 68) for more information.

该可选视图显示于状态条的下方 ,包括标题 ,按钮和分段控制。获取更多信息请见 “Navigation Bars”。

■ The tab bar. 标签条。

This optional view appears at the bottom edge of a screen and contains segments that activate different modes in the application. See “Tab Bars” (page 73) for more information.

此可选视图显示于屏幕的底部边缘，包含激活应用程序中的不同模式的片段。获取更多信息请见“Tab Bars”。

■ The toolbar. 工具条。

This optional view appears at the bottom edge of a screen and includes controls that perform specific actions in the current context of the application. See “Toolbars” (page 71) for more information.

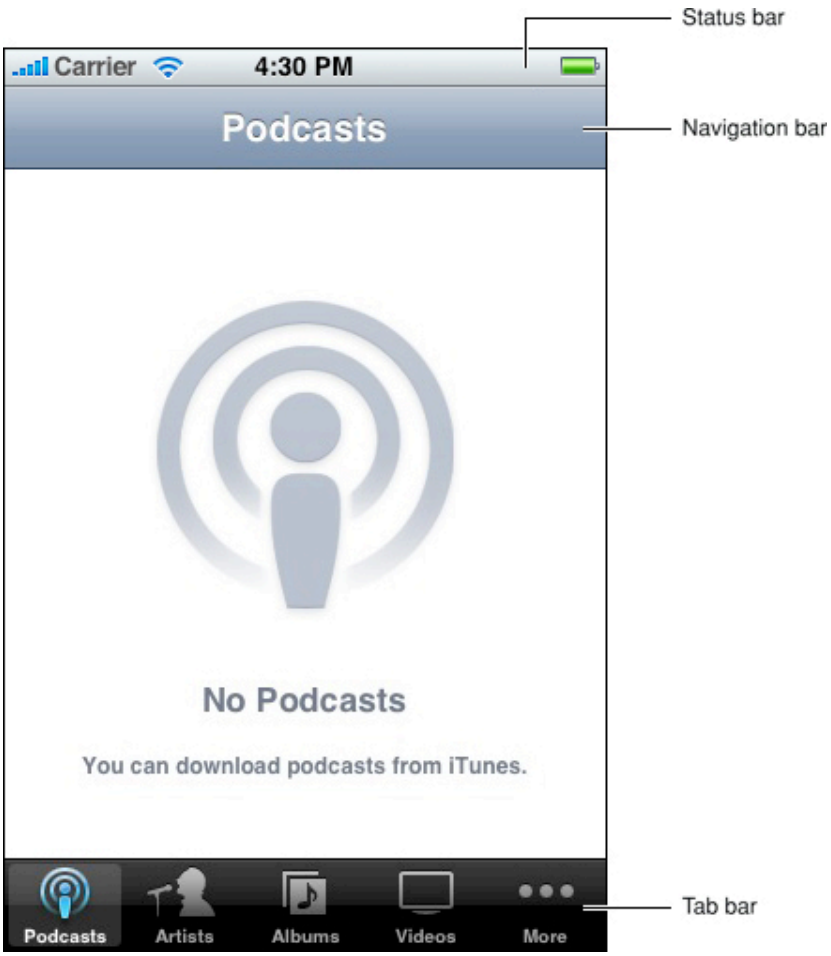
此可选视图显示于屏幕的底部边缘，包括执行应用程序在当前情境下的具体行动的控件。获取更多信息请见“Toolbars” 。

Figure 5-1 shows three of these views in an application screen. Note that if this application used a toolbar, it would appear in place of the tab bar.

图 5-1 显示了在某个应用程序界面三个视图。请注意，如果这个应用程序中使用了一个工具条，显示在标签条的位置。

Figure 5-1 An application screen that contains a status bar, a navigation bar, and a tab bar

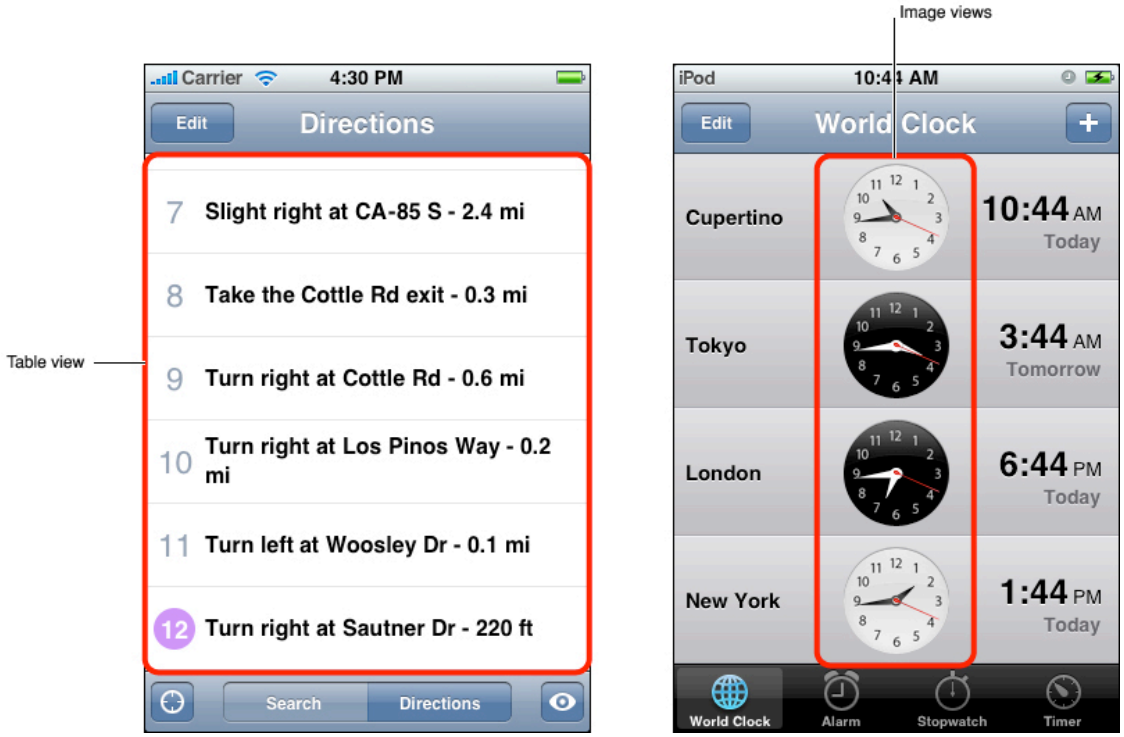
一个应用程序界面包含一个状态条，一个导航条，和一个标签条



In an application that displays some combination of these four views, you can think of the area between the bottom of the navigation bar and the top of the toolbar or tab bar as the content area. In this area, an application screen can contain arbitrary views to display content, such as table views, web views, and image views. Figure 5-2 shows examples of two of the content-area views available in iPhone OS: a style of table view and image views. To learn more about the behavior and appearance of some of these views, in addition to the controls associated with them, see “Table Views, Text Views, and Web Views” (page 89).

在一个有这 4 个视图的应用程序中，可以把导航条底部和工具条或标签条顶部之间的区域看作内容区。在这块区域，应用程序界面可以包含任意的视图来显示内容，如表格视图，文本视图，及图片视图。图 5-2 显示了 iPhone 操作系统中内容区的两个视图的例子：表格视图和图片视图。要了解有关这些视图的行为和外观（除了与他们相关的操控），请参见“Table Views, Text Views, and Web Views” 。

Figure 5-2 Two types of content-area views 两种类型的内容区视图



As mentioned above, there are some controls that are available only in specific views. An example of such a control is the disclosure indicator, which has a specific use in a table view. You can see an example of the disclosure indicator (it looks like >) in the left-hand list in Figure 8-1 (page 89). These controls are described in the sections that cover their associated views. In addition to these, however, there are a handful of controls, such as the detail disclosure indicator, that have a wider usage. See “Application Controls” (page 107) for more information on the controls available to you.

如上所述，有一些控件只能在特定视图中使用。其中一个例子是扩展标示符控件，该控件在表格视图中有着特定的用途。你可以在图 8-1 的左边列表看到一个扩展标示符控件（看起来就像）的例子。这些控件会在与它们相关的 views 章节被详细讲解。不过除了这些，也有极少数的控件，如细节标示控件，更广泛地被使用。请参阅“Application Controls” 以获得更多提供给您的有关操控的信息。

Using Views and Controls in Application Screens

在程序屏幕上使用视图与控件

In iPhone OS, UIKit determines the behavior and default appearance of views and controls. As much as possible, you should use the standard user interface elements UIKit provides and follow their recommended usages. Doing this helps you in two important ways:

在 iPhone 操作系统中，UIKit 决定了视图和控件的行为以及默认外观。您应尽可能地使用 UIKit 提供的标准用户界面元素，并遵照其使用建议。这样做可以在两个重要的方面帮助你：

- Users are accustomed to the look and behavior of standard views and controls. When you use

familiar user interface elements, users can depend on their prior experience to help them as they learn to use your application.
用户习惯于视图和控件的标准外观及行为。当你使用熟悉的用户界面元素，用户可以依靠自己以前的经验来帮助他们学会使用你的应用程序。

■ If iPhone OS changes the look or behavior of standard views or controls, your application continues to work and automatically looks up to date with little, if any, work on your part.
如果 iPhone 操作系统改变了视图和控件的标准外观和行为，你的应用程序将会自动作出小范围的更新。

Many controls support some kind of customization, usually in color or content (such as the addition of a text label or an image). If you’ re developing an immersive application, it’ s reasonable to create controls that are completely different from the default controls. This is because you’ re creating a unique environment, and discovering how to control that environment is an experience users expect in immersive applications.
许多控件都支持各类自定义，通常是颜色或内容（如增加一个文本标签或图像）。如果你正在开发一个沉浸式程序，创建与默认操控完全不同的控件是合理的。这是因为你正创造一个独特的环境，而发现如何操控这个环境是一种用户希望在沉浸式程序中得到的体验。
In general, though, you should avoid radically changing the appearance of a control that performs a standard action. If you use unfamiliar controls to perform standard actions, users will have to spend time discovering how to use them and will wonder what, if anything, your controls do that the standard ones do not.
然而一般情况下，你应该避免从根本上改变一个执行标准行为控件的外观。如果使用不熟悉的控件来执行标准动作，用户将不得不花费时间去了解如何使用他们，并且会认为你的控件不标准。

CHAPTER 6

Navigation Bars, Tab Bars, Toolbars, and the Status Bar

导航条，标签条，工具条及状态条

The status bar, navigation bar, tab bar, and toolbar are views that have specifically defined appearances and behaviors in an iPhone application. These bars are not required to be present in every application (immersive applications often don’ t display any of them), but if they are present, it’ s important to use them correctly.

状态条，导航条，标签条，和工具条在 iphone 应用程序中有特定的外观和行为定义。这些控件对于每一个应用程序来说不是必须的（沉浸式应用程序经常没有这些条），但如果它们有这些条，那么正确使用它们就变得很重要。

The reason is that these bars provide familiar anchors to users of iPhone OS–based devices, who are accustomed to the information they display and the types of functions they perform.

原因是这些控件对以 iPhone OS 设备的用户来说，已在对标准控件上的信息显示及其功能类型的应用方式非常熟悉。

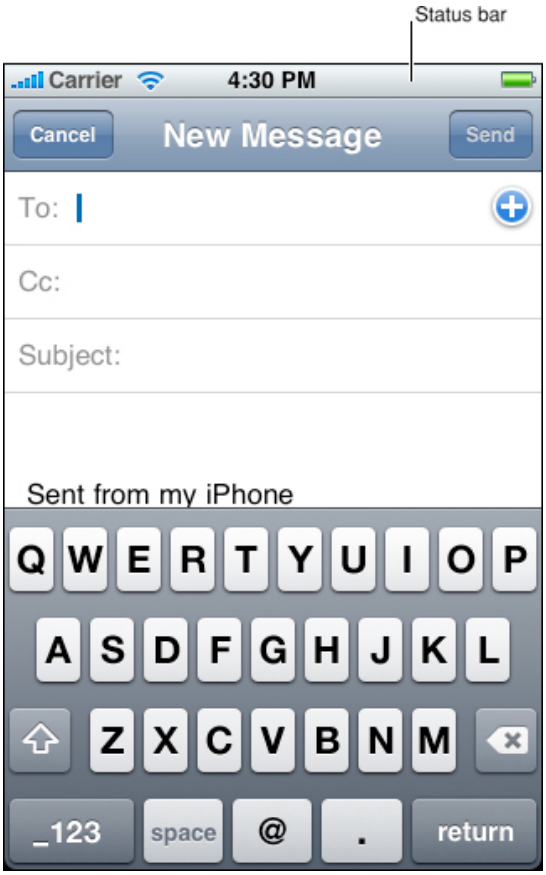
The Status Bar 状态条

The status bar shows users important information about their device, including cell signal strength, the current network connection, and battery charge. Figure 6-1 shows an example of a status bar.

状态条向用户显示该设备的重要信息，包括信号强度，当前的网络状态和电池电量。图 6-1 显示了一个状态条的例子。

Figure 6-1 A status bar contains important information for users

状态条向用户提供了重要信息



Although a full-screen, immersive application can hide the status bar, you should carefully consider the ramifications of this design decision. People expect to be able to see the current battery charge of their devices; hiding this information, and requiring users to quit your application to get it, is not an ideal user experience.
虽然沉浸式应用程序在全屏幕的时候能够将状态条隐藏起来，但是你得仔细考虑这样设计的结果。人们希望看到设备的电池电量;隐藏了那些信息，就需要用户退出全屏去看，这是一个不理想的用户体验。

For example, Photos displays individual photos from a camera roll in a full-screen view that fades out the status bar, navigation bar, and toolbar after a few seconds. This is appropriate because in Photos, users focus on viewing the content, not interacting with it. However, users can bring back the status bar, navigation bar, and toolbar with a single tap on the screen.
例如，全屏浏览照片的时候，状态条，导航条和工具条会在几秒钟之后淡出屏幕。但这是恰当的，因为用户在浏览照片的时候集中观看照片内容，而不是与它进行交互。然而，用户可以通过单一的一个动作将状态条，导航条和工具条带回屏幕。

If you sometimes hide the status bar in your application, you should take advantage of users' experience of this behavior and allow them to redisplay it with a single tap. Unless you have a very compelling reason to do so, it' s best to avoid defining a custom gesture to redisplay the status bar because users are unlikely to discover such a gesture or remember it.
如果你有时在应用程序中隐藏了状态条，你应该利用用户行为经验，并让它们能够通过一个简单的步骤

就重新显示出来。除非你有一个非常令人信服的理由那样做，所以最好避免定义一个自定义的动作来重新现实状态条，因为用户未必能够发现和记住这个动作。

Although you have little control over the contents of the status bar, you can customize its appearance and, to some extent, its behavior. Specifically, you can:
尽管你对于状态条的内容控制极少，但是在一定程度上你可以定制它的外观和行为。具体来说，你可以：

■ Indicate whether the network activity indicator should be visible. You should display the network activity indicator if your application is performing a network operation that will take more than a couple of seconds. If the operation will finish sooner than that, you don’ t have to show the network activity indicator, because it would be likely to disappear before users notice its presence. (In your code, you use the UIApplication method networkActivityIndicatorVisible to control the indicator’ s visibility.)
指定网络活动指示是否应可见。如果你的应用程序正在执行网络操作你就应该显示网络活动指示。如果操作非常短暂，你没有必要显示网络活动，因为它很可能在用户注意到这点的时候消失（在代码中，通过 UIApplication 方法 networkActivityIndicatorVisible 来控制网络活动显示。）

■ Specify the color of the status bar. You can choose gray (the default color), opaque black, or translucent black (that is, black with an alpha value of 0.5). Figure 6-2 shows these styles. (Note that you set a value in your Info.plist file to specify the status bar style; see iPhone Application Programming Guide for more information on how to do this.)
指定状态条的颜色。你可以选择灰色（默认色），黑色不透明，或者半透明的黑色（也就是用黑色 0.5 的 alpha 值）。图 6-2 显示了这些样式。（在 Info.plist 文件中通过设定一个值来指定状态条的样式。更多关于如何操作的信息见 iPhone Application Programming Guid ）

■ Set whether the change from the current status bar color to the new color should be animated. (Note that the animation causes the old status bar to slide up until it disappears off the screen, while the new status bar slides into place.)
设置状态条的颜色更改是否使用一个动画效果。（请注意，动画能够使旧的状态条下滑，直到它滑出整张屏幕，同时新的状态条上升到相应位置。）

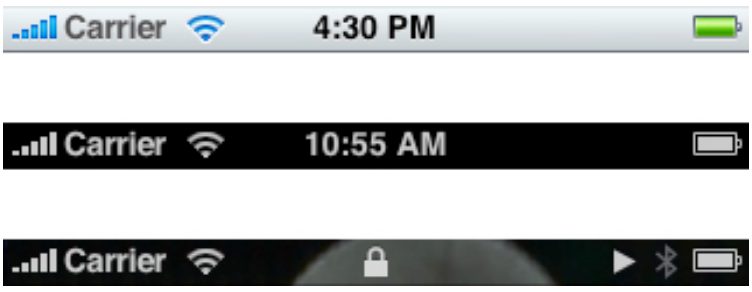


Figure 6-2 Three styles of status bars 三种样式的状态条

Be sure to choose a status bar appearance that coordinates with the rest of your application.For

example, avoid using a translucent status bar if the navigation bar is opaque.
一定要选择一款与你的应用程序其它部分相匹配的状态条外观。例如，当导航条是不透明的时候应该避免使用半透明（透明）的状态条。

Navigation Bars
导航条

A navigation bar appears at the upper edge of an application screen, just below the status bar. A navigation bar usually displays the title of the current view and can contain controls that act on the view' s contents, in addition to navigational controls when appropriate. Navigation bars are especially useful in productivity applications (described in "Productivity Applications" (page 19)), because these applications typically arrange information in a hierarchy.
导航条出现在应用程序屏幕的上边缘，状态条之下。一个导航条通常显示当前视图的名称，操控视图的控件并且可以根据需要添加导航控件。导航条在生产力辅助应用程序中尤其有用（在"Productivity Applications"中被这样描述过），因为这些应用程序通常将信息安排在一个层次结构中。

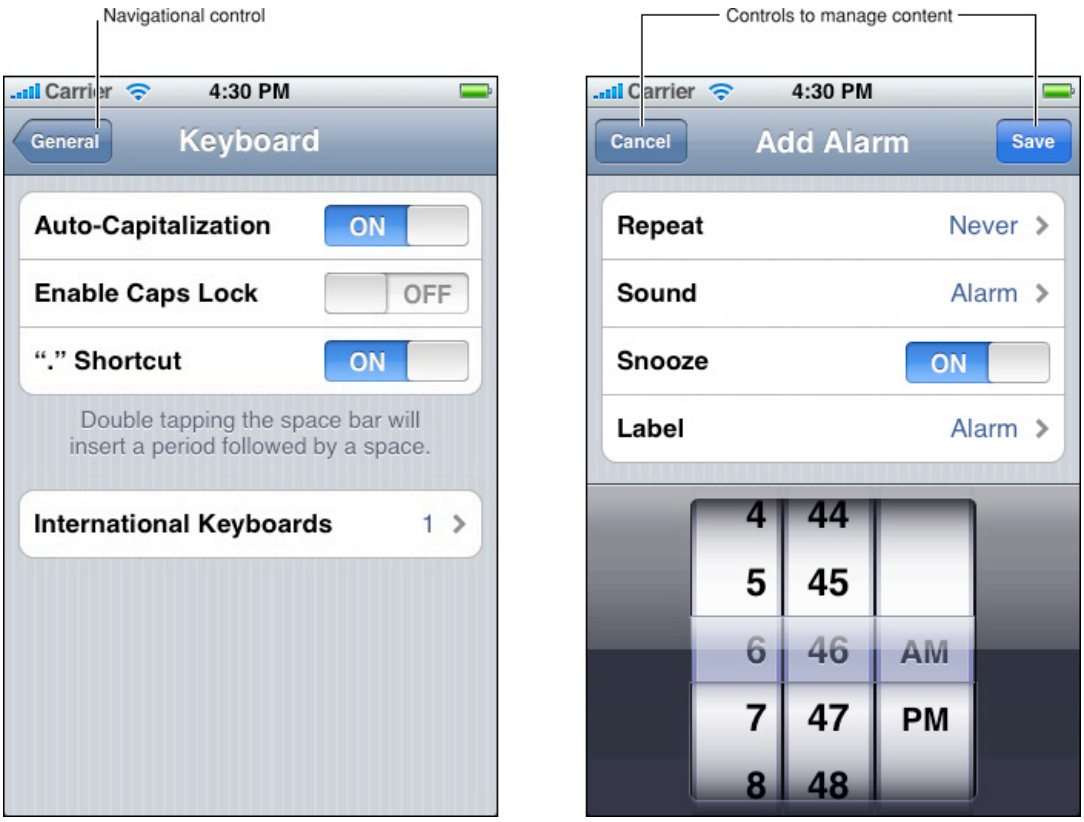
Navigation bars have two purposes:
导航条有两个作用：

- To enable navigation among different views in an application
能够对应用程序中不同视图进行引导。

- To provide controls that manage the items in a view
提供控件使之能够在视图中管理项目

Figure 6-3 shows examples of both these uses.
图 6-3 显示了这两种用途的例子。

Figure 6-3 Navigation bars can contain navigational controls and controls to manage content
导航条能够包含导航控件和管理内容的控件



Navigation Bar Contents

导航条内容

A navigation bar can display just the title of the current view, centered along its width, as shown in Figure 6-4. The initial view in a productivity application should include a navigation bar that displays only the titleof the first view because the user hasn’t yet navigated to another location.

导航条可以显示只是当前视图标题，根据它的宽度居中，如图 6-4 所示。在生产力辅助应用程序中最初的视图应该包括一个只显示最初视图标题的导航条，因为该用户还没有浏览到另一个位置。

Figure 6-4 A navigation bar displays the title of the current view

一个显示当前视图标题的导航条

As soon as the user navigates to another view, the navigation bar should change its title to the title of the new location, and should provide a back button labeled with the title of the previous location. For example, Figure 6-5 shows the navigation bar in Date & Time settings, which is in

General settings.
当用户导航到另一个视图的时候，导航条应该改变标题，使用新位置的标题，并应提供一个带有先前位置标签的返回按钮。例如，图 6-5 显示了 Date & Time 设置中的导航条，这是 General 设置中的内容之一。

Figure 6-5 A navigation bar can contain a navigational control
一个导航条可以包含一个导航控件

The standard back button gives users a reliable way to return to the previous screen, so it's important to avoid altering the button's behavior. In particular, you should avoid creating a multi-segment back button, such as the one shown in Figure 6-6.
标准的返回按钮为用户提供了一个可靠的方式返回到前一个屏幕，所以重要的是要避免改变按钮的行为。特别是，你应该避免创建多段后退按钮，如图 6-6 所示的。

Figure 6-6 A multi-segment back button is not recommended
多段后退按钮，不推荐

Using a multi-segment back button causes several problems:
使用多段返回按钮导致的几个问题：

- The extended width of a multi-segment back button does not leave room for the title of the current screen.
一个多段后退按钮会挤压当前屏幕上的标题显示空间。

- There is no way to indicate the selected state of an individual segment.
没有办法指出单独片段中被选择的部分。

- The more segments there are, the smaller the hit region for each one, which makes it difficult for users to tap a specific one.
片段越多，每一个片段所能够被点中的区域就越小，这使用户点击它们中的一个变得困难。

- Choosing which levels to display as users navigate deeper in the hierarchy is problematic.
用户导航到更深层次的浏览时，应该支持哪个层次是大问题。

If you think users might get lost without a multi-segment back button that displays a type of

breadcrumb path, it probably means that users must go too deeply into the information hierarchy to find what they need. To address this, you should flatten your information hierarchy.

如果你认为用户没有多段后退按钮会迷失，这可能意味着用户必须进入深层次的信息才能找到他们需要什么。为了解决这个问题，你应该简单化你的信息层次结构。

In addition to a back button, a navigation bar can also contain a second button to the right of the title. If you do not need to display a back button (because your application does not support hierarchical navigation), you can opt instead to display a button that affects the contents of the view, such as an Edit button, to the left of the title. Figure 6-7 shows an example of this.

除了一个返回按钮，导航条也可以包含第二个按钮，在标题的右边。如果你不需要显示返回按钮，(因为你的应用程序不支持分层导航)，你可以选择显示一个影响视图内容的按钮，比如位于标题左侧的编辑按钮。图 6-7 举了一个例子。

Figure 6-7 A navigation bar can contain controls that manage the content in the view

导航条可以包含管理视图内容的控件

To learn how to implement a navigation bar in your application, see “Navigation Controllers”.

要了解如何实现一个应用程序中的导航条，见“Navigation Controllers”。

As you can see in the illustrations above, buttons in a navigation bar include a bezel around them. In iPhone OS, this style is called the bordered style. All controls in a navigation bar should use the bordered style. In fact, if you place a plain (borderless) control in a navigation bar, it will automatically convert to the bordered style.

正如上面所看到的图标一样，导航条里的按钮有一圈带梯度边缘。在 iPhone 的操作系统中，这种样式被称作边框样式。导航条中所有控件都应该使用边框样式。实际上，如果你放置一个普通控件（无边框）在导航条里，它会自动转换成边框样式。

You can design your own icons for use in navigation-bar buttons, or you can take advantage of the predefined buttons iPhone OS provides. See “Standard Buttons for Use in Toolbars and Navigation Bars” (page 124) for more information on the buttons available to you.

你自己可以为导航条设计按钮，或者你可以使用由 iPhone 操作系统提供的预定义的按钮。见“Standard Buttons for Use in Toolbars and Navigation Bars”以获得更多关于按钮设定的信息。

Although you can specify a font for all text displayed in a navigation bar, it’s recommended that you use the system font for maximum readability. When you use the appropriate UIKit programming interfaces to create your navigation bar, the system font is used automatically to

display the title.
虽然你可以在导航条中指定文本字体，但还是推荐你使用系统字体以获得最大限度的易读性。当你用 UIKit 撰写界面代码来创建你的导航条时，系统字体会自动被应用到标题中。

Navigation Bar Size and Color
导航条的尺寸和颜色

Changing the device orientation from portrait to landscape can change the height of the navigation bar automatically (you should not specify the height programmatically). In landscape orientation, the thinner navigation bar provides more space for your screen contents. Be sure to take the difference in heights into account when you design icons for navigation bar controls and when you design the layout of your screens.
将设备方向从纵向更改为横向的时候，导航条高度会自动改变（你不应该指定其高度）。在横向的情况下，较薄的导航条可以为你的屏幕内容提供更多的空间。当你为导航条设计图标和设计屏幕布局的时候，要明确考虑其高度差。

You can specify the color and translucency of a navigation bar to coordinate with the overall look of your application and with the other bars in it (that is, toolbars, tab bars, and the status bar). You can use a custom color or choose one of the standard colors:
你可以指定导航条的颜色和透明度用来配合应用程序和其他控件的整体外观（即工具条，标签条和状态条）。你可以使用自定义颜色，或者选择标准颜色之一：

- Blue (the default color)
蓝色（默认色）
- Black
黑色

If it complements the look of your application, you can add translucency to the navigation bar. When you use a translucent navigation bar, the screen gives the impression of having a larger visible area, which is especially desirable in landscape orientation. Be sure to avoid mixing a translucent navigation bar with an opaque black status bar (although you can display a translucent navigation bar with an opaque gray status bar).
如果给导航条添加透明度可以使应用程序的外观更完善，那么就添加这种效果。当使用一个半透明的导航条时，用户会感到屏幕上给出了一个更大的可视面积，尤其是横向的情况下。一定要避免将一个半透明的导航条和一个不透明的黑色状态条混合（尽管你可以显示一个半透明的导航条和一个不透明的灰色状态条）

Strive for consistency in the appearance of navigation bars and other bars in your application. If you use a translucent navigation bar, for example, don’t combine it with an opaque toolbar.

Also, avoid changing the color or translucency of the navigation bar in different screens in the same orientation.
努力使导航条与其他控件在应用程序中保持一致性。如果你使用的是半透明的导航条，比如，不要将不透明的工具条和其结合。此外，在同一个方向不同屏幕上要避免改变导航条的颜色和透明度。

Toolbars 工具条

If your application provides a number of actions users can take in the current context, it might be appropriate to provide a toolbar. A toolbar appears at the bottom edge of the screen and contains buttons that perform actions related to objects in the current view. A toolbar should not be used to switch among different modes in an application; if you need to do this, use a tab bar instead (see "Tab Bars" (page 73) for more information).
如果你的应用程序提供了一些用户可以在当前内容下采取的动作，那么最好提供一个工具条。工具条显示在屏幕底部边缘，包含在当前视图下执行有关对象的行动按钮。工具条不应该用于切换应用程序下的不同模式；如果你必须这么做，那就用标签条，而不是工具条。（参考"Tab Bars"以获得更多信息）。

For example, when users view a message in Mail, the application provides a toolbar that contains items for deleting, replying to, and moving the message, in addition to checking for new mail and composing a new message. In this way, users can stay within the message-viewing context and still have access to the commands they need to manage their email. Figure 6-8 shows what this looks like.
例如，当用户查看邮件消息时，应用程序提供了工具条，包括删除项目，回复和移除消息项目，还有查看新邮件和撰写新邮件项目。这样，用户就可以在查看具体内容的同时进行管理与处理邮件了。图 6-8 显示如此。

Figure 6-8 A toolbar provides functionality within the context of a task
工具条当前任务下提供的功能。

Toolbar Contents 工具条内容

The toolbar displays toolbar items equally spaced across the width of the toolbar. It’ s a good idea to constrain the number of items you display in a toolbar, so users can easily tap the one they want. Remember that the hit-region of a user interface element is recommended to be 44 x 44 pixels, so providing five or fewer toolbar items is reasonable. Figure 6-9 shows an example of appropriate spacing of toolbar items in a toolbar.

根据工具条的宽度，工具条上的项目被平均水平分布。对于工具条上的项目个数有一个强制限制是个好主意，这样用户就能更容易地点击按钮。请记住，一个用户界面元件被点击的区域最好为 44X44 像素，所以提供 5 个或者更少的工具条项目是合理的。图 6-9 显示了在一个工具条中工具条项目合理分布的例子。

Figure 6-9 Appropriately spaced toolbar items

工具条项目的合适距离

The items in both Figure 6-8 (page 72) and Figure 6-9 do not include a bezel. In iPhone OS this style is called the plain style. (For an example of the bordered style, look at the buttons in Figure

6-7 (page 70).) Although you can use either the bordered or plain style for buttons in a toolbar, you should not mix both styles in the same toolbar.
在图 6-8 和图 6-9 中的项目都不包括内切面。在 iPhone 操作系统中这种样式被叫做平整样式。（边框样式的例子，参见图 6-7 中的按钮）。尽管在工具条中你可以使用边框样式和平整样式中的任何一款，但是你不应该在用一个工具条中混合使用它们两者。

You can design your own icons for use in toolbar buttons, or you can take advantage of the predefined buttons iPhone OS provides. (See “Standard Buttons for Use in Toolbars and Navigation Bars” (page 124) for more information on the buttons available to you.) If you choose to create custom toolbar buttons, be sure to make them as similar in size as possible to achieve a balanced, attractive appearance.
你可以为工具条按钮自己设计图标，或者可以使用 iPhone 操作系统预定义的按钮，（参见“Standard Buttons for Use in Toolbars and Navigation Bars”以获得更多有关按钮的信息）如果你选择自己创建工具条按钮，请务必让它们在尺寸大小方面尽可能的相似，以达到一定的平衡和美观。

Toolbar Size and Color
工具条的尺寸和颜色

Changing the device orientation from portrait to landscape can change the height of the toolbar automatically (you should not specify the height programmatically). The thinner toolbar available in landscape orientation leaves more room for your screen contents. Be aware of the difference in heights when you design icons for toolbar buttons and when you design the layout of your screens.
将设备方向从纵向改为横向的时候，工具条高度会自动改变（你不应该指定其高度）。在横向的情况下，较薄的工具条可以为你的屏幕内容提供更多的空间。当你为工具条设计图标和设计屏幕布局的时候，要明确考虑其高度差。

You can specify the color and translucency of a toolbar to coordinate with the overall look of your application and with the other bars in it (that is, navigation bars, tab bars, and the status bar). You can use a custom color or choose one of the standard colors:
你可以指定工具条的颜色和透明度用来配合应用程序和其他条的整体外观（即导航条，标签条和状态条）。你可以使用自定义颜色，或者选择标准颜色之一：

- Blue (the default color)
蓝色（默认色）
- Black
黑色

If it complements the look of your application, you can add translucency to the toolbar. When you use a translucent toolbar, the screen gives the impression of having a larger visible area,

which is especially advantageous in landscape orientation.
如果给工具条添加透明度可以使应用程序的外观更完善，那么就添加这种效果。当使用一个半透明的导航条时，用户会感到屏幕上给出了一个更大的可视面积，尤其是横向的情况下。

Strive for consistency in the appearance of toolbars and other bars in your application. If you use a translucent toolbar, for example, don’ t combine it with an opaque navigation bar. And, avoid changing the color or translucency of the toolbar in different screens in the same orientation.
努力使导航条与其他控件在应用程序中保持一致性。如果你使用的是半透明的工具条，比如，不要将不透明的导航条和其结合。同时应该避免在同一方向不同屏幕上改变工具条的颜色和透明度。

Tab Bars
标签条

If your application provides different perspectives on the same set of data, or different subtasks related to the overall function of the application, you might want to use a tab bar. A tab bar appears at the bottom edge of the screen.
如果你的应用程序对于同一套数据，或者和整体功能相关的不同子任务有不同的想法期望，你可能需要使用一个标签条。一个标签条显示在屏幕的底部边缘。

A tab bar gives users the ability to switch among different modes or views in an application, and users should be able to access these modes from everywhere in the application. However, a tab bar should never be used as a toolbar, which contains buttons that act on elements in the current mode (see “Toolbars” (page 71) for more information on toolbars).
标签条能够使用户在应用程序中就不同的模式和视图进行切换，而且用户应该能够在应用程序的各个位置都能进入这些模式中。但是，标签条绝不应该被当做工具条使用，其中包括在当前模式下起作用的按钮（参见“Toolbars”以获得更多关于工具条的信息）

For example, on iPhone, iPod uses a tab bar to allow users to choose which part of their media collection to focus on, such as Podcasts, artists, videos, or playlists. The Clock application, on the other hand, uses a tab bar to give users access to the four functions of the application, namely, World Clock, Alarm, Stopwatch, and Timer. Figure 6-10 shows how selecting a tab in a tab bar changes the view in Clock. Notice how the tab bar remains visible in the different Clock modes shown in Figure 6-10. This makes it easy for users to see which mode they’ re in, and allows them to access all Clock modes regardless of the current mode.
例如，在 iPhone 中，iPod 使用的一个标签条，允许用户选择自己较关注的媒体部分，比如播客，艺术家，视频或者播放清单。另一方面，时钟应用程序用了一个标签条使用户能够访问软件的四个功能，即，世界时钟，闹钟，秒表，定时器。图 6-10 显示了在时钟应用程序中标签条如何改变职能的图片。在图 6-10 中，请注意标签条是如何在不同的时钟软件模式下保留视觉效果的。这使得用户很容易看到他们所

处的模式，并允许他们访问除当前模式以外的模式。

Figure 6-10 A tab bar switches views in an application

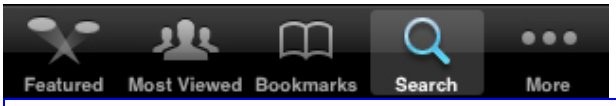
图 6-10　标签条在应用程序中转换视图

The tab bar displays icons and text in tabs, all of which are equal in width and display a black background. When a tab is selected, its background lightens and the image in the tab is highlighted. Figure 6-11 shows how this looks.

标签条显示图标和文字，所有的这些标签都具有相等的宽度，并且显示在黑色背景中。当一个标签被选中时，其背景淡化并且标签中的形象突出。图 6-11 显示

Figure 6-11 A selected tab in a tab bar

图 6-11　标签条中被选中的标签



Note: A tab bar does not change its opacity or height, regardless of orientation.

注：标签条不论方向，不改变其不透明度或高度。

iPhone OS provides a number of icons for tabs, such as the items labeled Featured and Bookmarks in Figure 6-11. If you choose to use these icons, be sure to use them in accordance with their documented meaning. For more information on the tab bar icons available to you, see “Standard Icons for Use in Tab Bars” (page 126).

iPhone 操作系统提供了一些标签条用的图标，如图 6-11 中的特征图标和书签图标。如果你选择使用这

些图标，一定要让图标和文件的意义相符合。如果需要了解更多关于标签条的图标，请参阅“Standard Icons for Use in Tab Bars”

Providing Additional Tabs
提供额外的标签

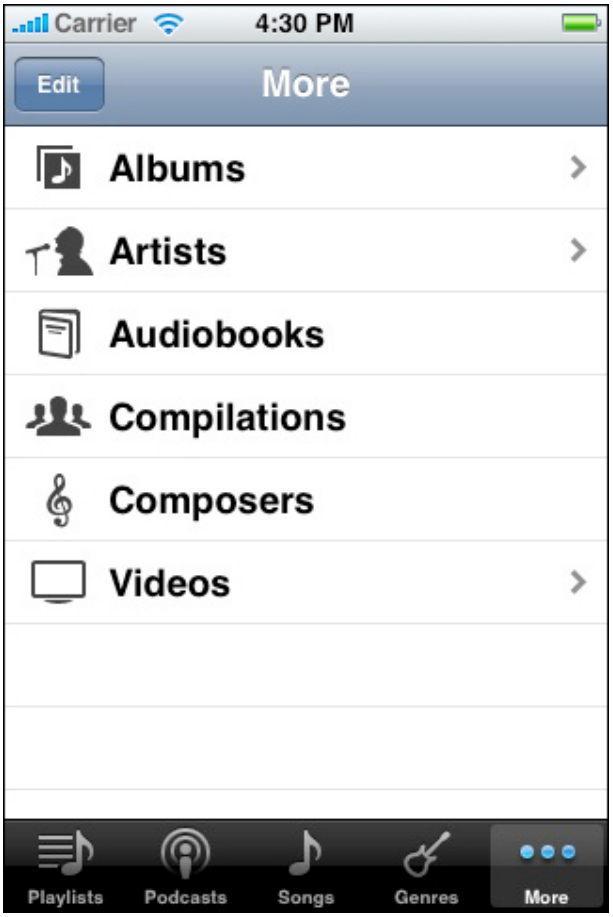
If your application's tab bar contains five or fewer tabs, iPhone OS displays all of them, equally spaced within the tab bar, as shown in Figure 6-12.
如果应用程序的标签条包含五个或更少的标签，iPhone 操作系统会显示全部标签，在标签条中等距离隔开，如图 6-12 所示。

Figure 6-12 iPhone OS displays up to five tabs in a tab bar
图 6-12　iPhone 操作系统在标签条中最多可显示五个标签

If your application’ s tab bar contains more than five tabs, iPhone OS displays four of them in the tab bar and adds a More tab, as shown in Figure 6-11 (page 74).
如果应用程序的标签条包含的标签超过五个，iPhone 操作系统就只会显示它们中的四个，并且添加一个“更多”标签，如图 6-11 所示。

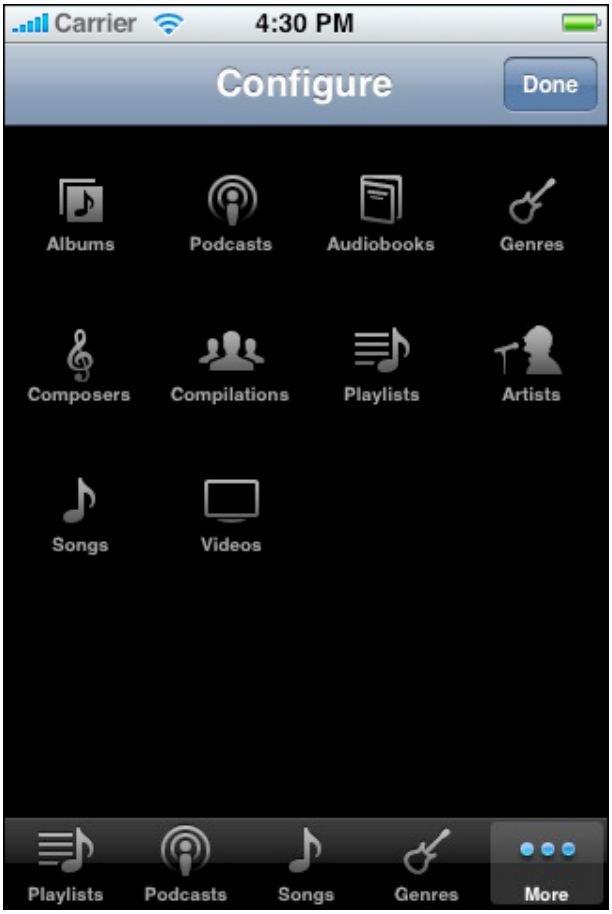
Users tap the More tab to see a list of additional tabs in a separate screen, as shown in Figure 6-13.
用户点击“更多”标签，在一个单独的屏幕上可以看到一个额外的标签列表，如图 6-13 所示。

Figure 6-13 Additional tabs are displayed when users tap the More tab
当用户点击 More 标签，其它隐藏的标签显示出来



The More screen can also include an Edit button that users can tap to configure the tab bar so that it displays the tabs they use most often. For example, Figure 6-14 shows the Configure screen users see after they tap the Edit button in the iPod application’ s More screen.
这个“More”屏幕还包括一个编辑按钮，用户可以点击配置标签条来显示他们最常使用的标签。例如，图 6-14 显示了用户点击 iPod 应用程序的编辑按钮后的配置屏幕。

Figure 6-14 When an application has more than five tabs, users can select their favorite tabs to display in the tab bar
当应用程序已超过了 5 个标签，用户可以选择自己喜欢的标签显示在标签条中。



Notice how iPod uses the same tab icons in all three places (the tab bar, the More screen, and the Configure screen). This helps users be confident that each icon means the same thing, regardless of where it’ s displayed.

应注意 iPod 在三个地方使用了相同标签图标（标签条，“更多”窗口，配置窗口）。这样就能够帮助用户明确那些图标各自的意义，无论它们显示在哪里。

Badging a Tab in a Tab Bar

标签条中标徽标签

You can display a badge on a tab to communicate with users in a nonintrusive, understated way. This type of feedback is suitable for communicating information that isn’ t critical to the user’ s task or context, but that is useful to know. The badge looks similar to the one Phone displays on the Voicemail tab to indicate the number of unheard messages: it is a red oval that appears near the upper right corner of the tab. The white text inside the oval provides the information.

你都能够在标签上面贴标徽，通过非侵入性和简单的方法和用户进行沟通。这种反馈类型适用于那些对于用户任务或情景并不重要但需知道的通信信息。这个标徽类似于语音信箱上显示的标签，表明了一些没看过的信息：这是一个在选项卡右上角的红色椭圆形。椭圆形里面的白色文本提供了信息。

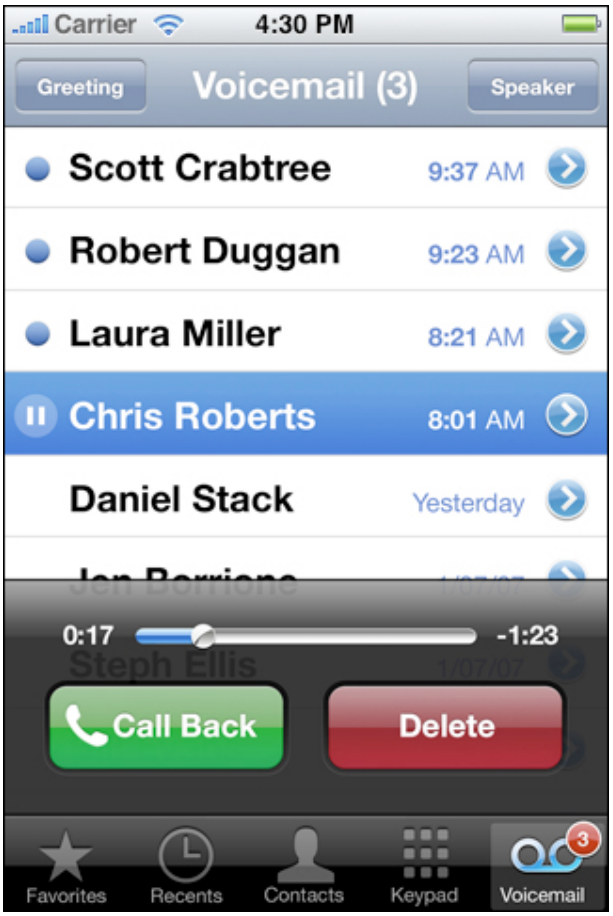
Associating a badge with a specific tab allows you to connect the information in the badge with

a particular mode in your application, even when that mode is not the current one. Figure 6-15 shows an example of a badge on a tab.

标徽与标签通过关联使你能够在应用程序中通过特别的模式来连接标徽中的信息，即使该模式不是当前的。图 6-15 显示了标签条上一个标徽的例子。

Figure 6-15 A badge conveys information in a tab bar

图 6-15　标签条上标徽传达的信息



Note that a badge can also be displayed on your application icon in the Home screen if you register for Apple Push Notification Service and users agree to allow badging. See “Enabling Push Notifications” (page 49) for more information on how this works.

请注意，如果你注册并允许了苹果的通知服务，那么标徽也可以在主屏幕上显示。参阅“Enabling Push Notifications”以获得更多关于这方面的信息。

CHAPTER 7

Alerts, Action Sheets, and Modal Views

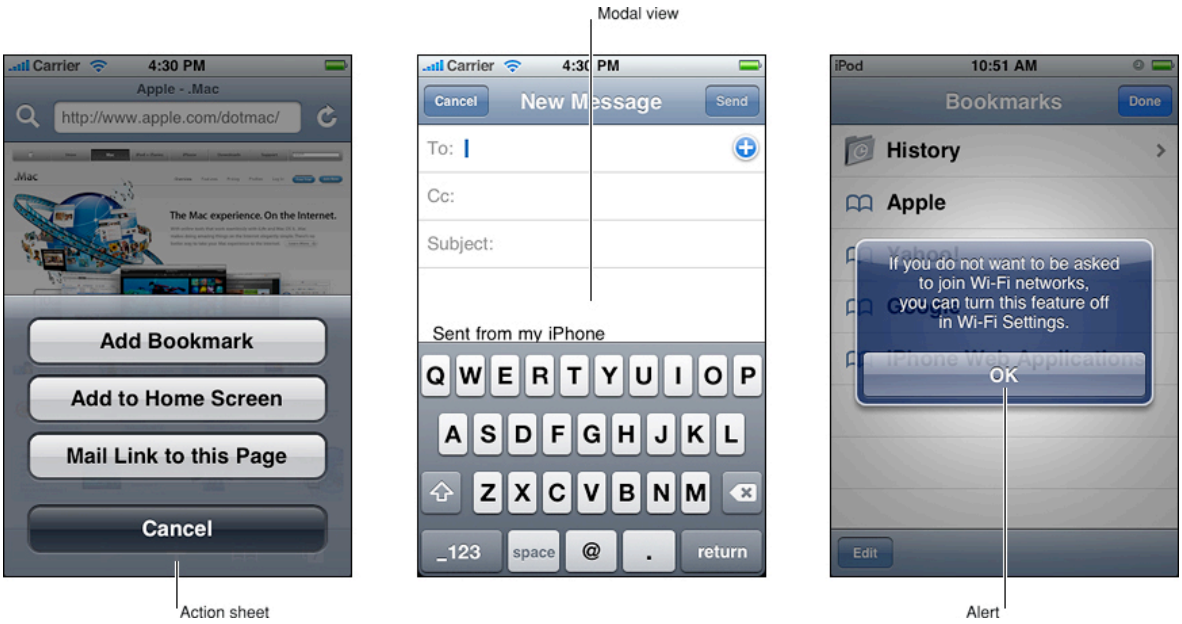
提醒，动作表单，及模态视图

Alerts, action sheets, and modal views are types of views that appear when something requires the user’ s attention or when additional choices or functionality need to be offered. Figure 7-1 shows examples of these
types of views.

提醒，动作表单，及模态视图这些视图类型，是在需要用户注意或者当有额外的选择或功能被提供时出现的。 图 7-1 为这些视图类型的例子。

Figure 7-1
An action sheet, a modal view, and an alert

动作表单、模态视图、提醒



To learn about implementing these types of views programmatically, see “Modal View Controllers”.

要了解这类视图代码的实现，请查看“Modal View Controllers”。

Usage and Behavior

用法与行为

Alerts, action sheets, and modal views are all modal, which means that users must explicitly

dismiss them, by tapping a button, before they can continue to use the application. Although there are times when you need to warn users of potentially dangerous actions or provide extra choices, it’ s important to avoid overusing these views. This is because:
提醒，动作表单，及模态视图都是一种模态，这意味着用户继续使用该应用程序前，必须通过点击一个按钮来明确选择以消除此模态。虽然有些时候，当用户有潜在危险动作，或者提供了额外选项时需要提醒一下用户，但是避免过度使用这些模态视图是重要的。这是因为：

■ All types of modal views interrupt the user’ s workflow.
所有类型的模态视图都会中断用户的工作流程。

■ The too-frequent appearance of a view requesting confirmation or acknowledgment is likely to be more annoying than helpful.
过于频繁地出现要求确认或给予肯定的模态视图不仅于事无补，更令人讨厌。

Alerts, in particular, should be used only rarely. When alerts appear too frequently, people are likely to dismiss them without reading them, just to get them out of the way.
特别是提醒，要慎用。当提醒出现过于频繁，人们很可能会忽略它，继续干他们的。

Alerts, action sheets, and modal views are designed to communicate different things:
通过设计提醒，动作表单，及模态视图，来传达不同事情：

■ Alerts give people important information that affects their use of the application (or the device).
The arrival of an alert is usually unexpected, because it generally tells people about a problem or a change in the current situation that might require them to take action.
提醒给人们影响他们使用应用程序（或设备）的重要的信息。 一个提醒通常是意外的到来，因为它通常要求用户马上采取行动，来解决当前状态下的一个问题或作出一个抉择。

■ Action sheets give people additional choices related to the action they are currently taking. People learn to expect the appearance of an action sheet when they tap a toolbar button that begins either a potentially destructive action (such as deleting all recent calls) or an action that can be completed in different ways (such as a send action for which users can specify one of several destinations).
动作表单给予用户及正在进行的动作提供额外的选择。当使用者点击一个工具条按钮开始一个有潜在破坏性的动作（例如删除所有最近通话），或者一个可以用不同方式来完成的动作（如发送动作，用户可以指定多个目的地之一 ）。在这些情况下，使用者会期望出现一个动作表单以供选择。

■ Modal views provide more extensive functionality in the context of the current task. Modal views
can also provide a way to perform a subtask directly related to the user’ s workflow.

模态视图对当前的情景下进行的任务提供更广泛的功能。模态视图还可以提供一种直接关系到用户的工作流程的子任务途径。

These types of views also differ in appearance and behavior, which underscores the difference in the messages they send. Because users are accustomed to the appearance and behavior of these views, it’ s important to use them consistently and correctly in your application. Read the following sections to learn more about using alerts, action sheets, and modal views.
这类模态在外观和行为上都有所区别，以强调所发送信息的差异。因为用户习惯了这些模态的外观和行为，所以在你的应用程序中正确使用并保持一致性很重要。阅读一下章节学习更多提醒、动作表单及模态视图的使用方法。

Using Alerts 使用提醒

An alert pops up in the middle of the application screen and floats above its views to give users critical information in a highly visible way. The unattached appearance of an alert emphasizes the fact that its arrival is due to some change in the application or the device, not necessarily as the result of the user’ s most recent action. An alert should display text that describes the situation and, ideally, give users a way to choose an appropriate course of action.
提醒会在屏幕中间弹出并浮动在所有其它视图之上，以一种高度明显的方式推送给用户一个重要信息。提醒用未强化的外观强调了是要通知用户，应用程序或者设备本身有了变动，而不一定指用户最近动作所带来的结果。提醒应该显示文本以描述情况，并给用户一个比较理想的和适当的动作选择。

Users are accustomed to seeing alerts from the device or from built-in applications that run in the background, such as Messages, but you should seldom need to use them in your application. For example, you might use an alert to tell users that the task they initiated is blocked. It makes sense to display an alert with this message, because it’ s important to tell users what the problem is and give them a choice of ways to handle it.
用户已经习惯看到来自设备和在后台运行的应用程序（如短信）的提醒。但你应该在自己的应用程序中很少使用到它们。例如，使用提醒告诉用户，他们在执行的任务被阻止了。这些提醒是合理的，因为它告诉用户是什么问题，提示用户用什么方法解决这个问题。

You can also use an alert to give users a chance to accept or reject an outcome that is potentially dangerous. When this is the case, the alert should display two buttons: one that dismisses the alert and performs the action and one that dismisses the alert without performing the action. Often, it makes sense to use the label "Cancel" for the button that dismisses the alert without performing the action. Note that if users press the Home button while such an alert is visible, the result, in addition to quitting the application, should be identical to tapping the Cancel button: That is, the alert is dismissed and the action is not performed.

也可以利用提醒，给用户选择接受或拒绝一个有潜在危险的操作。当这种情况下，提醒应显示两个按钮，一个是关闭提醒继续操作，另一个是关闭提醒并停止操作。通常情况下，是使用“取消”按钮来表示关闭提醒并停止操作。请注意，如果用户看到提醒后按下“HOME”按钮，这样的结果就是等同于同时按下取消按钮，并直接退出应用程序。也就是说提醒被关闭并停止动作执行。

The infrequency with which alerts appear helps users take them seriously. Be sure to minimize the number of alerts your application displays and ensure that each one offers critical information and useful choices. In general, try to avoid creating alerts that:
推送提醒以帮助用户认真对应事件的次数并不多。提醒的数量应该控制在最少的范围，而且要确保每一个都能够提供准确的信息和有用的选择。通常情况下，尽量避免出现以下的提醒：

■ Update users on tasks that are progressing normally. Instead, consider using a progress view or an activity indicator to provide progress-related feedback to users (these controls are described in “Progress Views” (page 115) and “Activity Indicators” (page 107)).
在用户进行操作的时候进行正常的任务更新。 相反，应该考虑使用进程视图或者活动标示为用户提供相关反馈（这些控制都会在“Progress Views”和“Activity Indicators”中进行描述）

■ Ask for confirmation of user-initiated actions. To get confirmation for an action the user initiated, even a potentially risky action such as deleting a contact, you should use an action sheet (described next in “Using Action Sheets” (page 81)).
要求用户主动确认开始的动作。 为了得到用户对于一个动作的主动确认，甚至是有潜在危险的动作例如删除一个联系人，你应该使用一个动作表单（详见“Using Action Sheets”）

■ Inform users of errors or problems about which they can do nothing. Although it might be necessary to use an alert to tell users about a critical problem they can’t fix, it’s better to integrate such information into the user interface, if possible. For example, instead of telling users every time a server connection fails, display the time of the last successful connection.
通知用户一些关于他们无能为力的错误或者问题。 尽管可能需要使用一个警示去告诉使用者一个他们无法修改的很严重的问题，但是最好还是将这样的信息整合到使用界面上去，如果可能的话，比如，用显示最后一个连接成功的的时间来替代每一次都告诉用户一个服务连接的失败。

Using Action Sheets
使用动作表单

An action sheet displays a collection of alternatives that are associated with a task users initiate by tapping a button in an application’s toolbar. An action sheet is an appropriate way to:
用户点击应用工具条上的一个按钮而显示的与该任务相关的选项组合，被称为动作表单。动作表单适合于下面：

■ Provide a selection of ways the task can be completed. In Photos, for example, users can tap the Send button when viewing an individual photo. An action sheet appears, giving users a choice of three destinations for the photo (in addition to a Cancel button, which cancels the send). It’ s useful to display an action sheet in a situation like this, because it allows you to provide a range of choices that make sense in the context of the current task, without giving these choices a permanent place in the user interface.

提供多条选择性的方法去完成任务。例如在 Photos 中 ,使用者点击发送按钮来发送正在查看的个人相片，然后一个动作表单会出现给了用户三个操作选择（（除了取消发送的按钮 ）。在这样的情况下动作表单将会非常的有用。因为它针对当前的任务情景提供了一系列选项，而不需要在用户界面上为各种选项设置固定的位置。

■ Get confirmation before completing a potentially dangerous task. For example, depending on Mail settings, an action sheet appears when users tap the Trash button in the Mail toolbar, allowing them to proceed with the deletion or cancel it. When you display an action sheet in a situation like this you ensure that users understand the dangerous effects of the step they’ re about to take and you can provide some alternatives. This type of communication is particularly important on iPhone OS–based devices because sometimes users tap controls without meaning to.

在完成一个有潜在危险任务前先获得确认。例如 ,在邮件的设置里 ,当使用者点击再邮件工具条中的 Trash button（ 删除邮件按钮 ）会出现一个动作表单，它允许使用者进行删除或者取消删除。当提供一个动作表单的时候，如上述情况可确保使用者能够理解他们即将进行的操作的危险性，并给予用户一些其它选择。这样的交流方式在装有 iPhone OS 的设备里显得尤为重要，因为有时候使用者会不小心点击控制按钮。

An action sheet always emerges from the bottom of the application screen and hovers over its views (as shown in the far left of Figure 7-1 (page 79)). Unlike an alert, however, the side edges of an action sheet are anchored to the sides of the screen, reinforcing its connection to the application and the user’ s most recent action.

一个动作表单总是从一个应用程序屏幕的底部弹出，然后覆盖在视图上。（ 如图 7-1 的左边部分 ）。和提醒不一样的是，动作表单的两端被固定在屏幕的边缘，加强它和应用程序，和使用者最近的动作之间的联系。

An action sheet contains a few buttons that allow users to choose how to complete their task. You should not have to add a message to an action sheet because the button labels, in conjunction with the task being performed, should provide enough context for the user to understand their choices. When users tap a button, the action sheet disappears. Because an action sheet should provide users with a choice of actions, an action sheet always provides more than one button.

一个动作表单包含一些任务按钮以允许使用者选择如何完成他们的任务。你没必要添加一条信息给动作表单，因为按钮标签对进行中的任务提供了足够的内容以帮助用户选择。当使用者点击一个按钮，动作

表单就消失了。因为动作表单应该提供给用户一系列动作的选择，而不仅仅是一个按钮。

Using Modal Views 使用模态视图

By default, a modal view slides up from the bottom edge of the screen and always covers the entire application screen (as shown in the middle of Figure 7-1 (page 79)). Because a modal view hides the current application screen, it strengthens the user’ s perception of entering a different, transient mode in which they can accomplish something.
默认情况下，模态视图从屏幕的底部边缘划上来，而且会涵盖整个应用程序屏幕（如图 7-1 中间所示）。由于一个模式视图隐藏了当前应用程序屏幕，它加强了用户进入一个不同的短暂模式的认知体验，使他们了解可以做些什么了。

A modal view can display text if appropriate, and contains the controls necessary to perform the task. In addition, a modal view generally displays a button that completes the task and dismisses the view, and a Cancel button users can tap to abandon the task.
模态视图在适当情况下可以显示文本，并包含执行任务必要的控件。此外，一个模态视图通常显示一个完成任务和退出视图的按钮，以及一个用户可以点击取消放弃任务的按钮。

A modal view supports more extensive user interaction than an action sheet. Unlike an action sheet, which accepts a single choice, a modal view supports multistep user interaction, such as the selection of more than one option or the inputting of information.
一个模态视图比一个动作表单支持更广泛的用户交互。不同于动作表单，只是接受一个简单的选择，一个模态视图支持多步骤的用户交互，如可以选择多个选项或输入信息。

Use a modal view when you need to offer the ability to accomplish a self-contained task related to your application’ s primary function. A modal view is especially appropriate for a multistep subtask that requires user interface elements that don’ t belong in the main application user interface all the time. A good example of a modal view is the compose view in Mail. When users tap the Compose button, a modal view appears that contains text areas for the addresses and message, a keyboard for input, a Cancel, and a Send button.
当您认为需要提供一个完成与应用程序的主要功能相关的独立任务功能（a self-contained task）功能时，可以使用一个模态视图。一个模态视图特别适用于并不属于应用程序主要用户界面的情况，但需要用户界面元素的多步骤子任务。一个很好的模态视图例子是邮件撰写视图。当用户点击撰写按钮，会出现一个模态视图，它包含文本区域，包括地址和内容信息，虚拟键盘，“取消”和“发送”按钮。

Designing an Alert 设计提醒

You can specify the text, the number of buttons, and the button contents in an alert. You can’

t customize the width or the background appearance of the alert view itself, or the alignment of the text (it’ s center-aligned).
可以指定文本、按钮的数量、在提醒中按钮的内容。您不能自定义提醒视图的宽度、背景色，以及文本对齐方式。

Note: As you read these guidelines, be aware of the following definitions:
注意：当你阅读这些指南，应注意以下定义：
■ Title-style capitalization means that every word is capitalized, except articles, coordinating conjunctions, and prepositions of four or fewer letters.
标题式大写是指除了冠词、连词及介词（四个或更少字母组成的），其他每个词是大写的。
■ Sentence-style capitalization means that the first word is capitalized, and the rest of the words are lowercase, unless they are proper nouns or proper adjectives.
句式的大写是指第一个单词大写，而其余的都是小写，除非是专有名词或专属形容词。

The alert title (and optional message) should succinctly describe the situation and explain what people can do about it. Ideally, the text you write gives people enough context to understand why the alert has appeared and to decide which button to tap.
提醒标题（和消息）应该简明扼要地说明情况，解释用户可以做什么。理想情况下，你写的文字应足够使用户理解为什么出现提醒，并决定按哪个按钮。

As you compose the required alert title:
当你撰写所需要的提醒标题时：

■ Keep the title short enough to display on a single line, if possible. A long alert title is difficult for people to read quickly, and it might force the alert message to scroll.
如果可能，保持标题足够短，以能在一行呈现。长标题难以让人们快速阅读，并可能迫使提醒消息需滚动查看。

■ Avoid single-word titles that don’t provide any useful information, such as “Error” or “Warning.”

避免一个字的标题，它不能提供任何有用的信息，如“错误”或“警告”。

■ Prefer a sentence fragment. A short, informative statement is often easier to understand than a complete sentence.

宁愿截取一个句子的片段。一个简短、内容丰富的阐述往往比一个完整的句子更容易。

■ Don’t hesitate to be negative. People understand that most alerts tell them about problems or warn them about dangerous situations. It’s better to be negative and direct than it is to be positive but oblique.

无需避免负面消息。人们知道大部分提醒告诉他们问题或警告他们有危险情况。直接了当的负面告知比隐晦不清的正面告知要好得多。

■ Avoid using “you,” “your,” and “me” as much as possible. Sometimes, text that identifies people directly can be ambiguous and can even be interpreted as an insult.

尽可能避免使用“你”、“你的”、和“我” 。有时候，直接标识人称的文字可能会表意不明确，甚至可能被理解为是一种侮辱。

■ Use title-style capitalization and no ending punctuation when:

使用标题样式的大写并不要带有结束标点，当：

/ The title is a sentence fragment 标题是一个句子片段

/ The title consists of a single sentence that is not a question

标题由一个不是问句的简单的句子组成

■ Use sentence-style capitalization and an ending question mark if the title consists of a

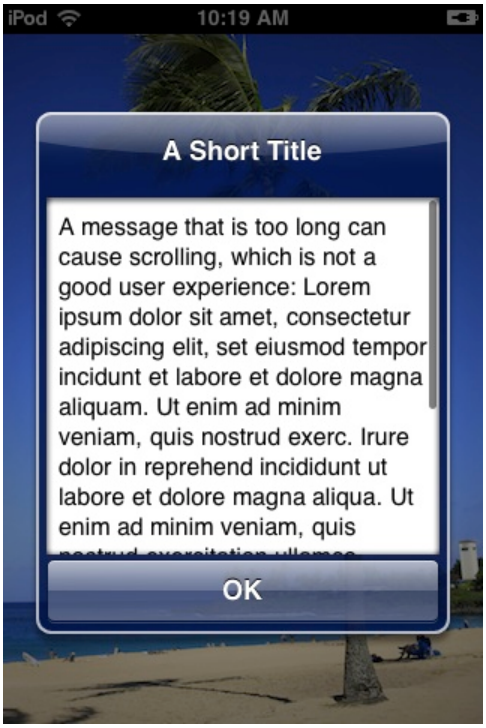
single sentence that is a question. In general, consider using a question for an alert title if it allows you to avoid adding a message.
如果标题的一句话是由一个问题组成，用句子式的大写和结束问号。在一般情况下，如果可以让你避免增加一条消息，那可以考虑使用问题作为提醒标题。

■ Use sentence-style capitalization and appropriate ending punctuation for each sentence if the title
consists of two or more sentences. A two-sentence alert title should seldom be necessary, although you might consider it if it allows you to avoid adding a message.
如果标题由两个或更多的句子组成，用句子式的大写和适当的终止标点符号。应尽量避免一个两句话提醒的标题，即使它可以让你避免增加一条消息从而你可能会考虑使用它 。

If you provide an optional alert message:
如果您提供一个可选的提醒消息：

■ Keep the message short enough to display on one or two lines, if possible. If the message is too long, it will scroll.
如果可能的话，保持信息足够短显示在一或两行。如果消息太长，将滚动。

■ Always use sentence-style capitalization and appropriate ending punctuation. The alert message should be a complete sentence.
始终使用句子式的大写和适当的结束标点。提醒的消息应是一个完整的句子。



Avoid lengthening your alert text with descriptions of which button to tap, such as "Tap View to see the information." Ideally, the combination of unambiguous alert text and logical button labels gives people enough information to understand the situation and their choices. However, if you must provide detailed guidance, follow these guidelines:

避免因为描述点击哪个按钮而使文本加长， 如"点击'视图'查看信息 "。理想情况下，通过表意明确的提醒文字和有逻辑的按钮标签这样的组合，能给人足够的信息了解情况，并作出他们的选择。但是，如果你必须提供详细指导，遵循以下原则：

■ Be sure to use the word "tap" (not "touch" or "click" or "choose") to describe the selection action.

一定要使用"点击"（不是"触摸"或"敲击"或"选择"）来形容选择的动作。

■ Don't enclose a button title in quotes, but do preserve its capitalization.

按钮标题不要用引号圈起来，但确保文字大写。

Be sure to test the appearance of your alert in both orientations. Because the height of an alert is constrained in landscape, it might look different than it does in portrait. It's recommended that you optimize the length of the alert text so that it looks good (and avoids scrolling) in both orientations.

一定要测试提醒设计在两个方向的外观。由于提醒高度在横向上有约束，它看起来比竖直画面中较不一样。建议优化提醒文本的长度，以便它在这两个方向看起来都不错（避免滚动）。

Prefer a two-button alert. A two-button alert is often the most useful, because it is easiest for people to choose between two alternatives. It is rarely a good idea to display an alert with a single button because such an alert is merely informative; it does not give people any control over the situation. An alert that contains three or more buttons is significantly more complex than a two-button alert, and should be avoided if possible. In fact, if you find that you need to offer people more than two choices, you should consider using an action sheet instead (see "Using Action Sheets" _self - 81(page 81) and "Designing an Action Sheet" _self - 84(page 84) for more information on this type of view).
推荐带有两个按钮的提醒视图。两个按钮的提醒往往是最有用的，因为它是人们最容易二选一。很少显示只有单一按钮的提醒，因为它仅仅是提醒信息，它不能让用户对当前状态施加任何控制。一个提醒包含三个或多个按钮明显比两个按钮的提醒更复杂，如果可能的话应避免。事实上，如果您发现您需要为人们提供两个以上的选择，你应该考虑使用动作表单（见"Using Action Sheets"和"Designing an Action Sheet"获取更多有关信息）。

Use alert button colors appropriately. Alert buttons are colored either dark or light. In an alert with two buttons, the button on the left is always dark-colored and the button on the right is always light-colored. In a one-button alert, the button is always light-colored.
适当地使用提醒按钮的颜色。提醒按钮的颜色或深或浅。在一个有两个按钮的提醒里 ，左边的按钮总是深色，右边的按钮始终浅色。在 一键式提醒里，按钮总是浅色。

- In a two-button alert that proposes a potentially risky action, the button that cancels the action should be on the right (and light-colored).
在有两个按钮的，发出一个潜在的危险动作警示的提醒里，"取消"按钮应放在右边（浅色）。

- In a two-button alert that proposes a benign action that people are likely to want, the button that cancels the action should be on the left (and dark-colored).
在有两个按钮的，发出一种人们想要的良性动作的提醒里，"取消"按钮应在左边（深色）。

Note: A Cancel button may be either light-colored or dark-colored and it may be on the right or the left, depending on whether the alternate choice is destructive. Be sure to properly identify which button is the Cancel button in your code (for more information, see UIAlertView Class Reference).
注意："取消"按钮可以是浅色或深色，可能是右侧或左侧， 这取决于另一个选项是否是有破坏性。在程序代码中一定要正确识别哪个是"取消"按钮 （有关按钮的详细信息，见 UIAlertView Class Reference）。

Give alert buttons short, logical titles. The best titles consist of one or two words that make sense in the context of the alert text. Follow these guidelines as you create titles for alert buttons:

给予提醒按钮简短而有逻辑的标题。最好的标题由一个或两个重要的词组成。创建提醒按钮的标题应遵循以下准则：

- As with all button titles, use title-style capitalization and no ending punctuation.

与所有按钮的标题一样，使用大写的标题样式并没有结束标点。

- Prefer verbs and verb phrases, such as “Cancel,” “Allow,” “Reply,” or “Ignore” that relate directly to the alert text.

优先用动词和动词短语，如“取消”、“允许”、“回复”或“忽略”等与提醒内容直接相关的词。

- Prefer “OK” for a simple acceptance option if there is no better alternative. Avoid using “Yes” or “No.”

如果没有更好的选择，优先用“OK”接受一个简单的选择。避免使用“Yes”或“No” 。

- Avoid “you,” “your,” and “me” as much as possible. Button titles that use these words are often both ambiguous and patronizing.

尽可能避免“你”、“你的”和“我” 。使用这些文字的按钮标题，往往表意含糊不清。

Designing an Action Sheet

设计动作表单

You choose the background of an action sheet to coordinate with the look of your application, and you can specify the number of buttons and their contents.

您选择动作表单的背景以配合应用程序的外观，并可以指定按钮数目和它们的内容。

Unlike an alert, an action sheet should not need to display a textual message. This is because an action sheet appears as the result of a user action, such as tapping a Delete or Send button, so there should be no need to explain its arrival.

和提醒不同，动作表单不应该显示一个文本消息。这是因为动作表单是作为一个用户操作的结果，如点击“删除”或“发送”按钮，没有必要解释该提醒的到来。

Action sheets can have two different background appearances. You need to ensure that the background of the action sheets in your application coordinates with the appearance of your application’s toolbars or navigation bars. If your application uses black navigation bars and toolbars, for example, the action sheet background should be translucent black. By default, iPhone OS displays action sheets with a standard blue background, which coordinates with the standard blue toolbars and navigation bars. All action sheets in your application should have the same background color, and that color should coordinate with the color of the navigation bars and toolbars.

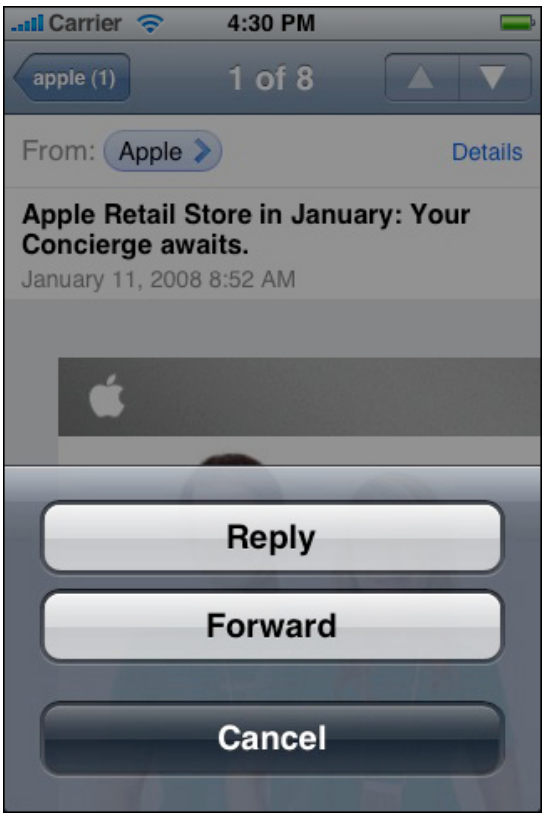
动作表可以有两种不同的背景。您需要确保应用程序中的动作表单背景与您的应用程序的工具条或导航条的外观相配。例如，如果您的应用程序使用黑色导航条和工具条，动作表单背景应该是半透明的黑色。默认情况下，iPhone 操作系统显示一个标准的蓝色背景的动作表单，这与标准蓝色的工具条和导航条相

配。你的所有动作表单应用程序应具有相同的背景颜色，而且颜色要协调于导航条和工具条颜色。

Be sure to display the Cancel button at the bottom of an action sheet. This encourages the user to read through all the alternatives before reaching the Cancel option.
一定要在一个动作表单底部显示“取消”按钮。这鼓励用户看过所有的选项后才看到“取消”。

Figure 7-2 shows an action sheet with the default background appearance and a Cancel button in the recommended location.
显示了一个动作表单，有默认背景和放在推荐位置的“取消”按钮。

Figure 7-2 A typical action sheet
一个典型的动作表单



If you need to provide a button that performs a potentially destructive action, such as deleting all the items in a user’ s shopping list, you should use the red button color. It’ s important to display such destructive buttons at the top of the action sheet for two reasons:
如果您需要提供一个按钮，来执行具有潜在破坏性的动作，诸如删除所有用户购物清单上的项目，你应该使用红色按钮。 这一点很重要，把这种具破坏性的按钮显示在动作表单顶部的原因有两个：

- The closer to the top of the action sheet a button is, the more visible it is.
按钮越接近动作表单顶部，就越明显。

■ Sometimes users mistakenly tap the bottom of the device screen when they’ re aiming for the Home button. By placing a destructive button away from the bottom of an action sheet, users are less likely to cause undesirable results if they mistakenly tap the screen instead of the Home button.

有时，用户想按“主界面”按钮，却误按了设备屏幕的底部。通过把具破坏性的按钮远离动作表单底部的放置，降低用户错误地点击到屏幕而不是“主界面”按钮时，所能造成的不良后果。

Figure 7-3 shows an action sheet with the translucent black background appearance and both a Cancel and a destructive button in their recommended positions.

图 7-3 显示了一个半透明黑色背景外观的动作表单，在他们建议的地方都放置了一个“取消”按钮和一个具破坏性的按钮。

Figure 7-3

A button that performs a destructive action should be red and located at the top of the action sheet

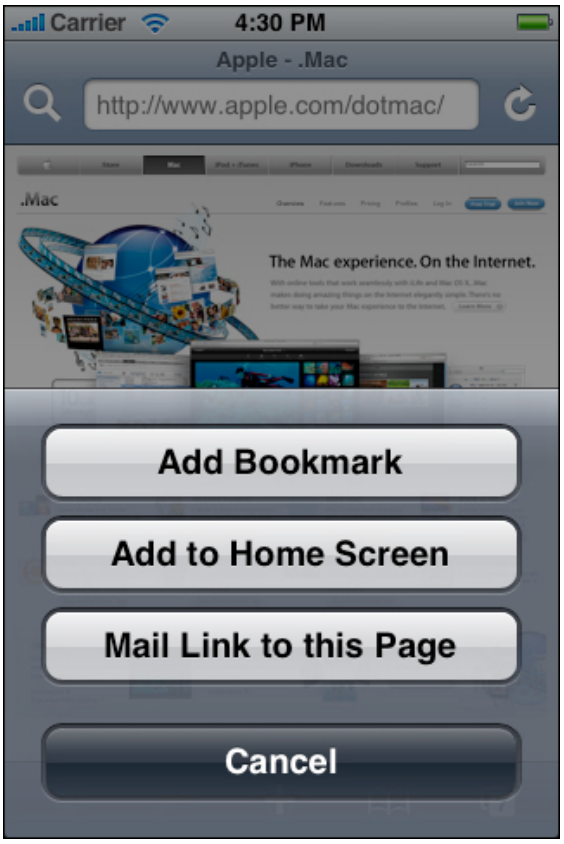
一个执行具破坏性动作的按钮应该是红色，并位于动作表单的顶部。

You can display several buttons in an action sheet, as long as you make sure each button is easily distinguished from the others. Figure 7-4 shows an action sheet with a background that matches the standard blue toolbar and that provides three alternatives in addition to Cancel.

您可以在一个动作表单中显示多个按钮，只要确保每个按钮都能很容易地区别于其他按钮。 图 7-4 显示了一个符合标准蓝色的工具条的动作表单，并提供了除”取消“以外的三个选择。

Figure 7-4 An action sheet with four buttons

有四个按钮选项的动作表单



Designing a Modal View

设计模态视图

The overall look of a modal view should coordinate with the application that displays it. For example, a modal view often includes a navigation bar that contains a title and buttons that cancel or complete the modal view’ s task. The navigation bar should have the same background appearance as the navigation bar in the application.

模态视图的整体效果应该和运用程序相协调。例如，一个模态视图视图通常包含一个导航条，其中有标题，取消或者完成模态视图的按钮。模态视图的导航条应与应用程序的导航条有相同的背。

A modal view should usually display a title that identifies the task in some way. If appropriate, you can also display text in other areas of the view that more fully describes the task or provides some guidance.

模态视图通常要有一个比较容易识别的标题。如果适合，你可以在视图的其它区域显示文本信息，以更

充分描述或提供导引。

For example, the Messages application provides a modal view when users want to compose a text message. This modal view, shown in Figure 7-5, displays a navigation bar with the same background as the application navigation bar and with the title New Message.
例如，当用户要撰写的短信时，短信程序提供一个模态视图。这个模态视图如图 7-5 所示，显示一个与应用程序相同的背景导航条及新短信标题

Figure 7-5 A modal view should coordinate with the application screen
模式的视图应该与应用程序屏幕相适应

In a modal view, you can use whichever controls are required to accomplish the task. For example, you can include text fields, buttons, and table views.
在一个模态视图中，你可以使用任何控件来完成任务，例如，文本方面，按钮，表格视图。

You can choose to reveal a modal view in a way that coordinates with your application and enhances the user’ s awareness of the temporary context shift the view represents. To do this, you can specify one of the following transition styles:
你可以选择一个模态视图，与应用程序相协调，而且增强了用户对当前视图状态变化的认识。要做到这一点，你可以采用下列过度的样式：

■ Vertical. The modal view slides up from the bottom edge of the screen and slides back down when dismissed. (This is the default transition style.)

垂直。模态视图从屏幕底部向上滑入；消失时则向下滑出。（这是默认过渡样式）

■ Flip. The current view flips horizontally from right to left to reveal the modal view. Visually, the modal view looks as if it is the back of the current view. When the modal view is dismissed, it flips horizontally from left to right, revealing the previous view.

翻转。当前视图由右至左的水平翻转以显示模态视图。视觉上，模态视图看上去好像它是当前视图的背面。 当模态视图消失，它水平地由左到右翻转，显示之前的视图。

If you decide to vary the transition styles of the modal views in your application, avoid doing so merely for the sake of variety. Be aware that users notice such differences and will assume that they mean something. For this reason, it’ s best to establish a logical, consistent pattern that users can easily detect and remember, and avoid changing transition styles gratuitously.

如果您决定更改应用程序中的模态视图过渡样式，避免这样做只是为了获得不同显示样式。请注意，用户注意到这样的区别，并认为这里有所含义。基于这个原因，最好建立一个逻辑的、一贯的模式，用户可以轻松地识别和记忆，避免过度地转变风格样式。

CHAPTER 8

Table Views, Text Views, and Web Views

表格视图，文本视图，及网页视图

Table views, text views, and web views are versatile elements that lend themselves to different uses in your iPhone application. For example, table views can be configured to display short lists of choices, grouped lists of detailed information, or long, indexed lists of items. Text views and web views are relatively unconstrained containers you can use to accept and display content.

表格视图，文本视图，和网页视图使你的iPhone应用程序有了不同用途的多种功能元素。例如，表格视图可以配置为显示短清单的选择，分组的详细信息清单，或长索引的项目清单。文本视图和网页视图是比较不受约束的容器，你可以使用它来接受和显示内容。

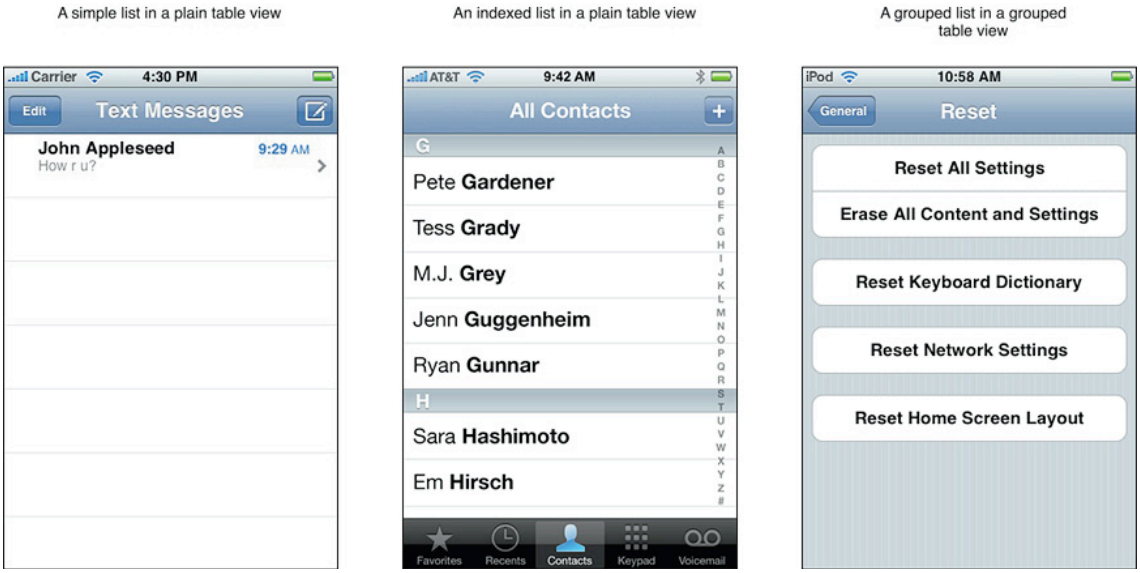
Table Views 表格视图

A table view presents data in a single-column list of multiple rows. Rows can be divided into sections or groups and each row can contain some combination of text, images, and controls. Users flick or drag to scroll through rows or groups of rows. Figure 8-1 shows how different styles of table views can display lists in different ways.

一个表格视图通过多行单列的表格显示数据。行可分为节或组，每行可以包含一些文本，图像和控件的组合。用户轻击或拖动来滚动行或行组。图8-1不同的表格视图样式可以多种不同的方式显示列表。

Figure 8-1 Three ways to display lists using table views

三种不同表格视图的显示方式



Usage and Behavior
用法和行为

Table views are extremely useful in iPhone applications because they provide attractive ways to organize both large and small amounts of information. Table views are most useful in productivity applications that tend to handle lots of user items, although utility applications can make use of smaller-scale table views, as well. An immersive application would probably not use a table view to display information, but it might use one to display a short list of options.
表格视图对iPhone应用程序极为有用，它提供许多组织各种大小类型信息的有吸引力的方法。尽管实用工具程序也可以利用规模较小的表格视图，表格视图对倾向于处理大量的用户项目的生产力辅助程序最有用。沉浸式应用程序一般不会使用表格视图来显示信息，但它可能使用它来显示一个简短的选项列表。

Table views provide built-in elements that allow users to navigate and manipulate information. In addition, table views support:
表格视图提供了允许用户浏览和处理的信息的内置元素。此外，表格视图还支持：

■ The display of header and footer information. You can display descriptive text above or below each section or group in a list, and above or below the list as a whole.
页眉和页脚的信息显示。你可以将说明性文字显示在每一部分或列表组的上方或下方，和高于或低于整个列表的地方。

■ List editing. You can allow users to add, remove, and reorder list items in a consistent way. Table views also support the selection and manipulation of multiple list items, which you might use to give users a convenient way to delete more than one list item at a time.
列表编辑。你可以允许用户以统一的方式来添加，删除，重新排列。表格视图也支持选择和操做多种列表项，这样你就可以给用户一个方便的方法一次删除多个列表项。

A table should always provide feedback when users select a list item. When an item can be selected, the row containing the item highlights briefly when the user selects it, providing feedback that the selection has been received. Then, an immediate action occurs: Either a new view is revealed or the row displays a checkmark to indicate that the item has been selected or enabled.
当用户选择一个列表项时，表格视图应始终提供反馈信息。当一个项目可以被选择，该项目所在的行高亮显示，并提供反馈表示选择已收到。然后，立即行动发生：要么是一个新的视图显示，要么该行显示一个标记，表明该项目已被选定或启用。

In rare cases, a row might remain highlighted when secondary details or controls related to the row item are displayed in the same screen. However, this is not encouraged because it is

difficult to display a list of choices, a selected item, and related details or controls without creating an uncomfortably crowded layout.
在极少数情况下，当相关的项目当次要的细节或控件是在同一个屏幕上显示，行可能仍然高亮显示。然而，这是不鼓励的，因为它难以将选择列表，选定的项目，以及相关的细节或控制项，统统显示在一个版面内，而不令人感到不安与拥挤。

If a row selection results in navigation to a new screen, the selected row highlights briefly as the new screen slides into place. When the user navigates back to the previous screen, the originally selected row again highlights briefly to remind the user of their earlier selection.
如果行选择的结果是到一个新的屏幕，选择行高亮显示，然后新屏幕滑入。当用户要回到前一个屏幕，原来选定的行再次高亮显示提醒他们早先的用户选择。

Note that you can also animate the changes users make to list items. Doing so is a good way to provide feedback and strengthen the user’ s sense of direct manipulation. In Settings, for example, when you turn off the automatic date and time setting (by selecting Off in Date & Time > Set Automatically), the list group expands smoothly to display two new items, Time Zone and Set Date & Time.
请注意，你也可以在用户使用列表项时使用动画效果。这样做是一个好的方法来提供反馈和加强用户对直接操纵的感觉。例如，当您关闭自动选择日期及时间的设置时(通过选择“日期与时间”>关闭“自动设置”)，列表组平滑展开以显示两个新项目，“时区”和“设置日期和时间”。

A table should display content immediately. If the table's content is extensive or complex, avoid waiting until all the data is available before displaying anything. Instead, fill the onscreen rows with textual data immediately and display more complex data (such as images) as they become available. This technique gives users useful information right away and increases the perceived responsiveness of your application.
表格视图应立即显示内容。如果该表的内容是广泛或复杂的，避免直到所有的数据加载完成之前不显示任何东西。相反，填充屏幕上的可见行文本数据应立即显示，更复杂的数据（如图像）可以等到就绪再显示。这种技术及时地为用户提供了有用信息，并提高了用户对于应用程序反应速度的感觉。

If your application displays data that changes infrequently, you might consider displaying “stale” data while waiting for new data to become available. This technique also allows users to see something useful right away, but it is not recommended for applications that handle data that changes frequently. Before you decide to do this, gauge how often the data changes and how much users depend on seeing fresh data quickly
如果应用程序显示的数据变化不频繁，你可以考虑在等待新的数据加载时，先显示“旧”数据。这种技术还允许用户马上看到一些有用的东西，但它不被建议放在处理数据频繁更改的应用程序上。在你决定这样做之前，估计仪表数据变化的频率和多少用户想要马上看到新的数据。

If it’ s difficult to display anything useful right away, it's important to avoid displaying empty rows, because this can imply that the application has stalled. Instead, the table should display a spinning activity indicator along with an informative label, such as “Loading...”, centered in the screen. If you can display older data, you don’ t have to worry about blank rows, but you should update onscreen rows as soon as possible. Both techniques provide feedback to users, letting them know that processing is continuing.
如果难以马上显示任何有用的信息，避免显示空行是重要的，因为这可以意味着该应用程序宕机了。相反，应在屏幕中心显示一个旋转的活动指标及标签信息，如“加载中...”。如果你能显示较旧的数据，你不必担心空白行，但你应该尽快更新屏幕上的数据行。这两种技术提供反馈给用户，让他们知道，处理仍在继续。

Table-View Styles
表格视图样式

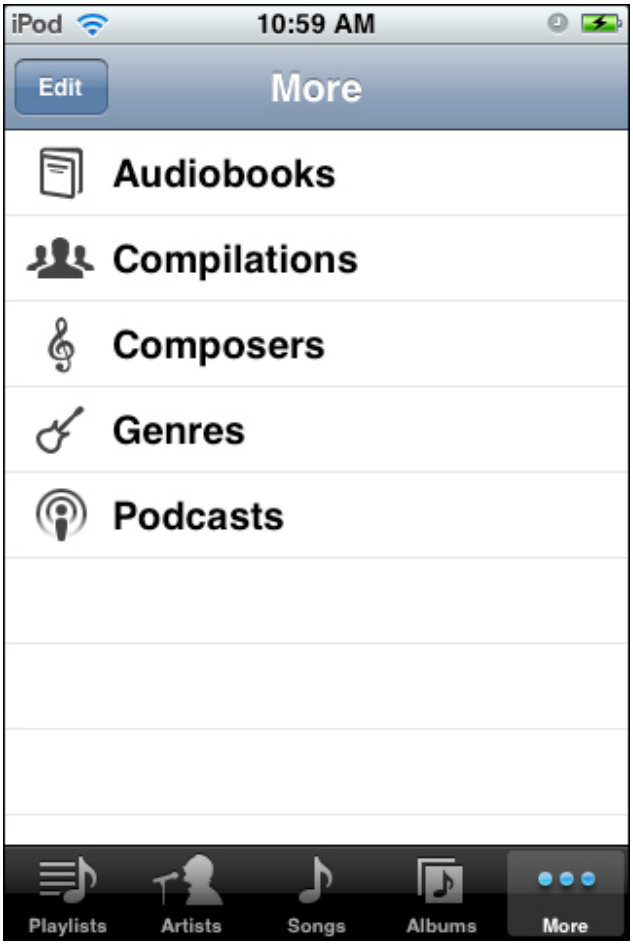
---

iPhone OS defines two styles of table views, which are distinguished mainly by appearance:
iPhone 操作系统定义了两种表格视图的样式，他们的主要区别在于外观样式：

Plain (UITableViewStylePlain). This table-view style displays rows that extend from side edge to side edge of the screen. The background of the rows is white. The rows can be separated into labeled sections and the table view can display an optional index that appears vertically along the right edge of the view.
普通。此表格视图样式显示的行从侧面边缘延伸到另一侧屏幕的边缘。该表格行背景为白色。该表格行可分成多个带标签的节，也可以在表格视图垂直右边缘显示一个可选索引。

Figure 8-2 shows a list in a plain table (without headers, footers, or an index) in the iPod application.
显示了一个普通的表格（没有页眉，页脚，或索引）在iPod中的应用。

Figure 8-2 A simple list in a plain table
普通表格中的简单列表



Grouped (UITableViewStyleGrouped). This table-view style displays groups of rows that are inset from the side edges of the screen. The groups are displayed on a distinctive vertically striped background, while inside the groups the background is white. A grouped table can contain an arbitrary number of groups, and each group can contain an arbitrary number of rows. Each group can be preceded by header text and followed by footer text. This style of table view does not provide an index.

分组。此表格视图样式显示的是从屏幕两侧边缘缩进的镶嵌行分组。这些分组显示在一个独特的垂直条纹背景中，分组内部的背景是白色的。带分组的表格可以包含任意数量的组。可在每个分组之前添加页眉文本，之后添加页脚文本。这表格视图样式不提供索引。

Figure 8-3 shows a list in a grouped table, in which each group contains one row. This list, in the Settings application, does not include header or footer text.

显示了分组表格的一个列表，其中每个分组包含一个表格行。在设置应用程序中这个列表不包括页眉或页脚文本。

Figure 8-3 A list of four groups in a grouped table

一个包含四个组列表的分组表格

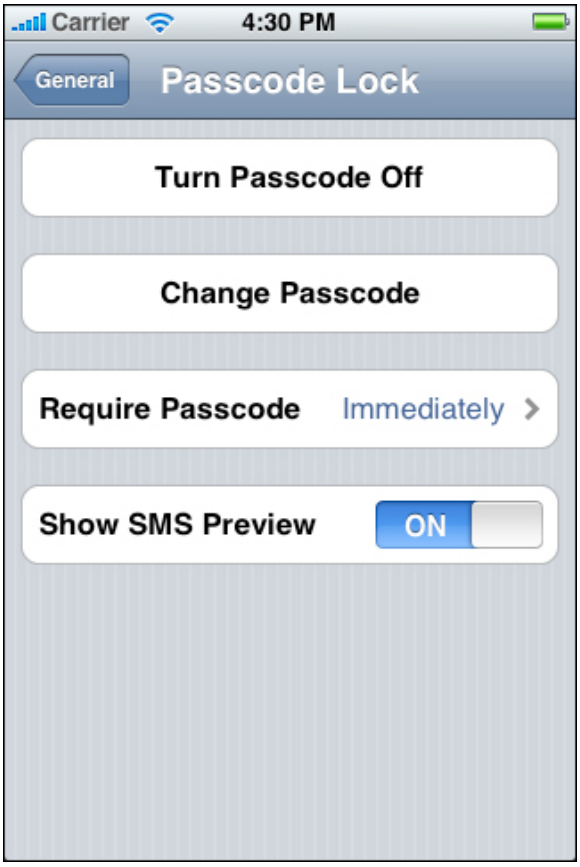


Table-Cell Styles

表格单元样式

---

iPhone OS 3.0 and later includes four predefined table-cell styles you can use to quickly and easily produce the most common layouts for table rows in both plain and grouped tables. Note that, programmatically, these styles are applied to a table view’ s cell, which is an object that tells the table how to draw its rows.

iPhone操作系统3.0及后续版本包括四个预定义的表格单元样式，您可以快速，方便地在所有的普通和分组表格里产生最常见的表格行版面。请注意，从程序上讲，这些样式应用到表格视图的单元里，它是一个对象用来告诉表格如何绘制其行。

When you use the standard table-cell styles, your application is consistent with the built-in applications, which benefits you in a couple of ways:

当你使用标准的表格单元样式，你的应用程序与内置应用程序是协调的，其好处是：

- Users more quickly understand how your application works

用户更快地了解你的应用程序是怎样工作的。

- Your application remains consistent without a lot of extra work on your part, if the standard table-cell styles are enhanced in the future

如果标准表格单元样式是在以后会被修改或加强，你无需花费大量额外工作来维护应用程序的一致性。

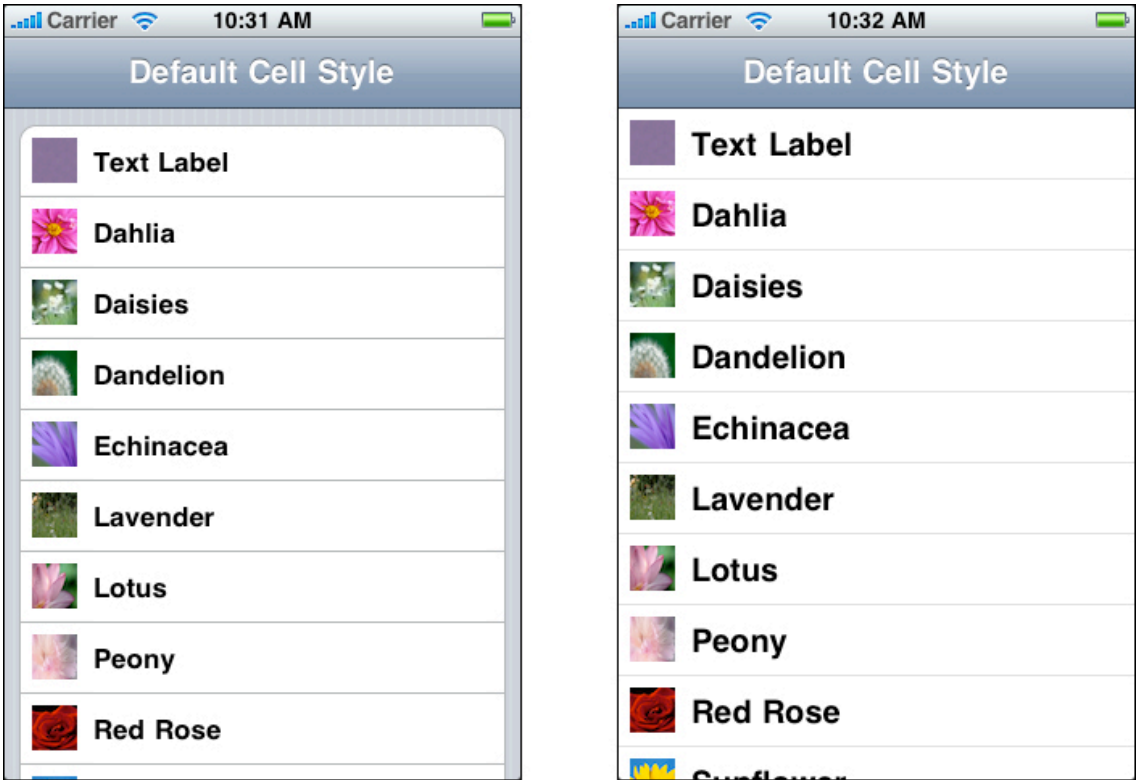
If you want to lay out your table rows in a nonstandard way, it’ s better to create a custom table-cell style than to significantly alter a standard one. “Customizing Cells” in Table View Programming Guide for iPhone OS helps you learn how to create your own cells.
如果你想展示一个非标准的表格方式，最好创造一个专门的自定义表格单元样式，而不是改变一个标准的表格单元样式。“Customizing Cells”在Table View Programming Guide for iPhone OS helps可帮助你了解如何创建自己的单元格。

Be aware that text truncation is automatic in all table-cell styles. Generally speaking, you should ensure that your text is as succinct as possible to avoid displaying truncated words or phrases that are difficult for users to understand. Specifically, text truncation can be more or less of a problem, depending on which cell style you use and on where truncation occurs.
请注意，在所有表格单元样式中，文本都会被自动截断。一般来说，你应该确保你的文字尽可能简洁，避免显示为用户很难理解的被截断的单词或词组。具体来说，文字截断是可大或可小的问题，这取决于哪个单元格样式的使用和在何处发生截断。

iPhone OS provides the following standard table-cell styles:
iPhone 操作系统提供以下标准表格单元样式：

- The default table-cell style (UITableViewCellStyleDefault) includes an optional image on the left, followed by a left-aligned text label in black.
默认的表格单元样式，包括左侧的可选图像，和左对齐的黑色字体文本标签。

Figure 8-4 The default table-cell style in a grouped table (left) and a plain table (right)
带分组的表格（左）和普通表格（右）的默认表格单元样式



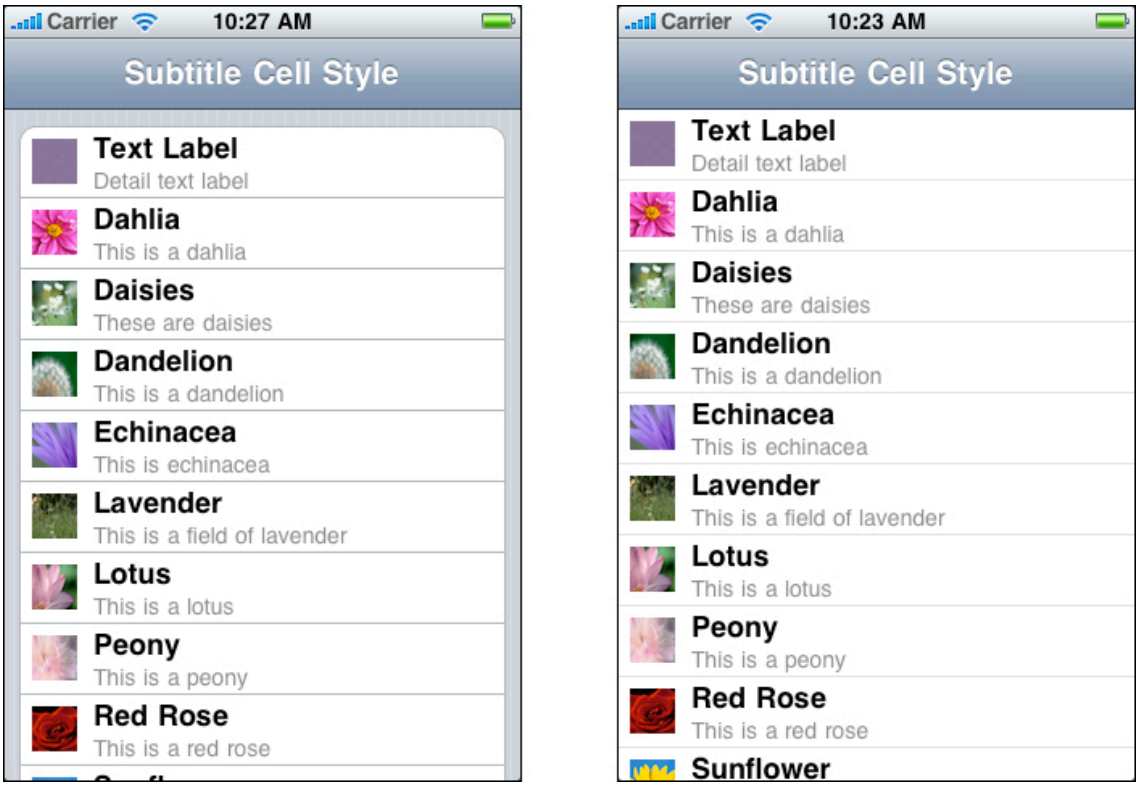
The text label’ s appearance implies that it represents an item name or title and its left-alignment makes the list easy to scan. This makes the default style good for displaying a list of items that do not need to be differentiated by supplementary information.
文本标签的外观意味着它代表一个项目的名称或者标题，左对齐使得列表易于扫视。这使得以默认的样式显示的项目不须补充信息来区别。

Short text labels are best, but if truncation is unavoidable, try to ensure that the most important information is contained in the first few words.
简短的文字标签是最好的，但如果截断是不可避免的，尽量确保最重要的信息是在前几个字。

■ The subtitle table-cell style (UITableViewCellStyleSubtitle) includes an optional image on the
left, followed by a left-aligned text label on one line and a left-aligned detail text label on the line below. The text label is in black and the detail text label is in a smaller, gray font.
副标题表格单元样式。包括一个左侧可选图像，左对齐的文本标签和在它下面的左对齐的详细文本标签。文本标签的字体是黑色的，细节文本标签是一个较小的，灰色的字体。

Figure 8-5 The subtitle table-cell style in a grouped table (left) and a plain table (right)

带分组的表格（左）和普通表格（右）的副标题表格单元样式



The prominent appearance of the text label implies that it represents an item name or title, while the subtle appearance of the detail text label implies that it contains subsidiary information related to the item. The left-alignment of the text labels makes the list easy to scan. This table-cell style works well when list items might look similar, because users can use the additional information in the detail text labels to help distinguish items named in the text labels.

文本标签的突出显示标识了一个项目的名称或标题，而详细的文字标签显示它包含的相关附加信息。左对齐的文本标签列表使用户易于扫视。此表格单元样式适用于列表项相似的情况，因为用户可以使用含有附加信息的详细文本标签，以帮助区分相似文本标签命名的项目。

Text labels should be short to avoid truncation. If truncation is unavoidable, focus on putting the most important information in the first few words. If the detail text label is truncated, users are not likely to mind too much because they view it as information that enhances or supplements the item named by the text label.

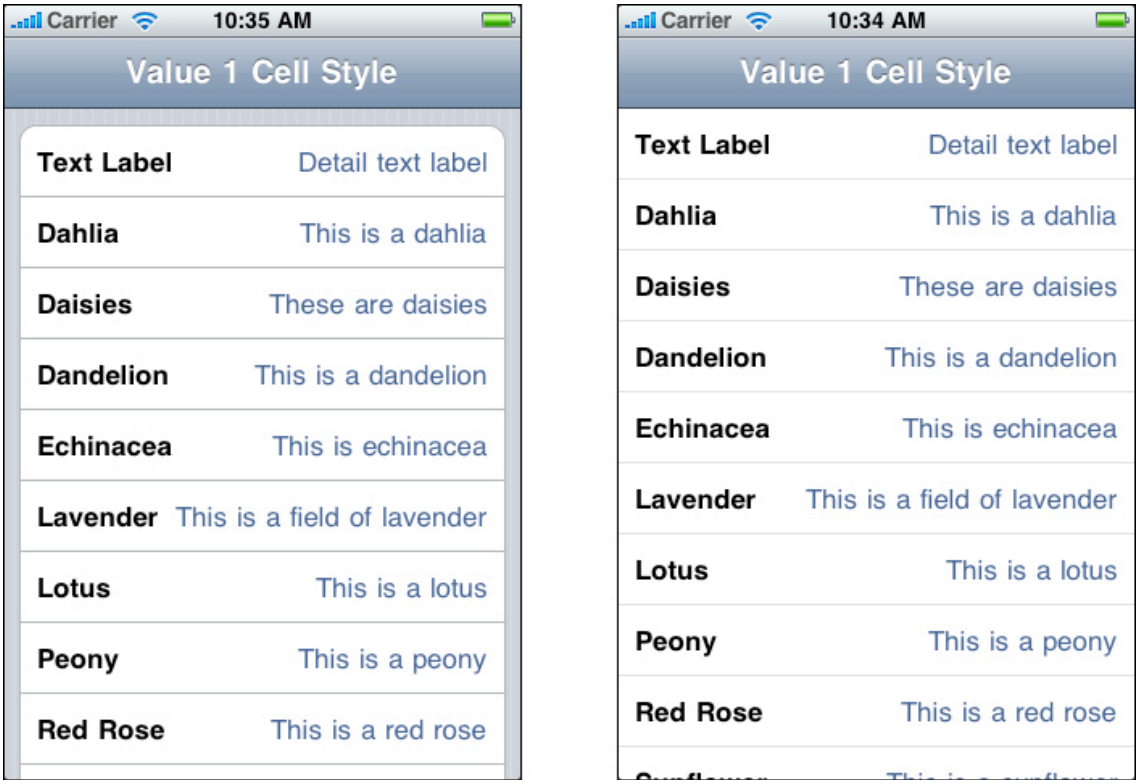
文本标签应简短，避免截断。如果截断是不可避免的，最重要的信息应放在前几个字。如果详细文本标签被截断，用户可能不会介意太多，因为他们认为这是文字标签命名项目的提高或补充资料。

■ The value 1 table-cell style (UITableViewCellStyleValue1) displays a left-aligned text label in blackon the same line with a right-aligned detail text label in a smaller, blue font. Images do not fit well in this style.

值1表格单元样式。显示为一行左对齐的黑色字体的文本标签和在同一行的右对齐的较小蓝色字体的详细文本标签。图像不适用于这种风格。

Figure 8-6 The value 1 table-cell style in a grouped table (left) and a plain table (right)

包含分组表格和普通表格的值1表格单元样式



The appearance of the text label implies that it represents an item name or title, while the appearance of the detail text label implies that it provides important information that is closely associated with the item.

文本标签的外观意味着它标识一个项目的名称或标题，而详细的文字标签的外观意味着它提供了与该项目有关的重要信息。

The left-alignment and font of the text label help users scan the list for the item they want, and the right-alignment of the detail text label draws their attention to the related information it provides. This table-cell style works well to display an item’ s current value, possibly selected from a sublist.

文本标签的左对齐和字体帮助用户扫视到他们想要的列表项，详细文本标签的右对齐提醒他们注意所提供的相关信息。此表格单元样式适用于显示一个可能选自子列表项目的当前值。

Text truncation can be difficult to avoid in this layout (because both labels are on the same line), but it’ s worth the effort. Otherwise, you lose the active space between the labels that helps users understand the relationship between the two pieces of information.

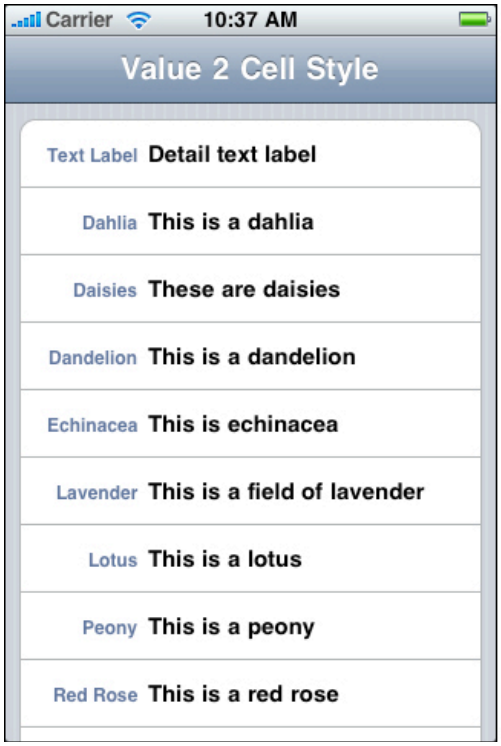
在此版面布局中，文本截断可能很难避免（因为这两个标签在同一行中），但值得努力。否则，你将失去帮助用户了解信息之间的关系的两个的标签之间的空间。

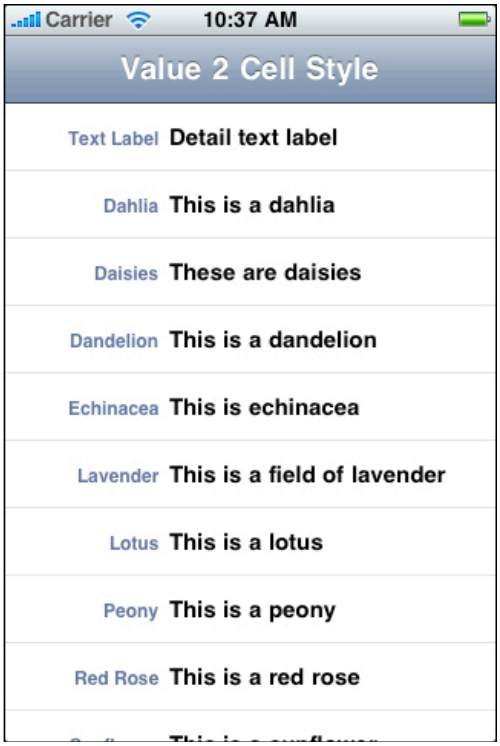
Although you can use the value 1 table-cell style in either a plain or a grouped table, its appearance is better suited to a grouped table. For example, the Usage screen in Settings uses the value 1 style in grouped tables:
虽然您可以在任何一个普通或分组表格中使用值1表格单元样式，但是其外观更适合于分组表格。例如，“设置”中“使用时间”选项屏幕的分组表格中使用了值1样式：

Figure 8-7 The value 1 table-cell style looks best in a grouped table
值1表格单元样式最适用于分组表格

■ The value 2 table-cell style (UITableViewCellStyleValue2) displays a right-aligned text label in a small blue font, followed on the same line by a left-aligned detail text label in a larger, black font. Images do not fit well in this style.
值2表格单元样式。显示为一行小的蓝色字体并右对齐的文本标签，然后在同一行上有一个左对齐的较大的黑色字体详细文本标签。图像不适用这种风格。

Figure 8-8 The value 2 table-cell style in a grouped table (left) and a plain table (right)
包含分组表格（左）和普通表格（右）的值2表格单元样式





The right-alignment, constrained width, and font of the text label imply that it functions as a heading or caption for the important information in the more prominent, left-aligned detail text label.

文本标签的右对齐，约束宽度，和字体意味着它可以标题，而左对齐的文本标签对重要信息详细说明。

In this layout, the labels are aligned towards each other at the same location in every row. This creates a crisp, vertical margin between the text labels and the detail text labels in the list, which helps users focus on the first words of the detail text label. If you allow the text labels to be truncated, you lose the sharpness of this vertical strip, which can make it harder for users to scan the information in the detail text labels.

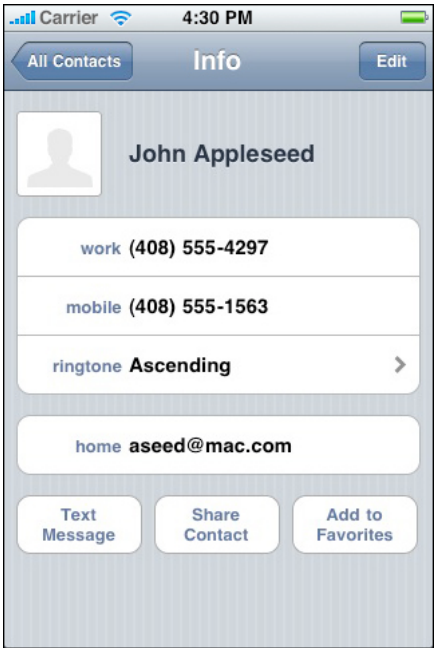
在此版面，标签对待同一地点的每一行是一致的。这将创建一个明快的风格，列表中垂直缘位于文本标签和详细文本标签之间，这可以帮助用户专注于详细文本标签的第一句话。如果你哦允许文字标签被截断，你就失去了这种垂直地带的清晰度，这使用户更难扫描详细文字标签中的信息。

Although you can use the value 2 table-cell style in either a plain or a grouped table, it looks much better in a grouped table. For example, the Info screen in Contacts uses the value 2 table-cell style in grouped tables:

虽然您可以在任何一个普通或分组表格中使用值2表格单元样式，但它更适用于分组表格中。例如，在联系人信息屏幕的分组表格中使用值2表格单元样式：

Figure 8-9 The value 2 table-cell style looks best in a grouped table

值2表格单元样式更适用于分组表格



Note: All standard table-cell styles also allow the addition of a table-view element, such as the checkmark or the disclosure indicator. Be aware that adding these elements decreases the width of the cell available for the title and subtitle.
注意：所有的标准表格单元样式都允许添加表格视图元素，如勾选标记或延展指示符。要注意，加入这些元素降低了表格单元的标题和副标题可用宽度。

You might be able to avoid text truncation by increasing the height of a table row to accommodate text wrapping, but this can be problematic:
为避免文本截断，你可能会通过增加表格行的高度来容纳文本，但这可能会产生问题：

- You have to programmatically check the text length and decide if wrapping might occur. You must make this determination for both portrait and landscape orientation, because the table width affects text wrapping.
你必须以编程方式检查文本的长度，并决定是否需要折行。你必须兼顾纵向和横向，因为表格的宽度会影响文本折行。

- You should avoid displaying wrapped text in one orientation, but not the other.
您应该避免一个方向上的文本出现折行，而另一个方向没有。

- Variable row heights can negatively impact the overall table view performance in your application, regardless of table-view style.
可变行高对你应用程序的表格视图总体性能产生负面影响，无论是哪种表格视图样式。

Finally, although variable row heights are acceptable in grouped tables, they can make a plain table look cluttered and uneven.
最后，虽然可变行高是可以接受的在分组表格中，但是他们可以使一个普通的表格变得混乱和不平衡。

Table-View Elements
表格视图元素

iPhone OS includes a handful of table-view elements that can extend the functionality of table views. Unless noted otherwise, these elements are suitable for use in table views only. Be sure to use these elements correctly in your application, because users are accustomed to their appearance and behavior in the built-in applications.
iPhone OS系统包括一些可扩展功能的表格视图元素。除非注明，这些元素只能在表格视图中使用。确保在你的应用程序中去正确使用这些元素，因为用户已经习惯于内置应用程序的外观和行为。

Note: Programmatically, table-view elements are implemented in different ways. Some are accessory views of the table cell (an object that tells the table how to draw its rows) and others can be displayed when the table view enters an editing mode. See Table View Programming Guide for iPhone OS to learn about the different ways to manage these elements.
注：从编程上讲，表格视图元素被以不同的方式实施。一些是表格单元的配置视图（控制表格如何绘制其行的对象）及其他则当表格视图进入看编辑模式后被显示。详细见Table View Programming Guide for iPhone OS，了解不同的方法来管理这些元素。

■ Disclosure indicator. When this element is present, users know they can tap anywhere in the row to see the next level in the hierarchy or the choices associated with the list item. Use a disclosure indicator in a row when selecting the row results in the display of another list. Don’t use a disclosure indicator to reveal detailed information about the list item; instead, use a detail disclosure button for this purpose.
继续展开标识符。当这个元素出现时，用户知道可以在该行的任何地方点击，进入一个平行屏幕，及选择与表格项目相关联的内容。当选择表格行需显示另一个列表时，使用继续展开标识符。不要使用继续展开标识符来显示列表项目的详细资料，这种情况应该选择细节展开标识符。

■ Detail disclosure button. Users tap this element to see detailed information about the list item. (Note that you can use this element in views other than table views, to reveal additional details about something; see “Detail Disclosure Buttons” (page 110) for more information.)
细节展开按钮。用户点击这个元素，查看有关的清单项目的详细信息。（注：你可以使用该元素在表格视图以外的其它视图，以展现额外的详细信息；见“Detail Disclosure Buttons”以获取更多信息。）

In a table view, use a detail disclosure button in a row to display details about the list item. Note that the detail disclosure button, unlike the disclosure indicator, can perform an action that is

separate from the selection of the row. For example, in Phone Favorites, tapping the row initiates a call to the contact; tapping the detail disclosure button in the row reveals more information about the contact.
在表格视图中，使用细节展开按钮以显示该列表项的详细信息。注意，与继续展开指示符不同，细节展开按钮可以执行选定行的特殊操作。例如，在电话收藏夹中，点击行能够呼叫联系人，点击行中的细节展开按钮可以显示更多关于该联系人的信息。

■ Delete button. Users tap this element to delete the list item. This element appears to the right of a list item when users swipe in the row or when they tap the delete control button while in an editing context. (See Figure 8-10 for an example of this element.)
删除按钮。用户点击此元素以删除列表项目。当用户在该行上用手指滑过或者当点击进入编辑状态时，该元素就会在列表项的右侧出现。见图 8-10 此元素的示例。

■ Delete control button. Users tap this element to reveal and hide the Delete button for each list item. To give additional feedback to users, the horizontal minus symbol inside this button becomes vertical when users tap it to reveal the Delete button.
删除控制按钮。用户点击此元素用来显示和隐藏在每个列表项目的删除按钮。为了让更多的用户进行反馈，当用户点击它来显示删除按钮，水平符号将变成垂直。

In a grouped table that supports a transitory editing mode, the delete control appears outside the table view, on the left. You can see this, for example, when editing an individual’ s information in Contacts. In a grouped table that is in a permanent editing mode (such as the grouped tables on the back of Stocks and Weather), the delete control appears inside the table, on the left.
在支持一个临时编辑模式的分组表格中，删除控件在表格视图外部的右侧。例如，当编辑个人通讯录中的资料时，便可以看到这一点。在分组表格中，在一个永久性的编辑模式（例如是在“股票”与“天气”应用程序背面的分组表格），删除控件显示在表格内部的左侧。

In a plain table, the delete control always appears inside the table, on the left, as shown in Figure 8-10.
在一个普通的表格中，删除控件总在表格内部的左侧显示，如图8-10。

■ Row insert button. Users tap this element to add a row to the list.
插入行按钮。用户点击该元素以在列表中添加一行

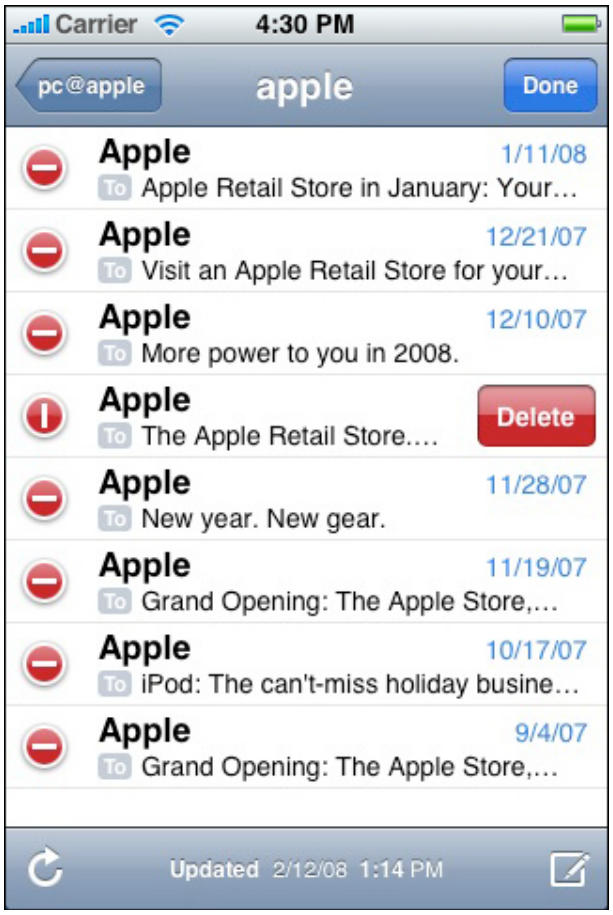
■ Row reorder control. When this element is present, users can drag the row to another location in the list.
行排序的控件。当这个元素存在，用户可以拖动行到表单的另一个位置中去。

■ Checkmark. This element appears next to a list item to show that it is currently selected.

选择标记。这个元素出现在一个列表项目旁边，表明它当前已被选定。

Figure 8-10 A table view can display the Delete button and the delete control button

表格视图可以显示删除按钮和删除控制按钮



Switch Controls

开关控件

A switch control presents to the user two mutually exclusive choices or states, such as yes/no or on/off. A switch control shows only one of the two possible choices at a time; users slide the control to reveal the hidden choice or state. Figure 8-11 shows examples of switch controls.

一个开关控件呈现给用户的是两种相互排斥性的选择或状态，例如是/否或开/关。一个开关控件在同一时间只显示了两种可能的选择的其中之一，用户滑动控件来显示隐藏的选择或状态。图8-11是一个开关控件的例子。

Figure 8-11 Switch controls in a table view

表格视图中的开关控件

Use a switch control in a grouped table view when you need to offer the user two simple, diametrically opposed choices. Because one choice is always hidden, it’ s best to use a switch control when the user already knows what both values are. In other words, don’ t make the user slide the switch control just to find out what the other option is.
当您需要提供给用户两个简单，但是截然相反的选项时，可利用分组表格视图来使用开关控件。因为一个选项总是隐藏的，最好当用户已经知道这个选项是什么时，再使用一个开关控件。换句话说，不要让用户滑动开关控件只是为了看看其他选项。

You can use a switch control to change the state of other user interface elements in the view. Depending on the choice users make, new list items might appear or disappear, or list items might become active or inactive.
你可以使用一个开关控件来改变其他视图的用户界面元素状态。根据不同的用户选择，新的列表项目可能会出现或消失，或者列表项目可能变为有效或无效。

Using Table Views to Enable Common User Actions
使用表格视图实现一般用户操作

Table views are particularly versatile user interface elements, because they can be configured in different ways to support different user actions, such as:
表格视图是非常通用的用户界面元素，因为它们可以通过不同的方式来配置，以支持不用的用户操作，比如：

- Selecting options.
  选择选项

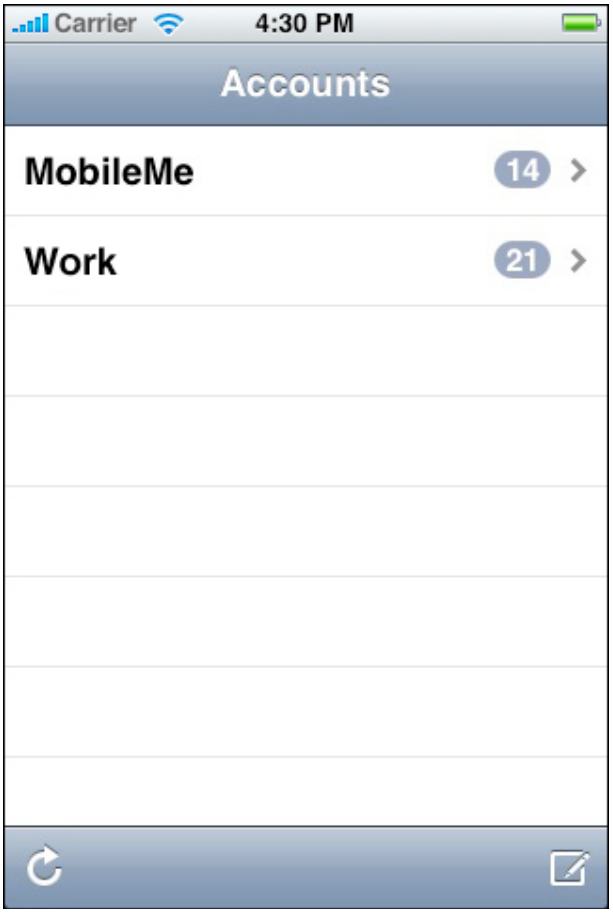
iPhone OS does not include multi-item selection controls analogous to menus or pop-up menus, but a table view works well to display a list of options from which the user can choose. This is because table views display items in a simple, uncluttered way. In addition, the table view provides a checkmark image that shows users the currently selected option (or options) in a list, as shown in Figure 8-12 (page 100).
iPhone OS系统不包括类似于菜单或弹出菜单的多项目选择控件，但一个表格视图适用于显示一系列选项以供用户选择。这是因为表格视图使用一种简单，整齐的方式来显示项目。此外，表格视图提供选择标记图像，来显示列表中用户当前选中的一个（或多个）项目。如图8-12所示。

If you need to display a list of choices users see when they tap an item in a table row, you can use either style of table view. But if you need to display a list of choices users see when they tap a button or other user interface element that is not in a table row, use the plain style.
如果当用户在表格行中点击一个项目时，需要显示选中列表，可以使用任意一种样式的表格视图。但是如果当用户点击一个不在表格行中的按钮，或别的用户界面元素时，需要为用户显示选中列表，则使用普通样式。

- Navigating hierarchical information.
  导航分层信息。

A table view works well to display a hierarchy of information in which each node (that is, list item) can contain its own subset of information, because each subset can be displayed in a separate list. This makes it easy for users to follow a path through the hierarchy by selecting one item in each successive list. The disclosure indicator element tells users that tapping anywhere in the row reveals the subset of information in a new list, as shown in Figure 8-13 (page 101).
表格视图适用于显示的信息的层次结构，其中每个节点（即，列表项目）可以包含其自己的信息子集，因为每个子集可以在一个单独的列表显示。这使得它方便用户顺着路径，在每个层级中选择一个项目。继续展开指示符元素告诉用户在任意行的点击都可以在一个新的列表中显示信息的子集，如图8-13所示。

Figure 8-13 A disclosure indicator indicates that a subset of information is on the next screen
继续展开指示符显示了在下一屏中可以显示一个信息的子集

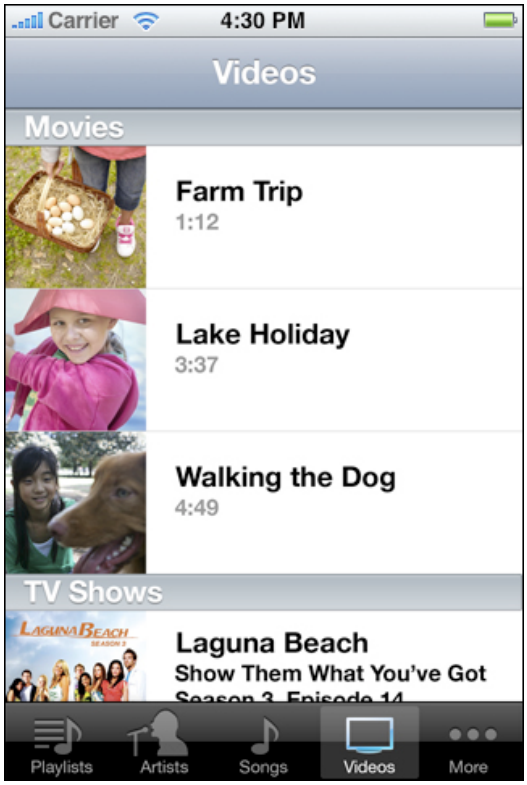


When a table is used for navigation, previously selected table rows do not remain highlighted when users retrace their steps through the hierarchy.
当一个表格被用于导航，在用户返回上一级层时，之前选定的表行将不再被高亮显示。

- Viewing conceptually grouped information.
  查看概念上的分组信息

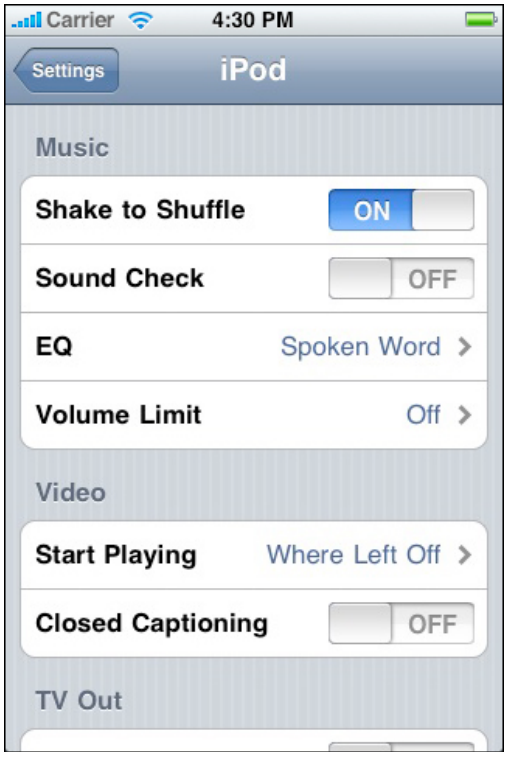
You can use either table-view style to cluster information into logical groups, such as work, home, or school. Both plain and grouped tables allow you to provide context for each section by supplying header and footer text, as shown in Figure 8-14 (page 102).
你可以使用表格视图样式来整合信息到逻辑分组中，如工作，家庭，或者学校。所有的普通表格和分组表格允许你为每个区段通过页眉和页脚文本来提供上下文关系，如图8-14所示。

Figure 8-14 Header text in a plain table divides a list into sections
页眉文本在普通表格中将列表分成若干区段



Generally speaking, grouped tables provide a clearer visual indication of grouping because it's easy for users to distinguish the rounded corners of the groups, even when scrolling quickly. Figure 8-15 (page 102) shows several conceptual groups of values in iPod settings.
一般来说，分组表格提供了一个清晰的分组视觉指示，因为它有很容易让用户区分群组的圆角效果，甚至在快速滚动中。图8-15显示在iPod设置中的几个概念组的价值标准。

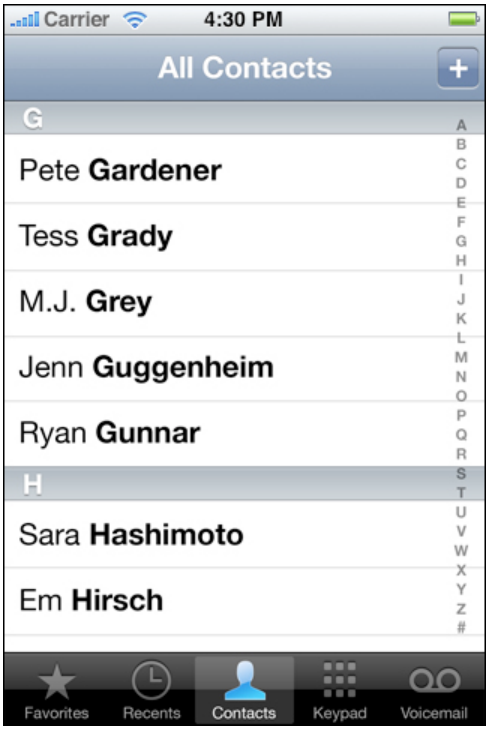
Figure 8-15 A grouped table can contain many separate groups
分组表格可以包含多个分组



- Looking up indexed information.
  查询索引信息

If you’ re using a plain table, you can display an index that helps users quickly find what they need. The index consists of a column of entries (usually letters in an alphabet) that floats on the right edge of the screen, as shown in Figure 8-16 (page 103). Users tap (or drag to) an index entry to reveal the corresponding area in the list. An index is most useful in a list that might span more than a few screenfuls.
如果使用一个普通的表格，你可以显示一个索引来帮助用户快速找到他们需要的东西。该索引包含一个载入的条目（通常是字母表中那些字母）浮在屏幕的右边缘，如图8-16所示。用户点击（或拖曳）索引项以显示在列表中的相应区域。索引最有用的地方是在需延伸数屏的列表中。

Figure 8-16 A plain table can include an index
普通表格可以包含一个索引



If you include an index in a plain table, avoid using table-view elements that display on the right edge of the table (such as the disclosure indicator), because these elements interfere with the index.

如果你在普通表格的包含索引，那么避免使用显示在表格右边缘的表格视图元素（如继续展开标识符），因为这些要素会干涉索引。

Text Views

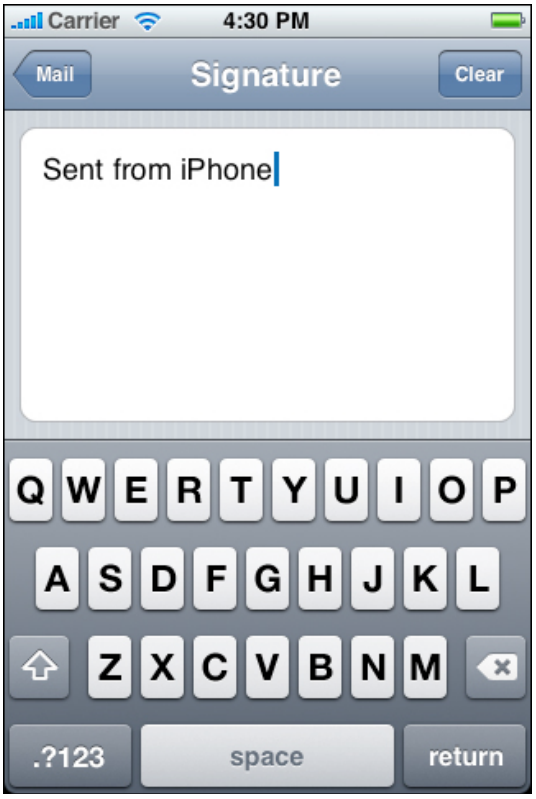
文本视图

A text view is a region that displays multiple lines of text and supports scrolling when the content is too large to fit inside its bounds. Mail uses a text view to allow users to create a signature that appears at the end of each email message they compose, as shown in Figure 8-17.

文本视图是一个区域，它可以显示多行文本，并当内容过大时支持滚动，以适应其范围。“邮件”就使用了文本视图让用户能够创建一个出现在电子邮件末尾的签名，如图8-17所示。

Figure 8-17 A text view displays multiple lines of text

文本视图显示多行文本



Although you might use a text view to display many lines of text, such as the content of a large text document, you can also use a text view to support user editing. If you make a text view editable, a keyboard appears when the user taps inside the text view. The keyboard’ s input method and layout are determined by theuser’ s language settings. When users tap the button labeled “.?123” (shown in Figure 8-17), the keyboard changes to facilitate the entry of numbers and punctuation. You can also specify different keyboard styles, depending on the type of content you expect users to enter. See “Text Fields” (page 120) for a description of the styles you can use.

虽然您可以使用一个文本视图显示多行文字，如一个大的文本文件的内容，你也可以使用文本视图来支持用户编辑。如果你要进行文本编辑，在点击文本视图区域后一个键盘会出现。键盘输入法和布局都取决于用户的语言设置。当用户点击按钮标记“.？123”（如图8-17所示），键盘就会变化，以方便数字和标点符号输入。你也可以指定不同的键盘样式，这取决于你期望用户输入的内容类型。见“Text Fields”，可以使用的样式描述。

You have control over the font, color, and alignment of the text in a text view, but only as they apply to the entirety of the text. In other words, you can’ t change any of these properties for only part of the text. The defaults for the font and color properties are, as you would expect, the system font and black, because they tend to be the most readable. The default for the alignment property is left (you can change this to center or right).

在文本视图中，你能控制字体，色彩，以及文本对齐方式，但它们只适用于文本的全部内容。换句话说，你不能改变这部分文本的任何属性。正如你所期望的，字体和颜色属性的默认值是系统字体和黑色，因为他们往往是最有可读性的。对齐方式是默认左对齐（您可以更改为居中或右对齐）。

If you must enable variable fonts, colors, or alignments within a view that displays text, you can use a web view instead of a text view, and style the text using HTML.
如果必须在一个视图上，显示各种字体，颜色或对齐方式，你可以使用Web视图和HTML样式的文本，而不是一个文本视图。

Web Views
Web视图

A web view is a region that can display rich, HTML content in your application screen. For example, Mail uses a web view to display message content, because it can contain elements in addition to plain text (Figure 8-18 shows an example of this)
Web视图是一个可以在你的应用程序屏幕上显示丰富的，HTML内容的区域。例如，邮件是使用Web视图显示信息的内容，因为它有纯文本不能包含的元素（图8-18显示了一个例子）。

Figure 8-18 A web view can display web-based content
Web视图可以显示基于Web的内容

In addition to displaying web content, a web view provides elements that support navigation through open webpages. Although you can choose to provide webpage navigation

functionality, it’ s best to avoid creating an application that looks and behaves like a mini web browser.
除了显示网页内容，网页视图还提供支持以打开网页中导航的元素。虽然你可以选择提供网页导航功能，但是最好避免创建一个看起来像微型浏览器的行为。

If you have a webpage or web application, you might choose to use a web view to implement a simple iPhone application that provides a wrapper for it. If you plan to access web content that you control, be sure to read Safari Web Content Guide to learn how to create web content that is compatible with and optimized for display on iPhone OS–based devices.
如果你有一个网页或Web应用程序，您可以选择使用Web视图来实现一个简单的iPhone应用程序，因为它可以提供封装。如果你计划访问你所控制的网页内容，一定要读Safari Web Content Guide，以了解如何创建Web内容，并对基于iPhone OS系统的设备提供兼容与显示优化。

CHAPTER 9

Application Controls

应用程序控件

iPhone OS provides several controls you can use in your application, most of which are already familiar tousers of iPhone OS–based devices. Many of these controls are intended for use in specific places, such as in a table view, but some are available for more general usage. This chapter describes the controls that you can use in arbitrary views in your application.
iPhone OS 系统为应用程序开发提供了一些控件，其中大部分已被基于 iPhone OS 系统设备的用户所熟识。很多控件需在特定的地方使用，如表格视图；但是有些可以被普遍应用。本章讲述的控件，你可以在应用程序的任意视图中使用。

As you design the user interface of your application, always remember that users expect familiar controls to behave as they do in the built-in applications. This is to your advantage, as long as you use these controls appropriately in your application.
当你在设计你的应用程序的用户界面时，一定要记住用户期望使用他们熟悉的控件，如同操控设备内置应用一样。随着你在你地应用程序中恰当地使用这些控件，这将成为你的优势。

Activity Indicators

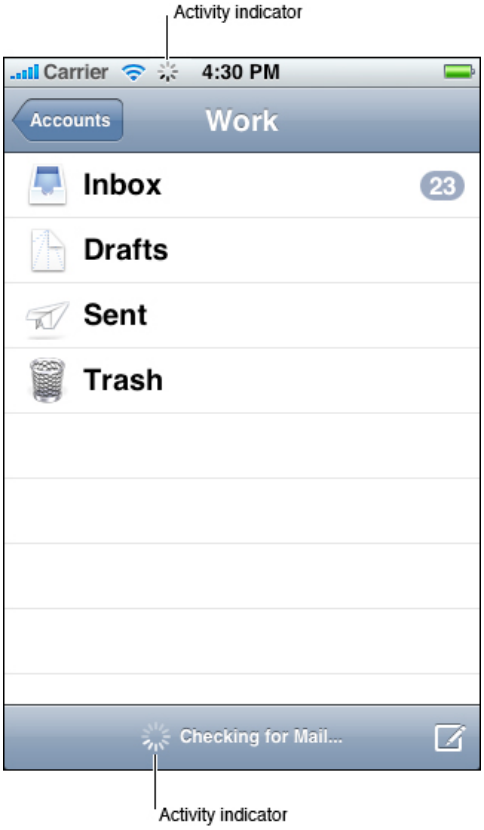
活动状态标示

An activity indicator shows the progress of a task or process that is of unknown duration. If you need to display progress for a task of known duration, use a progress view instead (see "Progress Views" (page 115) for more information about this control).
活动状态标示显示了一个任务的进程或持续时间未知的进程。而如果你需要显示一个已知明确时间的任务进度请用进程视图（见"Progress Views" 了解更多有关信息）。

The "spinning gear" appearance of the activity indicator shows users that processing is occurring, but does not suggest when it will finish.Figure 9-1 shows two types of activity indicators. The activity indicator in the status bar is the network activity indicator; it should be displayed when your application accesses the network for more than a couple of seconds. The larger activity indicator in the toolbar should be displayed if it will take more than a second or two for your application to perform the current task.
当活动状态标示的"旋转叶轮"图标出现并转动时，说明正在处理用户的操作，但并不说明何时会完成。图 9-1 显示了两种类型的活动状态标示。在状态条"旋转叶轮"标示是网络活动标示，它应在你的应用程序访问网络超过两秒钟后显示。而在工具条的大"旋转叶轮"标示应在应用程序来执行当前的任务超过两秒钟后显示。

Figure 9-1 Two types of activity indicators

两种类型的活动状态标示



An activity indicator is a good feedback mechanism to use when it's more important to reassure users that their task or process has not stalled than it is to suggest when processing will finish. You can choose the size and color of an activity indicator to coordinate with the background of the view in which it appears. By default, an activity indicator is white.

活动状态标示是一个很好的反馈机制，其重要性在于向用户告知反馈并使之安心：他们的任务或进程没有停滞，这比动作将完成时才告知要好得多。你可以选择活动状态标示的大小和颜色，从而使它在出现的时候能跟背景视图相协调。默认情况下，活动状态标示是白色的。

An activity indicator disappears when the task or process has completed. This default behavior is recommended, because users expect to see an activity indicator when something is happening and they associate a stationary activity indicator with a stalled process. To learn how to display the network activity indicator, see the networkActivityIndicatorVisible method in *UIApplication Class Reference*. To learn how to display the larger, non-network activity indicator in your code, see *UIActivityIndicatorView Class Reference*.

当活动状态标示消失时，表明已完成任务或进程。这是一个默认的行为，在有任务进行的情况下，用户希望看到一个活动的标示，因为他们会将把一个静止不动的活动指标与停滞不前的过程联在一起。要了解如何显示网络活动状态的标示，见 UIApplication Class Reference 中的

networkActivityIndicatorVisible method。要学习如何在你代码中实现较大的非网络活动状态标示，见 UIActivityIndicatorView Class Reference。

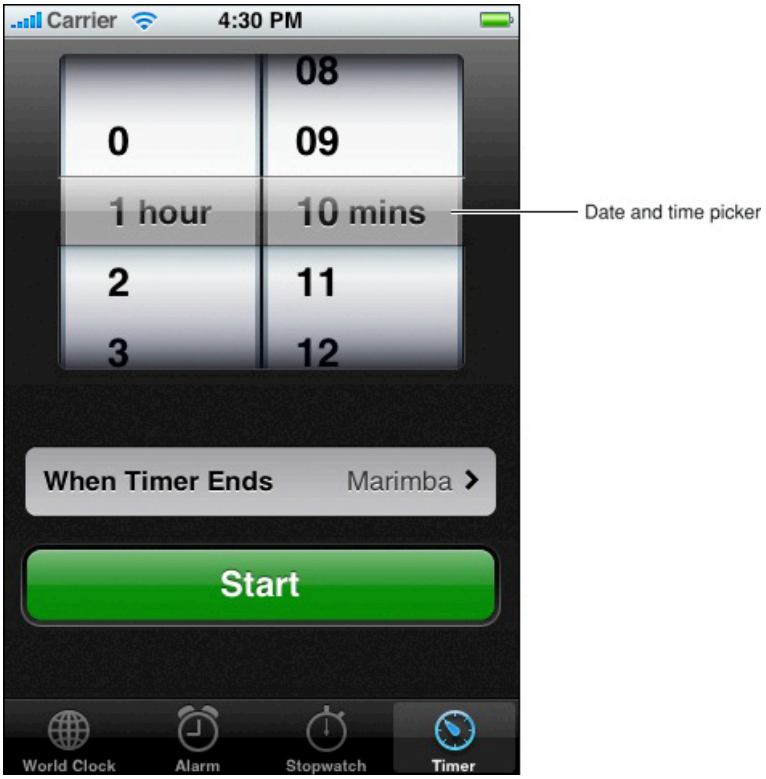
Date and Time Pickers
日期与时间拨选器

A date and time picker gives users an easy way to select a specific date or time. A date and time picker can have up to four independent spinning wheels, each of which displays values in a single category, such as month or hour. Users flick or drag to spin each wheel until it displays the desired value beneath the clear selection bar that stretches across the middle of the picker. The final value comprises the values displayed in each wheel. Figure 9-2 shows an example of a date and time picker.

日期和时间拨选器为用户提供了一个简单的方法来选择一个特定的日期或时间。一个日期和时间拨选器可最多有 4 个独立的飞轮，每个显示在一个独立类别的数值，如：月份、小时等。用户拨动或滚动每个轮子，直到飞轮中间区域的选择栏上出现所需的数值。最终的数值将包括每个车轮的显示值。图 9-2 展示了一个日期和时间拨选器的例子。

Figure 9-2 A date and time picker

图 9-2 日期和时间拨选器



Use a date and time picker to allow users to avoid typing values that consist of multiple parts, such as the day, month, and year of a date. A date and time picker works well because the values in each part have a relatively small range and users already know what the values are. Depending on the mode you specify, a date and time picker displays a different number of wheels, each with a set of different values. The date and time picker defines the following modes:
使用日期和时间拨选器,使用户避免了输入由多个部分组成的数值,如日、月、年。日期和时间拨选器效果很好,因为值在每个部分都有相对较小的范围，而且用户已经知道了数值是多少。根据你指定的模式,日期与时间拨选器显示了不同数量的轮子,每个都有一系列的数值。日期和时间拨选器定义了以下模式：

- Time. The time mode displays wheels for the hour and minute values, with the optional addition of a wheel for the AM/PM designation.
时间。显示时间的模式包含有小时和分钟数值的飞轮，外加一个飞轮用于设定 AM/PM。

- Date. The date mode displays wheels for the month, day, and year values.
日期.。日期模式包含有显示月、日和年数值的飞轮。

- Date and time. The date and time mode displays wheels for the calendar date, hour, and minute values,with the optional addition of a wheel for the AM/PM designation. This is the default mode
日期和时间。日期和时间模式包含有显示日历 ,小时和分钟数值的飞轮，外加飞轮可以设置 AM/PM。这是默认模式。

- Countdown timer. The countdown timer mode displays wheels for the hour and minute. You can specify the total duration of a countdown, up to a maximum of 23 hours and 59 minutes.
倒数计时器. 倒数计时器包含显示小时和分钟的飞轮。你可以指定一个倒计时数,最高达到 23 小时 59 分钟。

By default, a minutes wheel displays 60 values (0 to 59). However, if you need to display a coarser granularity of choices, you can set a minutes wheel to display intervals of minutes, as long as the interval divides evenly into 60. For example, you might want to display the quarter-hour intervals 0, 15, 30, and 45.
默认情况下,分钟飞轮有 60 个数值(0 ~ 59)。然而,如果你需要显示一个粗略选择,你可以设定一分钟的时间间隔来显示,只要区间划分均匀到 60。例如,你可能想显示一刻钟的时间间隔 0、15、30、45。

Regardless of its configuration, the overall size of a date and time picker is fixed, and is the same size as the keyboard. You might choose to make a date and time picker a focal element in your view, or cause it to appear only when needed. For example, the timer mode of the built-in Clock application displays an always-visible date and time picker because the selection of a

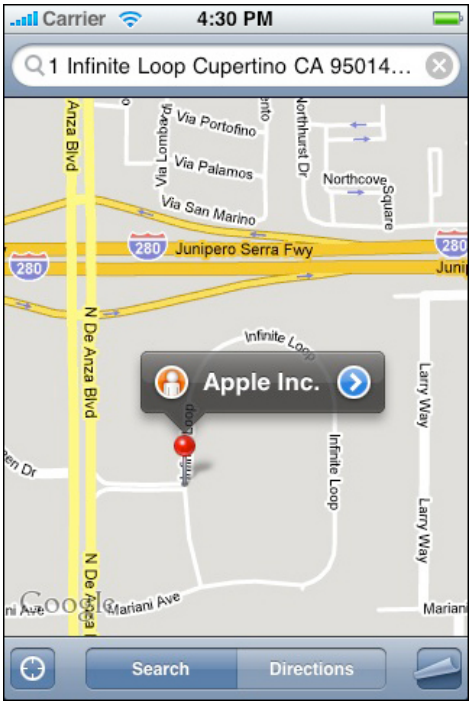
time is central to the function of the Timer. On the other hand, the Set Date & Time preference (available in Settings > General > Date & Time, when you turn off Set Automatically) displays transient date and time pickers, depending on whether users want to set the date or the time.
无论它的配置，日期和时间拨选器的总体尺寸都是固定的,如虚拟键盘一样有着固定的尺寸。你可以在视图中选择日期和时间拨选器做为一个焦点元素,或使它在需要的时候才出现。例如，内置的计时器应用程序显示一个始终可见的日期和时间拨选器，因为时间的选择是至关重要的计时器功能。另一方面，设置日期和时间偏好设置（在设置>一般>日期及时间，可当你关闭自动设置）显示临时日期和时间拨选器，这取决于用户是否要设置日期或时间。

To learn more about using a date and time picker in your code, see *UIDatePicker Class Reference*.
要了解有关在代码中实现日期和时间选取器，请见 *UIDatePicker Class Reference*。

Detail Disclosure Buttons
细节示意按钮

A detail disclosure button reveals additional or more detailed information about something. Usually, you use detail disclosure buttons in table views, where they give users a way to see detailed information about a list item (for more information about this usage, see “Table-View Elements” (page 98)). However, you can use this element in other types of views to provide a way to reveal more information or functionality.
For example, the Maps application displays a detail disclosure button users can tap to access more functionality related to the dropped pin. Figure 9-3 shows an example of a detail disclosure button.
细节示意按钮揭示了附加或者更详细的信息。通常，在表格视图中的细节示意按钮能帮助用户查看关于列表项目更详细的信息(更多关于这个用法的信息，见 “Table-View Elements”)。当然，你也可以在其他视图中使用这个元素，以提供一种方式来显示更多的信息或功能。例如，地图中的显示细节按钮。用户可以点击细节示意按钮来了解大头针所在位置的更详细信息。图 9-3 展示一个细节示意按钮

Figure 9-3A detail disclosure button reveals additional details or functional
一个细节示意按钮，显示更多细节和功能



To learn more about using a detail disclosure button in your code,seeUIButton Class Reference.

要了解更多关于你代码的细节示意按钮，见 UIButton Class Reference

Info Buttons

信息按钮

An Info button provides a way to reveal configuration details about an application, often on the back of the screen. For this reason, Info buttons are especially well suited to utility applications. You can see an example of an Info button in the lower-right corner of the Weather application (shown in Figure 9-4).

信息按钮提供一种方式来展示配置的细节,经常在视图背面页面上。因此,信息按钮特别适合实用工具程序。你可以在"天气"应用程序右下角看到一个信息按钮的例子(如图 9-4)。

Figure 9-4 An Info button reveals information, often configuration details

一个信息按钮显示信息,通常是配置的细节

Info buttons are available with a light background and a dark background. The light background style (which is shown in Figure 9-4) looks good on a view with a dark background. Conversely, an Info button with a dark background shows up well on a view with a light background. An Info button glows briefly when users tap it. When you use the Info button iPhone OS provides, you get this pressed-state appearance automatically. To learn more about using an Info button in your code, see *UIButton Class Reference*.

信息按钮可与淡色和黑色背景相搭配使用。淡色背景样式(如图 9-4)在深色背景的应用程序上看起来不错。相反地,一个深色背景信息按钮更适合显示在淡色背景的应用程序上。信息按钮在用户点击时会发光。当你在使用 iPhone OS 系统提供的信息按钮时，这种外观效果会随着你的操作自动产生。要了解更多有关于信息按钮的代码，见 *UIButton Class Reference*。

Labels

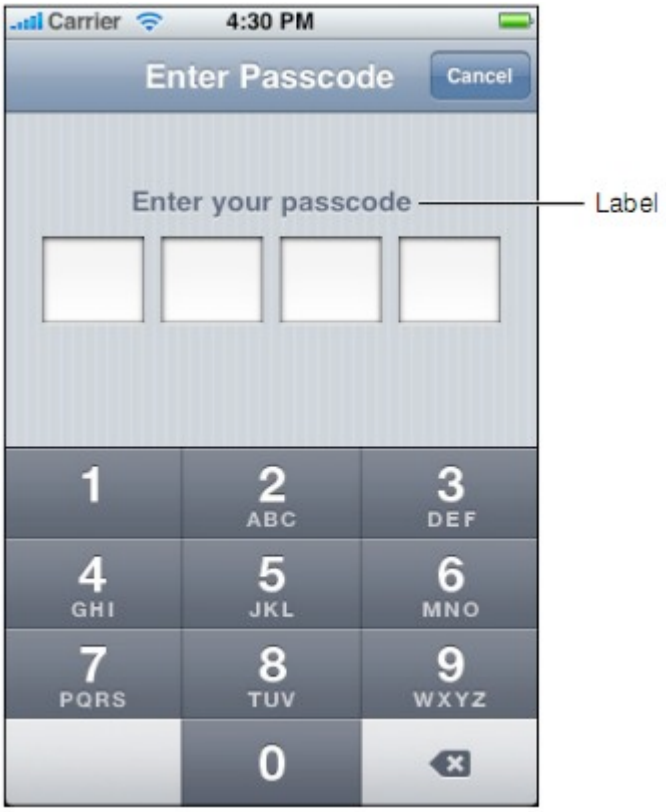
标签

A label is a variably sized amount of static text. Figure 9-5 shows an example of a label.

标签是一种不同大小的静态文本。图 9-5 展示了一个标签例子。

Figure 9-5 A label gives users information

为用户提供信息的标签



You can use a label to name parts of your user interface or to provide limited help to the user. A label is best suited to display a relatively small amount of text. You can determine various properties of the label's text, such as font, text color, and alignment, but above all, you should take care to make your labels legible. Don't sacrifice clarity for fancy fonts or showy colors.
你可以使用标签来命名用户界面的一些部分或给用户提供有限的帮助。标签是最好的相对文本较小的显示方式。你可以决定标签文本的不同属性，如字体、文本颜色和对齐方式，但最重要的是，让它们清晰地显示。不要因花式字体或艳丽的颜色而牺牲文字的清晰性。

As you compose the text of your labels, be sure to use the user's vocabulary. Examine the text in your application for developer-centric terms and replace them with user-centric terms. To learn more about using labels in your code, see *UILabel Class Reference*.
当你撰写你的标签文本时，一定要使用用户的语言。检查你的文本，把应用开发人员为中心的文本，改为以用户为中心的文本。了解更多关于标签的代码，见 *UILabel Class Reference* 参考。

Page Indicators
页面标示

A page indicator displays a dot for each currently open view in an application. From left to right, the dots represent the order in which the views were opened (the leftmost dot represents the first view). The currently visible view is indicated by a glow on the dot that represents it. Users

tap to the left or the right of the glowing dot to view the previous or next open view. Figure 9-6 shows an example of a page indicator.

页面标示会为应用程序的每一个当前打开视图显示一个标示点。这些点从左至右的排列代表了一个个的被打开的视图（最左边的点代表第一个视图）。发光的点就代表了当前显示的视图。用户可点亮左右的点来查看上一个或下一个视图。图 9-6 展示了一个页面标示的例子。

Figure 9-6 A page indicator

页面标示

A page indicator gives users a quick way to see how many views are open and an indication of the order in which they were opened; it does not help users keep track of the steps they took through a hierarchy of Views. Because the views in a utility application tend to be peers of each other, a page indicator is sufficient to help users navigate through them. A productivity application that displays hierarchical information, on the other hand, should offer navigation through the elements in the navigation bar (for more on this, see “Navigation Bars” (page 68)).

页面标示为用户提供了快捷的方法，查看有当前打开视图的数量及排列顺序；但它们不能帮助用户跟踪视图的层次结构。因为在一个实用程序中，视图间的关系是对等的，页面标示只是方便用户浏览它们。而一个生产力辅助应用程序会显示分层信息，并通过提供在导航条来帮助引导用户（了解更多，可以看“Navigation Bars” ）

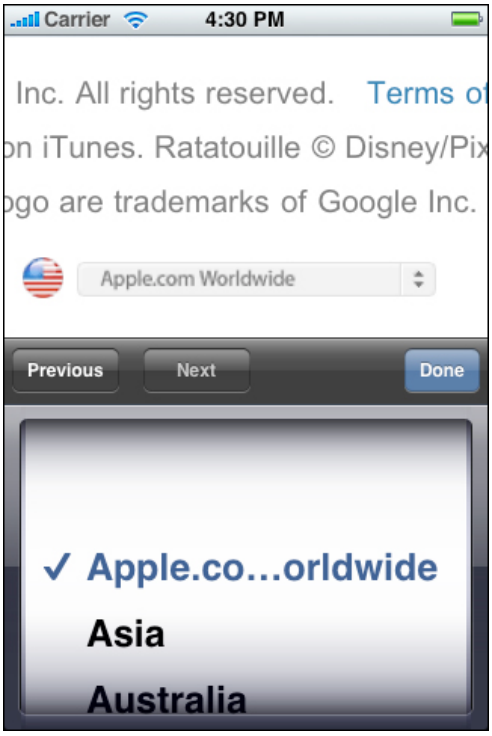
Typically, page indicators work well near the bottom edge of the application screen, below the it contains. This leaves the more important information (the view itself) in the upper part of the screen where users can see it easily. Be sure to vertically center a page indicator between the bottom edge of the view and the bottom edge of the screen.
通常情况下，页面标示适用于应用程序屏幕的底部边缘，界面视图之下。这使得更重要的信息（视图本身）在屏幕的上方，用户可以很清晰地看到它。一定要确保页面标示垂直居中于页面的底部边缘和屏幕底部之间。

Although there is no programmatic limit to the number of dots you can display in a page indicator, be aware that the dots do not shrink or squeeze together as more appear. For example, in portrait orientation, you can display at most 20 dots in a page indicator before clipping occurs. Therefore, you should provide logic in your application to avoid this situation. Although you can hide a page indicator when there is only one open view, the default behavior is to display it. To learn more about using a page indicator in your code, see *UIPageControl Class Reference*.
虽然没有限制页面指标的数量，但注意，当视图越来越多时，那些标示点不可收缩、挤压。比如，在纵向方向，你最多可以在页面标示上显示 20 个点在没有剪辑前。因此，你应该为你的应用程序提供一些逻辑，从而避免这种情况发生。虽然，你可以在只打开一个页面视图的时候隐藏页面标示，但是默认动作是显示它。想要了解更多关于页面标示的代码， 见 *UIPageControl Class Reference*。

Pickers
拨选器

A picker is a generic version of the date and time picker (see “Date and Time Pickers” (page 108) for more information about this control). You can use a picker to display any set of values. As with a date and time picker, users spin the wheel (or wheels) of a picker until the desired value appears. Figure 9-7 shows a picker with a single wheel.
拨选器是日期和时间的通用版本（见“Date and Time Pickers”更多有关这些控件的信息）。你可以使用拨选器来显示任何一个数值。作为一个日期和时间的拨选器，用户只需转动拨选器的飞轮，直到需要的数值出现为止。图 9-7 展示了有一个飞轮的拨选器例子。

Figure 9-7 A picker as displayed in Safari on iPhone
iPhone 上，Safari 浏览器中的拨选器



As you decide whether to use a picker in your application, consider that many, if not most, of the values in a wheel are hidden from the user when the wheel is stationary. This is not necessarily a problem, especially if users already know what those values are. For example, in a date and time picker, users understand that the hidden values in the month wheel can only be numbers between 1 and 12. If you need to provide choices that aren’ t members of such a well-known set, however, a picker might not be the appropriate control.
当你打算在应用程序中使用一个拨选器时，需考虑到当飞轮处于静止状态，大多数的数值在默认状态下是被隐藏的。这不一定是个问题，尤其是在用户已经知道这些数值是什么的情况下。例如，在日期和时间拨选器中，用户知道，隐藏在月份飞轮下的数值只能是介于 1 和 12 之间的这些数字。但如果你需要提供的选择并不是那些用户本来就能知道的数值，选择器可能就未必是适当的控件了。

If you need to display a very large number of values, you might want to list them in a table view instead of in a picker. This is because the greater height of a table view makes scrolling faster.
如果你需要显示的数值很大，可能就需要用表格视图来列表，而不是用拨选器了。这是因为表视图有更长的列表能使滚动更快。

If you need to provide context for a value in a picker, such as a unit of measurement, display it in the translucent selection bar horizontally across the center of the control. Do not display such labels above the picker or on the wheels themselves. For an example of the correct way to display labels, see the Timer function of the built-in Clock application, which displays “hours” and “mins” (or “min”) next to the values users select.
如果你需要提供拨选器上值的上下文关系，如计量单位方面，那就显示在半透明的横跨控件中心的选择栏众。请不要显示在拨选器上方的标签或飞轮本身中。关于显示标签的例子，看时钟应用程序中的计时

器，它在用户选择的数值旁显示“小时”和“分钟”（或“分”）。

As with a date and time picker, a generic picker can be visible all the time (as a focal point of your user interface) or it can appear only when needed. The overall size of a picker, including its background, is fixed, and is the same size as a keyboard.
To learn more about using a picker in your code, see *UIPickerView Class Reference*.
同带有日期和时间拨选器一样，一个通用的拨选器可以始终被显示（作为你用户界面的中心），或者只在需要时出现。一个拨选器的总体尺寸，包括它的背景，是固定的，并且与虚拟键盘的大小相同。 要了解更多关于拨选器的代码，见 *UIPickerView Class Reference*。

Progress Views
进程视图

A progress view shows the progress of a task or process that has a known duration. If you need to display progress for a task of unknown duration, use an activity indicator instead (see “Activity Indicators” (page 107)for more information about this control).
进度视图显示了一个任务的进展，或一个已知持续时间的进度。如果你需要显示一个未知持续时间的任务进度，请用活动状态标示代替（见 “Activity Indicators” 可以了解更多有关控件的信息）

iPhone OS provides two styles of progress view, which are the default style and the bar style. The appearance of each style is very similar, except for height:
iPhone OS 系统提供了两种进程视图的样式，这两种样式是默认样式和条栏样式。除了高度每个样式的外观上都非常相似的：

·The default style is intended for use in an application’ s main content area.
默认样式是用于应用程序中的主要内容区域。

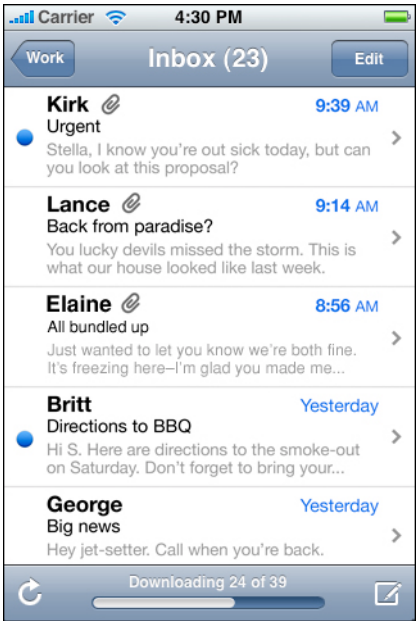
·The bar style is thinner than the default style, which makes it well-suited for use in a toolbar. For example, in Mail a bar-style progress view appears in the toolbar when users download new messages or send anemail message.
条栏样式比默认样式单薄，这使它非常适合用于工具条中使用。例如：当用户在下载新的邮件或发送电子邮件，在条栏样式的进程视图会显示在工具条中。

The behavior of both styles of progress view is the same. As the task or process proceeds, the track of the progress view is filled from left to right. At any given time, the proportion of filled to unfilled area in the view gives the user an indication of how soon the task or process will finish. Figure 9-8 shows an example of a bar-style progress bar
两种进程视图的行为模式是相同的。当任务或进程在进行，进程视图的轨道都是从左到右逐渐填满。在任何一个特定的时间，填满与未填满部分的比例为用户提供完成该进程所需时间的标示。图 9-8 显示了

一个条栏样式的进度条例子。

Figure 9-8 A bar-style progress view in a toolbar

在工具条上一个条栏样式的进程视图



A progress view is a good way to provide feedback to users on tasks that have a well-defined duration, especially when it’ s important to show users approximately how long the task will take. When you display aprogress view, you tell the user that their task is being performed and you give them enough informationto decide if they want to wait until the task is complete or cancel it.

进程视图是一种很好的方法，提供给用户一个所需时间的反馈，特别是当它很有必要让用户了解任务大约会多久时。当显示进度视图时，你告诉了用户任务正在执行并给他们足够的信息，去决定是否要等到完成任务或者取消它。

To learn more about using a progress view in your code, see *UIProgressView Class Reference*.

了解更多关于进程视图的代码，见 *UIProgressView Class Reference*。

Rounded Rectangle Buttons

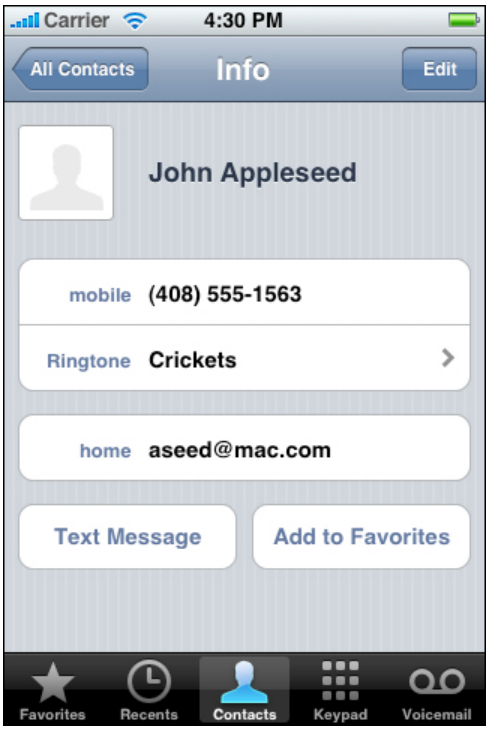
矩形倒角按钮

A rounded rectangle button is a versatile button you can use in a view to perform an action. You can see examples of this type of button at the bottom of an individual’ s Contacts view: The Text Message and Addto Favorites buttons are rounded rectangle buttons, as shown in Figure 9-9.

矩形倒角按钮是一个多功能键，你可以在一个视图中用它执行一个动作。你可以在个人电话本的视图底

部看到这种类型按钮的例子：文本消息按钮和添加收藏夹按钮都是矩形倒角按钮，如图 9-9.

Figure 9-9
Rounded rectangle buttons perform application-specific actions
矩形倒角按钮执行应用程序的具体动作



When you supply a title for a rounded rectangle button, be sure to:
当你给一个矩形倒角按钮提供标题时，一定要：

■ Use title-style capitalization (that is, capitalize every word except articles, coordinating conjunctions, andprepositions of four or fewer letters)
标题首字母要用大写（也就是说，除了冠词，并列连词，和四个字母的或者更少字母的介词外都需要大写）

■ Avoid creating a title that is too long. Overly long text gets truncated, which can make it difficult forusers to understand it.
避免创建一个太长的标题。过长的文本被截断后，会使用户难以理解它。

To learn more about using a rounded rectangle button in your code, see *UIButton Class Reference*.
了解更多矩形倒角按钮的代码，见 *UIButton Class Reference*

Search Bars
搜索条

A search bar is a field that accepts text from users, which your application can use as input for a search. Whenthe user taps a search bar, a keyboard appears; when the user is finished typing search terms, the input ishandled in an application-specific way. (For guidelines on handling search in your application, see "ProvidingSearch and Displaying Search Results" (page 51).)
搜索栏是一个接受输入用户文本的区域，这里应用程序可以输入文本以进行搜索。当用户点击搜索条，一个虚拟键盘就会出现；当使用者打完要搜索的内容，输入以应用程序特定的方式处理。（关于处理应用程序的搜索的导引，见"ProvidingSearch and Displaying Search Results" 。

By default, a search bar displays the search icon on the left side. In addition, a search bar can display a fewoptional elements:
默认情况下，搜索图标显示在搜索栏的左侧。此外，搜索栏可以显示一些可选的要素：

■ Placeholder text. This text might state the function of the control (for example, "Search") or remind users in what context they are searching (for example, "YouTube" or "Google").
占位符文本。文本可能是强调控件功能（例如，"搜索"）或是提醒用户在什么搜索引擎下搜索（例如，"YouTube"或"谷歌"）。

■ The Bookmarks button. This button can provide a shortcut to information users want to easily find again. For example, the Bookmarks button in the Maps search mode gives access to bookmarked locations,recent searches, and contacts.
书签按钮。这个按钮提供快捷，使用户更加容易再次找到信息。例如，在地图搜索模式书签按钮可以访问已收藏的位置，找到最近的搜索和联系人。

■ The Clear button. Most search bars include a Clear button that allows users to erase the contents of the search bar with one tap.
清除按钮。大多数的搜索栏有有一个清除按钮，允许用户一键清除搜索栏内的内容。

■ A descriptive title, called a prompt, that appears above the search bar. For example, a prompt can be ashort phrase that provides introductory or application-specific context for the search bar
表述性的标题，叫做提示，出现在搜索栏的上面。例如，提示可以是短句，提供介绍或在特定的应用程序情境。

Figure 9-10 shows a search bar that includes customized placeholder text, a Bookmarks button, and thedefault search icon.
图 9-10 显示了一个搜索栏，包括自定义的占位符文本，书签按钮，和默认搜索图标。

Figure 9-10　A search bar with optional placeholder text and a Bookmarks button

带有可选的占位符文本和书签按钮的搜索栏

By default, the Bookmarks and Clear buttons interact in the following ways:

默认情况下，书签和清除按钮通过以下方式交互：

■　When the search bar contains any non-placeholder text, the Clear button is visible so users can erasethe text. If there is no user-supplied or non-placeholder text in the search bar, the Clear button is hiddenbecause there is no need to erase the contents of the search bar.

当搜索栏包括任何非占位符文本时，清除按钮是可见的，因此用户就可以删除文本。如果搜索栏中没有用户提供的文本或非占位符文本，清除按钮被隐藏。

■　The Bookmarks button is visible only when there is no user-supplied or non-placeholder text in thesearch bar. This is because the Clear button is visible when there is text in the search bar that users might want to erase.

当没有用户提供的文本或者非占位符文本在搜索栏中，才能显示书签按钮。因为清除按钮只有当用户在搜索栏有想要删除的文本时才能显示。

You can customize a search bar by specifying one of the standard-color background styles, such as:

你能通过指定一种标准色背景样式来自定义搜索栏，例如:

■　Blue (the default gradient that coordinates with the default appearance of toolbars and navigation bars).The default background style is shown in Figure 9-10.

蓝色（默认的渐变色，和默认的工具条和导航条的外观一致）。默认的背景样式在图 9-10 有显示。

- Black
  黑色

In addition, you can display a scope bar below the search bar, which contains buttons that users tap to select a scope for the search. The scope bar adopts the same appearance you specify for the search bar, and you supply custom titles for the scope buttons.
此外,你可以在搜索栏下面显示一个范围栏, 其包含了一些按钮供用户点击以选择搜索范围。范围栏与指定的搜索栏采用一样的外观, 并由用户自定义范围栏按钮标题。

The scope bar displays below the search bar, regardless of orientation, unless you use a search display controller in your code (see *UISearchDisplayController Class Reference* for more information). When you usea search display controller, the scope bar is displayed within the search bar to the right of the search field when the device is in landscape orientation (in portrait orientation, it' s below the search bar).
范围栏显在搜索栏下面，不分方向，除非在你的编程时使用搜索显示控件。（更多信息见 *UISearchDisplayController Class Reference* ）当你使用搜索显示控件时，当装置横向显示时，范围栏在搜索栏中，搜索输入区右侧。（纵向显示时，在搜索栏下方）

To learn more about using a search bar and scope bar in your code, see *UISearchBar Class Reference*.
更多关于搜索栏和范围栏的代码实现，见 *UISearchBar Class Reference*

Segmented Controls
分段控件

A segmented control is a linear set of segments, each of which functions as a button that can display a different view. When users tap a segment in a segmented control, an instantaneous action or visible result should occur. For example, Settings displays different information when users use the segmented control to select an email protocol, as shown in Figure 9-11.
分段控件是一种线性分段集，每一个分段作为一个按钮以显示不同的视图，当用户点击分段控件中一个分段时，瞬间的动作或者明显可见的结果将会出现。例如，当用户利用分段控件选择一个邮件协议时，设置会显示不同的信息，如图 9-11。

Figure 9-11 A segmented control with three segments
具有三个分段的分段控件

The length of a segmented control is determined by the number of segments you display and by the size of the largest segment. The height of a segmented control is fixed. Although you can specify the number of segments to display, be aware that users must be able to comfortably tap a segment without worrying about tapping a neighboring segment. Because hit regions should be 44 x 44 pixels, it’ s recommended that a segmented control have five or fewer segments.
分段控件的长度取决于你显示的段数的数量和最大的分段的尺寸。分段控件的高度是固定的。虽然你可以指定的段数，注意用户可以很舒适地点击一个分段而不用担心误点到旁边的分段。因为触击区域必须是 44*44 像素，所以谨记分段数不要超过 5 个。

A segmented control can contain text or images; an individual segment can contain either text or an image, but not both. In general, it’ s best to avoid mixing text and images in a single segmented control.
分段控件能够包含文本或者图像，一个独立的分片能够包含文本或者图片中的其中一个，但不能同时包含两者。一般来说，最好能够避免文本和图片混合在一个单独的分段控件中。

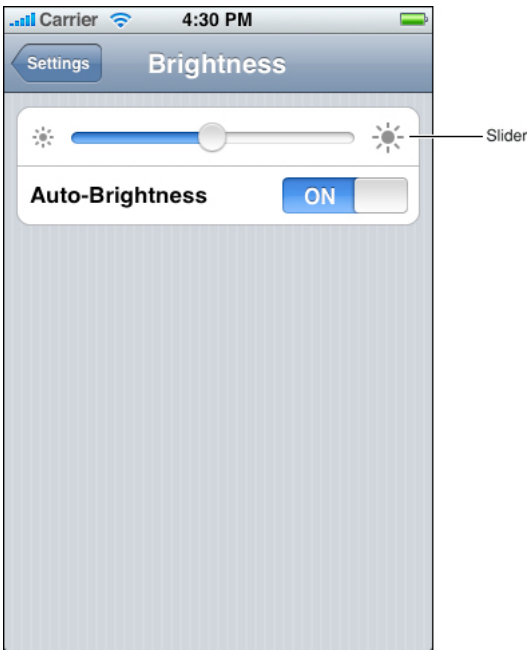
A segmented control ensures that the width of each segment is proportional, based on the total number of segments. This means that you need to ensure that the content you design for each segment is roughly equal in size.
分段控件保证了每个分段的宽度成比例，这取决于总的分段数。这意味着需要保证每个分段的设计大小基本要保持一致。

To learn more about using a segmented control in your code, see *UISegmentedControl Class Reference.*
了解更多有关分段控制的代码，见 *UISegmentedControl Class Reference*。

Slider 滑块

A slider allows users to make adjustments to a value or process throughout a range of allowed values. When users drag a slider, the value or process is updated continuously. Figure 9-12 shows an example of a slider with minimum and maximum images.
滑块允许用户在一个允许值的范围内调整一个值或进程。当用户拖动一个滑块，数值或进程随之更新。图 9-12 显示了一个滑块的例子及两头最大化和最小化的图形。

Figure 9-12 A slider
一个滑块



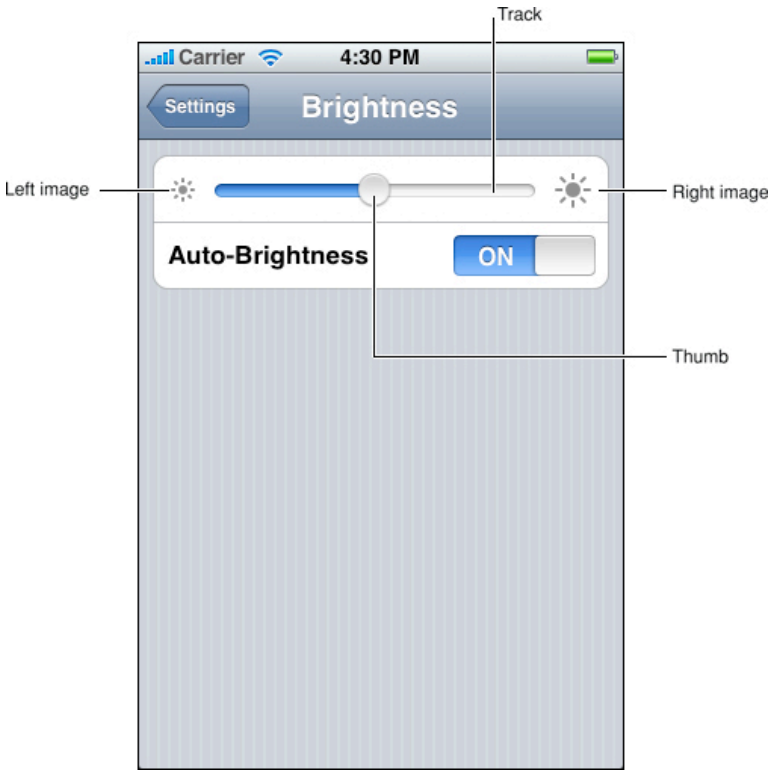
Sliders are useful in two main situations:
滑块主要在这两种情况使用：

·When you want to allow users to have fine-grained control over the values they choose
当你允许用户对他们选择的值进行精确的控制

·When you want to allow users to have fine-grained control over the current process
当你允许用户对当前过程进行精确的控制

A slider consists of a track, a thumb, and optional right and left value images. Figure 9-13 shows these parts of a slider.
滑块包含一个轨道，一个调节点和可选的左右数值图形组成。图 9-13 显示了滑块的这些部件。
Figure 9-13 Four parts of a slider

滑块的四个部件



You can set the width of a slider to fit in with the user interface of your application. In addition, you can display a slider either horizontally or vertically.
你可以设置滑块的宽度，以适应你的应用程序用户界面。此外，还可以设定水平或垂直显示的滑块。

There are several ways to customize a slider:
几种自定义滑块的方法：

■ You can define the appearance of the thumb, so users can see at a glance whether the slider is active.
你可以定义调节点的外观，让用户对滑块是否处于活动状态一目了然。

■ You can supply images to appear at either end of the slider (typically, these correspond to minimum and maximum values), to help users understand what the slider does.
你可以在滑块的任一端设置图形（通常，这些对应的最小和最大值），以帮助用户理解滑块的作用。

A slider that controls font size, for example, could display a very small character at the minimum end and a very large character at the maximum end.
例如，一个用于控制字体大小的滑块，可以显示一个非常小的字体在最小值的一端，而最大值的一端显示比较大的字体。

■ You can define a different appearance for the track, depending on which side of the

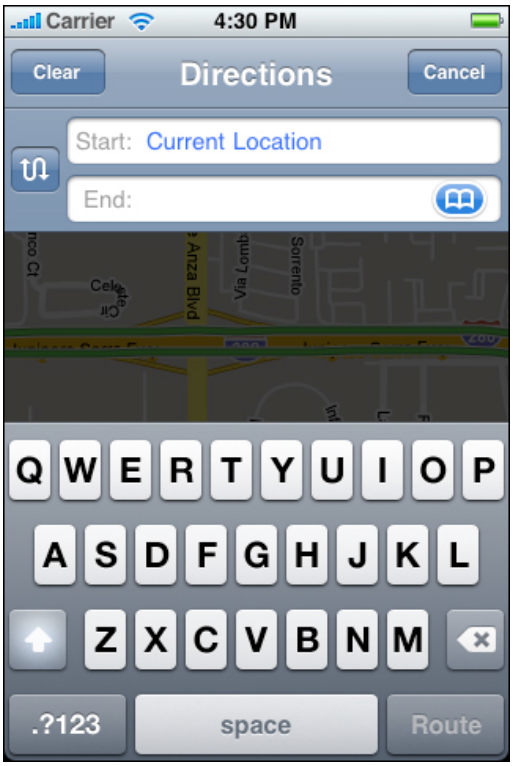
thumb it is on and which state the control is in.
你可以给轨道定义一个不同外观的，这取决于调节点所在的那一端以及控件的状态。

To learn more about using a slider in your code, see *UISlider Class Reference*.
了解更多关于滑块的代码，见 *UISlider Class Reference*。

Text Fields 文本输入框

A text field is a rounded rectangular field that accepts user input. When the user taps a text field a keyboard appears; when the user taps Return in the keyboard, the text field handles the input in an application-specific way. A text field can contain a single line of input. Figure 9-14 shows two text fields in the Maps application.
文本输入框是一个圆角矩形区域，来接收用户的输入。当用户点击文本输入框时，虚拟键盘就会出现；当用户点击键盘上的回车键时，文本输入框用应用程序指定的方式来处理输入。文本输入框可以包含一行简单输入。图 9-14 显示了地图应用程序的两个文本输入框。

Figure 9-14 A text field can accept user input
文本输入框可以接受用户输入



You can customize a text field to help users understand how they should use it in your application. For example, you can display custom images in the left or right sides of the text field, or a system-provided button, such as the Bookmarks button shown in Figure 9-14. In

general, you should use the left end of a text field to indicate its purpose and the right end to indicate the presence of additional features, such as bookmarks.
你可以自定义文本输入框，以帮助用户理解他们该如何在应用程序中使用它。例如，你可以在文本输入框的左边或右边显示自定义图形，或一个系统提供的按钮，如图 9-14 显示的书签按钮。一般而言，你应该使用文本输入框的左边标示其用途，而右边标示附加特性，如书签。

You can also cause the Clear button to appear in the right end of a text field. When this element is present, tapping it clears the contents of the text field, regardless of any other image you might display over it.
你也可以让清除按钮显示在文本输入框的右边。当该元素出现，点击它可以清除文本输入框中的内容，不用考虑其他任何你可能会显示在这里的图形。

Sometimes, it helps users understand the purpose of a text field if it displays a hint, such as "Name." A text field supports the display of such placeholder text, which can appear when there is no other text in the field. To learn more about using a text field and customizing it to display images and buttons, see *UITextField Class Reference*.
有时候，会显示提示以帮助用户理解文本输入框的目的，如"名称"。在文本输入框里没有文字时，在文本输入框里会显示一个文本占位符。要了解更多关于文本输入框和自定义文本输入框以显示图形和按钮，见 *UITextField Class Reference*。

You can specify different keyboard styles to accommodate different types of content you expect users to enter. (Note that you have no control over the keyboard's input method and layout, which are determined by the user's language settings.) For example, you might want to make it easy for users to enter a URL, a PIN, or a phone number. iPhone OS provides several different keyboard types, each designed to facilitate a different type of input. To learn about the keyboard types that are available, see *UIKeyboardType*. To learn more about managing the keyboard in your application, read "Managing the Keyboard" in iPhone Application Programming Guide.
你可以定义不同的键盘样式来适应你期望用户输入的不同类型内容。(注意你不能控制键盘的输入方式和布局，这由用户的语言设置决定。)例如，你可能想更方便用户输入 URL，PIN 码或电话号码。iPhone OS 系统提供多种不同的键盘样式，每种设计适合于一种不同的输入类型。要了解可供使用的键盘类型，见 *UIKeyboardType*。要了解更多如何在应用程序里管理键盘的信息，见 *iPhone Application Programming Guide* 中的 *"Managing the Keyboard"*。

CHAPTER 10

System-Provided Buttons and Icons

系统预制按钮与图标

To promote a consistent user experience (and to make your job easier), iPhone OS provides numerous standard buttons for use in navigation bars and toolbars, and icons for use in tab bars.

为了促进一致的用户体验（并让你的工作更容易），iPhone OS 系统提供许多用于导航条和工具条的标准按钮，以及用于标签条的图标。

This chapter describes the standard icons and buttons available to you and provides guidelines on how to use them appropriately.You should familiarize yourself with the buttons and icons in this chapter regardless of the type of application you’ re developing, so that you can:

本章介绍了供给您的标准的图标和按钮提，并提供有关如何适当使用的指南。不管你正在开发应用程序的类型是什么，都应该熟悉本章中的按钮和图标，以便可以：

- Use the system-provided items correctly

正确使用系统提供的按钮与图标项目

- Avoid designing a custom icon that looks too similar to a system-provided icon

避免设计一个与系统提供的图标过于相似的自定义图标

Using System-Provided Buttons and Icons

利用系统提供的按钮和图标

iPhone OS makes available many of the standard toolbar and navigation bar buttons, tab bar items, and general-use buttons used throughout the built-in applications.You can see a handful of standard toolbar buttons in the Mail toolbar, shown in Figure 10-1.

iPhone OS 系统，在其内置应用程序中提供并使用了大量标准的工具条和导航条按钮，标签条项目以及常规使用的按钮。你可以看到在“邮件”工具条上的很多标准的工具条按钮，如图 10-1.

Figure 10-1 Standard buttons in the Mail toolbar

在邮件工具条中的标准按钮

Buttons such as the Refresh, Organize, Trash, Reply, and Compose buttons shown in Figure 10-1 are used consistently in many of the built-in applications, so users are well-acquainted with what they mean and how to use them.This means that if your application supports these functions, you can take advantage of users' familiarity to streamline the application's user interface.It also means that if you associate these buttons with other tasks, you're likely to confuse and irritate users by promising functionality they expect, but delivering something else.
如图 10-1 所示的刷新，组织，删除，回复，及撰写的按钮在内置应用软件中经常被使用，所以用户对它们的意义及用法都非常熟悉。这意味着，如果你的应用程序支持这些功能，便可以利用用户熟悉功能的好处，来优化应用程序的用户界面。相反地，如果你将这些按钮和其他任务联系起来，很可能会使你的用户困惑和急躁，因为你承诺了用户希望的功能，但提供了别的东西。

In addition to the benefit of leveraging users' prior experience, using system-provided buttons and icons imparts two other substantial advantages, specifically:
除了能利用用户的先前使用经验的好处之外，使用系统预制的按钮和图标还有另外两个很大的好处，具体如下：

■ Decreased development time, because you don't have to create custom art to represent standard functions.
缩短开发时间，因为你不用去创建自定义图形来去呈现标准功能。

■ Increased stability of your user interface, even if future iPhone OS updates change the appearances of standard icons. In other words, you can rely on the semantic meaning of a

standard icon remaining the same, even if its appearance changes.
增强您的用户界面的稳定性，即使将来 iPhone OS 系统升级后改变标准图标的外观。换句话说，你可以依靠这个特征即使它的外观发生了变化，标准图标的语义仍旧可以保持不变。

It bears repeating that to realize the advantages of user familiarity, shorter development time, and semantic consistency of the user interface, you must use the buttons and icons appropriately. Specifically, this means that you should use a button or icon in accordance with its documented meaning and recommended placement, and *not* according to your interpretation of its appearance. See *"Standard Buttons for Use in Toolbars and Navigation Bars,"* *"Standard Icons for Use in Tab Bars,"* and *"Standard Buttons for Use in Table Rows and Other User Interface Elements"* for meaning and placement information for the system-provided buttons and icons.
再次重申，为了得益于用户的熟悉程度，更短的开发时间，及用户界面语义的一致性，你必须恰当地使用这些按钮和图标。具体来说，这意味着你在使用按钮或图标时应该与它的文档释义和推荐位置保持一致，而不是根据自己对其外观的理解。参见“Standard Buttons for Use in Toolbars and Navigation Bars,” “Standard Icons for Use in Tab Bars,” 和 “Standard Buttons for Use in Table Rows and Other User Interface Elements” 来获取系统预制按钮和图标的含义和位置信息。

Interface Builder makes it easy to use the system-provided buttons and apply system-provided icons to your controls.See the appearance-related information in "Interface Objects" in *Interface Builder User Guide* for guidance.
Interface Builder（Xcode 开发环境的组件，译者注）使运用系统预制按钮和图标到你的应用程序控件上变得很容易。参见 Interface Builder User Guide 中“Interface Objects”的相关外观信息作为指导。

If you can't find a system-provided toolbar or navigation bar button or tab bar item icon that has the appropriate meaning for a specific function in your application, you should design a custom button or icon. "Icons for Navigation Bars, Toolbars, and Tab Bars" (page 133) gives some guidelines to help you do this.
如果你不能找到与你的应用程序中有特殊功能含义相当的系统预制工具条或者导航条按钮或标准栏项目图标，你就应该设计一个自定义按钮或图标。“Icons for Navigation Bars, Toolbars, and Tab Bars”提出了一些指导方针，帮助你做这些。

Standard Buttons for Use in Toolbars and Navigation Bars
用于工具条与导航条的标准按钮

iPhone OS makes many of the standard buttons users see in toolbars and navigation bars available to you.These buttons, shown in Table 10-1 (page 125), are available in two styles, each of which is appropriate for the specific usages described here:
iPhone OS 系统提供了大量用户在工具条和导航条可见到的标准按钮。表 10-1 显示的这些按钮，有两

种样式，分别用于如下情况：

■ Bordered style—for example, the Add button in the Phone Contacts navigation bar. This style is suitable for both navigation bars and toolbars.
带边框的样式—例如，在电话 / 联系人 / 导航条中的“增加”按钮。这种样式对于导航条和工具条来说都很适用。

■ Plain style—for example, the Compose button in the Mail toolbar. This style is suitable for toolbars only. In fact, if you specify the plain style for a button in the navigation bar, it will be converted to the bordered style.
普通样式—例如，在邮件 / 工具条中的“撰写”按钮。这种样式只适合工具条。事实上，如果你在导航条中指定一个普通风格的按钮，它将被转换为带边框样式。

As with all system-provided buttons, you should avoid using the buttons described in Table 10-1 to represent actions other than those for which they are designed.In particular, avoid choosing a button based on its appearance, without regard for its documented meaning.See “Using System-Provided Buttons and Icons” (page123) for a discussion of the reasons why it’s important to use these icons correctly.(Information on symbol names and availability for these buttons is available in documentation for *UI Bar Button System Item*.)
对于所有系统预制的按钮，你应该避免使用表 10-1 中描述的按钮来呈现不同于其预定义的操作。尤其要避免以按钮的外观来选择，而不顾其文档释义。请参“Using System-Provided Buttons and Icons”来了解正确使用这些图标重要性的原因。（关于这些按钮符号的名称与可用性，见 UI Bar Button System Item 的文档。）

Table 10-1 Standard buttons available for toolbars and navigation bars (shown in the plain style)
用于工具条和导航条的标准按钮（以普通样式显示）

Button 按钮	Meaning 含义	Name 名称
	Opens an action sheet that allows users to take an application-specific action 打开动作列表，使用户能够执行特定应用程序相关的动作	Action
	Opens an action sheet that displays a photo picker in camera mode 打开动作列表，以显示拍照模式下的照片拨选器	Camera
	Opens a new message view in edit mode 在编辑模式下打开新消息视图	Compose
	Show application-specific bookmarks 显示应用程序特定的书签	Bookmarks

	Display a search field 显示搜索框	Search
	Create a new item 创建新项目	Add
	Delete current item 删除当前项	Trash
	Move or route an item to a destination within the application, such as a folder 将项目移动或传送至应用程序的目标位置，如文件夹	Organize
	Send or route an item to another location 将某项发送或传送至其他位置	Reply
	Stop current process or task 停止当前进程或任务	Stop
	Refresh contents (use only when necessary; otherwise, refresh automatically) 刷新内容（仅在必要时使用，否则自动刷新）	Refresh
	Begin media playback or slides 开始播放媒体或幻灯片	Play
	Fast forward through media playback or slides 在播放媒体或幻灯片时快进	FastForward
	Pause media playback or slides (note that this implies context preservation) 暂停播放媒体或幻灯片（暗示要保存上下文关系）	Pause
	Move backwards through media playback or slides 在播放媒体或幻灯片时快退	Rewind

In addition to the buttons shown in Table 10-1, you can also use the system-provided Edit, Cancel, Save, and Done buttons shown in Table 10-2 to support editing or other types of content manipulation in your application.(Information on symbol names and availability for these buttons is available in documentation for *UI Bar Button System Item*).These buttons are suitable for both navigation bars and toolbars, and are available in the bordered style only If you specify the plain style for one of these buttons, it will be converted to the bordered style..
除了在表 10-1 所示的按钮，你也可以使用系统提供的编辑，取消，保存，及完成按钮来在应用程序中编辑或其他类型的内容操作，如表 10 - 2 所示。（关于这些按钮的符号名称和可用性，见 UI Bar Button System Item 文档。）这些按钮适合于导航条和工具条，但仅对具有带边框样式按钮。如果你为这些按钮指定普通样式，它会自动转换成带边框样式。

Table 10-2 Bordered action buttons for use in navigation bars

在导航条中使用有边框的的动作的按钮

Button 按钮	Meaning 含义	Name 名称
Edit	Enter an editing or content-manipulation mode 进入编辑或内容处理模式	Edit
Cancel	Exit the editing or content-manipulation mode without saving changes 退出编辑或内容处理模式，但不保存更改	Cancel
Save	Save changes and, if appropriate, exit the editing or content-manipulation mode 保存更改，如果合适的话，退出编辑或内容处理模式	Save
Done	Exit the current mode and save changes, if any 退出当前模式，并保存所做的更改	Done

Standard Icons for Use in Tab Bars
标签条的标准图标

iPhone OS provides the standard icons described in Table 10-3 for use in tab bars.Information on symbol names and availability for these icons is in documentation for *UI Tab Bar System Item*.
iPhone OS 系统提供用于标签条的标准图标，如表 10-3 所示。关于这些按钮的符号名称和可用性，见 *UI Tab Bar System Item* 文档。

As with all standard buttons and icons, it’ s essential to use these icons in accordance with their documented meanings. In particular, take care to base your usage of an icon on its semantic meaning, not its appearance.This will help your application’ s user interface make sense even if the icon associated with a specific meaning changes its appearance.See “ Using System-Provided Buttons and Icons” (page 123) for further reasons why it’ s important to use these icons correctly.
对于所有的标准按钮和图标，按照文档释义来使用是非常重要的。特别要注意要在图标的语义基础上选择，而不是根据它的外观。这将帮助应用程序用户界面表现得合理，即使相关图标的外观被改变了。请参阅“Using System-Provided Buttons and Icons”理解正确地使用这些图标的原因。

Table 10-3 Standard icons for use in tab bar tabs
在标签条中使用标准图标

Icon 图标	Meaning 含义	Name 名称
	Show application-specific bookmarks 显示特定应用程序相关的书签	Bookmarks

	Show Contacts 显示联系人	Contacts
	Show downloads 显示下载	Downloads
	Show user-determined favorites 显示用户设定的收藏夹	Favorites
	Show content featured by the application 显示应用程序推荐的内容	Featured
	Show history of user actions 显示用户的动作历史	History
	Show additional tab bar items 显示更多标签条项目	More
	Show the most recent item 显示最近一次使用的项目	MostRecent
	Show items most popular with all users 显示所有用户最关注的项目	MostViewed
	Show the items accessed by the user within an application-defined period 显示在应用程序设定的用户访问过的项目	Recents
	Enter a search mode 进入搜索模式	Search
	Show the highest-rated items, as determined by the user 显示用户评份最高的项目	TopRated

Standard Buttons for Use in Table Rows and Other User Interface Elements
用于表格行和其他用户界面元素的标准按钮

iPhone OS provides a few buttons for use in table rows and other elements.These buttons, described in Table 10-4, should be used semantically correctly, as with all standard buttons and icons.In particular, avoid choosing a button based on its appearance, without regard for its documented meaning.See “Using System-Provided Buttons and Icons” (page 123) for a discussion of the reasons why it’s important to use these buttons correctly.
iPhone OS 操作系统提供在表格行和其他元素中使用的按钮。这些按钮，在表 10-4 中所述，所有标准按钮和图标应该按正确的语义使用。尤其要避免出现按钮的外形来选择，而不顾它的文件意义。见“Using System-Provided Buttons and Icons”来了解正确地使用这些图标很重要的原因。

Although the detail disclosure button is usually used in table rows, it can be used elsewhere. For

more information about this button, see “Detail Disclosure Buttons” (page 110).iPhone OS also provides a set of controls for use in table rows only; for more information about these, see “Table-View Elements” (page 98).
虽然细节显示按钮经常在表格行中使用，它也能在任何地方使用，要了解这一按钮的更多信息，见“Detail Disclosure Buttons”。iPhone OS 操作系统也提供一套只在表格行中使用的控件，要了解相关更多信息，见“Table-View Elements”。

For more information on symbol names and availability for these buttons, see documentation for *UIButtonType*. （For information on the symbol name and availability for the detail disclosure table-view element, see documentation for *UITableViewCellAccessoryDetailDisclosureButton*）.
有关这些按钮的符号名称和可用性，参阅 *UIButtonType* 文档。（有关细节显示表格视图元素的符号名称和可用性信息，请查看 *UITableViewCellAccessoryDetailDisclosureButton*）。

Table 10-4 Standard buttons for use in table rows and user interface elements
表格行和其他用户界面元素的标准按钮

Button 按钮	Meaning 含义	Name 名称
	Display a people picker to add a contact to an item 显示联系人拨选器以将联系人加至一个项目	ContactAdd
	Display a new view that contains details about the current item 显示包含当前项详细信息的新视图	DetailDisclosure
	Flip to the back of the view (usually in a utility application) to display configuration options or more information. 翻转至视图背面（通常是在实用工具应用程序中）以显示配置选项或更多信息。 Note that the Info button is also available as a light-colored “i” in a dark circle. 注：也可以是深色圆圈浅色字母“i”。	Info

CHAPTER 11

Creating Custom Icons and Images

定制图标和画面

The user interfaces of iPhone applications are characterized by beautiful images and lush color. As an application designer, you want to fit into this environment by providing an aesthetically pleasing user interface. Although iPhone OS provides a wide range of elegant and attractive user interface elements, there are two custom elements every application needs: an application icon and a launch image. In addition, applications should provide a small icon for iPone OS to display in Spotlight search results.

iPhone 应用程序的用户界面特征是漂亮的画面和丰富的色彩。作为应用程序设计者，你需要能够制作出美观的，令人合意的用户界面以匹配这样的用户环境。虽然 iPhone OS 提供了大量优雅，引人入胜的用户界面元素。但是，有两个自定义元素是每一个应用程序都需要的。一个是应用程序图标，另一个是启动画面。此外，还需要为应用程序提供一个小图标给 iPhone OS，用来在 Spotlight（Spotlight 是 iOS 的一项搜索控件）上的搜索结果中显示。

Some applications need custom icons to represent application-specific functions and modes in navigation bars, toolbars, and tab bars. When you follow the guidelines for creating these icons, you can achieve results that both harmonize with the built-in icons and subtly accentuate your application’ s style.

一些应用程序需要定制一些自定义图标用来在导航条，工具条和标签条中表示特定程序功能和模式。当你遵循这些导引去定制图标时，就能使设计的图标与内置的图标协调一致并能准确的强调出你所设计应用程序的风格。

Note: The standard bit depth for icons and images is 24 bits (8 bits each for red, green, and blue), plus an 8-bit alpha channel. The PNG format is recommended, because it preserves color depth and supports an embedded alpha channel. You can use the Preview application to produce the final PNG file.

注意:画面和图标的标准位深是 24 位(每 8 位颜色分别为红，绿和蓝)，加上一个 8 位的 Alpha 通道。PNG 格式是被推荐的，因为它能保存色深和支持一个内嵌的 Alpha 通道。你可以使用“预览”程序来生成最终的 PNG 文件。

You do not need to constrain your palette to web-safe colors. Although you can use alpha transparency in the icons you create for navigation bars, toolbars, and tab bars, do not use it in application icons.

你不需要限制你的色板来适应网页安全色。虽然你能用 Alpha 透明处理来制定导航条，工具条和标签条的图标，但不能把它运用在应用程序图标的制作中。

Application Icons
应用程序图标

An application icon is an icon users put on their Home screens and tap to start an application. This is a place where branding and strong visual design should come together into a compact, instantly recognizable, attractive package.
应用程序图标是放在主屏幕上的图标，点击它能启动应用程序。在此品牌推广和强烈视觉冲击的设计应该需要整合于精简、可识别且引人注意中去。

Users choose which application icons they want to display on their Home screens, so you should design an icon that is:
用户选择他们想在主屏幕上显示的应用程序图标，因此，图标设计应按以下原则：

- Attractive, so users feel compelled to keep it on their Home screens
  吸引人的，使用户不想更换你为他们设计的图标
- Distinctive, so users can easily find it among all other icons
  有特色的，与众不同，用户能在所有图标中轻易地找到你设计的图标

Try to balance eye appeal and clarity of meaning in your icon so that it’ s rich and beautiful, but still conveysthe essence of your application’ s purpose. Also, it’ s a good idea to investigate how your choice of image and color might be interpreted by people from different cultures.
努力使你设计的图标即能吸引眼球，又能有清晰的含义，以便图标达到丰满和精美的效果，更能传达你应用程序的实质。而且，研究自不同文化背景的人们对你所选择的画面和颜色的理解是一个好想法。

When a user decides to display your application icon on the Home screen, iPhone OS automatically addssome visual effects so that it coordinates with the built-in icons. Specifically, iPhone OS adds:
当用户决定把你的应用程序图标显示在主屏幕上时，iPhone OS 会自动的增加一些视觉效果，以使其与系统内置图标相协调。iPhone OS 具体添加地有：

- Rounded corners
  倒圆角
- Drop shadow
  投影
- Reflective shine
  光照反射

For example, Figure 11-1 shows a simple icon as it might be provided by an application.
例如，图 11-1 展示了一个简单的图标在成为一个应用程序图标之后，可能会出现的效果。

Figure 11-1 A simple application icon before it is displayed on a Home screen
一个简单的应用程序图标显示在主屏幕之前的效果

Figure 11-2 shows the same icon as it is displayed on a Home screen by iPhone OS.
展示了同样的图标由 iPhone OS 主屏幕显示的效果。

Figure 11-2 A simple application icon displayed on a Home screen
在主屏幕上显示的一个应用程序图标

Application icons that include a discernible background look best on the Home screen. This is primarily because of the rounded corners iPhone OS adds: uniformly rounded corners ensure that all the icons on a user's Home screen have a consistent appearance that invites tapping. If you create an icon with a black background that disappears when it's viewed on the Home screen, users don't see the rounded corners. Such icons often don't look tappable and tend to interfere with the orderly symmetry of the Home screen that users appreciate.
包含清晰背景的应用程序图标应该在主屏幕上有最佳的视觉效果。这主要是 iPhone OS 增加了倒圆角：统一倒圆角确保了所有图标在用户主屏幕上有一致的外观。如果你创建了一个带有黑色背景的图标，那么当它在主屏幕上显示时就会消失不见，用户就会看不到倒圆角。这些图标通常看上去不易被点击，并常常会干扰用户对主屏幕的整齐匀称的需求。

To ensure that your icon can take advantage of the visual enhancements iPhone OS provides, produce an image in PNG format that:
为了确保你的图标能利用 iPhone OS 所提供的视觉增强功能，生成 PNG 图像格式的规则是：

- Measures 57 x 57 pixels, with 90 degree corners (if the image measures other than this size, iPhone OS scales it)
  57x57 像素， 90 度直角（如果图像的尺寸不符，iPhone OS 会进行缩放）
- Does not have any shine or gloss
  不要有任何的亮斑或光泽
- Does not use alpha transparency
  不要使用 Alpha 透明处理

Name your icon file Icon.png and place it at the top level of your application bundle. To learn more about the contents of the application bundle, see "*The Application Bundle*" in *iPhone Application Programming Guide*.
将你的图标文件命名为 Icon.png 并把它放置在你应用程序包的顶层。了解更多关于应用程序包的内容，阅读在 *iPhone Application Programming Guide* 中的"*The Application Bundle*"部分。

Note: If you choose, you can prevent iPhone OS from adding the shine to your icon. To do this, you need to add the UIPrerenderedIcon key to your application's Info.plist file (read "The Information Property List" in iPhone Application Programming Guide to learn about this file).
注意：你可以阻止 iPhone OS 将光泽增加到你的图标上。为了做到这一点，你需要添加 UIPrerenderedIcon 键到你应用程序的 Info.Plist 文件中(见 *iPhone Application Programming Guide* 中 *"The Information Property List"* 学习更多)。

Your icon should still measure 57 x 57 pixels, regardless of whether you take advantage of the added shine.
你的图标仍然应是 57x57 像素，无论你是否利用了额外的光泽。

When you submit your application, you must include a 512 x 512 pixel version of your application icon for display in the App Store. Although it's important that this version be instantly recognizable as your application icon, it should be subtly richer and more detailed. In other words, you should not simply scale up your application icon to create an icon for the App Store.
当你提交设计的应用程序时，里面必须包含一个能显示在苹果软件应用商店的 512x512 像素版本的应用程序图标。尽管作为你应用程序的图标必须是能易于识别的，这是非常重要的，它仍然应该是更精细丰富和详细的。换句话说，你不能把按比例放大的应用程序图标来作为苹果软件应用商店的图标。

You must also provide a 512 x 512 pixel version of your application icon if you're developing an application for ad-hoc distribution (that is, to be distributed in-house only, not through the App Store). In the ad-hoc case, name the icon file iTunesArtwork (no file extension) and place it at the top level of your application bundle. This icon identifies your application in iTunes.
如果你是为一个私下发布（及不通过 App Store）而开发的应用程序，仍然须另外提供一个 512x512 像素版本的应用程序图标。在私下发布情况中，命名图标文件为 iTunesArtwork（没有文件扩展名）并把它放置在你应用程序包的顶层。这个图标能在 iTunes 中辨认你的应用程序。

Small Icons 小图标

Every application should supply a small icon that iPhone OS can display when the application name matches a term in a Spotlight search.

每一个应用程序都应该给 iPhone OS 提供一个小图标，这样当应用程序名字符合 Spotlight 搜索条件时才能显示。
Applications that supply settings should also supply this icon to identify them in the built-in Settings application.
提供设置的应用程序也应该提供这些选项的图标，以便能在内置的设置程序选项中标识他们。

Your small icon should clearly identify your application so that users can easily recognize it in a list of search results. To do this, create a streamlined, attractive icon that:
你的小图标应该能清楚的标识出你应用程序的含义，以便用户能轻易地在搜索结果列表中识别出来。为了做到这一点，就要制定出一个合理且有有吸引力的图标，还要符合以下规则：

- Uses the PNG format.
  使用 PNG 格式
- Measures about 29 x 29 pixels.
  尺寸约为 29x29 像素

Name your icon file Icon-Small.png and place it at the top level of your application bundle. To learn more about the contents of the application bundle, see "The Application Bundle" in *iPhone Application Programming Guide*.
图标文件命名为 Icon-Small.png 并把它置于你应用程序包的顶层。阅读在 *iPhone Application Programming Guide* 中的“The Application Bundle”部分，学习更多关于应用程序包的内容 。

Note: If you do not provide an icon named Icon-Small.png, and your application bundle does not contain a previous version of a small icon named Icon-Settings.png, iPhone OS shrinks your application icon for display in search results and in Settings.
注意：如果你不提供一个命名为 Icon-Small.png 的图标，那么你的应用程序包就不会包含上一个以 Icon-Small.png 命名的小图标的版本，那么 iPhone OS 会将应用程序图标缩小显示在搜索结果和设置选项上。

If your application bundle already contains a small icon named Icon-Settings.png, and does not contain an icon named Icon-Small.png, iPhone OS displays the settings icon in the search results. However, you should update the application bundle so that it contains only the Icon-Small.png file.
如果你的应用程序包已经包含了一个名为 Icon-Settings.png 的小图标，却没有包含一个名为 Icon-Small.png 的图标，那么 iPhone OS 会在搜索结果中显示设置图标。不管怎样，你应该修正你的应用程序包，以便它只包含名为 Icon-Small.png 的文件。

Launch Images
启动画面

To enhance the user’ s experience at application launch, you should provide a launch image. A launch image looks very similar to the first screen your application displays. iPhone OS displays this image instantly when the user taps your application icon on the Home screen. As soon as it’ s ready for use, your application displays its first screen, replacing the launch placeholder image.
为了增强在应用程序启动时的用户体验，你应该提供一个启动画面。启动画面看起来类似于应用程序地第一屏画面。当用户在主屏幕上点击你的应用程序图标时，iPhone OS 会立即显示这个启动画面。当应用程序准备就绪后，iOS 就会显示应用程序的第一屏画面来替换启动占位画面。

It’ s important to emphasize that the reason to supply a launch image is to improve user experience; it is not an opportunity to provide:
这里重要强调的是之所以运用启动画面是为了改善用户体验。而不是用于提供：

- An “application entry experience,” such as a splash screen
  一个“应用程序进入体验”，例如版权页
- An About window
  一个关于窗口
- Branding elements, unless they are a static part of your application’ s first screen
  品牌推广元素，除非他们本身就是你应用程序第一屏画面中的一个静态部分

Because users are likely to switch among applications frequently and quickly, you should make every effort to cut launch time to a minimum, and you should design a launch image that downplays the experience rather than drawing attention to it.
因为用户可能会频繁的，快速的在应用程序之间切换，所以你应该尽一切努力把启动时间缩减到最小值，并设计一个启动画面能使用户弱化程序切换时的感受，而不是引起用户的注意。

To do this, you should design an image in PNG format that:
为了做到这些，你应该设计一个 PNG 格式的图像：

- Measures 320 x 480 pixels. Including the status bar area allows you to display the status bar color you’ ve chosen immediately, instead of displaying it after your application has finished starting.
  尺寸为 320x480 像素。包括状态条区域，使你选择状态条颜色立刻显示，而不是等待应用程序启动完成成启动后才显示。

- Is identical to the first screen of the application, except for:
  应用程序的第一屏是相同的，除了下列以外：

- Text. The launch image is static, so any text you display in it will not be localized.

文本。启动画面是静态的，所以任何文本的显示都不能被本地化。

- User interface elements that might change. Avoid including elements that might look different when the application finishes launching, so that users don't experience a flash between the launch image and the first application screen.

可以改变用户界面的元素。避免包括那些当应用程序启动后，看起来会与之前不一样的元素，以便用户感受不到在启动画面和应用程序第一屏间的切换。

Name your launch image file Default.png and place it at the top level of your application bundle. To learn more about the contents of the application bundle, see *iPhone Application Programming Guide*.

把你的启动画面的文件命名为 Default.png，并把它置于你应用程序包的顶层。阅读在 *iPhone Application Programming Guide* 中的“The Application Bundle”部分，学习更多关于应用程序包的内容 。

If you think that following these guidelines will result in a very plain, boring launch image, you're right. Remember, the launch image is not meant to provide an opportunity for artistic expression; it is solely intended to enhance the user's perception of your application as quick to launch and immediately ready for use. The following examples show you how plain a launch image can be.

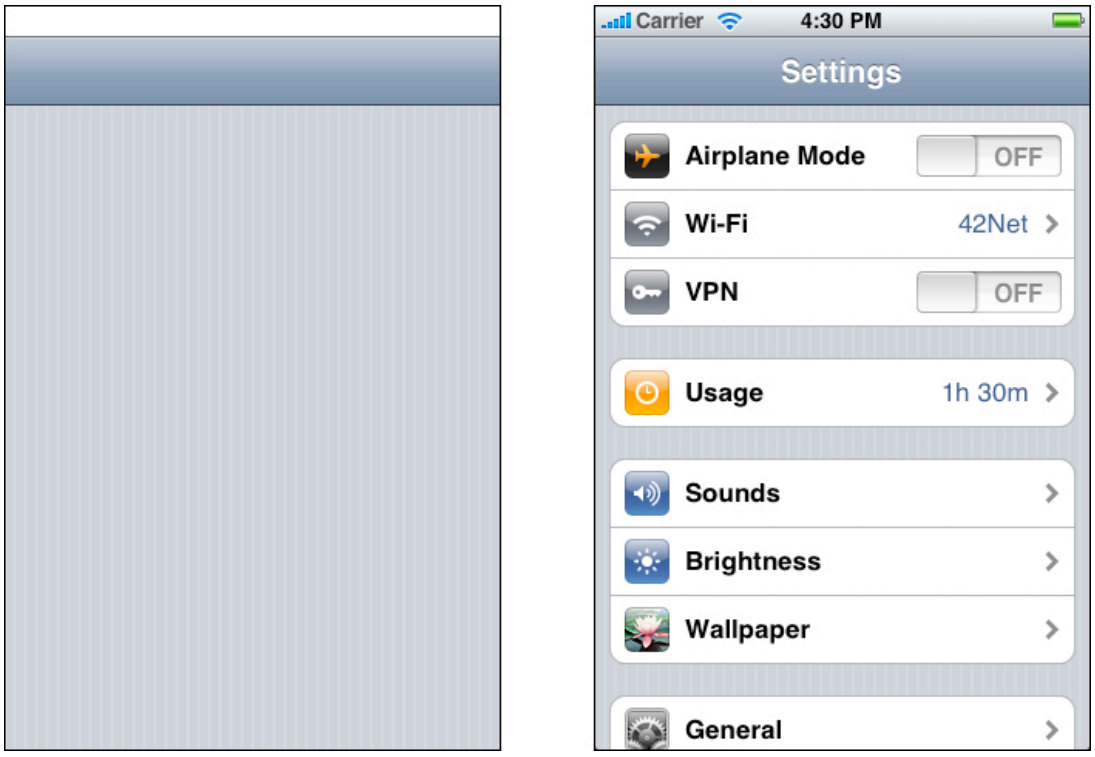
如果你认为遵循这个设计导引会产生一个非常平淡，乏味的启动画面，那么你对了。记住，启动画面不是意味着提供一个静态艺术效果的体验，它只是为了提高用户对应用程序能立即快速启动的体验。看看下面这些例子，了解启动画面能多无趣。

The first example is the launch image for the built-in Settings application, shown in Figure 11-3. The Settings launch image displays only the background of the application, because no other content in the application is guaranteed to be static.

第一个例子是内置应用程序的启动画面，如图 11-3。该启动画面只显示应用程序的背景，因为应用程序的其他内容都不是静态的。

Figure 11-3　The launch image for the Settings application

“设置”应用程序的启动画面

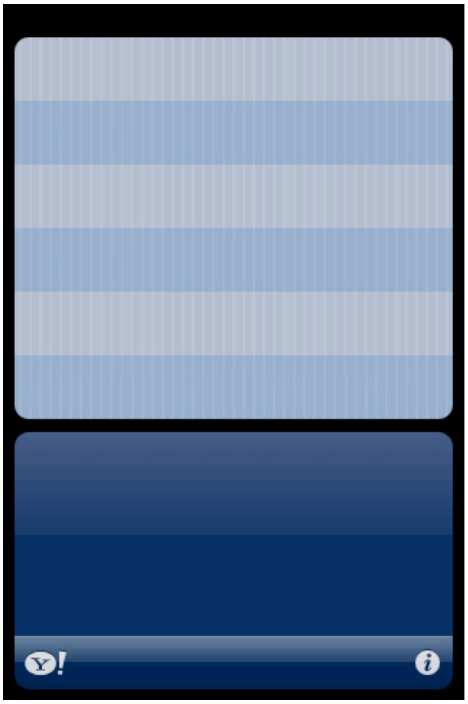


Another launch image example comes from the built-in Stocks application, shown in Figure 11-4. Note that the only images included in the launch image are static images, which are always visible in the front view of the Stocks application.

另一个启动画面的例子是来自内置的股票软件，如图 11-4。要注意的是，包含在启动画面在内的图像都是静态的，通常位于股票软件前部视图。

Figure 11-4 The launch image for the Stocks application

股票软件的启动画面



Icons for Navigation Bars, Toolbars, and Tab Bars

导航条，工具条和状态条的图标

When possible, you should use the system-provided buttons and icons in navigation bars, toolbars, and tab bars. iPhone OS provides a wide range of standard buttons and icons that users associate with standard tasks and modes supported in the built-in applications. If your application supports standard functions, such as refreshing a content-area view or deleting an item, or displays different subsets of data, such as contacts or bookmarks, be sure to use the appropriate system-provided button or icon to represent it. For a complete list of standard buttons and icons, and guidelines on how to use them, see *"System-Provided Buttons and Icons"* (page 123).

在导航条，工具条和状态条中尽量使用系统预置的按钮和图标。iPhone OS 提供了大量的在内置应用程序中使用并被用户熟悉地标准按钮和图标。如果你设计的应用程序能支持标准功能，例如刷新视图内容或者删除某一项目；或者显示数据地具体部分，例如联系人或书签。则应当使用恰当的系统预置按钮和图标来表示。 关于完整的标准按钮和图标及其使用方法。见*"System-Provided Buttons and Icons"*。

Of course, not every task your application performs is a standard one. If your application supports custom tasks users need to perform frequently, you need to create custom icons that represent these tasks in your toolbar or navigation bar. Similarly, if your application displays a tab bar that allows users to switch among custom application modes or custom subsets of data, you need to design tab bar icons that clearly describe these modes or subsets. This section

gives you some guidance on how to design icons that work well in navigation bars, toolbars, and tab bars.
当然，不是每一个应用程序执行的功能任务都是标准的。如果你设计的应用程序提供了用户频繁去执行的特殊操作，那么你需要自定义图标来显示在工具条或导航条上。同样的，如果你设计的应用程序在标签条上需要显示切换自定义程序模式或自定义数据集时，那么你就需要设计能够清楚描述这些模式和数据集的标签条图标。本节能够在如何设计出好的导航条图标，工具条图标和状态条图标上给你一些指引。

Before you create the art for your icon, you need to spend some time thinking about what it should convey. As you consider designs, aim for an icon that is:
在你为图标制作艺术效果之前，需要花一些时间去思考你的图标去传达些什么。你需要为你的设计做以下的思考：

- Simple and streamlined. Too many details can make an icon appear sloppyor indecipherable.
  简洁的，合理的。太多的细节会使得图标变的杂乱或者难以辨认。

- Not easily mistaken for one of the system-provided icons. Users should be able to distinguish
  your custom icon from the standard icons at a glance.
  不要使人误认为这是系统图标。用户应能在系统图标中一眼就辨别出你设计的图标。

- Readily understood and widely acceptable. Strive to create a symbol that most users will interpret correctly and that no users will find offensive.
  易于理解并能普遍被接受的。努力争取去定制一个能被大多数用户正确理解并不会感到厌恶的图标。

After you’ ve decided on the appearance of your icon, follow these guidelines as you create it:
当确定了你的图标外观后，继续遵循以下这些引导去制作它：

- Use the PNG format.
  使用 PNG 格式。
- Use pure white with appropriate alpha.
  使用纯白色伴随适当的 alpha 通道。
- Do not include a drop shadow.
  不加投影。
- Use anti-aliasing.
  使用抗锯齿。
- If you decide to add a bevel, be sure that it is 90°（to help you do this, imagine a light source positioned at the top of the icon).

如果你决定要增加一个斜角，那必须确保它是 90 度的（为了帮助你能做到这一点，想像光源在图标的顶部位置）。

- For toolbar and navigation bar icons, create an icon that measures about 20 x 20 pixels. For tab bar icons, create an icon that measures about 30 x 30 pixels.

为工具条和导航条定制的图标尺寸是 20x20 像素。为标签条定制的图标尺寸是 30x30 像素。

Note: The icon you provide for toolbars, navigation bars, and tab bars is used as a mask to create the icon you see in your application. It is not necessary to create a full-color icon.
注意：你为工具条，导航条和标签条提供的图标通常是作为图标的蒙版在你设计的应用程序中显示的。因此，没有必要去做制作一个全色图标。

iPhone OS automatically provides the pressed or selected appearance for items in navigation bars, toolbars, and tab bars, so you only need to provide a single version of an icon. Because these visual effects are automatic, you cannot change their appearance.
iPhone OS 自动为导航条，工具条和状态条上项目提供了按下与选中的小关外观，因此，你只需要提供一个图标即可。由于一些效果是自动添加的，所以你不能改变它们的外观。