

# iPhone 人机界面设计规范

iPhone Human Interface Guidelines

来源: UX Guide

整理人: Camel

整理日期: 2011-03-18

# 目录

- [简介](#)
- [第一部分：设计你的 iPhone 软件产品](#)
- [第二部分：为你的 iPhone 软件设计用户界面](#)

## 简介

iPhone 和 iPod touch 是融合了革命性的多点触摸技术和多种强大功能的复杂尖端设备，其功能包括电子邮件和即时通讯功能、全功能浏览器、iPod、以及——对于 iPhone 来说——移动电话。iPhone OS 是运行于 iPhone 和 iPod touch 上的系统软件。随着 iPhone SDK 的到来，其强大的功能进一步延伸，为开发人员提供了大量的机会。除了为基于 iPhone OS 的设备提供 Web 内容外，开发人员还可以通过 iPhone SDK 来创建本地应用程序，使人们能够在设备上直接储存和使用。

通过这篇文章你可以了解到，你可以为 iPhone OS 创建何种类型的应用程序，以及为优秀的 iPhone 应用程序奠定基础的人机界面设计原则。你将了解到如何在为 iPhone 应用程序设计精彩的用户界面和用户体验时遵循这些原则。无论你是否是一位经验丰富的计算机应用程序开发人员、还是经验丰富的移动设备应用程序开发人员、或者是这个领域的新手，本文中的准则都将帮助你创建出用户想要的 iPhone 应用程序。

**注：**本文概述了用于基于 iPhone OS 设计的基于 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 Web Applications》。

## 文章结构

《iPhone 人机界面设计规范》分为两个部分，各包含若干章节：

- 第一部分：[“设计你的 iPhone 软件产品”](#)。该部分讲述了 iPhone OS 的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创建你的 iPhone 应用程序。
- 第二部分：[“为你的 iPhone 软件设计用户界面”](#)。该部分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的准则。

## 第一部分：设计你的 iPhone 软件产品

iPhone 人机界面设计规范的这个部分描述了设计和开发 iPhone OS 软件的方法。通过这个部分的阅读，你将了解到可以为 iPhone OS 创建的软件类型以及你可以在工作中应用的设计准则。同时你将得知如何在软件的各个方面细化的应用这些准则，从而开发出具有漂亮易用界面的杰出产品。

- iPhone OS 平台：无尽的可能
  - 牢记平台间的差异
  - 你的选择是什么？
  - 三种应用程序样式
  - 选择一种应用程序样式
  - 当你已经有了一个计算机应用程序时
  - 案例研究：将桌面应用程序移植到 iPhone OS 上
- 人机界面原则：创建优秀的用户界面
  - 隐喻 Metaphors
  - 直接操纵 Direct Manipulation
  - 即看即点 See and Point
  - 反馈 Feedback
  - 用户控制 User Control
  - 美学集成度 Aesthetic Integrity
- 设计 iPhone 应用程序：从产品定义到品牌宣传
  - 创建产品定义说明
  - 优秀 iPhone 应用程序的共有特征
  - 适当地手势支持 Support Gestures riately
  - 慎重地体现品牌化元素
- 处理常见任务 Handling Common Tasks
  - 启动 Starting
  - 停止 Stopping
  - 管理设置或配置选项 Managing Settings or Configuration Options
  - 支持复制和粘贴 Supporting Copy and Paste
  - 支持撤消和重做 Supporting Undo and Redo
  - 启用推送通知 Enabling Push Notifications
  - 使你的程序具有无障碍特性 Making Your Application Accessible
  - 提供搜索功能并显示搜索结果 Providing Search and Displaying Search Results
  - 利用用户的位置 Using the User' s Location
  - 处理方向变化 Handling Orientation Changes
  - 使用声音 Using Sound
  - 提供选择 Providing Choices
  - 提供许可协议 Providing a License Agreement

## 第二部分：为你的 iPhone 软件设计用户界面

iPhone OS 的用户界面元素包括视图和控制。**视图**为内容提供定义良好的功能集。**控件**是引起即时行动或可视结果的图形对象。尽管所有的视图和控件都包含于应用程序窗口，用户在屏幕上察看它们并且与它们交互，这些**屏幕**大致对应于应用程序不同的视觉状态。

iPhone OS 定义这些用户界面元素的标准外观，并提供与用户期望一致的行为。阅读第二部分的各个章节，可以了解所有可用的用户界面元素类型以及如何使用它们来构建你的应用程序的用户界面。

- 应用程序用户界面简介 A Brief Tour of the Application User Interface
  - 应用程序页面和其中的内容 Application Screens and Their Contents
  - 在应用程序界面中使用视图和控件 Using Views and Controls in Application Screens
- 导航栏、标签栏、工具栏和状态栏 Navigation Bars, Tab Bars, Toolbars, and the Status Bar
  - 状态栏 The Status Bar
  - 导航栏 Navigation Bars
  - 工具栏 Toolbars
  - 标签栏 Tab Bars
- 警告、操作列表和模态视图 Alerts, Action Sheets, and Modal Views
  - 用法和行为 Usage and Behavior
  - 设计警告 Designing an Alert
  - 设计操作列表 Designing an Action Sheet
  - 设计模态视图 Designing a Modal View
- 表格视图、文本视图和 Web 视图 Table Views, Text Views, and Web Views
  - 表格视图 Table Views
  - 文本视图 Text Views
  - Web 视图 Web Views
- 应用程序控件 Application Controls
  - 活动指示器 Activity Indicators
  - 日期时间选择器 Date and Time Pickers
  - 详情展开按钮 Detail Disclosure Buttons
  - 信息按钮 Info Buttons
  - 标签 Labels
  - 页面指示器 Page Indicators
  - 选取器 Pickers
  - 进度视图 Progress Views
  - 圆角矩形按钮 Rounded Rectangle Buttons
  - 搜索栏 Search Bars
  - 分段控件 Segmented Controls
  - 滑块 Sliders
  - 文本框 Text Fields

- 系统自带的按钮与图标 System-Provided Buttons and Icons
  - 使用系统自带的按钮与图标 Using System-Provided Buttons and Icons
  - 用于工具栏与导航栏的标准按钮  
Standard Buttons for Use in Toolbars and Navigation Bars
  - 用于标签栏的标准图标 Standard Icons for Use in Tab Bars
  - 用于表格行及其他用户界面元素中的标准按钮  
Standard Buttons for Use in Table Rows and Other User Interface Elements
- 创建自定义图标和图像 Creating Custom Icons and Images
  - 应用程序图标 Application Icons
  - 小图标 Small Icons
  - 启动图像 Launch Images
  - 用于导航栏、工具栏和标签栏的图标 Icons for Navigation Bars, Toolbars, and Tab Bars

## 第一部分：设计你的 iPhone 软件产品

iPhone 人机界面设计规范的这个部分描述了设计和开发 iPhone OS 软件的方法。通过这个部分的阅读，你将了解到可以为 iPhone OS 创建的软件类型以及你可以在工作中应用的设计准则。同时你将得知如何在软件的各个方面细化的应用这些准则，从而开发出具有漂亮易用界面的杰出产品。

## 1.1 iPhone OS 平台：无尽的可能

iPhone OS 支持多种类型的软件，从使用 Safari 浏览的网页延伸至可在 iPhone 操作系统设备上自然 (natively) 运行的应用程序。本章概括了针对于基于 iPhone OS 设备创建软件的多种不同解决方案。

如果你是 iPhone 操作系统平台的新手，推荐从第一小节的《牢记平台间的差异》开始阅读。虽然这部分的信息不是很全面，但是它涉及到你设计一个 iPhone 应用必须要意识到的问题。

现在，为了帮助你规划一个 iPhone 应用程序，这一章讲述了如何思考不同应用程序的风格以及定义他们特点的方法。这一章还讲述了怎样将一些捆绑在 Mac OS X 上的应用程序转换为对应版本的 iPhone OS 应用程序。如果你要把现成的电脑应用程序再设计使之适用于 iPhone OS，理解这个过程是很关键的。

- [牢记平台间的差异 Platform Differences to Keep in Mind](#)
- [你的选择是什么？What Are Your Options?](#)
- [三种应用程序样式 Three Application Styles](#)
- [选择一种应用程序样式 Choosing an Application Style](#)
- [当你已经有了一个计算机应用程序时 When You Have an Existing Computer Application](#)
- [案例研究：将桌面应用程序移植到 iPhone OS 上  
Case Studies: Bringing a Desktop Application to iPhone OS](#)

### 1.1.1 牢记平台间的差异 Platform Differences to Keep in Mind

基于 iPhone OS 的设备并非台式计算机或笔记本，且 iPhone 应用程序也不同于桌面应用程序。虽然这些似乎只是常识性的说明，尽管如此，但当你基于这些设备开发软件的时候，最重要的是记住这些说明。

为 iPhone 操作系统的设备设计软件需要一种思维方式，这种思维方式或许不是你所习惯的。尤其是当你的大部分经验都来源于开发桌面应用程序，你应该意识到设计移动平台软件和设计电脑软件的显著性差异。

这一部分概述了对于你设计决策潜在影响最大的不同点。有关如何处理你的 iPhone 应用程序开发过程中各种问题，参见《[iPhone 应用程序编程指南](#)》（英文）的详细信息。

#### 紧凑的屏幕尺寸 Compact Screen Size

小巧，高分辨率的屏幕使 iPhone 操作系统设备成为具有强大显示能力且适合用户装进口袋使用的设备。对于用户来说这是个优势，但是对于你，开发人员来说这是个很大的挑战。因为这意味着你必须设计出一个不同于你习惯设计的用户界面。

要记住屏幕的尺寸只有 480×320 像素，所以要把重点放在用户界面必须具备的东西。你没有任何展示设计元素的空间，况且这些元素是完全没有必要的。拥挤的用户界面会使你的应用程序失去吸引力并且难以使用。

## 内存是有限的 Memory is Not Unlimited

内存是 iPhone 操作系统的重要资源,所以控制你的应用程序所占的内存是至关重要的。由于 iPhone 操作系统的虚拟内存模型不包括磁盘 交换空间,所以你必须小心,不要给你的应用程序分配过多的内存。当内存不足的情况发生的时候, iPhone 操作系统会对正在运行的程序发出警告,如果问题 依然存在的话可能会终止程序。请确保你的应用程序能够即时响应内存使用警报并即时清理内存。

当你设计应用程序的时候,要严格地尽量降低应用程序的内存占用,例如,消除内存泄漏,尽量压缩资源文件大小,缓慢的装载资源等。《iPhone 应用程序编程指南》中有大量有关如何设计 iPhone 应用程序使之合理的占用内存的资料。

## 同一时间只能显示一个页面 One Screen at a Time

iPhone 操作系统与电脑操作系统在操作环境上最大的不同是窗口的模式。除了一些模态视图之外,用户在 iPhone 操作系统设备上同一时间只能看到单一的页面。iPhone 应用程序能尽可能的包含所需要的不同的页面,但是用户是渐进方式看到它们,而不是同时看见。

如果你的应用程序对应的桌面版本要求用户同时浏览几个窗口,你就要决定是否有另外一种方法允许用户只打开一个页面或者打开一系列页面就可以 完成任务。如果不能的话,你就要把你这个 iPhone 应用程序的重点放在这个电脑应用程序的下属子功能上,而不是去试图复制一个广泛的功能集合。

## 同一时间只能运行一个应用程序 One Application at a Time

同一时间只能运行一个 iPhone 应用程序,并且第三方应用程序不会在后台运行。这意味这当用户切换到其他应用程序的时候,如接听电话或者 查看邮件等,他们现在之前所用的应用程序将会退出。确保用户不会因为遇到这种情况而感受到负面的影响是非常重要的,因为这是很自然的操作流程。换句话说, 不应该让用户感觉到退出 iPhone 应用程序稍后再打开,会比在电脑程序之间切换更困难。

为了确保用户能够拥有一个良好的程序切换体验,你能做的最有效的事情就是降低应用程序的载入时间。关于如何尽可能的减少程序载入时间请参阅 [《iPhone 应用程序编程指南》\(英文\)](#)。

一般情况下,用户通过切换到其他应用程序来退出你的应用程序,他们是不会做退出程序的确切行为。因此,不要期望用户会选择菜单中的“退出” 选项或者点击关闭按钮。这种使用习惯就意味着:你的应用程序有可能在没有任何预警的情况下被关闭,所以你要时刻准备好尽可能快的保存用户做出的改动。这样 做会使程序之间的转换快速而稳定,并且可以确保你的应用程序再次被打开的时候依然还是上次关闭前的状态。

这种单一应用程序模式的另一个重要方面是你处理用户定制的偏爱的应用程序。在 iPhone 操作设备中,用户在“设置”这个应用程序中设置选 项。你的 iPhone 应用程序也可以使用这种设置选项,但这意味着当用户想进行设置时,需进入“设置”应用程序,且必须退出你的应用程序。如果你遵循这个 准则并提供设置选项,用户只要用过一次,就会逐渐的习惯并再次使用,你的应用程序将会拥有一个流畅的用户体验。

## 有限的用户帮助 Minimal User Help



移动用户在使用你的应用程序之前没有时间去阅读大量的“帮助”内容。更何况你也不想放弃宝贵的屏幕空间去显示它们或存储这些内容。iPhone 操作设备的一个标志性设计就是易用,所以有一点很重要,满足用户的期望并使应用程序的用法一目了然。想实现这些你要做的有以下几点:

- 正确使用标准控件。用户已经对内置应用程序中的标准控件非常熟悉了,所以他们清楚的知道该如何使用它们。
- 确保通过你给出的信息得到的路径是符合逻辑且易于用户预知的。另外,确保提供像后退按钮这样的标记,这样用户可以知道他们在哪和如何追溯他们的操作步骤。

### 1.1.2 你的选择是什么?

在你决定怎样将你的产品呈现给 iPhone 操作系统的用户时,你需要了解你的选择范围。根据你提出的产品定义的实施细节及其目标人群,某些类型的软件可能比其他的软件更适合你的需求。

这一部分将基于 iPhone 操作设备的软件的实现方法划分成 3 个类型,粗略地说,您可以创建:

- **An iPhone application, iPhone 的应用程序**,这是一个使用 iPhone 软件开发包开发出的可在 iPhone 操作设备上完美运行的应用程序。
- **Web-only content, 仅在网站上的内容**,包括在线应用程序——一种表现与内置应用程序相似但实际上是网站的应用。
- **A hybrid application,混合应用程序**,这是一个主要通过网上浏览区域来提供接入网站内容的 iPhone 应用程序,但包括一些 iPhone 操作系统的用户界面元素。

### iPhone 应用程序 iPhone Applications

**iPhone 应用程序**类似与 iPhone 操作设备的内置应用程序,它们被预装到设备中,并能方便容易的使用 iPhone 操作系统环境中的功能。用户安装 iPhone 应用程序跟他们使用内置程序一样,如股票,地图,计算器,邮件等。

一个 iPhone 应用程序应该是载入飞快且易于使用的。无论这个应用程序能够完成能发送邮件或者是娱乐任务,它必须具有以下特点:响应快,简单,美观且流线型的用户界面。

### 仅用于 Web 的内容 Web-only Content

针对 iPhone 系统的用户,当说起仅在网站上的内容的应用程序的时候,你有一些不同的选择:

- **Web 应用**  
**Web applications** 网页能为完成一个任务提供一个集中的解决方案,并且符合某些已知的网络应用程序的界面规范,因为他们的操作表现和 iPhone 操作系统的内置程序很相似。一个网络应

用程序和所有仅在网站上的内容一样都要在 iPhone 上的 Safari 浏览器运行; 用户无需在设备上安装他们, 只需访问网络应用程序的网址即可。

- **优化的网页**

**Optimized webpages** 在 Safari 中, 网页的显示和操作, 都将有计划的被优化。(但不包括任何使用支持技术的元素, 如插件, Flash 和 Java)。此外, 优化后的网页将 缩放网站内容使之适合设备的屏幕, 并且网页应该检测当前是否正在 iPhone 系统内浏览[UI](#), 以便对应地调整内容。

- **兼容网页**

**Compatible webpages** 网页的显示和操作, 都将有计划的被兼容。(但不包括任何使用支持技术的元素, 如插件, Flash 和 Java)。一个兼容的网页并不用 更多的措施去优化阅读体验, 但是 iPhone 设备通常能够正常的显示他们。

如果你有一个现成的网站或者网络应用程序, 你首先要确保它能够在 iPhone 操作设备上良好的运行。同时, 你应该考虑创建一个个性图标, 用户通过粘贴你的 URL 地址即可把应用的快捷方式放至到桌面屏幕上。。实际上, 这样的话, 用户可以把你的应用的书签放置在桌面屏幕上, 且看起来像一个完美的 应用程序图标。想要更多有关创造自定义图标以及怎样使你在 iPhone 操作设备上运行的网站内容看上去更优秀, 请参阅《iPhone 网络应用程序的人机 界面指南》。

## 混合应用程序 Hybrid Applications

你可以在 iPhone 操作系统上创建一个结合本地程序以及网页功能的应用程序。**混合应用程序**是一个通过网络浏览区域提供大部分的结构和功能的本地 iPhone 应用程序, 但也包含了标准的 iPhone 操作系统的用户界面元素。

混合应用程序允许用户可以通过网络视图(参见 [Web 视图](#))元素获取到网络内容(了解“[网络视图](#)”)。而具体在你的应用程序中如何应用网络视图是你的责任, 但很重要的一点是避免让用户认为你的应用程序仅仅只是一个小型网络浏览器。混合应用程序的操作行为及其呈现方式应该像本地 iPhone 应用程序一样, 不应该只注意到它是取决于网络资源的事实。(所以就照搬网页的用户界面)

### 1.1.3 三种应用程序样式 Three Application Styles

这一段文档基于视觉, 行为特征, 数据模型以及用户体验定义了应用程序的三个样式。在你进一步阅读之前, 需要强调的是: 这些命名以及描述这些 特征只是为了帮助你清楚地确认设计方案, 但这不表示所有的 iPhone 软件必须严格的遵守这些分类计划, 这很重要。相反, 描述这些样式可以帮助你认清针对不同样式应用程序的信息与功能, 他们的设计方案是多么的不同。

**注:**请记住, 应用程序的风格并不指示程序实现方法(implementation method)。这文档专注于设计天然的 iPhone 应用程序, 而不是在这里探索是不是应用程序的风格可以再任意类型的软件上实现。

当你阅读这三个应用程序的样式时, 仔细思考如何把每一个可能增强应用程序功能组和用户体验的特点传递给用户。为了帮你发现最适合你的应用程序的特性搭配, 当你学习不同 iPhone 应用程序设计样式时, 请将几个问题默念在心中:

- 你期望用户是出于什么动机去使用你的应用程序？
- 你打算在使用应用程序的时候获得怎样的用户体验？
- 你的应用程序的目标或者说专注点在哪？
- 你的应用程序是怎样组织和显示那些人们关心的信息的？是否跟应用程序的主要任务有天然的关系？

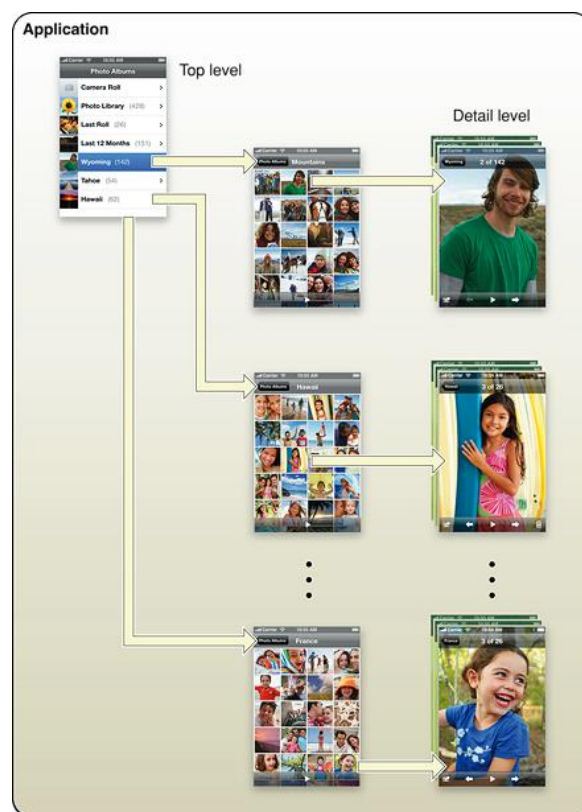
## 生产力应用程序 Productivity Applications

**生产力应用程序**可以完成那些对具体信息的组织与处理的任务。人们使用生产力应用程序是为了完成重要的任务。比如邮件是一个生产力应用程序的典型例子。

任务的严肃性并不意味着为了生产力应用程序应该试图表现得严肃，而提供一个枯燥的且不讨好的用户体验，相反的是，它（任务的严肃性）意味这用户会欣赏一个流线型且使用起来没有阻碍的用户体验。为了达到这个目的，生产力 应用程序应该专注于主任务的用户体验，以便人们能尽快找到他们所需求，容易地执行必要的动作，完成任务后再关注其他东西。

生产力应用程序通常会对用户数据分层次的组织。这样，人们可以通过逐步从大概到具体的选项中来搜索信息，直到他们达到想要那种详细级别。 iPhone 操作系统提供的表格元素，使这（搜索信息）一步骤在基于 iPhone 操作系统的设备上表现得极为高效( 更多有关这些用户界面元素的信息, 参见 [“表格视图”](#) )。图 1-1 显示了一个这种数据结构形式的例子。

图 1-1 生产力应用程序倾向于通过层级划分来管理信息



通常, 在一个生产力应用程序中用户交互模式包括:

- 组织列表
- 增加至列表或从列表中移除
- 逐步让信息更加详细直到达到目标的详细级别, 然后执行与该详细级别对应的任务。

生产力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。用户界面倾向于简单而整洁, 并且是由标准的视图和控件组成。生产力应用程序不太倾向于自定义界面, 因为程序的专注点应该是信息和任务, 而不是使用环境或操作体验。

在所有的 iPhone 应用程序中, 生产力应用程序是最有可能在设置应用程序中提供具体的用户可操作的选项或者设置功能。这是因为生产力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, 用户很少需要更改这些设置, 所以, 这里不应该包含那些简单的用户在主界面即可更改的基本配置。

### 实用应用程序 Utility Applications

**实用应用程序**能够完成那些简单的, 对用户的输入要求最低的任务。人们启动实用程序来快速查看概览信息, 或者是对 (屏幕中) 有限的几个对象简单操作。天气应用程序(见图 1-2)实用应用程序的典型例子, 因为它严谨地聚焦并呈现了一个易于浏览的概览信息。

图 1-2 实用应用程序举例: 天气



实用程序有吸引人的视觉效果, 在某种程度上, 这种效果加强了信息的呈现, 而不是遮蔽呈现。人们使用实用程序来检查一些事情的状态或者查找一些东西, 所以他们希望可以快速并且方便地定位他们感兴趣的东西。为了达到这一点, 实用程序的用户界面整洁有序, 提供简单的, 往往是标准的视图和控件。

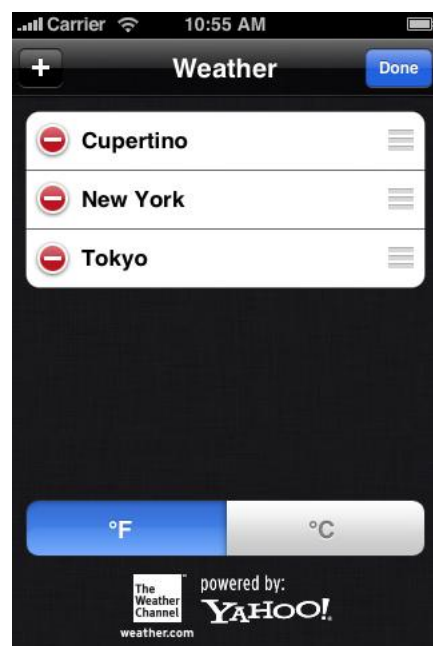
实用程序往往把信息组织成一个扁平<sup>[1]</sup>的项目列表；用户往往不需要逐级深入信息层次。通常，实用程序的每一个视图都提供相同结构的数据和细节深度，但（这些数据）可以是不同的来源。这样，用户可以打开单独一个实用应用程序，看到对多个主题的共同处理。一些实用程序标明了打开视图的数目；用户可以通过这些数字一个接一个地导航到这些视图。图 1-3 展示了这种数据组织的一个例子。

图 1-3 实用程序往往在一个扁平的列表里呈现数据



实用应用程序的用户交互模型很简单：用户打开应用程序来浏览信息的概览，然后可以选择更改信息的设置或者来源。实用程序可能需要支持频繁修改设置或者信息来源，因此常常在主视图的后面提供这些设置操作的集合。用户点一下主视图右下角熟悉的信息按钮来查看后面（的设置视图）。修改完毕后，用户点一下完成按钮回到正面的主视图。在实用程序里，主视图后面的配置选项是应用程序的功能的一部分，不是一组用户访问过一次之后，就很少或者再也不访问的偏好类型的设置。由于这个原因，实用程序不应该把特定设置放在设置应用程序里。图 1-4 显示了天气应用程序在主视图的后面提供了配置选项。

图 1-4 用户可以在天气应用程序后面调整设置





## 沉浸式应用程序 Immersive Applications

**沉浸式应用程序**提供全屏的, 有丰富视觉效果的环境, 专注于内容和用户对于内容的体验。人们常常用沉浸式应用程序来娱乐, 不管是玩游戏, 浏览富媒体内容, 或者执行一个简单任务。

显而易见, 游戏很适合这种 iPhone 应用程序类型, 但是, 你还是能想象一下沉浸式应用程序的特点如何去增强其它类型的任务。那些提供独特 的环境, 不显示大量文本信息, 鼓励用户专注的任务, 是沉浸式应用程序的一个不错的选择。例如, 有个应用程序复制了使用气泡水平仪的体验, 它在图形丰富, 全 屏幕的环境下工作得很好, 尽管它不符合游戏的定义。在这样的应用程序里, 就像在游戏里一样, 用户的注意力集中在视觉内容和体验, 而不是其背后的数据。图 1-5 显示了一个沉浸式应用程序的例子, 复制了真实生活体验, 可以完成一个简单任务。

图 1-5 沉浸式应用程序不一定要是一个游戏



**注意:** 虽然横屏的应用程序应该以主屏幕按钮在右侧的方向启动, 但是上图 1-5 所示的气泡水准仪应用程序恰恰在相反的方向启动。这是因为要确保设备边缘的物理按钮不干扰测量。更多启动准则见 [“开始”](#)。

沉浸式应用程序往往隐藏设备原有的界面, 取而代之的是自定义的用户界面, 因为增强了用户的融入应用程序的世界的感觉。用户希望沉浸式应用程序的体验的一部分是寻求和发现, 所以使用非标准的控件显得更为合适。

沉浸式应用程序可能要处理大量的数据, 但通常不组织和展示出来给用户顺序查看或逐级深入了解。相反, 沉浸式应用程序在游戏, 故事, 或体验这 几种情况下呈现信息。也是因为这个原因, 沉浸式应用程序通常采用自定义的导航方法以配合环境, 而不是使用像在实用应用程序和生产力应用程序中的标准的, 数据驱动的方法。

沉浸式应用程序的用户交互模型是由应用程序带来的体验决定的。虽然游戏不太可能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后面提供配置选项。

### 1.1.4 选择一种应用程序样式

在了解了生产力应用程序、实用程序和沉浸式应用程序样式后，考虑一下你的应用程序显示的信息类型和所能完成的任务。理论上说，你创建的应用程序类型是对于你来说是显而易见的，你可以开始动手做了；但在实践中，并不总是那么简单。这里有一个假设的情景来帮助你考虑你的决定。

如果你想探索一个主题，思考与该主题相关的对象和任务。想象一下人们对于这个主题的不同看法。拿棒球做一个例子。棒球使我想起很多，其中有 球队，比赛，统计数字，历史和球员。棒球作为一个应用程序的主题可能太宽泛了，所以让我们只考虑球员。现在想象一下如何创建一个关于球员的应用程序——例如，使用棒球卡上他们的肖像。

你可以开发一个 productivity application，帮助那些认真的收藏家管理他们的棒球卡收藏。使用列表的格式，你可以在一个层次等级里显示卡片，首先是球队，然后球员，再然后赛季。在最详细的视图里，你可以使用户能够记录他们从哪里获得了卡片，花了多少钱，卡片目前的市场价值是多少，他们有多少张拷贝。由于这个应用程序的重点是 定义那些收藏的数据，用户界面精简了寻求和添加信息的任务。

你也可以开发一个 utility application，显示特定的棒球卡目前的市场价值。每个视图都可以看起来像一张棒球卡，上面添加了它目前的价值，并且可以允许用户在主视图的后面 选择特定的卡片来追踪和显示。这个应用程序的重点是单张的卡片，因此，用户界面强调卡的外观，并为用户寻找新卡片提供一两个简单的控件。

或者，当然，你可以开发一个游戏。也许游戏将着重于用户对于单张棒球卡的某些统计数据的知识，或者对著名卡片的识别能力。也许只是在另一种 类型的游戏里使用棒球卡，比如滑动拼图。在这些情况下，应用程序的重点是在棒球卡的图像和游戏。用户界面配合这一点，展示一些以棒球为主题的控件，并隐藏 iPhone OS 的用户界面。

需要重申的是，不要局限于一个应用程序样式。你可能会发现，对于你的应用程序的想法得益于不同应用程序样式的特性的组合。

如果有疑问，就简化问题。把功能列表削减到最少，并创建只完成一个简单任务的应用程序（参见 [“创建产品定义声明”](#) 就如何集中你的应用程序给出的建议）。当你看到人们如何使用并响应应用程序，你可以选择创建一个稍微转移了焦点，或更改了显示的应用程序的另一个版本。或者，你可能会发现用户需要一个关于同样主题的更多（或更少）细节的版本。

### 1.1.5 当你有一个现成的电脑应用程序

如果你已有一个现成的计算机应用程序，别只是将其硬塞进去 iPhone 系统。用户在使用基于 iPhone 操作系统的设备的时候，与他们平时使用桌面或是手提电脑有着很大的不同，并且对用户体验有着不同的期望。

请记住,用户在使用基于 iPhone OS 的设备的时候,经常是处于一种移动的,并且是充满干扰的环境。这通常意味着他们希望能够在打开应用程序后,迅速而便捷的完成当前的任务,从而可以进行 别的活动。如果你的应用程序需要用户长时间的集中注意力才可以完成任务,当你打算把他移植到 iPhone 上面时,你最好重新审视一下这个现成的应用程序的信息结构和任务目标。

如果你的桌面应用程序允许用户完成一个或者多个连串的复杂任务,请研究用户是如何使用它的,以便找出一些用户希望在移动终端上也能够处理的 小任务。例如,一个商务应用程序,可以支持项目计划、结算和支出报告功能。应用于 iPhone 的平台上,就可以转化为一个 iPhone 实用应用程序用来帮助用户查看项目进程摘要,或者是一个 iPhone 生产力应用程序来帮助用户跟踪相关的支出信息。

如果你正在思考如何将桌面应用程序置入 iPhone ,在设计你的应用程序时,可使用 80-20 原则。它的意思是绝大部分的用户(至少 80% )只会应用到一个程序的非常有限的功能,只有很少的用户(不超过 20% )才会用到所有的功能。然后,仔细考虑下你的应用程序是否需要提供那些尽管很强大但只有少数用户需要的功能。你要清楚的认识到的,桌面应用程序可能更 适合来提供这些功能,因此最好让你的 iPhone 应用程序专注于绝大多数用户都会需要的核心功能。

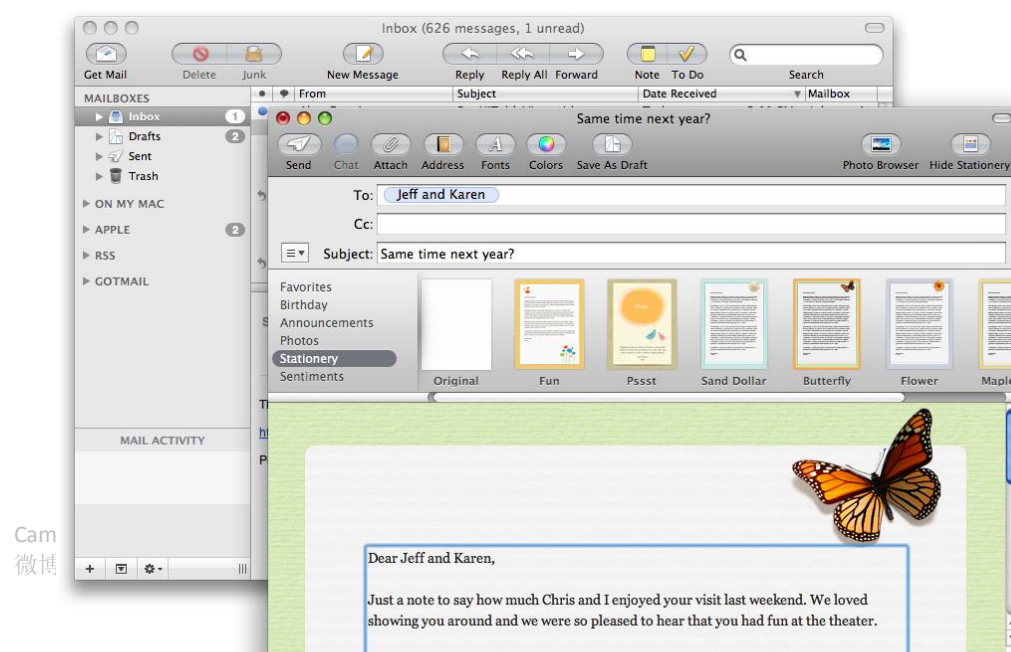
### 案例分析:将桌面应用程序移植到 iPhone 操作系统

为了帮助你把创建一个桌面应用程序的 iPhone 系统版这一过程可视化,本节介绍一些常见的苹果 Mac 操作系统中的应用程序与其对应的 iPhone 操作系统版在设计上的差异。当你了解了每个应用程序中哪些特点和功能是适合 iPhone 操作系统的时候,那么,当你在开发你的 iPhone 应用程序的时候,你将可以洞悉到哪种设计方案是你应该做的。

#### Mail

在 Mac OS X 中,Mail 是最明显的,被充分使用并且受用户喜爱的应用程序之一。它也是一个非常强大的程序,允许用户撰写,收取,分类和存储电子邮件,跟踪每项任务和事件,创建备注和邀请。Mail 在一个由多面版的窗口中提供了其大部分的功能。这对于使用台式电脑的人很方便,因为他们可以一直把 Mail 窗口保留在屏幕内(或最小化到任务栏)并可在任意时刻切换回来。图 1-6 显示了在桌面上,邮件阅读窗口和撰写窗口中提供的许多功能。

图 1-6 桌面 Mail 应用程序在两三个窗口内提供了许多不同的强大的功能



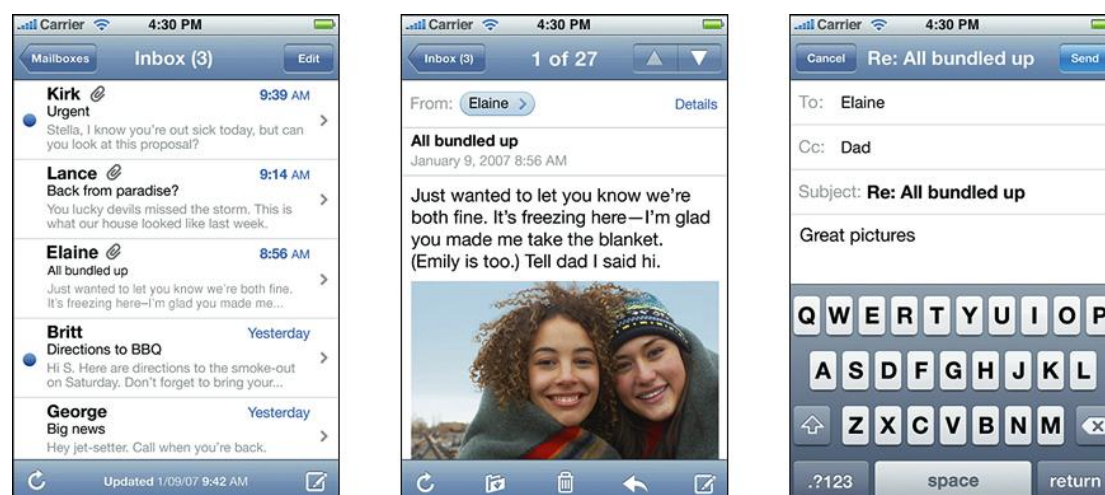


但是,当人们在移动时,他们需要一个更加简单且能够快速完成核心功能的电子邮件应用程序。因此,在基于 iPhone 操作系统的设备中,邮件应用程序专注于那些对于用户最重要的功能:收取,撰写,发送和管理邮件。要做到这一点,iPhone 的邮件系统采用了一个大大简化的用户界面,让帐户和邮箱的管理更加简洁,让用户用户的注意集中到邮件上。

邮件是 iPhone 操作系统中生产力应用程序完美的例子:为了让用户在内容之间的导航更加方便,iPhone 操作系统的邮件引用程序利用了邮件自然分级管理的优势,使用一连串的页面显示账户,邮箱,邮件列表,邮件具体内容。用户逐级深入,通过点击某项进入与该项相连的信息这种页面流方式,从总体(帐户列表)到具体(邮件内容)。要了解更多有关 iPhone 生产内容风格的应用程序,请参阅[“生产力应用程序。”](#)

此外,在 iPhone 操作系统中,邮件应用程序允许多种动作,如创建和发送,它通过显示几个容易被轻触且熟悉的控件来完成。图 1-7 显示了在 iPhone 操作系统内,邮件应用程序是如何让查看与发送电子邮件变得简单。它注释了,位于每个屏幕上方的视觉元素是如何让用户更容易获知到在应用程序中他们当前与上一步的位置。

图 1-7 邮件应用程序让查看和发送电子邮件在 iPhone 操作系统中变得简单



## iPhoto

另外一个从 Mac 操作系统改造至 iPhone 操作系统的应用程序的教科书般的例子是 iPhoto。在桌面端中,iPhoto 支持全面的搜索和管理,强大的编辑能力,以及富有创意的打印选项。当人们在台式电脑或手提电脑上使用 iPhoto 时,他们十分乐于浏览并组织他们全部的照片,调整照片,并且通过多种方式进行操作。尽管 iPhoto 的主要关注点是用户的内容(照片),它在其窗口内也提供了丰富的功能。图 1-8 显示了桌面上的 iPhoto 的用户界面。

图 1-8 iPhoto 的用户界面

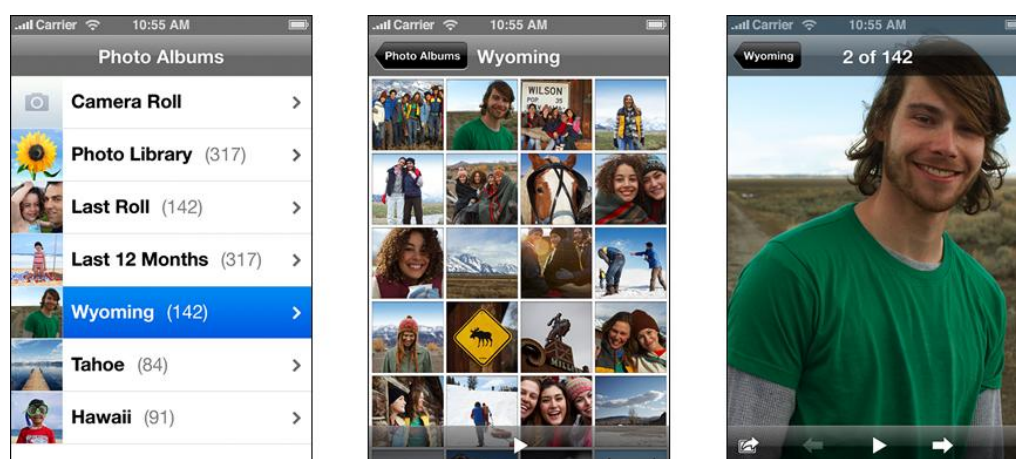


但是,当他们(用户)在移动时,人们没有时间来编辑自己的照片(而且他们也不希望打印出来);相反,他们希望能够快速地浏览和分享他们的照片。

为了满足基于 iPhone 操作系统的设备对这一功能的需求,苹果公司提供了照片应用程序,其主要功能在于浏览和与他人分享照片。这一照片应用程序的用户界面,完全以照片为核心,这种程度,事实上强烈到可以允许部分用户界面隐藏。所以,当用户选择以幻灯片的方式查看他们的照片时,照片应用程序会隐藏导航栏,工具栏,甚至是状态栏。当用户需要看到这些工具栏时,这些元素又以半透明状态显示出来。

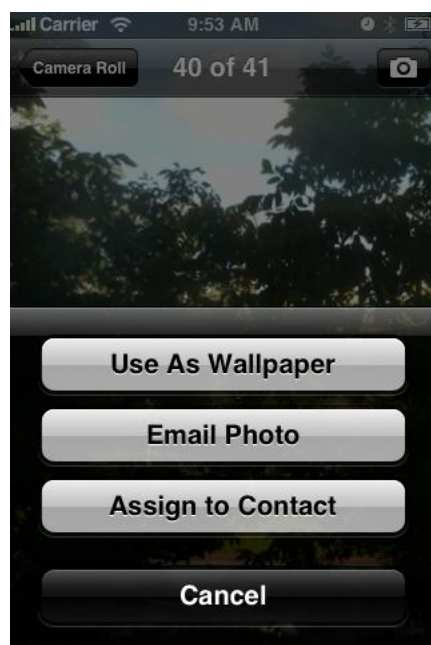
照片管理程序通过分级标记的方法使得用户很容易组织和查找他们的照片:用户先选择一个包含很多照片的专辑,然后从专辑中选择一张照片。从这个意义上讲,照片管理程序是一个有代表性的应用程序,它结合了生产力应用程序风格和沉浸式应用程序风格(想要了解更多关于这些样式的信息,请参阅“[三种应用程序样式](#)”)。

图 1-9 照片管理程序的三个界面



此外,照片管理程序提供短暂的预览,称为操作列表(详情查阅[“警告、操作列表和模态视图”](#)),这样,用户不退出照片浏览模式就能使用一些附加功能。图 1-10 显示了照片管理程序提供选项使用具体的单独照片。

图 1-10 照片管理程序在一个操作列表中给出的用户选项



## 1.2 人机界面原则：创建优秀的用户界面

一个优秀用户界面应该遵循一些人机界面设计原则,这些原则来源于人——用户——思考和工作的方式,而不是设备的性能。如果一个用户界面没有吸引力,令人费解,或不合逻辑,那么即使是一个伟大的应用程序也会因此让用户觉得烦琐。但是,一个美观的,无需用户思考且引人入胜的用户界面不仅可以改善应用程序功能,还能激发用户对应用程序产生正面情感和依赖感。

即便你已经熟悉这些基本的原则,你也应该阅读本章,因为本章的重点是讲如何在 iPhone 应用程序中体现它们。

- [隐喻 Metaphors](#)
- [直接操纵 Direct Manipulation](#)
- [即看即点 See and Point](#)
- [反馈 Feedback](#)
- [用户控制 User Control](#)
- [美学集成度 Aesthetic Integrity](#)

### 1.2.1 隐喻 Metaphors

如果可能, 将你的应用程序的对象和操作按照现实世界中的对象与操作仿造。特别是对于新手用户, 这项技术能够帮助他们快速领会你的应用程序是如何工作的。文件夹是一个经典的软件隐喻。在现实世界中人们把文档放到文件夹中, 所以他们很快会明白把数据放到计算机的文件夹中。

隐喻在 iPhone 操作系统中包括, iPod 的回放控制, 轻触控制以触发事件, 滑动解锁开关, 快速查看在选取器滚轮内的数据。

尽管隐喻暗示了各种对象和操作在 iPhone 操作系统界面中的用途, 但是这些用途并不限制隐喻在软件中的实施。再回到文件夹这个例子, 在软件中一个文件夹的存储能力与其在真实世界中的物理容量完全无关。

当您设计您的应用程序时, 请认识到 iPhone 操作系统存在的隐喻, 并且不要对这些隐喻重新定义。同时, 检查你的应用程序执行任务时的表现, 看看是否有自然的隐喻可以使用。请记住, 相比为您的应用程序单独增加一个合适的符合真实世界的对象和操作的界面, 使用标准的控件和操作反而更好。除非你选择的隐喻能被您的绝大部分用户理解, 包括那些会增加而不是减少用户困惑的隐喻。

### 1.2.2 直接操纵 Direct Manipulation

直接操纵意味着人们觉得他们都在控制切实的而不是抽象的东西。遵循直接操纵的原则的好处是, 用户更容易了解他们的操作对于操纵对象所产生的结果。

因为其多点触摸界面, iPhone 操作系统的用户享受着高度的直接操纵的愉悦感受。使用手势操作让用户对他们在屏幕上看到的对象, 感觉上更具亲和力, 和控制感, 因为他们不必使用任何中间设备(如鼠标)操纵它们。

为了提高你的 iPhone 应用程序中的直接操纵感, 请确保:

- 当用户对在屏幕上的对象执行操作时, 要保持他们可见
- 用户的操作结果是立即地显而易见

### 1.2.3 即看即点 See and Point

iPhone 应用程序是比一个人更善于记忆一些选项列表, 命令, 数据等。利用这个优势, 以列表的形式给用户选择或选项, 所以用户可以方便地扫描然后做选择。尽量减少文本输入使用户不必花费大量的时间打字, 这样也避免了应用程序执行错误检查。



提供选择给用户,而不是要求更多开放性的输入,同时,允许可以让用户集中精力在你的应用程序完成任务,而不是去记忆如何操作它。

### 1.2.3 反馈 Feedback

用户除了要看到他们操作的结果,还需要得到操作控件时的即时反馈,以及在长时间操作中的状态报告。您的应用程序应提供明显的变化来响应每个用户操作。例如,确保列表选项时在用户点击时暂时的高亮。声音提示也很有效,但它不能成为主要或唯一的反馈机制,因为人们可能会在无法听到声音,或他们必须关闭声音的地方,使用携带 iPhone 操作系统的设备。况且,您也不想与 iPhone 操作系统的声音进行竞争,因为用户常把它认为是警告。

当系统暂时繁忙时,iPhone 操作系统自动地提供操作反馈:显示活动指示器;当系统操作持续超过好几秒时,你的应用程序应显示消耗时间进度,并且,如果有的话,显示解释性信息。

动画是向用户提供反馈的一种良好方式,它微妙而富有含义。动画贯穿于整个 iPhone 操作系统中,甚至在非沉浸式应用程序中。但作为提供反馈的一种手段,动画只是常用于提高用户体验,它本身并不是用户体验的焦点。

### 1.2.4 用户控制 User Control

让用户,而不是您的应用程序,去触发并控制操作。保持操作的简单和直接,因此用户能够很容易地理解并记住他们。尽可能地使用标准的控件和用户已经熟悉的行为。

在操作执行之前,应提供充分的取消操作的选择,确保当用户触发一个潜在危险的操作时,与用户进行确认。尽可能地使用户能够舒服地中止一个操作的执行。

### 1.2.5 美学集成度 Aesthetic Integrity

尽管一个应用程序的最终目的是为了完成某个任务,甚至可能只是玩游戏,我们也不应低估应用程序外观的重要性。这是因为外观对功能有很大的影响:如果一个应用程序看起来很混乱或不合逻辑的话,也必然是很难理解和使用的。

美学集成度并不是一个用于衡量您的应用程序的美丑。它是用于衡量外观与功能相结合的优劣。例如,一个生产力应用程序应该在背景处保持微小且精致的装饰元素,用于突出由标准控件与行为组成的任务。

一个沉浸式的应用程序是刚好实在美学集成度的另外一边，用户期望一个美丽的外观，保证乐趣和鼓励发现。尽管一个沉浸式应用程序侧重于以提供 消遣为焦点，它的外观仍然需要和任务结合起来。要确保当你在设计一个这样的应用程序的用户界面元素时，保持谨慎，这样它们才能提供给用户一个内部的完整且 一致的体验。

## 1.3 设计 iPhone 应用程序：从产品定义到品牌化

当开发 iPhone 应用程序的时候，你需要了解 iPhone 操作系统和移动手机环境的各个方面给你的设计方案的影响。本章涵盖了一系列 iPhone 应用程序设计议题的规范，从产品定义到品牌化，以及在一个 iPhone 应用程序如何去运用这些规范。

- [创建产品定义说明 Create a Product Definition Statement](#)
- [优秀 iPhone 应用程序的共有特征 Incorporate Characteristics of Great iPhone Applications](#)
- [适当地手势支持 Support Gestures Appropriately](#)
- [慎重地体现品牌化元素 Incorporate Branding Elements Cautiously](#)

### 1.3.1 创建产品定义说明 Create a Product Definition Statement

在你开始设计自己的应用程序之前，必不可少的是精确定义你的应用程序是做什么的。一个比较好的方式是用心撰写一份**产品定义说明**——一份简洁描述你的应用程序的主要目的与目标用户的文档。创建产品定义声明并不仅仅是个练习，相反的是，这是将一大串功能特征转化为一个连贯性产品的最佳方式。

首先，花些时间定义你的目标用户：他们是专家还是新手，是正式的还是随意的，是为了达成目标而寻求帮助还是在寻求娱乐。了解用户的这些事情可以帮助你定做符合他们独有需求的用户体验与用户界面。

因为你是创建一个 iPhone 应用程序，你已经了解许多关于你的用户的信息。比如：

- 他们是在移动的
- 他们希望能够尽快的打开应用程序并立即看到对他们有用的信息
- 通过你的应用程序，他们只需简单点击几下即可完成事项。

现在思考下，是什么样的特性可以将你的用户群和其他所有 iPhone 操作系统用户群区分开来。他们是商人、青少年还是退休人员？他们是否会在每天结束前，或每次检查邮件的时候，亦或任何有一些空闲时间的时候，使用你的应用程序？你对目标用户的定义越精确，你在用户界面的外观、感觉和功能性方面做的决策就会越精确。

举例来说,如果你的程序是要帮助商务人士跟踪记录他们的开支,那么你的界面就要专注于提供正确的分类,并易于输入的开支数目,避免去问一些与核心任务无关的细节。除此之外,你最好为界面选择一种微小精致的颜色搭配方案,令其看起来显得专业且保证用户在一天使用多次仍感觉愉悦。

或者,如果你的应用程序是以青少年为目标用户的游戏。那么你就需要换成一种令人兴奋的,语言能表现的独一无二的,颜色搭配方案符合当前时尚的用户界面。

最后,来计算下你打算提供的功能特征集合。在脑海中想象着你的目标用户,试着将功能特征列表提取到一份单独的说明上,这就是一份产品定义说明,它描述了你的产品可以提供的解决方案和你的目标用户。例如,桌面版的 iPhoto 应用程序,除了别的之外,还允许用户整理,编辑,共享,打印和查看照片。但一份好的产品定义声明不仅仅是只把重点放在功能特征上面,它还要描述你的目标用户。因此,一份健全的 iPhoto 产品定义说明可以是“一个为业余摄影爱好者设计的简单易用的图片管理应用程序”。注意一下在产品定义说明中描述你的目标用户是多么重要:你只要想象一下如果 iPhoto 是“为专业摄影爱好者设计的图片管理程序”,那将有多大的不同。

一份优秀的产品定义说明就像一个工具,你应该在整个产品开发进程中使用它来判断提供的功能特征,工具和术语是否合适。他尤其重要的作用是排除那些不符合产品定义说明的元素,因为 iPhone 应用程序没有多余的空间提供给那些不是已主任务为核心的功能。

想象一下,例如,你想开发一款当用户在杂货店购物的时候可以使用的 iPhone 应用程序。在计划阶段,你可能需要考虑到许多各种不同的用户可能去做的活动,如:

- 获得特定事物的营养信息
- 寻找电子优惠券或特价优惠
- 创建和使用购物列表
- 定位商店
- 查阅食谱
- 比较价格
- 随时查看购物总金额

然而,你相信你的用户最关心的是要记得他们想买的每一样东西,同时尽可能的省钱,同时他们很可能是急着想把东西买回家。根据这些对用户的定义,你可以为你的应用程序制作一个产品定义说明,如“为匆忙的人提供的一个可创建购物列表并查找折扣信息的工具”。使用这份产品定义说明,去过滤你创建的潜在功能特征列表,你决定首要的应该是专注于让购物列表容易创建,保存和使用。同时,你还提供给用户查找与购物列表上商品对应的折扣信息的功能。即使其他功能也是有用的(并且可能成为其他应用程序的主要功能特征),但是他们不符合这个应用程序的产品定义说明。

当你确立了一个固定的产品定义说明并且已经开始把它作为过滤器去筛选你的提议的功能特征,你大概也希望用它来确定关于你最初的应用程序样式的决定也是正确的。如果你在脑海中计划开始了一个特定应用程序的开发进程,你可能会发现定义产品定义声明的过程改变了全局。(关于你所能开发的不同类型的应用程序请参见[“三种应用程序样式”](#))

### 1.3.2 优秀 iPhone 应用程序的共有特征

优秀的 iPhone 应用程序提供给用户需要的用户体验的, 同时又精确地满足了用户的需求。为了帮助你在你的应用程序中取得这样的平衡, 本节调查分析了一些优秀的 iPhone 应用程序特性, 并提供如何在你的产品建立这些特性的建议。

#### 基于简单性和易用性建立 Build in Simplicity and Ease of Use

简单性和易用性是所有类型软件设计的基本原则, 但在 iPhone 应用程序中他们极度重要。iPhone 操作系统的用户在使用你的应用程序的同时, 很可能正在干其他事情。如果用户不能很快弄清楚如何使用你的应用程序的话, 他们很可能会转而使用其他竞争对手的 应用程序, 并且不会再尝试你的应用程序。

当你设计你的应用程序的流程及其用户界面的时候, 清遵从以下规范来基于简单性和易用性建立:

- 使你的应用程序的使用方法显而易见。
- 浓缩常用的与高级别的信息, 并把它放置为靠近屏幕顶部的位置
- 最少的文字输入
- 简洁地表达基本信息
- 为所有可以点击的元素提供一个指尖大小的目标区域

以下每个小节将在更多细节上介绍简单性和易用性中的每个规范。

#### 显而易见 Make It Obvious

你不能假设用户有多余的时间 (或有多余的注意力) 去探索你的应用程序如何工作。因此, 你应该努力让你的应用程序能立即被用户所理解。

你的应用程序的主要功能应该是一目了然的。为了达到此目的, 你应该将用户不得不用的那些控件的数量减到最少, 同时使用清晰地用文本标签描述 它, 这样用户就能理解他到底在做什么。例如, 在内置的秒表功能中 (钟表应用程序的一部分), 如图 3-1 所示, 用户一眼就可以看到哪个是停止按钮, 哪个是启动秒表的按钮, 以及哪个是记录每圈时间的按钮。

图 3-1 内置的秒表功能的使用方法显而易见





## 从上到下思考 Think Top Down

人们可使用他们的手指或者拇指来点击基于 iPhone OS 设备的屏幕。当使用手指的时候,人们倾向于用非惯用手握着设备(或者放在其表面),然后用惯用手的手指来点击屏幕。当使用拇指的时候,人们会用手掌握住设备,然后用(同一只手的)拇指点击,或者将设备握在双手中间,使用双手的拇指点击。但是无论人们使用哪种方式,屏幕的最上方总对于用户而言,都是最显眼的。

由于有这些使用模式,在设计你的应用程序的用户界面时应该将常用的与高级别的信息,并把它放置为靠近屏幕顶部的位置,因为那是最显眼和易于访问的。当用户从屏幕顶部浏览到底部的时候,信息的显示应该是一个从概括到具体,从高级到低级的过程。

## 最少的文字输入要求 Minimize Required Input

不论用户是点击控件或使用键盘,输入信息都会占用他们的时间和精力。如果你的应用程序要求用户输入一大堆信息才可以提供一些有用的东西,那么它就会降低用户的效率并使他们对继续使用你的应用程序感到气馁。

当然,你通常需要用户提供一些信息,但是你应该在它与你提供给用户的信息之间取得平衡。换句话说,对于用户提供的任何一小块信息,都努力争取返回尽可能多的信息或者功能性的东西。这样一来,当用户使用你的应用程序的时候他们会感到一直在取得进展,没有被拖延。

当你要求用户输入信息的时候,考虑使用表格视图(或选择器)来代替文本输入。对于用户来说从列表中选择一项比键入单词要容易的多。关于表格视图以及选择器的更多细节,请分别参见[“表格视图”](#)以及[“选择器”](#)。

## 简洁地传递信息 Express Information Succinctly

当你用户界面上的文本简短且直接的话,用户可以很快很容易的理解。因此,确定什么是最重要的信息后,简洁的表达它们,突出显示它们,这样用户无需为了找到想要的信息或者是下一步该做什么而去阅读大段文字。

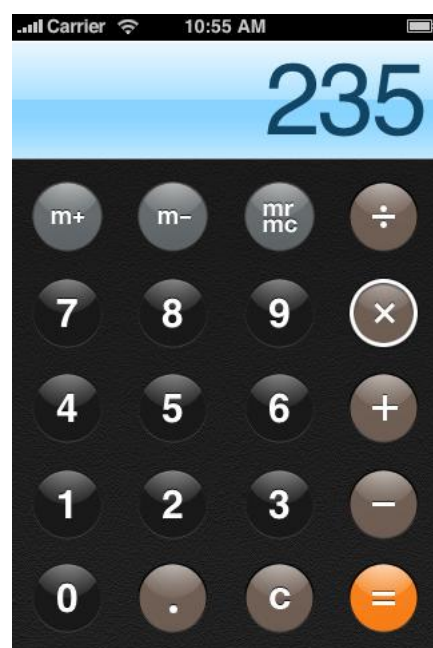
为了帮助你做到这点,像一个报纸编辑那样思考并且努力用压缩标题那样的方式去表达信息。为控件加上简短的标签(或者使用易于理解的符号),这样用户只需看一眼即可知道如何使用。

## 提供指尖大小的目标 Provide Fingertip-Size Targets

如果你将控件都布局挤在一起的话,用户就得小心的在他们点击的地方,花费更多的时间和注意力,并且,他们极容易点击错误。一个简单,易用的用户界面应在控件与其它的用户交互元素之间保留空间,以使用户只需最小的精力即可精确地点击。

例如,内置的计算器程序的按键显示着大且易用的控件,每个的目标大概是 44x44 像素。图 3-2 是内置的计算器程序。

**图 3-2** 内置计算器显示着指尖大小的控件



### 将重点放在主要功能上 Focus on the Primary Task

一个围绕首要功能特征为理念创建和坚持的 iPhone 应用程序是令人满意的，且使用起来令人感到愉悦。所以，当在设计你的应用程序时，要始终以你的产品定义说明为核心，并确保产品中每一项功能特征和用户界面元素都是与其相符的。关于如何创建一个产品定义说明请参见 [“创建产品定义说明”](#)。

一种实现重点的好方式是，决定在每个上下文中，什么是最重要的。当你决定每个单独页面要显示什么的时候你要不停问自己：这是用户现在需要的 最关键的信息或功能性吗？或者在具体条件下想一想，这个信息或功能性是用户在商场购物的时候，或赶着会议的时候所需要的吗？如果不是的话，判断一下这个信息或功能性是否属于另外一个上下文的关键信息，抑或是它根本就不重要？例如，一个记录汽车里程的应用程序如果同时提供记录汽车经销商地点的功能的话，它就失去重点了。

当你遵循这些规范让你的应用程序变得简单与易用的同时，你实际上是让你的解决方案更加聚焦集中了。特别是当你想让使用你的应用程序的变得简单明了并且最小化用户输入时。（遵循这些规范）会使用户更简单的快速到达你应用程序的最重要的那部分，并且这部分是紧密的以你的解决方案为核心的。（关于 这些规范的细节问题，请参见 [“基于简单性和易用性建立”](#)）。

例如，内置的日历程序（参见图 3-3）将着重于日期及其所发生的事情上。用户可以使用那些已经被标识清晰的按键去用标记日期，选择查看选项，添加事件等。最重要的信息，是，日期和与它相关的事件，它们是最显眼的。用户输入同样很简单，它允许用户选择从事件时间列表，重复间隔，提醒选项中作出选择，而不是要求用户全部使用键盘输入。

**图 3-3** 内置日历的着重于在日期和事件上



### 有效地沟通 Communicate Effectively

沟通和反馈在 iPhone 应用程序与桌面电脑应用程序中同样很重要。用户需要知道他们的请求是否正在处理中以及他们的行为是否会导致数据丢失或其他的一些问题。尽管如此,我们还是 应该避免过度沟通,例如,避免在并不是很严重的情况下出现警告提示或者过多的出现确认提示。

动画是一个很好的有效地沟通方式,但前提是它不会出现在用户任务中或者拖慢他们。微小精致且适当的动画可以传达状态,提供有用的反馈,并帮助用户可视化他们操作的结果。但是过多或者无意义的动画会阻碍应用程序的操作流程,拖慢其性能并惹恼用户。

对于所有与用户基于文本的沟通,要确保使用以用户为中心的术语,特别要避免在用户界面上使用技术行话。通过你对用户的了解,来判断你要使用的词语或短语是否合适。例如,在 WIFI 网络选项页面中使用清楚,非技术性的语言来描述设备是怎样连接网络的,参见图 3-4。

图 3-4 在你应用程序的用户界面中使用以用户为中心的术语



### 1.3.3 适当地手势支持 Support Gestures Appropriately

人们使用他们的手指来操作基于 iPhone 操作系统的设备所具有的独特多点触摸界面,使用点击,滑动,捏的方式去选择,导航,和阅读网页内容与使用应用程序。使用手指操作存在真正的优势:他们支持许多不同的动作,同时他们给用户一种直观的与设备关联的感觉,而这对于外接输入设备来说不几乎不可能的事不能实现的,如鼠标。

但是手指有一个主要的缺点,跟鼠标指针相比,无论是在大小,形状还是灵敏度方面都要差得多。在屏幕显示方面,手指永远不可能像鼠标那样精确。

幸运的是,一个优秀的用户界面的设计,足以应对以手指为基础的输入系统带来的挑战。在大多数情况下,这意味着你要确保你的布局能容纳一个手指尖的平均大小,同样也意味着用户指尖移动时要获得他们需要的反应。

用户执行某些具体的移动方式并获得独有的反馈结果,我们称之为手势。例如,用户点击一个按钮去选择它,以及滑动或拖动一个很长的列表。iPhone 用户很了解这些手势因为 iPhone 内置的应用程序一致化的使用了这些手势。因此,得益于用户对这些手势的熟悉,同时为了避免混淆,你应该适当的在你的应用程序中使用这些手势。

还有一些更复杂的手势,像横扫或放大,也是一致化的在内置程序中应用,但他们并不常见。一般来说,这些手势被用作快捷方式来加速任务完成,而不是执行任务的方法。例如,当查看邮件列表的时候,用户删除了一条正在显示的邮件,然后点击了预览行的删除按钮。用户可以以两种不同的方式来激活删除按钮:

- 点击导航条上的编辑菜单,在每个预览行都会出现一个删除控件。然后在特定的一行点击删除控件就会显示这条信息的删除按钮。
- 在特定的预览行使用横扫手势就会显示出这条信息的删除按钮。

第一种方式需要额外的步骤,然是很容易被发现。因为它仅仅需要点击已经标识清楚的编辑按钮来开启这一功能。第二种方法更快,但是它要求用户学习并记住更为特殊的滑动手势。

为了确保你的应用程序易于使用和发现,所以,试着限制你需要的手势,只保留最熟悉的手势——点击和拖动。你应该避免让像横扫或放大缩小这些不常见的手势成为执行动作的唯一方法。无论在什么情况下(你的应用程序)总是应该有一个简单直接了当的方法去执行一个操作,即使这意味着要多点击一次或者两次。

在大多数应用程序中,避免定义新的手势同样重要,特别这些新手势执行的操作是用户已经认为它是与标准手势相关联的。但是对于这个建议,最主要的例外是沉浸式应用程序,对它而言,自定义的手势是被认为合适的。例如,一个生产力程序要求用户做出一个圆形的手势来显示一行表格的删除按钮是混乱且难以使用的,但是另一方面,对于一个游戏来说要求用户做出一个圆形的手势来转动游戏的一部分就显得比较合理。

表 3-1 列举出了用户可以使用的标准手势。一定要避免重新定义这些手势；如果你在应用程序中支持这些操作，一定对这些手势提供适当的响应，且符合与这些操作。想了解更多关于如何创建手势触发事件的内容，请参见[“iPhone 应用程序编程指南”](#)。

**表 3-1** 用户可用于与基于 iPhone 操作系统的交互的手势

手势 Gesture	操作 Action
点击 Tap	按下或选择一个控件或条目（类似于普通的鼠标点击） To press or select a control or item (analogous to a single mouse click).
拖动 Drag	滚动或平移 To scroll or pan.
滑动 Flick	快速的滚动或平移 To scroll or pan quickly.
横扫 Swipe	在一个表格视图的行中，显示删除按钮 In a table-view row, to reveal the Delete button.
双击 Double tap	放大并将内容或图片置于中央 To zoom in and center a block of content or an image. 缩小（如果已经放大） To zoom out (if already zoomed in).
(双指捏的手势)放大 Pinch open	放大 To zoom in.
(双指捏的手势)缩小 Pinch close	缩小 To zoom out.
长按 Touch and hold	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

### 1.3.4 慎重地体现品牌化元素 Incorporate Branding Elements Cautiously

当品牌化是微小精致且易于理解的时候，他是最有效果的。人们使用你的 iPhone 应用来完成某件事情或消遣一下；他们不想产生一种强迫看广告的感觉，因此，你应该努力通过以一种高雅的，不唐突的方式体现融合你品牌的颜色和图片。例如，在视图与控件上，可以使用定义的配色方案。

例外的是你的应用程序的图标，它应该是集中于表现你的品牌。（用户在安装完你的应用程序后，在主界面能看到的图标），因为用户会频繁地看到你的图标，所以，花点时间使图标的外观吸引力与品牌认知保持平衡。关于如何设计一个应用程序图标，请参考[“应用程序图标”](#)

## 1.4 处理常见任务

如果你有使用桌面或笔记本电脑, 你会发现 iPhone 应用程序处理常见任务的方法是不同的。本章从人机界面观点出发, 描述这些常见任务, 关于用代码实现这些准则的技术细节, 参见 [“iPhone 应用程序编程指南”](#)。

- [启动 Starting](#)
- [停止 Stopping](#)
- [管理设置或配置选项 Managing Settings or Configuration Options](#)
- [支持复制和粘贴 Supporting Copy and Paste](#)
- [支持撤消和重做 Supporting Undo and Redo](#)
- [启用推送通知 Enabling Push Notifications](#)
- [使你的程序具有无障碍特性 Making Your Application Accessible](#)
- [提供搜索功能并显示搜索结果 Providing Search and Displaying Search Results](#)
- [利用用户的位置 Using the User's Location](#)
- [处理方向变化 Handling Orientation Changes](#)
- [使用声音 Using Sound](#)
- [提供选择 Providing Choices](#)
- [提供许可协议 Providing a License Agreement](#)

### 1.4.1 启动 Starting

iPhone 应用程序应迅速地启动, 这样用户无需延迟即可开始使用它们。启动时, iPhone 应用程序应:

- S 指定适当的状态栏样式 (关于可用样式的更多信息, 参见“状态栏”(第 67 页))。
- 显示一个与应用程序第一个屏幕极度相似的启动图像。这有助于减少应用程序被感知的启动时间。(有关更多信息, 参见“启动图像”(第 123 页))。
- 避免显示“关于”窗口, 版权协议屏幕, 或任何其他妨碍用户立即使用你的应用程序的启动体验。
- 默认情况下, 在竖向启动。如果你希望你的应用程序仅适用于横向, 就使用横向, 无需考虑当前设备的方向。如有必要, 允许用户旋转设备到横向。

一个只支持横向屏幕的应用程序, 应该同时支持两种横向方式, 也就是说, 主页按钮可以在右边也可以在左边。如果该设备本身就只是横向的, 只支持横向屏幕的应用程序就应该在这个方向启动。否则, 只支持横向使用的应用程序默认应该照着主页按钮右边的这个方向启动。

- 恢复你的应用程序最后一次运行时的状态。



**重要提示：**当用户安装你的应用程序后，不要让用户重新启动或重新开机他们的设备。如果你的应用程序因为内存使用或其他的问题导致它只能在系统刚启动后才可运行，那么，你需要解决这些问题。例如，参见 [iPhone 应用程序设计指南](#) 中的 [“高效使用内存”](#) 以获得一些关于开发高度优化的应用程序的指导。

### 1.4.2 停止 Stopping

当用户打开另外一个应用程序或设备级功能,如电话时,当前的 iPhone 应用程序将停止。特别一点的是：用户无需通过点击关闭按钮，或从菜单中选择退出。iPhone 应用程序应：

- 随时准备好接收退出或结束的通知。因此，尽快和尽可能频繁地合理保存用户的数据。
- 在停止时应尽可能全面的保存当前状态。例如，若你的应用程序正在显示滚动的数据，（在停止时，）保存当前滚动的位置。

iPhone 应用程序永不应该程序化的退出，因为这样用户会以为是程序崩溃了。但有的时候，外部环境阻止您的应用程序的发挥预定的功能。最好的处理办法是，显示一个有吸引力的屏幕，描述问题，并建议用户应当如何纠正它。这对用户有两个好处：

- 它提供的反馈，能让用户放心地知道你的应用程序并没有出错。
- 它把控制权交予用户，让他们决定是采取纠正措施，并继续使用你的应用程序还是按下主页按钮，并打开另外一个应用程序。

若在某些特定情况下应用程序的部分功能无法工作，当用户触动该功能时，你可以提供显示一个屏幕或警告。尽管在警告在设计上缺乏灵活性，但你若按下面的方法，不失为一个好的选择：

- 简洁的描述一下当前情况。
- 提供一个可以执行纠正操作的按钮。
- 仅当用户尝试使用那些无法运行的功能时，才显示警告。

对于所有的警告，越少显示他们，他们也就越有效。有关创建警告的更多信息，请参阅 [“使用警告”](#)。

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

iPhone 应用程序可以提供用于定义应用程序行为的设置，或者是用于改变一些影响应用程序功能性的选项。设置应该表示一些信息，例如帐户 的名称，用户只需设置一次，就很少会再去改动。用户在内置的设置应用程序设置中浏览特有的应用程序的设置。而配置选项，用户可能会经常改变他们的值。比如，显示一个列表的类别，配置选项应该在应用程序当中即可使用(而不应该把它放置到内置的设置应用程序中)。

你应该考虑到设置和选项是相互排斥的。也就是说,你不应该在你的应用程序中的同时提供设置和配置选项。

不需要用户指定任何设置的 iPhone 应用程序,是最佳的。用户无需被要求提供初始设置信息,即可马上使用这些应用程序。你可以采纳以下这些设计决策达到这个目的:

- **把你的解决方案集中在满足 80% 的用户的需求上。**当你这样做时,大多数用户就不需要提供设置了,因为你的应用已经建立在大多数用户所期望的操作行为方式上。如果有些功能,只有一小部分的用户可能想要,或者大多数用户可能只需要用一次,那就要放弃它。
- **尽可能多从其他来源获得更多信息。**如果你可以从内置的应用程序或设备的设置中获得任何用户提供信息,那就从系统查询这些值;不要让用户再次重新输入。
- **如果你必须得到初始的设置信息,那么提示用户在你的应用程序里输入。**然后,尽快的将这些信息保存在你的应用程序的设置中。这样,用户不会在还没开始感受你的应用程序的好处时,就被迫退出你的应用程序并打开设置。如果用户以后需要再次修改这些信息,他们可以在任何时候到应用程序的设置里修改。

让用户无需先退出你的应用程序即可打开设置应用程序,那是不可能的,并且你不应该鼓励他们这样操作。系统没有提供任何图标或控件来支持这一操作,并且也建议你不要创建自定义图标或控件的去支持。如果你确定必须为你的 iPhone 应用程序提供设置,请在 [iPhone 应用程序设计指南](#)中参阅“[设置绑定](#)”,以了解如何在你的代码中实现它。

**注:**特定应用设置不应该包含用户帮助内容。

与设置不同,配置选项经常性的被改变,比如当用户选择从新的来源或不同排列中察看资料。你可以提供动态反馈让用户作出选项,因为用户访问它们时并不会停止你的应用程序。

你可以在主用户界面或在之外的屏幕里提供配置选项。你可通过考虑这个选项代表的功能的重要性,以及他可能被使用的频率次数来判断,使用哪种方式提供配置会更加合理。

例如,日历允许用户按天,周,月的方式浏览他们的日程。这些选项其实也是可以在其他之外的屏幕中提供,但翻看日历的不同部分是其首要功能,并且用户可能会频繁改变他们的关注点。(所以,这些选项应该在主用户界面中提供。)

另一方面,天气的主要功能是显示城市当前天气情况以及 6 日内的天气预测。尽管选择是在温度摄氏还是在华氏显示是很重要的,但是用户不太可能经常更改此选项,所以把它们放在主用户界面里是说不通的。在其他之外的屏幕中提供温度单位的选项,可以使人们方便地察看,而且唐突。

#### 1.4.4 支持复制和粘贴 Supporting Copy and Paste



iPhone 操作系统提供了一个编辑（或粘贴板）菜单，用来支持剪切，复制，粘贴，选择，选择全部这些操作，它应用于文本，网页和图片的视图中。让此菜单显示的一种方式：用户长按屏幕，呈现放大视图（它允许用户移动光标或者是选区点至正确的位置）后释放手指即可。如果菜单在当前环境下可用，那么当手指释放后，他就会显示。在菜单选择“选择”这个菜单项，即可在视图中选择单词或有定义的应用程序项。用户可通过拖动选择区域边缘的手柄扩展选区，当选区确定后，就会自然而然的显示一个提供剪切，复制或粘贴命令的菜单。

你可以调整编辑菜单的某些行为以适应你的应用程序。（关于如何用代码实施这些行为的信息，请参见在[iPhone 应用程序编程指南](#)中的[“复制和粘贴操作”](#)。）例如，你可以指定菜单显示命令子集，或决定菜单应在哪里呈现。但是你不能支配菜单的颜色或本身的形状。

在编辑菜单显示的命令在当前的环境下应是合理的。举例来说，如果未选中任何内容，菜单上就不应包含“复制”或“剪切”，因为它们是针对选区执行的命令。同样，如果已选中某些内容，菜单上就不应包含“选择”这个命令。如果你的自定义视图支持编辑菜单，你有责任确保该菜单的命令在当前视图的显示是恰当的。请注意，你不能指定在菜单中显示自定义的命令。

UIKit 在光标或者选区的上方或下方显示编辑菜单，它取决于屏幕的可用空间和菜单指针的位置，因为这样便于用户看到菜单的命令与上下文的关系。你可以使用程序化的方式在菜单出现之前确定它的位置，可防止应用程序用户界面的重要组成部分被其遮挡。（如果可能被遮挡的话）

请注意，虽然长按手势是用户使编辑菜单显示主要的方式，他们还可以在文本浏览中双击某个单词来选择它，并同时使编辑菜单显示。如果你的应用程序允许在自定义视图中提供编辑菜单，就应该同时支持这两种手势。此外，你可以定义当用户双击时默认选中的对象。

避免建立一个按钮，提供编辑菜单中的命令。例如，相比提供一个复制按钮，让用户使用编辑菜单执行复制操作的方式更好，因为用户会对你的应用程序存在两种方式做相同的事情而感到困惑。

你可以启用对静态文本的选区功能，但应只在静态文本表示的含义对用户有实际作用的时候才启用。例如，用户可能想要复制图像的标题，但他们不太想复制标签项或屏幕标题的文本标签，例如帐户。在文本视图中，应该是默认按单词来选择的（而不是字母）。

按钮标题是不能够被选择的，因为对用户来说不触发按钮而显示菜单是困难的。一般来说，具有按钮行为的元素是无需被选择的。

如果你的应用程序中支持剪切，复制，粘贴命令，你也应该支持撤消和重做（参照[“支持撤消和重做”](#)）。这是因为编辑菜单不要求在操作之前执行进行确认，并且当用户改变主意时，他们往往希望能够撤消最近的操作。

#### 1.4.5 支持撤消和重做 Supporting Undo and Redo

iPhone 操作系统支持用户在文本视图输入时，使用撤销和重做。用户摇晃设备启动撤销，接着设备将显示警告，允许用户撤销他们刚刚输入的内容，重做之前撤销的内容，或取消撤销。

UIKit 允许你在你的应用程序里以更正规的方式支持撤销（关于如何在代码中实施这一行为，请参见["撤销架构"](#)）。你可以指定：

- 哪些操作是可以被用户撤销或重做的
- 什么时候你的应用程序应该把摇晃事件解释为撤销手势
- 最多支持多少次撤销

为了保证你的应用程序中的撤销和重做功能具有良好的用户体验，你应该：

- **用简短的描述性词组，准确地告知用户他们正在撤销或重做的是什么。** UIKit 自动使用“撤销”和“重做”作为撤销警告按钮的标题，但你需再增加一两个单词描述用户撤销或重做的操作。（请注意，取消按钮文字不能更改。）例如，你可能增加诸如“删除名称”或“地址变更”这样的单词，这样“撤销删除名称”或“重做地址更改”将作为按钮标题。

一定要避免提供太长的文本：按钮标题过长导致被截断，用户就难以解读。此外，由于这个文本是一个按钮标题，应使用标题风格的大写规则，并且不加标点。（简单的说，标题风格的大写规则意思是，除去冠词，并列联句，和四个或少于四个字母的介词之外，其他单词首字母都大写。）

- **避免过度使用摇晃手势。** 尽管你可以程序化地设置你的应用程序什么时候把摇晃事件解释为撤销手势，你还是面临着让用户迷惑的风险，因为用户可能使用摇晃手势去执行另外的操作。

摇晃是用户期望启动撤销和重做的首要方法，但如果可以的话，你也可以在导航栏中包含系统提供的撤销和重做按钮。如有必要，你可以在屏幕上显示一个明确的、专用的按钮用于在你的应用程序中执行这些功能（撤销与重做），但是这不是常用的方法。

- **考虑你所允许的撤销或重做操作的环境。** 一般来说下，用户期望他们的变化和行为立即生效。尽可能保证撤销和恢复的能力应明确与用户当前环境相关，而不是与更早前环境相关。

#### 1.4.6 启用推送通知 Enabling Push Notifications

当你为你的应用程序注册苹果推送通知服务时后，当接收到新的数据后，即可安排提醒予以用户（提醒新数据），即使你的应用程序没有运行。当设备接收到某个未运行应用程序的信息时，它可以这样通知用户：

- 在应用程序的主屏幕图标上显示徽章
- 播放提醒声
- 显示一个提醒信息

或者任意组合上述的通知方式。用户对通知的反应可能有，启动此应用程序去管理一下新数据，或者只是了解通知的到达（而不采取任何操作）。（要了解如何用代码处理推送通知，参见[苹果推送通知服务编程指南](#)。）

**注意：**推送通知的发送不是有保证的。同时，用户可以全系统地拒绝接收通知。推送通知的目的是通知用户有新数据，并不会向你的应用程序写入重要数据。

内置的设置应用程序中的通知设置部分主导着推送通知的设置，它对每个注册了苹果推送通知服务的应用程序都提供了设置。对于每个应用程序，iPhone 操作系统都提供给用户允许或不允许，标记，声音，提醒的三种方式的通知。

请花一些时间想想什么类型事件的通知是用户希望的。通知应给予用户那些有用的，可付诸操作的信息，即使他们没有使用你的应用程序。

当你已经确定应用程序的哪些事件用户可能会关心之后，如果有的话，你也应该允许你的用户决定哪些事件使用什么类型的通知。如果你不允许用户自定义应用程序中的推送通知的体验，你就容易让用户陷入与他们不感兴趣的通知的纠缠之中。

用户可以选择他们想要接受哪些类型的通知，所以你应该支持所有三种类型：

- **标记。**标记是打扰程度最小的方式告知用户有新的他们感兴趣的内容。标记出现是在主界面应用图标右上角的一个红色小椭圆。你不能对标记的出现做任何控制，它只包含数字，没有字母和标点。

当你想要告诉用户有多少通知项在等待他们的注意时，标记很有效。例如，在一个标记里数目可能会显示未读消息，新分配的任务，或者是当前远程参与者。

- **声音。**你可以提供一个自定义提醒声音，也可以使用内置的提醒声。如果你创建一个自定义的声音，请确保它是短暂的，独特的，专业化制作的。（要了解有关这个声音的技术要求，在[苹果推送通知服务编程指南](#)中参见“[准备自定义提醒声音](#)”。）请注意你不能强迫设备发出通知时振动；应该由用户控制是否警告伴随振动。

在通知到达的时候，提供一个容易识别的声音是十分理想的选择，因为它提供足够的信息给予用户反应。例如，一个用于多人协作完成任务的管理系统，可以使用一个独特的声音，伴随成员任务的完成。仅仅听到这个声音，用户就会知道已经完成了任务。

- **提醒。**提醒是打扰程度最大的通知用户有新内容的方式。在提醒的上方显示你应用程序的名称，下方显示的是信息，底部有一至两个按钮。如果你指定两个按钮，应在左侧显示关闭按钮，右侧显示查看按钮（用户可以通过点击查看按钮，关闭该提醒同时并启动应用程序）。如果你指定一个按钮，提醒只显示一个确定按钮。关闭按钮和确定按钮的作用都是关闭提醒，并且不打开应用程序。

提醒会干扰用户的工作流程，因此最好少用他们，除非是用来为你提供有关事件简短的，重要的信息。尤其是，一定要避免提醒消息中的包含任何广告内容。

## 1.4.7 使你的程序具有无障碍性 Making Your Application Accessible

无障碍性指的是应用程序应该让残疾用户也能成功的使用,或者是在辅助应用程序或设备的帮助下成功的使用。基于 iPhone 操作系统的设备包含许多方便用户(包括残疾用户)的特性,例如可视化语音邮件,放大,以及语音控制。并且,你无需在你的应用程序中介入任何东西,即可确保用户从这些特性中受益。

说道 VoiceOver,这个故事略有不同。VoiceOver 是苹果的创新屏幕阅读技术,利用这种技术,用户无需看到屏幕,就可以控制自己的设备。为了确保 VoiceOver 用户可以尽情地使用你的应用程序,你可能需要提供一些关于用户界面中视图和控件的自定义信息。

幸运的是,UIKit 控件和视图在默认就具有无障碍性,所以当你以一个完全标准的方法应用标准元素时,你基本只有一点儿(如果有的话)额外的工作要做。你的用户界面自定义程度越高,需提供的自定义信息就越多,这样 VoiceOver 才能向那些有视觉缺陷的用户正确地描述你的应用程序。

**重要提示:** 使你的应用程序具有无障碍性的工作包括向 VoiceOver 提供所需的信息,以帮助人们使用它。这种工作不包括改变用户界面的视觉设计以适应 VoiceOver。

使 VoiceOver 的用户能无障碍地访问你的 iPhone 应用程序是一件正确的事情。它也可以增加你的用户群,帮助您落实由各种监管组织的辅助功能规范。

## 1.4.8 提供搜索功能并显示搜索结果 Providing Search and Displaying Search

### Results

在你开始搜索时,UIKit 提供了一致性界面的搜索栏控件,但在你的应用程序中,实现搜索功能则是属于你自己的责任。(要了解更多关于搜索栏的信息,请参阅[“搜索栏”](#),要了解更多关于如何在代码中对待搜索结果的信息,请参阅[UISearchDisplayController Class 参考](#)。)为了确保让搜索对于用户来说是一个有用且方便的体验,你应花点时间考虑如何落实搜索进程和如何显示搜索结果。

一般来说,你应该:

- 为你的数据建立索引,以便随时查找。
- 对本地数据进行即时及时筛选,一旦用户开始输入,你就可以显示结果,随着用户继续输入而逐步缩小范围。
- 如果可能,同时筛选用户的远端数据,如果响应时间被推迟多于一两秒钟,就需要用户的许可(是否筛选远端数据)。
- 在列表上面显示搜索栏或在列表里显示索引。
- 避免使用标签作为搜索,除非它是应用程序中的主要功能必须用一个有区别的方式显示。

虽然及时的数据筛选通常会带来出色的用户体验,但这并不总是具有实用性。如果是这种情况下(非即时筛选数据),当用户敲击了键盘上的搜索按钮后,你才能开始执行搜索进程了。如果你这样做,一定要提供搜索进程情况的反馈,以使用户知道该进程并没有停滞不前。一种方法是,尽快的先显示文本搜索结果,而对于需要花点时间才能从数据中检索完毕的内容,先显示一个占位符(后再慢慢显示具体检索内容)。

在 YouTube,例如,用户点击搜索按钮开始视频搜索。如果网络连接速度慢,YouTube 上首先显示 正在载入... 的信息和一个旋转指示符,以使用户知道搜索在进行中。然后,YouTube 以列表的形式显示出结果,以文本的形式填充每一行,例如视频标题和评分,以及 一个点线描边自定义图像框。当用户在浏览视频标题列表时,视频的缩略图也在同时下载,当下载完完成后,他们会机会替换掉点线框。当更多数据在被下载的时候,先显示局部搜索结果,能够尽快地给用户提供有用的信息。

如果你要处理本来就属于不同种类的数据,你可以提供一个范围栏。一个范围栏包含四个范围选择按钮,分别代表一个类别。例如,邮件提供了一个 范围控件栏,让使用者可将其搜索集中在,发件人,收件人或邮件的主题,或把搜索扩大到全部。如果它能够帮助用户让搜索更集中,或者是能显著地减少搜索结果数,那么就应该提供范围栏。

#### 1.4.9 利用用户的位置 Using the User's Location

用户喜欢应用程序提供的允许用户使用物理位置来自动标记内容,寻找当前附近的朋友的特色功能。当用户不想与他人分享位置信息时,他们也希望能有禁用的功能。用户可以在“设置”应用程序中的常规菜单中,全系统地设置允许或者拒绝通过位置服务获取他们的物理位置。

如果用户关闭位置服务,随后使用应用程序某个需要提供位置的功能,他们将会看到一个提醒,告诉他们必须修改预设,才可以使用该功能。提醒不 提供给用户在当前应用程序修改预设的功能,反而,他们必须进入“设置”应用程序,然后改变他们的预设。这可确保用户充分认识到,他们在全系统地允许系统使用他们的位置信息。

为了帮助用户了解为什么他们需要打开位置服务,最好的方式是,在他们打开某个需要知道他们当前位置的功能的时候,给他们一个提醒。例如,当位置服务是关闭时,人们依然还是可以使用地图服务的,但他们想获取或者是跟踪他们当前的位置的时候,他们将会看到一个提醒。

如果位置服务是关闭的,当你的应用程序第一次尝试访问位置信息时,iPhone 操作系统将显示提醒。核心位置(Core Location)框架提供了一种方法让你可以获得用户对位置服务的预设,避免你不必要的或不适当的触发此提醒。(要了解更多信息,参见 [Core Location Framework Reference](#)。)

获取了用户对位置服务的预设以后,你就可以只为那些非常紧密地非常可能需要用户位置信息的功能,触发提醒,或者说,完全地避免这些提醒。

- 如果你的应用程序主功能没法在没有位置信息的环境下工作,最好就是让用户在启动它时就看到提醒。用户不会因此而感觉不快,因为他们明白,你的应用程序的主功能的执行依赖于他们自己的位置信息。

- 如果用户位置不是你的应用程序实质功能的一部分, 你可以选择仅仅只是限制那些要使用用户位置的功能。例如, 当位置服务是关闭的, “相机” 应用程序会自动关闭在他们拍摄的照片中增加用户位置信息的功能。如果用户不改变他们的预设, “相机” 应用程序并不会阻止用户拍照。因为在照片中添加位置信息是额外功能, 并非实质性的。
- 如果一个功能需要位置信息才能运行, 请确保除非用户真的使用这个功能, 否则不要执行任何触发提醒的编程的调用之前, ( 获取用户对位置服务的预设的调用并不会触发提醒。 ) 这样, 你就可以避免使用户产生如 “明明你的应用程序看起来在做一些完全无关的事情, 却需要我的位置信息” 的疑惑。

#### 1.4.10 处理方向变化 Handling Orientation Changes

用户随时可能旋转搭载 iPhone 操作系统的设备, 同时他们也期望当前浏览的内容能随之响应适当的调整。在你的 iPhone 应用程序中, 一定要:

- 要注意加速度值 ( 要了解更多关于加速度和有关编程接口的参考信息, 参见 iPhone 应用程序编程指南 )。如果合适, 你的应用程序应对设备所有的方向变化作出响应。
- 如果你的应用程序中某些用户界面只能在一个方向显示, 合理的办法是, 让它保持在指定方向的显示, 而不必响应设备的方向变化。例如, 当用户选择观看一个 iPod 视频, 不论当前的设备是什么方向, 它都横向显示。这就暗示用户要旋转设备, 以便于观看视频。在这个例子中重要的一点是, iPod 没有提供 “旋转” 按钮, 而是用户知道要旋转设备, 因为视频横向显示。

允许用户旋转装置以便于正确察看你的应用程序中, 那些需要一个明确方向的用户界面。避免为了告诉用户旋转设备而去创建一个控件或定义一个行为。

- 充分利用一步到位的方向改变进程, 顺畅, 快速地执行旋转。然而, 如果你的屏幕布局很复杂, 当方向发生变化时, 你可以选择淡入淡出的过渡这种方式替代。要了解如何在你的代码中支持这个步骤, 参见 “UIViewControllerClass” 参考。
- 用户经常旋转自己的设备至横向, 因为他们希望 “看到更多”。如果您只是以比例放大内容的方式响应用户, 那就未能达到用户的期望。相反, 你应该重新调整文本的换行线, 有必要的, 重新安排用户界面的布局, 以便让更多的内容显现在屏幕上。

#### 1.4.11 使用声音 Using Sound



用户期望从搭载 iPhone 操作系统的设备听到美妙的声音,无论他们在听系统声音,如铃声和提醒声音,或应用程序声音,如媒体播放,环境声音,以及音轨。此外,用户也希望设备发出的声音符合他们的预设和意愿。

声音的大小与是否播放,完全是由用户决定的。但是,有的时候,尽管用户目前的设置说明他们更希望处于静音模式,但是用户还是希望听到某些声音。例如,用户总是希望听到他们设置的提醒声。从本质上讲,用户是想听到他们需要的声音,但不想听到他们不需要的声音。

为了帮助你适应这种情况,iPhone 操作系统提供了以下编程接口,你可以使用:

- 描述你的应用程序的声音应该如何与设备上的其他声音相协调
- 确保你的应用程序的声音是根据用户的期望在播放

在你决定如何处理应用程序中的声音前,你需要了解,当用户在调整设备控制选项和外部设备的时候,他们是期望应用程序和设备应该如何表现他们的。调整控制和使用外部设备,如耳机和耳麦。

### 铃声/静音开关——用户的期望 The Ring/Silent Switch—What Users Expect

用户在他们的设备上使用铃声/静音开关时,他们想:

- 避免被不期望的声音打断,如“电话”铃声,信息到达音。
- 避免听到用户操作的附属声音,如键盘声音或其他反馈声音,偶然的聲音,或应用程序启动声音。
- 避免听到游戏声音,包括附属声音和音轨,这是并不是玩游戏时必不可少的声音。

例如,在剧院的时候,用户把设备的设置为静音,是为了避免打扰在剧院的其他人。在这种情况下,用户仍然希望能够在他们的设备上使用应用程序,但他们不想听到那些他们不期望或未明确请求的声音,如铃声或信息到达音。那样会让他们感觉惊讶。

但是,铃声/静音开关不会关闭那些由用户单独且明确的操作引起的声音。例如:

- 在只播放媒体的应用程序中的媒体播放,不会因为铃声/静音开关而关闭,因为媒体播放是用户明确的请求。
- 闹铃是会被铃声/静音开关关闭的,因为它是由用户明确设置过的。
- 在语言学习应用程序中的声音片段是不会被铃声/静音开关关闭的,因为用户明确地想听到它。
- 在音频聊天应用程序中,交谈的声音是不会被铃声/静音开关关闭的,因为用户打开这个应用程序的唯一目的就是语音聊天。

此行为遵循用户控制的原则,因为它是由用户而不是设备,来决定听到用户明确要求的声音是否合理。

## 音量按钮——用户的期望 Volume Buttons—What Users Expect

用户使用设备的音量按钮来调节设备所能播放的任何声音的音量，包括歌曲，应用程序声音和设备声音。这意味着用户可以使用音量按钮来调节任何声音至安静，而不用管铃声/静音开关的状况如何。

在某些情况下，在应用程序的用户界面上提供音量设置的功能也是合适的。例如，YouTube 为用户提供了一个音量滑块来调节他们正在观看的视频的音量。YouTube 在运行的时候，用户可以使用音量滑块或者音量按钮交替着来调节视频的音量。这是因为当应用程序在运行的时候，音量滑块就相当于音量按钮的代理：调节滑块同时影响应用程序的音量和整个系统的音量，不过不会影响铃声音量。

如果你需要显示一个音量滑块，一定要使用系统提供的在 [MPVolumeView](#) 类里的音量滑块。注意如果当前的声音输出设备不支持音量控制（比如 A2DP 设备），音量滑块就不会显示，取而代之的是设备名称。

用音量按钮来调节应用程序当前正在播放的音频的音量，同时也会更改整个系统的音量，不过这不会影响铃声音量。（假如当前没有播放任何音频，那么音量按钮调节的是铃声音量。）

此行为遵循用户控制的原则，因为设备应该发出多大的声音应该是一直由用户决定的。

有些时候，应用程序可能需要调节相关的，或是独立的多个响度（volume level）来制造最好的混音输出效果。但是最终输出的音量始终应该是由系统音量管理的，无论是使用音量按钮还是音量滑块。这意味着应用程序音频输出的控制权在用户手里，这也是理所应当的。

## 耳麦和耳机——用户的期望 Headsets and Headphones—What Users Expect

用户使用耳麦和耳机来获得私人的声音体验，并解放双手。在使用和不使用附件的情况下，用户对于应用程序的行为有不同的期望。

当用户插上耳麦和耳机，他们是打算继续听当前的音频，但是是私人的。为此，他们期望应用程序继续播放当前音频，没有停顿。

当用户拔下耳麦和耳机，他们不希望应用程序自动把他们刚才听的与他人分享。为此，他们期望应用程序可以暂停当前正在播放的声音，允许他们在就绪之后再重新开始播放。

## 无线音频——用户的期望 Wireless Audio—What Users Expect

用户很喜欢无线耳麦的便捷之处，比如蓝牙 A2DP 设备。人们使用无线耳麦和耳机的原因与使用有线的相同：希望获得私人的声音体验，并解放双手。

同样，对无线耳麦的用户体验的期望也很相似。

- 当用户连接到无线音频设备，他们是打算继续听当前的音频，但是是私人的。为此，他们期望应用程序继续播放当前音频，没有停顿。



- 当用户断开与无线设备的连接（或者该设备超出了服务的范围，或者被关闭了），他们不希望应用程序自动把他们刚才听的与他人分享。为此，他们期望应用程序可以暂停当前正在播放的声音，允许他们在就绪之后再重新开始播放。

虽然人们不会实际物理上的插上或者拔下无线音频设备，但是他们还是期望可以选择不同的音频路径。为了解决这一点，iPhone OS 自动提供允许用户选择输出音频路径的控制界面。由于选择不同的音频路线是用户创建的操作，用户希望当前正在播放的声音继续播放，不要停止。

### 定义应用程序的声音行为 Defining the Audio Behavior of Your Application

如果声音用于加强你应用程序的用户体验或者功能，抑或是，它对于这两方面来说是必不可少的，你就需要考虑你的音频如何与设备的音频环境饱和协调，并用用户的操作提供反馈。举例来说，你需要决定这些方面：

- 当设备被锁定或者被切换到静音，你的音频是否应该继续播放
- 你的音频是否应该和正在播放的其它音频混合（比如 iPod 里的歌曲）
- 你的应用程序是否需要兼顾处理音频的输入和输出，不管是按顺序的处理还是同时处理
- 你的音频是否应该在被中断之后自动恢复播放

关于如何支配你的应用程序的音频应在上诉情况下如何表现，使用音频会话服务（Audio Session Services）或者 [AVAudioSession](#) 类。这些编程接口并不创建声音；而是帮助你表现你的音频应该如何同设备上的音频相互作用，应该如何对中断和配置选项的改变作出响应。音频会话服务（Audio Session Services）控制由 AV Foundation 框架，Audio Queue Services，OpenAL 和输入输出音频单元等创建的声音。

**注：**如果你的应用程序只是创建与功能相对应的附属用户界面音效，你可以使用系统声音服务（System Sound Services）。系统声音服务（System Sound Services）是 iPhone 操作系统中，用于创建提醒音、用户界面声和振动的技术；它不适合任何其它的用途，并且它创建的声音不受音频会话服务（Audio Session Services）控制。关于使用这项技术的例子，参见示范项目 [SysSound](#)。

**重要：**无论你使用何种程序来创建声音，或是你怎样定义声音的行为，来电能在任何时候打断当前运行的应用程序。这是因为任何应用程序都不应该阻止用户接听来电。

音频会话是应用程序和系统之间的音频中介。站在用户体验的角度上来看，音频会话最重要是定义了应用程序音频行为的类别。

为了提供极好的音频用户体验，您需选择对你应用程序音频的最佳诠释的类别。请确保您的选择是基于类别的语义，而不是基于其明确的行为集合。这保证了你的应用程序的行为符合用户的期望。此外，即使以后对确切的行为集合进行了再优化，它可让你的应用程序依然正常工作。

在极少数情况下，你可能需要通过给音频会话添加一个属性来加强或改进某个类别的标准行为。例如，你可以添加 `kAudioSessionProperty_OtherMixableAudioShouldDuck` 属性来使你的应用程序的音频比其它音频都大声（来电铃声除外）。如果你认为当用户在播放其他音频时，让其听到你的应用程序的音频

是至关重要的，那么你就可 以这么做。但是，你必须意识到一个类别的标准行为代表着绝大多数用户的期望，因此在添加属性，改进行为之前必须深思熟虑。参见[音频会话编程向导](#)的“[微调类别](#)”来了解更多有关音频会话属性的内容。

你可以基于当前设备的音频环境来选择类别。举例来说，你可能会想让用户在使用你的应用程序时，可听取其他音频，不一定非得是你应用程序提供 的。如果对于你的应用程序来说这是合理的，那就一定要在启动你的应用程序之时，避免强迫用户停止听音乐，或者是让他们选择明确的音轨。参见“[整合](#)”中的情景二来学习应该怎么做。

当应用程序运行时，改变类别也是可能的，尽管很少必需这么做。这么做的首要原因是某些应用程序需要支持在不同的时间录音和回放。对于这样的应用程序，让其 根据需要在录音( Record )与回放( Playback )类别之间进行切换，要比直接选择录音( Record )与回放( Playback )类别类别更 好。这是因为选择录音( Record )类别能够保证在录音时，设备将屏蔽提醒音（比如短信到达提醒音）。

表 4-1 列出了你可以使用的音频会话类别。iPhone 操作系统将“独占环境”(Solo Ambient)分配为音频会话的默认类别。

**注** 为了节省空间 表 4-1 只显示了每个类别名的最后部分。类别实际的名字由 AVAudioSessionCategory 开头。如 MixWithOthers 属性的实际名称是 kAudioSessionProperty\_OverrideCategoryMixWithOthers。

**Table 4-1** Audio session categories you can use to define sound behavior in your application

**表 4-1** 可用于在应用程序里定义声音行为的音频会话类别

Category 类别	Meaning 意义	Silenced by Ring/Silent switch and locking 支持铃声/静音开关与 锁定切换至静音	Mixes with other audio 和其它音频混合
SoloAmbient 独占环境	Sounds enhance application functionality, and should silence other audio 声音用于提高应用程序功能性，并且 它在播放时其它声音都转为静音	Yes 是	No 否
Ambient 环境	Sounds enhance application functionality, but should not silence other audio 声音用于提高应用程序功能性，它在 播放时不会使其它声音转为静音	Yes 是	Yes 是
Playback 回放	Sounds are essential to application functionality, and might mix with other audio 声音对于应用程序功能性至关重要， 并且可以与其它音频混合	No 否	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	No	No
录音	用户录音	否	否
	Sounds represent audio input and output, sequentially or simultaneously	No	No (default)
PlayAndRecord	声音代表着音频的输入和输出, 按顺序的(音频的输入和输出)或者是同时的	否	否 ( 默认 )
播放与录音			Yes (when the MixWithOthers property is added)
	Application performs hardware-assisted audio encoding (it does not play or record)	-	是 ( 当添加了 MixWithOthers 属性时 )
AudioProcessing	执行硬件辅助的音频编码 ( 不录也不播音 ) 的应用程序		No
音频处理			否

## 整合 Putting it All Together

下面的几个情景阐明了如何选择音频会话类别来提供令用户满意的声音体验。

**情景 1** 想象一下你正在开发一个帮助人们学习一门新语言的教育应用程序。当用户敲击特定的控件时，你要提供反馈的声音；当用户想要听正确发音的例子时，你要提供单词和词组的发音。

对于此应用程序里，声音对其的首要功能性是至关重要的。人们使用这个应用程序来听他们要学习的语言里的单词和词组发音，因此，就算设备被设置为静音或者锁定，应用程序的声音仍然应该播放。由于用户需要清楚地听到声音，他们可能期望正在播放的其它音频转为静音。

为了提供用户对这种应用程序所期望的声音体验，你要使用“回放”（Playback）类别。虽然这个类别可以重新定义来允许和其它音频混合（如表 4-1 所述），该应用程序必须使用其默认行为，确保其它音频不会和用户选择收听的学习内容冲突。

**情景 2** 想象一下你正在开发一个游戏，游戏里用户需要指引屏幕上的人物完成许多不同的任务。你提供各种游戏音效和一段音乐音轨。

在这个应用程序里，声音能极大地增强了用户体验，但是对于首要任务来说并不是必需的。而且，用户可能喜欢在没有声音的状态下玩游戏，或者在玩游戏的同时听他们自己曲库里的音乐。

最佳策略是在你的应用程序启动的时候查明用户是否在收听其它的音频。不要让用户来选择是收听其它的音频还是你游戏的音轨。相反地，使用音频会话服务功能 [AudioSessionGetProperty](#) 来查询

[kAudioSessionProperty\\_OtherAudioIsPlaying](#) 属性的状态。根据查询的结果,你可以选择“环境”(Ambient)或者“独占环境”(Solo Ambient)类别(两种类别都允许用户在无声状态下玩游戏)。

- 如果用户正在听其它音频,你应该假设他们想继续听正在听的音频,而不希望被迫去听游戏的音轨。在这种情况下,应该选择“环境”(Ambient)类别。
- 如果当你的应用程序启动的时候,用户没有在任何其它音频,选择“独占环境”(Solo Ambient)类别。

**情景 3** 想象一下你正在开发一个为用户前往目的地提供精确的,实时的导航指引的应用程序。针对行程的每一步提供语音指示,还有一些反馈的声音。此外,你认为人们希望在使用你的应用程序仍然可以听取听取其他音频。

在这个应用程序里,语音的导航指引属于主要任务。为此,你要使用“回放”(Playback)类别,允许音频在设备被锁定或者静音的时候也能播放。

你可以添加 [kAudioSessionProperty\\_OverrideCategoryMixWithOthers](#) 属性,来允许人们在使用你的应用程序时可以听其它的音频。但是,你也想要保证语音指引在其他播放的音频至上,这利于用户收听。你可以把 [kAudioSessionProperty\\_OtherMixableAudioShouldDuck](#) 属性运用到音频会话上。这保证你的音频会比其它所有正在播放的音频大声(除了来电铃声)。

**情景 4** 想象一下你正在开发一个允许用户向中心网站上传文本和图形的博客应用程序。你的应用程序可能会包含一个短暂的启动声音,多种不同的短暂的用于附和用户操作的声音效果(比如说文件上传完毕时播放的声音),和一个表示博客发布失败的提醒声音。

在这个应用程序里,声音增强了用户体验,但这是非主要的。主要任务与声音无关,且用户并不需要听到任何声音就可成功地操作。在这个情景里,你要使用“系统声音服务”(System Sound Service)来制造声音。这是因为这个应用程序里的所有声音都符合这项技术的预期目的:生成符合用户期望的用户界面声音效果和提醒声音,且在设备被锁定和静音状态下,保持静音。

#### 1.4.12 提供选择 Providing Choices

iPhone 操作系统包含了一些界面元素提供给用户作出选择。当当你需要在应用程序里提供选择时,你应该提供这样选择方式,因为用户已经很熟悉(这些方式)的行为。一般来说,你不应该把桌面应用程序的选择控件的外观和行为直接复制过来,比如应用程序菜单或者一组单选按钮。iPhone OS 提供以下界面元素供你使用至给用户选择:

- **列表**(即表格视图)。用户点击列表里的一行来选择一个项目。列表几乎适用于展示任何数量的选择。关于在应用程序里如何使用表格视图的具体内容,参见[“表格视图”](#)。

- **选取器**，包括日期和时间选取器。用户拨动选取器内的滚轮，直到每个滚轮都显示了期望的那个组合的数值，比如一个日期组合包含年，月，日的数值。关于在 iPhone 应用程序里如何使用选取器的更多信息，参见 [“日期时间选取器”](#) 和 [“选取器”](#)。
- **开关控件**。用户将开关控件从一端滑动到另外一端，表示选择两个值中的一个。开关控件是用来在列表中提供简单选择的。关于开关控件的更多内容，参见 [“开关控件”](#)。

### 1.4.13 提供许可协议 Providing a License Agreement

有的时候，应用程序会要求用户在接受了许可协议后才可开始使用。这种协议（常被称作最终用户许可协议 或者 EULA）通常包含各种条款和条件，用以描述责任保险描述责任限制与管理用途。如果你的 iPhone 应用程序有许可协议，请注意他们将在 App Store 内显示，以使用户在获取你的应用程序之前即可阅读。

如果可能的话，尽量避免在用户第一次使用你的应用程序的时候，要求他们接受最终用户许可协议（EULA）。这样，用户无需受阻即可享受你的应用程序。不过，虽然这样做是用户体验所提倡的，但是它可能并非在所有情况下都可行。如果你的应用程序必须显示许可协议，试着在将界面调整得协调，并尽量降低给用户带来的打扰。

## 第二部分：为你的 iPhone 软件设计用户界面

iPhone OS 的用户界面元素包括视图和控件。**视图**为内容提供定义良好的功能集。**控件**是引起即时行动或可视结果的图形对象。尽管所有的视图和控件都包含于应用程序窗口，用户在屏幕上察看它们并且与它们交互，这些**屏幕**大致对应于应用程序不同的视觉状态。

iPhone OS 定义这些用户界面元素的标准外观，并提供与用户期望一致的行为。阅读第二部分的各个章节，可以了解所有可用的用户界面元素类型以及如何使用它们来构建你的应用程序的用户界面。



## 2.1 应用程序用户界面简介

在你去深入了解详细的视图和控制之前，如果你能对这些元素共同工作的方式，和用户是如何期望他们表现的有个高层次的理解，那会是非常有帮助的。本章介绍了大部分应用程序的建设模块中用到的视图，并描述了它们应该出现在哪里，他们应该如何被使用。

想要了解关于各种界面元素外观、行为以及用法的知识，请你一个要阅读接下来的这章内容。理解每个界面元素是如何被设计出来使用的，能帮助你正确的使用它们，如果有需要，你还可以定制它们达到你的需求。

- [应用程序界面及其内容](#)
- [在应用程序界面中使用视图和控制](#)

### 2.1.1 应用程序界面及其内容 Application Screens and Their Contents

无论是何种类型的应用程序，都会有一个应用程序窗口。通常还说，应用程序窗口为程序内的所有信息提供了一个背景平台。但是多数情况下，用户并不会意识到窗口的存在。取而代之的是，用户通过从一个页面切换到下一个页面来体验你的程序。

这里所说的页面并非是一个具体的概念，你可以认为它就是应用程序里面一幅静态的画面或样式。当用户从上至下的浏览信息、点击不同的标签、或是点击按钮弹出配置页面，用户都将看到一个独立的页面。

你的页面数量可多可少，这是由你的应用程序类型所决定的。拿邮件系统来讲，邮件系统会有账号界面，用于显示对应于不同账号的邮箱。会有页面 用于显示邮箱中的邮件，还会有界面显示每个邮件中具体的邮件内容，再加上一个撰写邮件界面。如果是股票应用程序的话就会有两个页面，一个页面用于显示各个公司的名称以及对应的股票走势图，另一个页面用于应用程序配置。

多数情况下，用户认为应用程序页面和设备页面是相同的。其实不然，应用程序界面完全可以超出设备界面的范围，而这个时候，用户就要滚动页面以看到整个页面内容。举例来说，手机中的联系人页面是一个很简单的界面，即使如此也是需要多个设备页面才能完全显示所有的联系人。

每个应用程序界面都包含了各种各样的视图和控制。一些视图中有自己独有的控件，而一些控件则可应用于多个视图中。

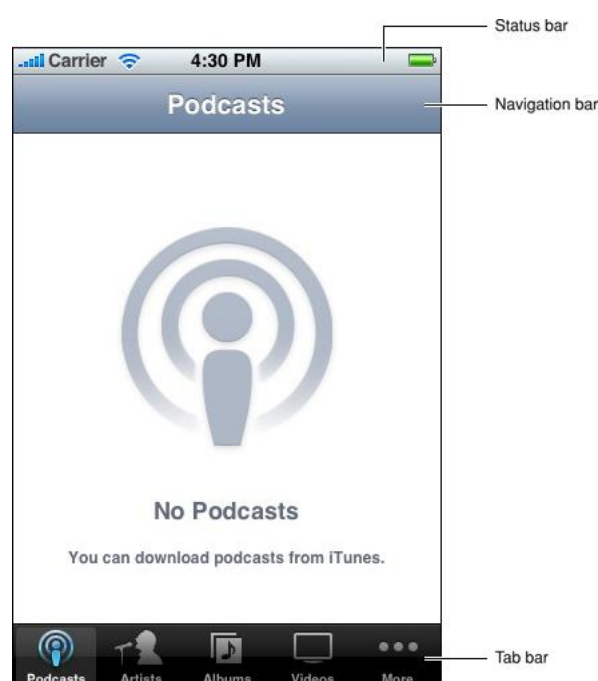
Alerts、Action Sheets 和 Modal 的视图与出现在应用程序中的其他视图不同。他们的视图浮现于应用程序之上。关于这些视图的更多信息，参见 [“Alerts、Action Sheets 和 Modal 的视图”](#)。

在应用程序的界面中，有四种视图有着特殊的状态。虽然这些视图并非在所有的应用程序中都出现或可见，我们这里还是把他们都列举出来：

- **The status bar. 状态栏.** 虽然应用程序可以在一定程度上自定义状态栏的外观, 这个独特视图并非常规应用程序窗口中的一部分。详情参见 [“状态栏”](#)。
- **导航栏 The navigation bar.** 这个可选的控件出现在状态栏之下, 可以包含标题、按钮、分割控件。详情参见 [“导航栏”](#)。
- **标签栏 The tab bar.** 这个可选控件出现于应用程序的页面底部, 用于切换程序的不同状态。详情参见 [“标签栏”](#)。
- **工具栏 The toolbar.** 这个可选控件出现于应用程序的页面底部, 包含了可对当前页面尽心操作的各种控件。详情参见 [“工具栏”](#)。

图 5-1 显示了以上视图中的其中三个。在这个应用程序中, 工具栏是在标签栏之内显示的。

**图 5-1** 一个包含了状态栏、导航栏、标签栏的应用程序页面



在一个包含了这 4 个视图的界面中, 可以将导航栏之下, 工具栏之上的部分定义为内容区域。在该内容区域内, 应用程序可以包含任意的视图来展现 内容, 如表格、网页以及图片等。图 5-2 显示了两个 iPhone OS 中的内容区域: 一种表格和图片。想要了解更多的关于这些视图的外观和行为方式, 以及和它们相关联的空间, 参见 [“表格、文字和网页”](#)。

**图 5-2** 两种在内容区域中的视图



正像之前所提到的，这些都是在特定的视图中才会出现的控件。扩展指示控件就是其中在表格中有着特殊用途的一个。你可以在图 8-1 左手边上看到扩展指示控件（看起来像>）。这些控件都在与他们相关的视图部分中讲到。除此之外，还有更多被广泛使用的控件，如细节指示控件。详情参见[“应用程序控件”](#)。

### 2.1.2 在应用程序界面中使用视图和控件

在 iPhone OS 中，视图和控件的默认外观和行为方式是由 UIKit 所决定的。你应该尽可能多的应用这些 UIKit 中的标准用户界面元素，并遵照它们的习惯用法。这样做将会给你带来如下两种好处：

- 用户已经习惯了那些标准视图和控件的外观和行为方式。当他们第一次使用的时候，他们就可以根据自己以往的经验，来帮助他们学习使用新的应用程序。
- 如果 iPhone OS 更新原本标准视图和控件的外观和行为方式，你只需要花很少的时间就能对应至新的标准。

很多控件都在颜色和内容方面支持一定程度的自定义（如添加文字标签或图片）。如果你是在开发一个 immersive 应用程序，采用与默认控件完全不同风格的新控件是完全可行的。这是因为对于 immersive 应用程序而言，需要的是一个专家级用户能够发现并掌控整个系统的特殊环境。

总的说来，你需要尽量避免过多的更改那些控件的标准外观和用法。如果你使用用户所不熟悉的空间，那么用户就要花费更多的时间来学习如何使用他，并且尝试着去挖掘你的控件在功能上是否和标准控件有所不同。

## 2.2 导航栏、标签栏、工具栏和状态栏

在 iPhone 应用程序中,导航栏、标签栏、工具栏和状态栏这些视图的外观和行为方式都是有明确定义的。这些视图并不会在所有的 iPhone 应用程序中出现(immersive application 中通常不会有这些视图),但是一旦他们出现,正确的使用这些视图就变得非常重要。原因就是,对于基于 iPhone OS 设备的用户而言,这些视图所显示的信息已及它们功能的应用方式都已是非常熟悉。

### 2.2.1 状态栏 The Status Bar

状态栏告诉用户有关他们设备的重要信息,包括手机信号强度、当前网络连接和电池电量等。图 6-1 显示了状态栏的一个例子。

图 6-1 状态栏包含了对用户很重要的信息



虽然状态栏在全屏的 immersive application 中是可以被隐藏的,但是对于这种设计你一定要谨慎考虑。用户希望能够看到他们设备的当前电量等信息。把这些信息隐藏起来,让用户只有在退出应用程序的时候才能看到,这并非理想的用户体验。

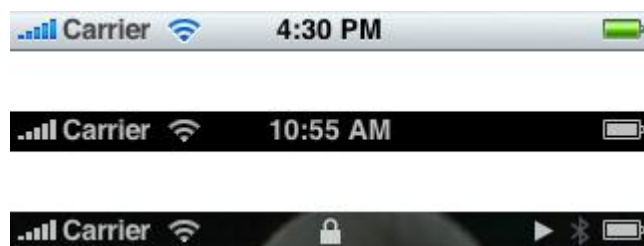
举例来说,iPhone 中的图片程序可以用几秒钟的时间隐藏掉状态栏、导航栏和工具栏等,以全屏的方式显示耽搁图片。这种情况是合理的是因为在图片程序中,用户关注于图片内容,而非与之互动。而且用户只需要轻点屏幕,就可以马上看到状态栏、导航栏和工具栏等。

如果你在应用程序中隐藏了状态栏，那么一定要是出于优化用户体验的考虑去这么做，并且允许用户轻点屏幕就可以重新看到它。除非你有特别令人信服的理由，最好避免自定义一个其他的触控方式来重新显示状态栏，因为用户不会去发现这个新的触控方式或是记住它。

虽然对于状态栏的内容你极少可以控制，但你可以改变它的外观，或是在一定程度上改变它的行为方式。具体的来说你可以：

- 选择是否显示网络活动指示器。当你的设备在进行较长时间的网络操作时，网络活动指示器是非常必要的。当设备的网络操作结束时间比较短时，就没有必要显示网络活动指示器，因为用户还没有注意到的时候它就已经消失了。（在你的代码中，通过 `UIApplication` 方法中的 [networkActivityIndicatorVisible](#) 来设置网络活动指示器的显示。）
- 改变状态栏的颜色。你可以选择灰色（默认颜色）、不透明黑或是半透明黑（Alpha 值为 0.5 的黑色）。图 6-2 显示了这些风格的状态栏。（通过在 `Info.plist` 文件中写入值来改变状态栏的风格，详情参见 [《iPhone 应用程序编程指南》](#)。）
- 选择颜色的改变是否以动画方式改变。（动画效果为：旧颜色的状态栏渐渐滑出屏幕，新颜色的状态栏渐渐滑入。）

图 6-2 三种风格的状态栏



要注意的是，选择的状态栏外观一定要与应用程序的其他部分相一致。比如，如果你的导航栏是不透明的，那就要避免使用半透明风格状态栏。

## 2.2.2 导航栏 Navigation Bars

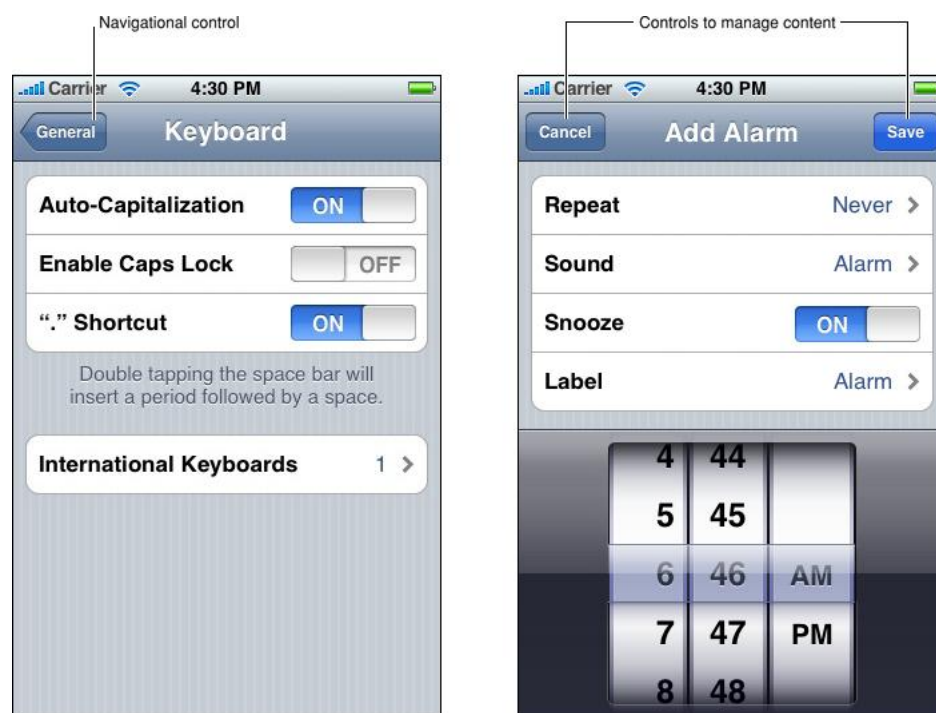
导航栏出现于屏幕的上方，位于状态栏之下。导航栏通常包括当前页面的名称，并且包含了可对页面进行操控的控件，除此之外还可添加导航控件。导航栏在 productivity application（参见 [“productivity application”](#)），因为这些应用程序通常以一种层级的方式组织信息。

通过导航栏的使用可以达成两个目的：

- 在应用程序的不同页面中进行导航
- 提供对当前页面进行操管理作的控件

图 6-3 显示了这两种用法的例子。

**图 6-3** 导航栏可包含导航性控件和页面管理控件



### 导航栏内容 Navigation Bar Contents

一个导航栏可以只是居中显示当前页面的名称，如图 6-4 所示。在一个 productivity application 的起始页面中，由于这个时候用户还没有导航到其他的页面，因此导航栏应该只显示起始页面的名称。

**图 6-4** 只显示当前页面名称的导航栏



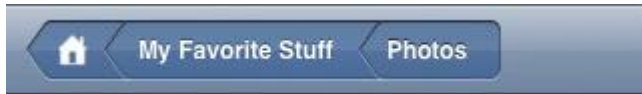
当用户导航到其他的页面上时，导航栏的名称就应相应的变为新的页面名称，并且提供一个以之前位置名称命名的后退按钮。举例来说，图 6-5 显示了 Date&Time 设置中的导航栏，Date&Time 设置是 General 设置的一部分。

**图 6-5** 包含了导航性控件的状态栏



后退按钮为用户回到上一个页面提供了一种可靠的方式，要尽量避免改变该按钮的行为方式。应当尤其避免创建多个部分的后退按钮，如图 6-6 所示。



**图 6-6** 不推荐使用多个部分的后退按钮

使用多个部分的后退按钮会引起很多问题：

- 多个部分后退按钮所占的空间将会使页面名称在状态栏中无处显示
- 无法显示单个部分被选中的状态
- 后退按钮所支持的部分越多，单个部分的点击区域就越小，用户点击的难度就越大。
- 当用户导航到更深层次的页面时，后退按钮应该支持到那个层级就成了个问题。

如果你担心没有这种类似于面包屑导航的多部分后退按钮之后，用户可能会在程序中迷失。那么就说明用户需要非常深入到你的信息结构中才能发现他们所需要的信息。那么你最好重新审视并组织下该程序的信息层级，使其变个更加平缓。

除了后退按钮之外，导航栏还可在页面名称的右方显示第二个按钮。如果你不需要一个后退按钮（因为你的应用程序并不支持层级式的导航），你可以在页面名称的左边使用对当前页面的操控按钮来替代，如“编辑”按钮。图 6-7 显示了这样的一个例子。

**图 6-7** 导航栏可包含页面内容操控按钮。

想要学习如何在你的应用程序中应用导航栏，参见 [“导航控件”](#)。

如上图所示，导航栏中的按钮包含了围绕着他们的斜面。这种风格在 iPhone OS 系统中被称为描边风格。所有在导航栏中的控件都应该使用描边风格。事实上，放入导航栏内的控件都会被自动转换为描边风格。

你可以自己设计导航栏中按钮的图标，也可以利用 iPhone OS 系统中已经定义好了的按钮。参见 [“工具栏和导航栏中的标准按钮”](#) 以获得更多你可以利用的按钮。

虽然你可以为导航栏中的所有文本指定一个字体，但最好还是使用系统字体以获得最大的可读性。当你使用适当的 UIKit 编程界面来创建导航栏时，系统字体会被自动用于显示页面名称。

### 导航栏的大小和颜色 Navigation Bar Size and Color

当把 iPhone OS 设备有竖向改为横向放置时，导航栏的高度会被自动改变（不能通过编程语言指定高度）。在横向放置时，更窄的导航栏为页面内容提供了更多的显示空间。在为导航栏设计图标和为页面设计布局时，一定要注意横屏和竖屏是的高度变化。

你可以为导航栏指定颜色和透明度，从而和你应用程序的整个外观和其他功能栏（工具栏、标签栏和状态栏等）的风格保持一致。你可以自定义一个颜色或是从标准的颜色中选择一个来使用：

- 蓝色 (默认)
- 黑色

如果半透明效果会使你的应用程序变得更加漂亮,你可以将这种效果加入导航栏。当导航栏是半透明效果时,页面的显示空间就会显得更大,这对设备横屏时的显示大有裨益。要避免将半透明的导航栏和不透明的黑色状态栏一起使用(虽然你能够这么做)。

要努力使你的状态栏和其他功能栏的外观保持一致,如果你已经使用了半透明的导航栏,那么就不要再和不透明的工具栏一起使用。此外,如果在设备放置方向不变的情况下,要避免在不同的页面中改变导航栏的颜色和透明度。

### 2.2.3 工具栏 Toolbars

如果你的应用程序允许用户对当前的页面进行一系列的操作,那么最好是为用户提供一个工具栏。工具栏出现于屏幕的最下方,提供可以对当前页面对象进行的操作。工具栏不应当用于转换程序的各种模式。如果你需要模式的转换,可以使用标签栏(详情参见[“标签栏”](#))。

举例来说,当用户在邮件系统中阅读一条消息时,应用程序就会提供一个工具栏,里面包含了一系列的按钮,用于删除、回复和移动该消息,还有查看新邮件和撰写新消息等。通过这种方式,用户可以一直停留在消息查看页面,于此同时进行和管理邮件相关的操作。如图图 6-8 所示。

图 6-8 工具栏为当前的任务提供了多种功能。



#### 工具栏内容 Toolbar Contents

工具栏中的按钮以均匀方式水平分布。对工具栏中的按钮数目进行限制可以使用户更容易点击到他们需要的按钮。推荐的界面元素点击域大小为 44×44 像素，因此少于或等于五个的工具栏按钮是合理的。图 6-9 显示了一个工具栏中合理分布按钮的例子。

图 6-9 合理分配的工具栏按钮



在图 6-8 和图 6-9 里的按钮都没有斜面。在 iPhone OS 中这种风格被称为平整风格。（描边风格参见图 6-7（第 70 页）中的按钮）虽然可以在工具栏中同时使用描边和平整风格，但这种情况要避免发生在同一个工具栏中。

你可以为工具栏上的按钮设计特有的图标，也可以利用 iPhone OS 系统中已经定义好了的按钮。（参见“工具栏和导航栏中的标准按钮”（第 116 页）以获得更多你可以利用的按钮。）如果你选择创建自定义的工具栏按钮，务必要使他们的大小保持一致以形成外观上的平衡和美观。

### 工具栏的尺寸和颜色 Toolbar Size and Color

将设备从纵向改变到横向的同时会自动改变工具栏的高度（你不应该特别指定一个高度）。窄的工具栏可以在横向布局的时候为你的内容留出更多空间。在设计工具栏上的图标或屏幕的布局时要注意高度的不同。

你可以为你的工具栏指定颜色和透明度来使它的整体外观跟你的应用程序以及其他栏（即导航栏、标签栏、状态栏）一致。你可以使用自定义颜色或者选择一种标准颜色：

- 蓝色（默认颜色）
- 黑色

你可以在工具栏中加入半透明的效果，前提是这样做能优化你的应用程序的外观。当你使用半透明的工具栏时，屏幕就有了更大的可视化区域，尤其是在横向使用的时候具有很大优势。

尽量保持你工具栏的外观和你应用程序的其他栏的一致性。例如，如果你使用半透明的工具栏，就不要把它融入不透明的导航栏中。并且要避免在同一个方向（横向或纵向）的不同页面中改变工具栏的颜色或透明度。

### 2.2.4 标签栏 Tab Bars

如果你的应用程序为相同的数据设定提供不同的透视，或者与应用程序所有功能相关联的子功能，你可能需要用到标签栏。标签栏出现在屏幕的底部边缘。

标签栏让用户能够在应用程序中的不同模式或视图中切换,并且用户应该可以在应用程序的任何位置进入到这些模式中。然而,标签栏绝对不应该被当做工具栏(包含控制当前模式中元素按钮)使用。(有关工具栏的更多信息请参见[“工具栏”](#))

例如,iPhone 和 iPod 使用标签栏让用户选择将重点放在他们媒体收集的哪个部分,如播客,艺术家,视频或播放列表。另一方面,在时钟 程序中,标签栏让用户得以进入到四大功能,即世界时钟,闹钟,秒表和定时器。图 6-10 展示了如何在标签栏中选择一个标签来改变时钟的视图。注意在图 6-10 中不同时钟模式中标签栏是如何显示的。用户可以轻而易举的知道他们在哪个模式,并且不管当前是在哪个模式下都允许用户进入到所有其他模式。

图 6-10 在应用程序中使用标签栏切换视图



所有标签栏显示图标和文字的标签都是相同的宽度并且显示黑色背景。当标签被选中后,它的背景淡化并且标签中的图片变亮。图 6-11 显示出了它们的样子。

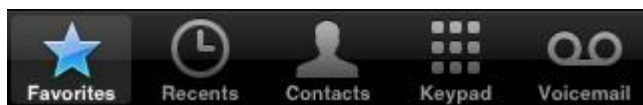
图 6-11 标签栏中一个被选中的标签



注: 标签栏不论哪个方向都不会改变其透明度或高度。

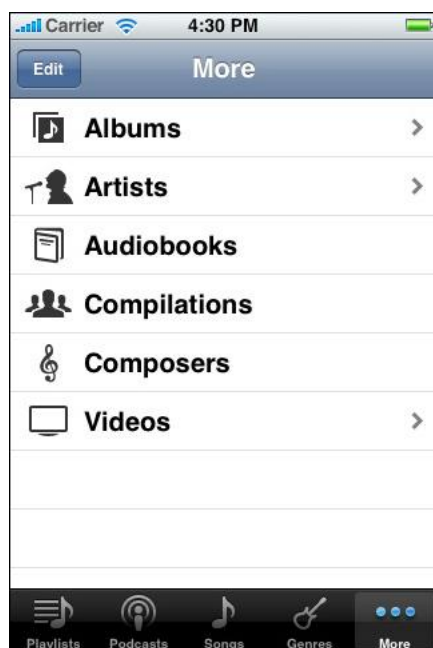
### 提供额外的标签 Providing Additional Tabs

如果应用程序的标签栏有五个或者少于五个的标签,iPhone 操作系统可以用相同的大小显示所有标签,如图 6-12 所示。

**图 6-12** iPhone 操作系统在标签栏中显示不多于五个的标签

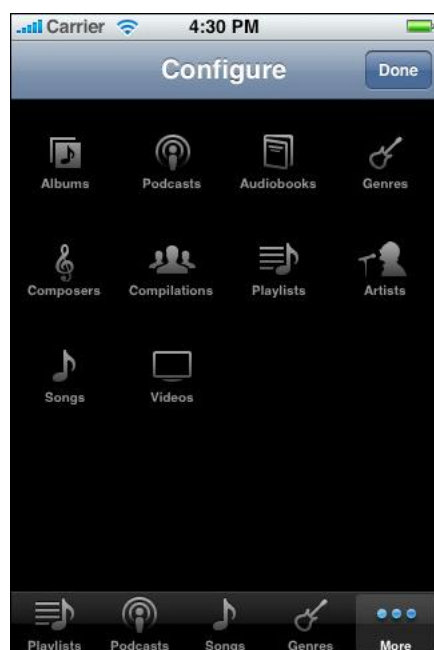
如果应用程序的标签栏包含五个以上的标签, iPhone 操作系统会显示其中的四个并增加一个“更多标签”, 如图 6-11 所示

用户可以查看一个单独在屏幕上显示的其他选项的列表来查找更多标签, 如图 6-13 所示。

**图 6-13** 当用户点击更多标签的时候其他标签被显示出来

在其他选项的页面也可以包含一个编辑按钮, 用户可以点击这个按钮区设定标签栏来显示他们最常用的标签。例如, 图 6-14 显示的就是用户点击其它选项页面中的编辑按钮后所看到的设定页面。

**图 6-14** 当应用程序拥有五个以上的标签时, 用户可以选择他们让他们最喜欢的标签显示在标签栏上。



注意 iPod 在这三个地方使用了相同的标签图标（标签栏其它选项页面，设置页面）。这有助于用户确信无论在哪里显示，每个图标代表的都是相同的一件事情。

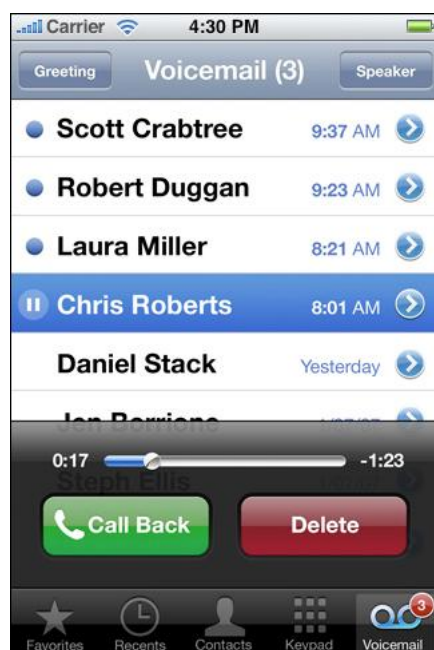
### 在标签栏中标记一个标签 Badging a Tab in a Tab Bar

你可以在标签上显示一个标记以便与用户以无干扰，朴素德方式进行沟通。这种类型的反馈适用于通话信息，虽然对用户的项目或其范围内的东西并不重要，但了解这些是很有用的。这种标签类似于在语音信箱标签上显示一个信息来表明未听的信息：它是一个在标签右上角的红色椭圆形的标记。椭圆形里面白色 的文字提供了信息。

将一个标记与一个特定的标签关联，这使你可以在你的应用程序中用一种特别的方式将标记中的信息联系起来，甚至这种方式不是当前方式。图 6-15 所示的是一个标签上标记的例子。

**图 6-15** 在标签栏上，一个标记传达了某种信息





请注意，如果你注册了苹果推送通知服务并且用户同意使用标记的话，标记同样可以显示在主页中你的应用程序图标上。关于更多有关苹果推送通知服务时如何工作的请参阅“启动推送通知”。

## 2.3 警告、操作列表和模态视图

### Alerts, Action Sheets, and Modal Views

警告，操作列表和模型视图是显示当有东西需要用户注意或者当有额外的选择或功能需要被支持的情况下的视窗类型。图 7-11 所示的是这些视图类型。

图 7-1 警告，操作列表和模态视图



要学习有关这些视图类型的编程，请参阅 [“模型视图控制器”](#)。

- [用法和行为 Usage and Behavior](#)
- [设计警告 Designing an Alert](#)
- [设计操作列表 Designing an Action Sheet](#)
- [设计模态视图 Designing a Modal View](#)

### 2.3.1 用法和行为 Usage and Behavior

警告、操作列表和模态视图都是视图，这就意味着用户必须明确的点击一个按键来消除他们，否则他们将不能继续使用程序。虽然有些时候你需要警告用户潜在的危险或者提供额外的选择，但是避免过度使用这些视图是很重要的，因为：

- 所有模式视图的类型都会中断用户的工作流程。
- 过于频繁的出现要求确认或承认的视图会惹恼用户而不是帮助他们。

警告要特别注意应该谨慎的使用。当警告过于频繁出现的时候，用户可能为了正常工作，不去看就将他们关闭了。

警告、操作列表和模态视图是设计用来传递不同的东西的：

- **警告给予用户影响他们使用应用程序（或设备）的重要信息。**警告通常是用户不希望出现的，因为通常它们会告诉用户出现了问题或者要求用户采取行动来改变当前的情况。
- **操作列表给了用户和他们当前正在进行的动作有关的额外的选项。**当用户点击工具栏中的一个按钮来开始一个潜在的有破坏性的动作（想删除所有最近的通话记录）或完全可以通过别的方式完

成的动作（像为那位用户发送的动作可以指定多个目标中的一个）时，他们会期望出现一个操作列表。

- **模态视图在当前任务的范围内提供了更广泛的功能。** 模态视图也可以提供一个执行跟用户工作流程直接相关的子任务的途径。

这些类型的视图在外表和行为上也是不同的，其中强调了他们发送信息的差异。因为用户习惯了这些视窗的外表和行为，所以在你的应用程序中利用用户的一致性和正确性很重要。接下来的章节将介绍更多关于警告、操作列表和模态视图的用法。

### 警告的用法 Using Alerts

警告窗口在应用程序页面的中间位置弹出并浮动，它用一种高度明显的方式给用户危险的信息。警告中未附加的外表强调了现在需要确认一些程序或设备上的改动的事实，不一定是用户最近的操作。警告弹窗应该显示解释有关情况的位子，最好是能为用户提供一个适当操作的选择方法。

用户已经习惯看到设备或者后台运行的内部程序发出的警告，像消息一类的，不过你应该很少需要在你的应用程序中用到他们。例如，你可能需要警告窗口来告诉用户他们开始执行的任务被阻止了。用消息的形式来显示一个警告时符合逻辑的，因为告诉用户问题是什么病给他们处理它的方法是非常重要的。

你也可以使用警告来给用户一个接受或拒绝有潜在危险结果的机会。在这种情况下，警告窗口应该有两个按键：一个是关闭警告窗口并继续操作，一个事关闭警告窗口并停止操作。通常使用标签为“取消”的按键关闭窗口并停止操作，这样比较符合逻辑。请注意当窗口弹出的时候用户按下“Home”键的结果就是关闭应用程序，这和点击“取消”按钮的结果是一样的：那就是警告窗口会关闭并停止执行操作。

使用警告帮助用户认真考虑的次数并不多。一定要尽可能的在你的应用程序中减少警告的数量并确保每个警告都给用户提供了重要的信息或有用的选择。一般来说，在下列情况下避免警告：

- 用户正在进行任务的时候正常的更新

相反，考虑使用一个 progress view 或者一个 activity indicator 来为用户提供进度相关的反馈。关于这些控件的描述请参阅 107 页的 [“Progress Views”](#) 和 [“Activity Indicator”](#)

- 询问并确认用户开始的操作

想确定一个用户已经开始的操作，甚至一个潜在的危险操作如删除联系人，你应该使用操作列表（在接下来的“操作列表的用法”会有详细描述）。

- 告知用户他们没办法解决的错误或问题

虽然有时候可能必须要用一个警告窗口来告诉用户他们遭遇了不可修复的问题，但是如果可能的话尽量将这些信息整合到用户界面当中。例如，显示最后一次成功连接服务器的时间，而不是在每次连接失败的时候都告诉用户连接失败。

### 使用操作列表 Using Action Sheets

操作列表显示的是与一个用户点击应用程序工具栏按键开始的任务相关的选项的集合。操作列表适用于以下情况：

- 提供可以完成任务不同方式的选择。例如，在图片中，当用户浏览一个特别的图片时可以点击发送按键，然后会出现一个操作列表，给用户三个发送图片目的地的选项（不包括取消发送的“取消”按钮）。

在这种情况下弹出一个操作列表是非常有用的，因为它允许你提供一些列选项，这样会使当前任务的运行环境更加合理，而且不需要在用户界面上给这些选项一个永久的空间。

- 在完成一个具有潜在危险的任务之前得到确认。例如，在邮件的设置中，当用户点击邮件工具栏上的“Trust”按键时一个操作列表会出现，让他们进行删除或取消。

当你在这种情况下显示你的操作列表时你要确保用户知道他们要进行的这个步骤可能会造成危险的影响并且你可以提供一些其他选项。在 iPhone 操作设备中这种沟通时尤为重要的，因为有时候用户会没理由的点击一些控件。

操作列表总是从应用程序的页面底部弹出的并且覆盖住当前页面（如图 7-1 的左图所示）。然而不像警告窗口那样，操作列表的两端是固定在屏幕的边缘的，加强了它与应用程序以及用户最近的操作之间的联系。

一个操作列表有几个按键并允许用户选择怎样完成他们的任务。你不应该在操作列表上加入信息，因为跟当前正在执行任务有关的按键标签应该可以提供足够的信息来帮助用户理解他们的选择。当用户点击一个按键的时候，操作列表会消失。因为操作列表应给提供给用户一些操作上的选择，所以一个操作列表总是提供不止一个按键。

### 使用模态视图 Using Modal Views

默认情况下，一个模型视窗从屏幕的底部边缘滑出，通常会覆盖应用程序的整个页面（如 79 页的图 7-1 中图所示）。因为模型视窗将当前应用程序的页面隐藏起来，这种做法会加深用户的一种认知，就是他们进入到了一个短暂，不同的模式中，在这个模式中他们可以有所作为。

一个模型视窗在适当的情况下可以显示文本，并包含必要的控件来执行任务。此外，模型视窗通常会显示一个按键来完成任务并关闭窗口，还会显示一个用户可以点击的“取消”按键来放弃任务。

模型视窗比操作列表支持更为广泛的用户交互。与操作列表只显示一个单一选项不同，模型视窗支持多步骤的交互，像多个选项或信息输入的选择。

当你需要提供一个完成与应用程序的主要功能相关的整套任务的能力时，那么请使用模型视窗。模型视窗特别适合多步骤的用户界面元素不属于应用程序主界面的子任务。邮件程序中写信模式是有关模型视窗的一个优秀例子。当用户点击“写邮件”按键的时候，会出现一个包含地址和信息的文本区域，输入的键盘，“取消”“发送”按键的模型视窗。

### 2.3.2 设计警告 Designing an Alert

你可以定义文字的内容, 按钮的数量, 以及按钮上显示的文字, 但是你不能定义警告窗口的宽度、背景色以及文字的对齐方式( 已经被预定于为居中对齐 )

**注意:** 当你阅读这些规范时, 需注意以下定义:

- 标题式大写: 除了冠词、连词、介词( 由四个或者更少的字母组成 ) 外的每个单词都大写.
- 句子式大写: 第一个单词大写, 其余单词小写( 除非是专属名称或者专属形容词 ).

警告窗口的标题( 警告信息也同样 ) 应该精炼的描述目前的情形并且向用户解释, 他们能够做些什么, 理想情况下, 你写的文字能够给用户足够的上下文信息, 去帮助他们理解为何会出现警告, 并且接下去应该去点击哪个按钮.

**当你写警告窗口的标题时:**

- K 如果可能, 尽可能简短, 以便能在一行显示, 冗长的让用户不能快速阅读, 并且造成文本区域内的滚动
- 不要只用一两个单词, 例如 “错误”、“警告” 等, 这样无法提供有用的信息
- 使用句子片段, 一条简短并且有信息量的描述比一个完整的句子更容易理解
- 无需避免消极或者否定的词汇。人们可以理解, 大部分的警告窗口都是在告诉他们有麻烦了或者警告他们当前的危险情况, 因此直接和消极的陈述要比拐弯抹角好的多
- 尽可能避免使用 “你” “你的” “我”, 有时, 直接指向人们的词语会被曲解甚者被认为是一种侮辱
- 下列情形使用没有结束标点的标题式大写
  - 标题是句子片段
  - 标题是一个句子, 但非疑问句
- 如果标题由一个疑问句组成, 使用有结束标点的句子式大写。一般情况下, 如果能够避免增加一条信息的话, 可以尽可能的使用疑问句作为警告窗口的标题
- 如果标题由两个以上句子组成, 使用句子片段式大写, 并且为每个句子加上结束标点。很少情况下会使用到两条句子作为警告窗口的标题, 但是如果避免再增加一条信息, 可以考虑使用

如果你使用可供选择的警告窗口

- 尽量保持信息简介, 以便于在一行或者两行内显示。如果信息过长, 文本区将不得不滚动

- 尽量使用句子片段式大写，警告信息应该是一个完整的句子

**不要描述应该点击哪个按钮而导致文字加长**，比如“点击按钮去查看信息”，理想情况下，明确的警告文字和按钮文字，能够告诉用户足够的信息去理解目前的情况并做出正确的选择，但是，如果你必须提供更多的细节指导，请遵循：

- 使用单词“点击”（而不是“触摸”“触碰”“选择”）来描述动作
- 不要用引号把标题引起来，但是一定要大写

**要在横向和竖向两个方向测试警告窗口。**因为警告窗口的高度是在横向方向限制的，这会在竖向上看起来不太一样，推荐的方法是优化文字内容，以便于在两个方向看起来都没有问题（避免滚动）

**推荐使用两个按钮的警告窗口。**两个按钮的警告窗口是最有效的，因为它能让人们更容易选择；如果只给用户一个按钮并不是一个好主意，因为这样的警告窗口只能提供信息，用户对当前情况无法作出任何选择和控制；三个或者更多的按钮明显比两个按钮更复杂，应当尽量避免使用。事实上，如果你需要用户做出两个以上的选择，你应该考虑使用操作列表

**按钮颜色应该适当。**警告按钮有两种颜色：深色和浅色，在有两个按钮的警告窗口中，左边的按钮通常是深色的，右边的按钮通常是浅色的，在只有一个按钮的警告窗口中，按钮颜色通常是浅色的

- 在包含两个按钮的警告窗口中，如果这个窗口是有风险操作的警告，取消按钮应该是放在右边（浅色）
- 在包含两个按钮的警告窗口中，如果这个窗口是用户想要开始的一个动作，取消按钮应该放在左边（深色）

**注意：**取消按钮是深色或者浅色都可以，位置在左边在右边也都可以，这主要取决于另外的按钮触发的行为是否具有毁坏性，要正确识别在代码中哪个按钮是取消按钮

- 所有按钮的标题，使用没有结束标点的标题式大写
- 和警告文字相连的词语，推荐使用动词或者动词词组，比如“取消”“允许”“回复”或者“忽略”
- 对于一个简单接受性的选择，推荐使用“OK”，而不是“Yes”或者“No”
- 尽量避免使用“你”“你的”“我”，按钮标题使用这些字眼，不但模糊而且不敬

### 2.3.3 设计操作列表 Designing an Action Sheet

你可以为操作列表选择一种配合应用程序外观的背景，并且你可以指定按键的数量跟他们的内容。



跟警告窗口不同,操作列表不需要显示文本信息。应为操作列表的出现是用户操作之后的结果,如点击“删除”或“发送”按键,所以没有必要解释操作列表的出现。

操作列表可以拥有两种不同的背景显示。你要确保应用程序中操作列表的背景跟应用程序中工具栏和导航栏中的显示是搭配的。例如,如果应用程序使用的是黑色的导航栏和工具栏,操作列表的背景就应该是半透明的黑色。默认情况下,iPhone 操作系统使用标准蓝色工具栏以及导航栏搭配的标准蓝色背景来显示操作列表。应用程序中的所有操作列表都应该具有相同的背景颜色,这种背景颜色应该与导航栏以及工具栏的颜色相搭配。

一定要在操作列表的底部显示“取消”按键。这样做会鼓励用户在读到最下边“取消”选项之前阅读其他所有选项。

图 7-3 所示的是一个默认背景的操作列表以及“取消”按键的推荐位置。

**图 7-3** 一个典型的操作列表



如果你要提供一个按键来执行具有潜在危险的操作,像删除所有用户购物列表中的条目,这时候你应该使用红色的按键。在操作列表的顶部显示这种具有危险性的按钮是非常重要的,因为:

- 越接近操作列表顶部的按键越明显可见。
- 有时候当用户想要点击“主界面”按键的时候他们会不小心点到屏幕底下。将操作列表有危险的按键放置在远离底部的位置,这样当用户没有点到“主界面”按键而点到屏幕上的时候他们就不大可能会造成不希望看到的结果。

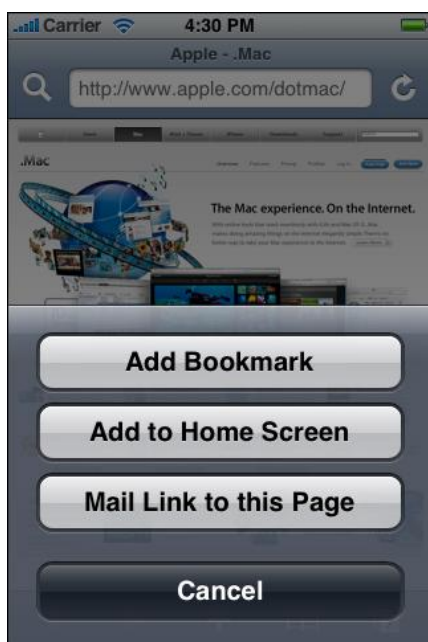
图 7-4 显示了一个黑色半透明背景的操作列表以及“取消”按键和另外一个具有危险性的按键的合适位置。

**图 7-4** 一个执行具有破坏性操作的按键应该是红色的并且定位在操作列表的顶部



你可以在操作列表中显示若干按键，只要确保每个按键和其他的相比很容易辨认。图 7-5 所示的是一个背景与标准蓝色工具栏相搭配的操作列表并提供了三个备选选项以及“取消”按键。

图 7-5 拥有 4 个按键的操作列表



### 2.3.4 设计一个模态视图 Designing a Modal View

模态视图的整体外观应该与显示它的应用程序相协调。例如，模态视图通常包含一个导航栏，导航栏中包含一个标题和一些用于取消或完成模态视图中的任务的按钮。模态视图的导航栏应与应用程序中的导航栏有相同的背景。

模态视图通常应该显示一个标题，用于定义它的任务。如果必要，还可以在视图中的其它位置显示文字，以更充分的描述任务或提供一些引导。例如，当用户想撰写文字信息时，邮件应用程序会提供一个模态视图。如图 7-6 所示，这种模态视图的导航栏显示了一个与应用程序导航栏相同的背景，在标题中还有新信息的文字提示。

图 7-6 模态视图应该和应用程序界面相协调



在一个模态视图中，你可以使用任何一种控件来完成任务。例如，你可以包含文本域，按钮和表格视图。

你可以这样来使用一个模态视图，它既与你的应用程序相协调，又能提升用户对于视图所代表的当前内容变化的认知。要做到这一点，你可以指定以下这些视图间过渡的方式：

- **Vertical. 垂直。**显示的时候，模态视图从屏幕底部向上滑动；消失的时候，再从顶部向下滑动到底部。（这是默认的过渡方式。）
- **Flip. 翻转。**当前视图从右向左翻转，以显示模态视图。从视觉上看，模态视图好像是当前视图的背面。当模态视图消失时，沿水平方向，从左至右翻转，然后显示原来的视图。

如果你决定更改应用程序中的模态视图的翻转样式，要避免仅仅为了获得不同的形式而这样做。请注意，用户会觉察到这种差异，并推断出这种差异 有其他什么寓意。由于这个原因，最好建立一个合乎逻辑的，一致的模式，这样用户可以轻松地识别和记忆，避免在不必要的时候改变了视图间翻转的风格。

## 2.4 表格视图、文本视图和 Web 视图

### Table Views, Text Views, and Web Views

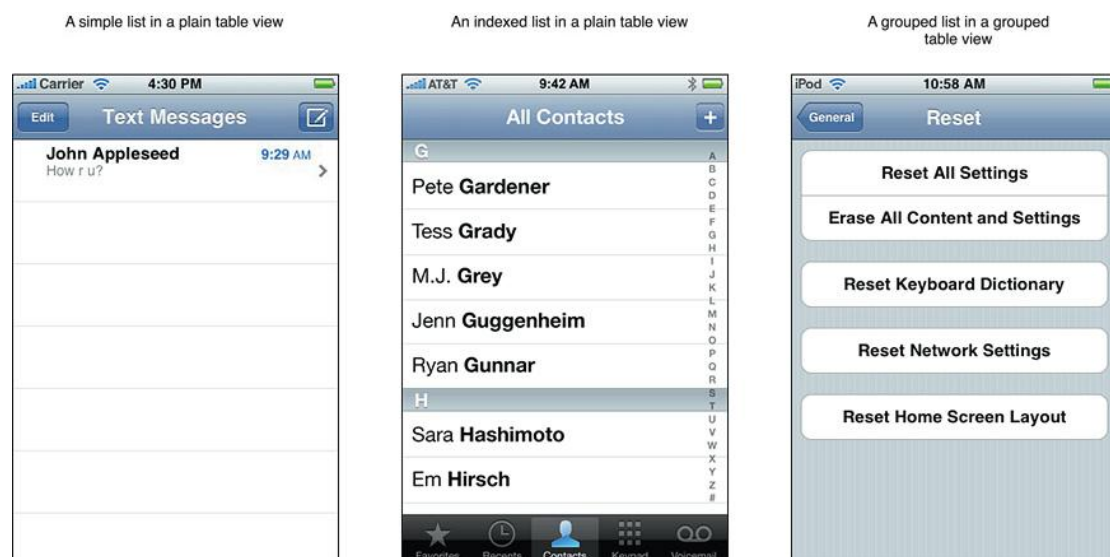
通用元素表格视图、文本视图和 Web 视图在 iPhone 应用程序中有着不同的用途。例如,表格视图可用于显示简短的列表选项、详细信息分组列表或者较长的项目索引列表。文本视图和 Web 视图则可以相对不受限制地用于接受和显示内容。

- [表格视图 Table Views](#)
- [文本视图 Text Views](#)
- [Web 视图 Web Views](#)

#### 2.4.1 表格视图 Table Views

**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字,图片和控件。用户可以在行间或者分组之间滑动或拖移来进行滚动。图 8-1 给出了不同样式的表格视图在显示列表时的不同方式。

图 8-1 使用表格视图显示列表的三种方式



#### 用途和行为 Usage and Behavior

表格视图在 iPhone 应用程序中相当有用,因为无论信息数量多少,都能提供具有吸引力的组织方式。尽管实用程序也能利用一些小规模的表格视图,但表格视图通常在那些需要处理大量用户数据的生产力应用

程序中最有用。沉浸式应用程序一般不会使用表格视图来显示信息，但它可能会用它来显示较短的选项列表。

表格视图提供了让用户浏览和处理信息的内置元素。此外，表格视图还支持：

- 显示页眉页脚信息。你可以在列表中的每一节或组的上方或下方，或者是整个列表的上方或下方显示说明文字。
- 编辑列表。你可以允许用户以统一的方式对列表项进行添加、删除及排序。表格视图还支持多项选择和操作，这就使得你能够向用户提供方便的一次删除多个列表项的操作方式。

**当用户选择一个列表项时，表格应始终提供反馈。**如果一个项目可以被选择，那么一旦用户进行选择，包含这个项目的行应高亮一下，给用户反馈，表示该选择已经收到。然后，立即采取行动：要么显示新视图，要么该行显示复选标记以表明该项目已被选中或启用。

在极少数情况下，当与该行项目有关的附加详细信息或控件显示在同一屏上时，该行可能仍然会持续高亮显示。不过，并不鼓励这样做，因为很难将选项列表、已经选定的项和相关的详细信息或控件同时显示而又不显得拥挤难受。

如果因为选择了一个行而转到一个新的屏幕，那么选中行应当先高亮一下，然后新屏幕再滑入。当用户返回前一屏时，原先选定的行应该再次高亮一下，提示用户他们先前所做的选择。

注意，你也可以用动画展现用户对列表项所做的更改。这是提供反馈并强化用户直接操纵感的好方法。例如在“设置”中，当你关闭自动设定日期和时间（通过在“日期与时间” > “自动设置”中选择关闭），列表分组将平滑展开，以显示“时区”和“设置日期和时间”两个新项。

**表格应立即显示内容。**如果表格中的内容非常庞杂，不要等到所有数据都加载完成再显示。相反，立即在屏幕可见的行中显示文本数据，而较复杂的数据（例如图片）则当其就绪后再显示。这个技巧能够立即向用户提供有用的信息，感觉上也提高了你的应用程序的响应速度。

如果应用程序显示数据变化不是很频繁，你可以考虑在等待新数据加载的过程中先显示“过期”的数据。这个技巧让用户也能立即看到一些有用的信息，但对于处理数据频繁改变的应用程序则不推荐使用。在你决定这么做之前，要先估计数据变化的频率，以及有多少用户希望迅速看到更新的数据。

如果很难立即显示有用的内容，那么避免显示空行也非常重要，因为空行可能意味着该应用程序死机。相反，应在屏幕中央显示一个旋转的活动指示器和一个信息描述标签，如“正在加载...”。如果你能显示旧的数据，就不必担心空行，但应当尽快更新可见区域的数据行。这两种技巧都向用户提供反馈，让他们知道程序正在继续进行。

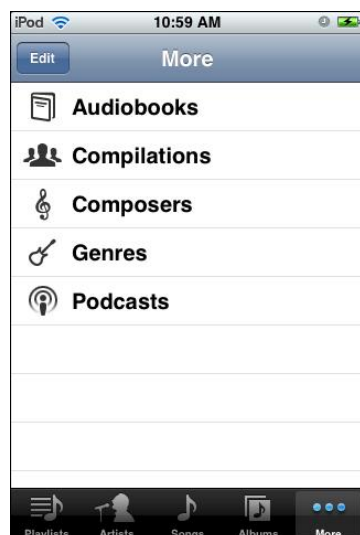
## 表格视图的样式 Table-View Styles

iPhone OS 定义了两类表格视图样式，其主要区别在于外观：

**Plain 普通**([UITableViewStylePlain](#))。在这种表格视图样式下,表格行伸展至屏幕两端,行背景为白色。表格行可以被划分为若干个带标签的节,也可以在表格视图的右端垂直显示可选的索引。

图 8-2 显示了 iPod 应用程序中的普通表格(无页眉、页脚、索引)。

图 8-2 普通表格中的简单列表



**Grouped**([UITableViewStyleGrouped](#))**分组**。在这种表格视图样式下,每组表格行显示时会相对屏幕两侧缩进一些。这些分组显示在独特的垂直条纹背景上,而组内背景则是白色的。带分组的表格可以包含任意数量的分组,每个分组可以包含任意数量的行。可在每个分组之前添加页眉文本,之后添加页脚文本。这种表格视图样式不支持索引。

图 8-3 显示了带分组表格中的列表,其中每组包含一个表格行。“设置”应用程序中的这个列表不包含页眉和页脚文本。

图 8-3 在群组表格中一个包含四个组的清单





## 单元格样式 Table-Cell Styles

iPhone OS 3.0 及后续版本包含了四种预置的单元格样式，可以简单快速地针对普通和分组表格生成常用的表格行布局。注意，从编程上来说，这些样式是应用在表格视图单元格上的，单元格用于控制表格如何绘制各行。

当使用标准单元格样式时，你的应用程序将与内置应用程序保持一致，其好处在于：

- 用户可以更快地了解如何使用你的应用程序
- 如果将来标准单元格样式发生更改，你无须花费大量的精力即可维持你应用程序的一致性。

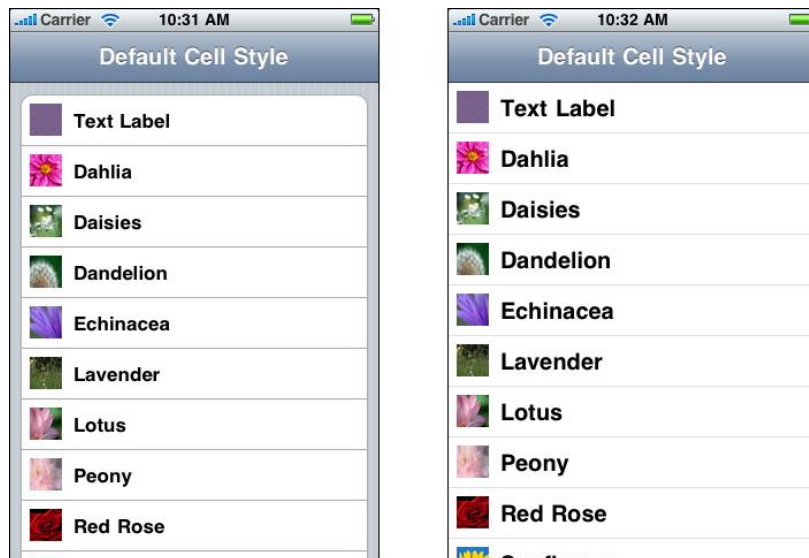
如果你希望使用非标准的表格行布局方式，最好创建一个自定义的单元格样式，不要对标准样式进行大规模的扩充修改。[Table View Programming Guide for iPhone OS](#) 中的 [“Customizing Cells”](#) 包含如何创建自己的单元格样式的信息。

注意，在任何单元格样式中，文本都会被自动截断。一般来说，你应当确保你的文本尽可能简洁以避免单词或短语被截断后对于用户来说难以理解。文本截断的问题可大可小，具体取决于你使用的单元格样式和截断出现的位置。

iPhone OS 提供下列标准单元格样式：

- **默认**单元格样式 ([UITableViewCellStyleDefault](#)) 包含一个位于左侧的可选图像，右侧是左对齐的黑色文本标签。

图 8-4 带分组的表格（左）和普通表格（右）中的默认单元格样式

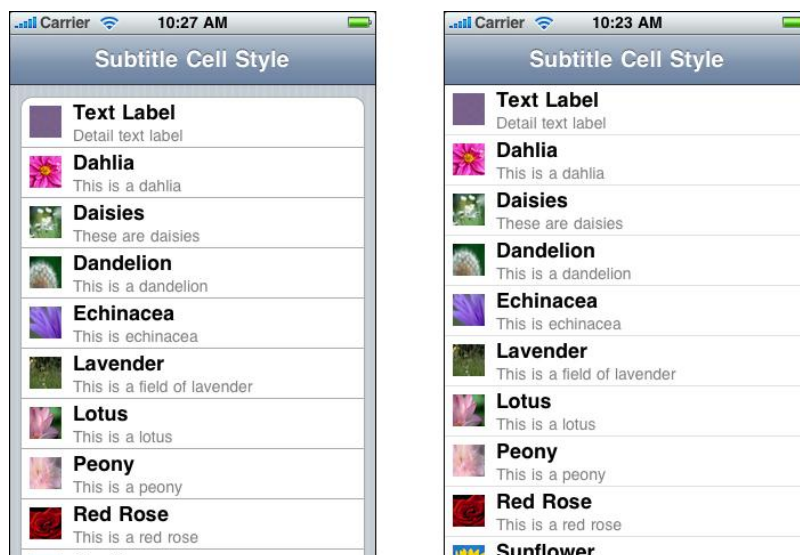


文本标签的外观意味着它所代表的是项目名称或标题，左侧对齐使得列表易于扫视。默认样式适用于显示一组无须通过附加信息进行区别的项目。

文本标签越短越好,但如果截断不可避免,尽可能保证最重要的信息出现在最开头的几个单词中。

- The **subtitle** 副标题单元格样式 ( [UITableViewCellStyleSubtitle](#) ) 包含一个位于左侧的可选图像,右侧是一行左对齐的文本标签,下方是一行左对齐的详细文本标签。文本标签是黑色的,详细文本标签则为稍小的灰色字体。

图 8-5 带分组的表格 ( 左 ) 和普通表格 ( 右 ) 中的副标题单元格样式

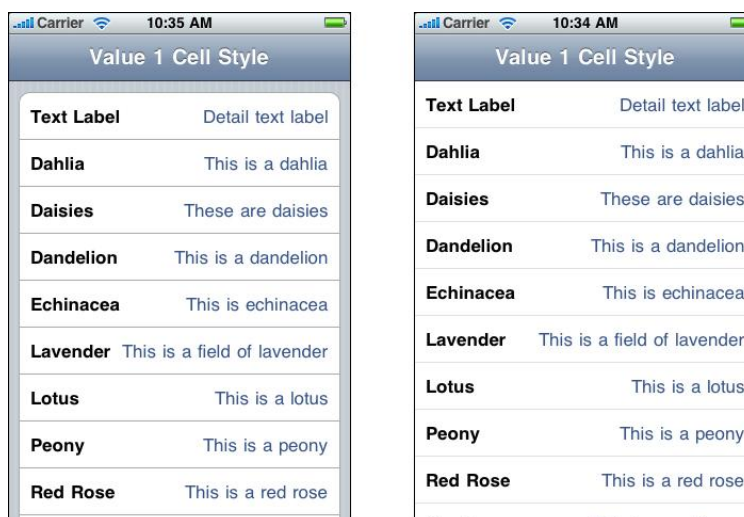


文本标签的突出外观意味着它代表了项目名称或标题,而详细文本标签的细微外观意味着它包含的是与此项相关的附属信息。文本标签左对齐可使列表更易于扫视。该单元格样式适用于列表项互相较为相似的情况,用户可以通过详细文本标签中的额外信息来帮助区分列表项。

文本标签应当简短避免截断。如果截断不可避免,尽量让最重要的信息出现在最开头的几个单词中。如果详细文本标签被截断,用户通常不会太在意,因为他们只是将其视为对文本标签所给出的条目名称的一种加强和补充。

- 值-1 型单元格样式 ( [UITableViewCellStyleValue1](#) ) 是在同一行显示黑色的左对齐文本标签,并以稍小蓝色的字体显示右对齐的详细文本标签。该样式不包含图片。

图 8-6 带分组的表格 ( 左 ) 和普通表格 ( 右 ) 中的值-1 型单元格样式



文本标签的外观样式意味着它表示的是条目名称或标签,而详细文本标签的外观意味着它提供了与条目相关的重要的信息。

文本标签的左对齐方式及其字体能够方便用户扫视列表以寻找条目,详细文本标签右对齐则能够使用户的注意力集中在所提供的相关信息上。该单元格样式适用于显示条目的当前值,可能是选自子列表。

在这种布局下,文本截断很难避免(因为一行要显示两个标签),但还是值得努力尝试的。否则,左右标签之间将没有留白,用户不容易理解两组信息之间的关系。

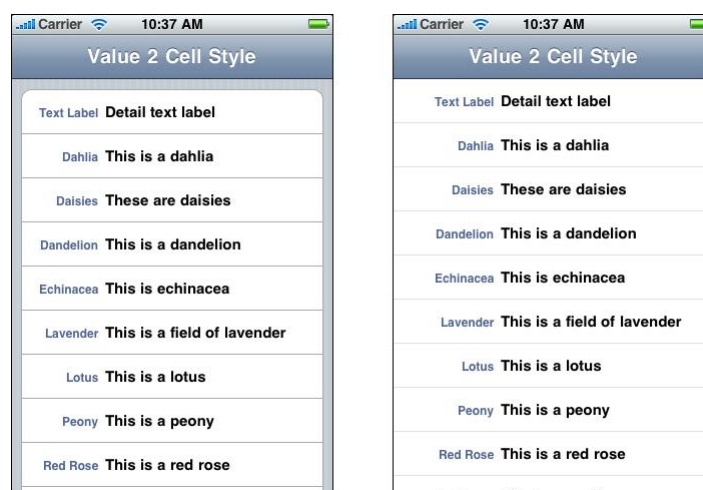
尽管值-1 型单元格样式在普通或者分组表格中均可使用,但其外观更适用于分组表格。例如,“设置”中的“使用情况”屏幕即在分组表格中使用了值-1 型样式。

图 8-7 值-1 型单元格样式最适用于分组表格



- 值-2 型单元格样式 ([UITableViewCellStyleValue2](#)) 在左侧以较小的蓝色字体显示右对齐的文本标签,在右侧同一行上以较大的黑色字体显示左对齐的详细文本标签。该样式不包含图片。

图 8-8 带分组的表格(左)和普通表格(右)中的值-2 型单元格样式



文本标签的右侧对齐方式、受限宽度及其字体意味着它是作为标题来使用的,而更突出、左侧对齐的详细文本标签则包含更重要的信息。

在这种布局方式下,每行的左右标签都在同一个位置相互靠齐,在列表的文本标签和详细文本标签之间产生了清晰的垂直间距,使用户能够专注于详细文本标签的首字词。如果文本标签被截断,就会影响这种垂直条带的清晰度,用户扫视详细文本标签中的信息则可能变得困难。

尽管值-2 型单元格样式在普通或者分组表格中均可使用,但其外观更适用于分组表格。例如,“联系人”中的“信息”屏幕即在分组表格中使用了值-2 型样式。

图 8-9 值-2 型单元格样式最适用于分组表格



**注：**所有的标准单元格样式都允许添加表格视图元素,比如复选标记或者继续展开指示符。要注意添加这些元素会减少标题及副标题可用的单元格宽度。

你可以通过增加表格行的高度支持文本折行来避免文本截断,但这也可能会有问题：

- 你需要通过编程方式来检查文本长度并确定是否需要折行。你必须在纵横两个方向都进行确定,因为文本折行受表格宽度的影响。
- 你应当避免一个方向上出现文本折行而另一个方向上没有。
- 不同的行会对你应用程序中表格视图的整体性能产生负面影响,无论使用何种表格视图样式。

最后,即使在分组表格使用不同的行高还可以接受,但在普通表格中这会显得杂乱不均。

## 表格视图的元素 Table-View Elements

iPhone OS 包含一些能够扩展表格视图功能的**表格视图元素**。除非另有说明,否则这些元素只在表格视图中适用。一定要确保这些元素在应用程序中的正确使用,因为用户已经习惯了它们在内置应用程序中的外观和行为。

**注:**从编程上来说,表格视图元素是以不同方式实现的。有些是单元格(控制表格如何绘制各行的对象)的附属视图,而其他的则当表格视图进入编辑模式后显示。要了解管理这些元素的不同方式,参见 [iPhone OS 表格视图编程指南\(英文\)](#)。

- **Disclosure indicator.继续展开指示符。**当此元素出现时,用户知道他们可以点击该行的任何位置以查看下一级信息,或是与该列表项相关的选项。

当选择表格行将显示另一个列表时,应当使用继续展开指示符。不要使用继续展开指示符去表示该列表项的详细信息,在这种情况下应该使用详情展开按钮。

- **Detail disclosure button.详情展开按钮。**用户点击该元素以查看关于该列表项的详细信息。(注意,你也可以在除表格视图以外的其他视图中使用该元素,以展现某物额外的详细信息。更多信息,参见[“详情展开按钮”](#))。

在表格视图中,在表格行中使用详情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,详情展开按钮可以执行与选定该行不同的操作。例如,在电话收藏夹中,点击行能够呼叫联系人,点击行中的详情展开按钮可以显示更多关于该联系人的信息。

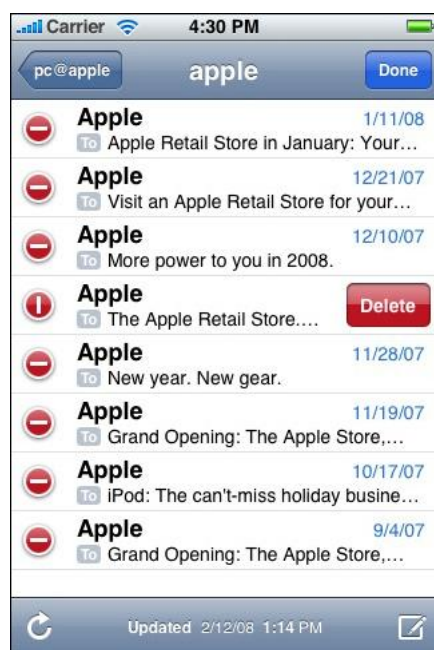
- **Delete button.删除按钮。**用户点击此元素以删除列表项。当用户在该行上扫过或者当他们在编辑状态下点击删除控制按钮时,该元素就会在列表项的右侧出现。(参见元素的示例:图 8-10)。
- **Delete control button.删除控制按钮。**用户点击此元素以显示和隐藏每个列表项的删除按钮。为了向用户提供额外的反馈,当用户点击删除控制按钮时,其中的减号会由横向变为纵向。图 8-10 给出了该元素的一个示例。

在一个支持临时编辑模式的分组表格中,删除控件显示在表格视图外部的左侧。例如,你可以在“通讯录”中编辑具体某个人的信息时看到。在处于持续编辑模式(比如“股票”和“天气”背后的分组表格)下的分组表格中,删除控件显示在表格内部的左侧。

在普通表格中,删除控件始终显示在表格内部的左侧,如图 8-10 所示。

- **Row insert button.插入行按钮。**用户点击此元素向列表中添加一行。
- **Row reorder control.行排序控件。**当此元素出现时,用户可以将表格行拖移到列表中的另一个位置。
- **Checkmark.复选标记。**该元素显示在一个列表项的右边,表明它当前已被选定。

**图 8-10** 表格视图可以显示删除按钮和删除控制按钮



## 开关控件 Switch Controls

一个**开关控件**向用户呈现了两个互斥的选项或状态，如是/否或者开/关。开关控制一次只能显示两个可选项中的一个，用户滑动控件以显示另一个隐藏的选项或状态。图 8-11 给出了开关控制的一些示例。

图 8-11 表格视图中的开关控件



当你需要向用户提供两个简单而又截然相反的选项时，可以在分组表格视图中使用开关控制。因为其中一个选项始终是隐藏的，所以最好是在用户已经知道两个选项是什么时才使用开关控件。换句话说，不要让用户仅仅为了弄清另一种选项是什么，而去滑动开关控件。



你可以使用开关控件来改变视图中其他用户界面元素的状态。根据用户的选择不同,新的列表项可能出现或消失,或列表项可能变为有效或无效。

### 使用表格视图实现普通用户操作 Using Table Views to Enable Common User Actions

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配置方式来支持不同的用户操作,例如:

- **Selecting options. 选择选项。**

iPhone OS 并不包括类似于菜单或弹出菜单的多项选择控件,而表格视图适用于显示可让用户进行选择选项的列表。这是因为表格视图简洁且不拥挤。此外,表格视图提供了复选标记图像,用于显示用户当前在列表中选中的一个(或多个)选项。

如果你需要显示的是用户在点击列表中某一行时,所看到的选项清单,你可以使用表格视图的任意一种样式。但是如果你需要显示的是,当用户为了看到的选项清单,而点击列表外的一个按钮或其他用户界面元素时,请使用简洁的风格。

- **Navigating hierarchical information. 导航层次信息。**

表格视图适用于显示具有层次结构的信息,其中每个节点(即列表项)可以包含自己的子信息,这是因为每个子元素可以显示在单独的列表中。这样用户可以很容易地沿着层次结构中的路径,从每层列表中选择一项。继续展开指示符元素意味着用户在表格行的任意位置点击即可在新的列表中查看子信息。

当表格用于导航时,在用户沿着层次结构返回时,先前选中的表格行不应当保持高亮显示。

- **Viewing conceptually grouped information. 查看在概念上进行分组的信息。**

你可以使用任意一种表格视图样式对信息进行逻辑分组,如工作、家庭或学校。普通和分组表格中都可以通过添加页眉和页脚文本为每节添加上下文。

一般来说,分组表格为分组提供了清晰的视觉识别,因为用户能够很容易地区分分组的圆角,即使在快速滚动的时候。

- **Looking up indexed information. 查找索引信息。**

如果使用的是普通表格,可以通过显示索引来帮助用户快速找到他们需要的东西。索引由浮在屏幕右边缘的一列条目(通常是字母表中的字母)组成。用户点击(或拖移至)某个索引项以定位至列表中的相应区域。索引在当列表可能延伸数屏时最为有用。

如果你在普通表格中使用索引,应当避免使用显示在表格的右边缘的表格视图元素(如继续展开指示符),因为这些元素会影响索引的使用。

## 2.4.2 文本视图 Text Views

**文本视图**是一个能够显示多行文本的区域，当内容太多而超出其适应范围时，支持滚动。邮件使用一个文本视图以使用户创建签名，签名显示在每封电子邮件的结尾处，如图 8-12 所示。

图 8-12 一个文本视图显示多行文本



虽然你可以使用一个文本视图显示多行文本，如一个大文本文件的内容，你也可以使用文本视图以支持用户编辑。如果你使用可编辑的文本视图，当用户在文本视图点击时就会出现一个键盘。键盘的输入法和布局是由用户的语言设置决定的。当用户点击按钮“?.123”（如图 8-12 所示）时，键盘的发生 改变以便于输入数字和标点符号。你也可以指定不同的键盘风格，这取决于你希望用户输入的文本类型。见 [“文本区域”](#) 有你可以参考的样式描述。

你可以在文本视图控制文字的字体，颜色和对齐方式，但它们只适用于文本的全部内容。换句话说，你不能改变某一部分文本的任何属性。文字的字 体和颜色属性的默认值，如你所期望的，系统字体，黑色，因为他们往往是最可读的。对齐属性默认为左对齐（你可以更改为中心对齐或右对齐）。

如果你必须要在一个视图，显示各种字体，颜色，或对齐方式的文本，你可以使用 HTML 来控制文本而不是一个文本视图。

## 2.4.3 Web 视图 Web Views

一个 **Web 视图** 是一个能够在应用程序的屏幕中显示丰富的 HTML 内容的区域。例如，邮件使用网页视图显示邮件的内容，因为它可以包含纯文本不能包含的元素（图 8-13 显示了一个例子）。

**图 8-13** 一个网页视图可以显示基于 Web 的内容



除了显示网页内容，网页视图提供了一些在打开的页面中支持导航的元素。虽然你可以提供网页导航的功能，但是最好避免造成的应用程序看起来像一个微型浏览器的行为。

如果你有一个网页或基于网络的应用程序，你可以选择使用网页视图来实现一个简洁的 iPhone 应用程序，因为网页视图可以为它提供封装。如果你计划访问你所控制的网页内容，请务必阅读 Safari 网页内容指南，以了解如何创建，在基于 iPhone 操作系统的设备上兼容并优化显示的网页内容。

## 2.5 应用程序控件

### Application Controls

iPhone 操作系统提供了一些你可以使用在你的应用程序中的控件，其中大部分已被基于 iPhone 操作系统的设备的用户所熟知。这些控件有很多是为了在特定的地方，如在表格视图中使用，但有些是适用于一般用途的。本章介绍的控制，可适用于你的应用程序中的任意视图。

当你设计应用程序的用户界面时,永远记住,用户希望对于他们熟悉的控件,要使用与他们的内置应用程序相同的控件行为。只要适当地使用这些应用程序的控件,这就会是你的优势。

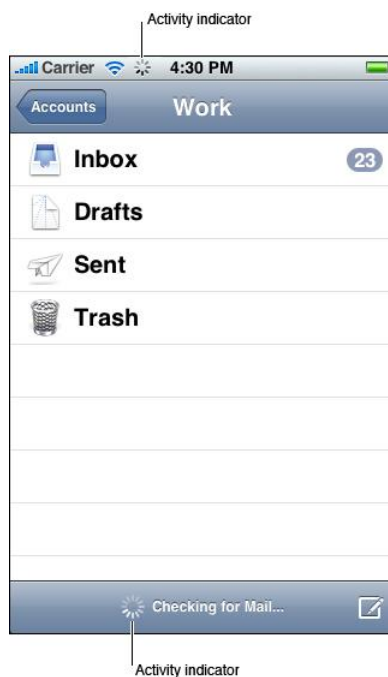
- [活动指示器 Activity Indicators](#)
- [日期时间选择器 Date and Time Pickers](#)
- [详情显示按钮 Detail Disclosure Buttons](#)
- [信息按钮 Info Buttons](#)
- [标签 Labels](#)
- [页面指示器 Page Indicators](#)
- [选取器 Pickers](#)
- [进度视图 Progress Views](#)
- [圆角矩形按钮 Rounded Rectangle Buttons](#)
- [搜索栏 Search Bars](#)
- [分段控件 Segmented Controls](#)
- [滑块 Sliders](#)
- [文本框 Text Fields](#)

### 2.5.1 活动指示器 Activity Indicators

一个活动指示显示了一项任务的进度,或是一个持续时间不明的进程。如果你需要显示一个已知时间的任务进度,使用进度视图(详细信息请参见[“进度视图”](#))。以“spinning gear”方式显示的活动指示向用户说明了,正在处理中,但并不能说明何时会完成。

图 9-1 显示了两种类型的活动指示。在状态栏里的活动指示是一个网络活动指示,它应在你的应用程序访问网络超过几秒钟后显示。工具栏上的较大的活动指示应在你的应用程序执行当前任务超过 1 秒或 2 秒时显示。

图 9-1 两种类型的活动指示



一个活动指示是一个很好的反馈机制，它在让用户确认它们的任务或进程并没有停滞不前时使用，比在告知用户处理即将完成时使用，起到更为重要的作用。

你可以选择一个活动指示的大小和颜色，以适应使用它的视图的背景。默认情况下，活动指示是白色的。

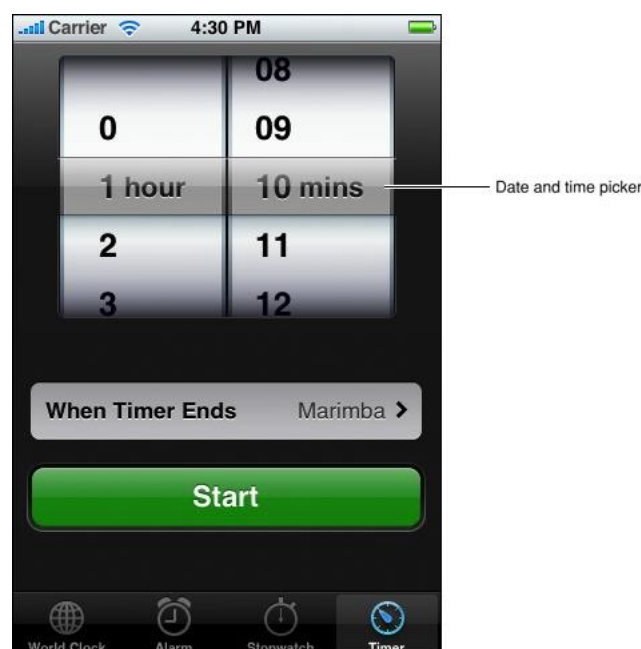
一个活动指示在任务或进程已完成时就会消失。这种默认行为是被推荐的，因为用户希望当有事情发生时看到一个活动指示，他们会把一个停止活动指示与一个已经停止的进程联系起来。

要了解如何显示网络活动指示，请参阅 [UIApplication 类型参考](#) 中的 [networkActivityIndicatorVisible](#) 方法。要了解如何在代码中显示较大的，非网络活动指示，见 [UIActivityIndicatorView 类型参考](#)

## 2.5.2 日期时间选择器 Date and Time Pickers

一个日期和时间选择器为用户选择特定的日期或时间提供了一种简单的方法。一个日期和时间选择器最多可以有 4 个独立的可旋转的轮子，每个轮子 在一个单独的类里显示数值，比如月份或小时。用户以弹或拖动的方式旋转每个轮子，直到它在横跨选择器并位于选择器中间的选择栏下方显示所需的值。最后的值 包括每个轮子中显示的值。图 9-2 显示了一个日期和时间选择器的例子。

图 9-2 日期和时间选择器



使用日期和时间选择器，使用户避免了输入由多个部分组成的值，如一个日期包含日、月和年。一个日期和时间选择器运作良好因为在每一部分的值有一个相对小的范围，并且用户已经知道这些值是什么。

根据你指定的方式，日期和时间选择器显示不同数量的轮子，每个含有一些列的值。日期和时间选择器定义了下列模式：

- **Time.时间。**时间模式包含显示小时和分钟数值的轮子，外加一个可选的轮子用于设定 AM/PM。
- **Date.日期。**日期模式包含显示月份、日和年数值的轮子。
- **Date and time.日期和时间。**日期和时间模式包含显示日历日期、小时和分钟值的轮子，外加一个可选的轮子用于设定 AM/PM。这个是默认模式。
- **Countdown timer.倒计时计时器。**倒计时计时器模式包含显示小时和分钟值的轮子。你可以指定一个倒计时的总时间，最长时间为 23 小时和 59 分钟。

默认情况下，分钟轮子包含 60 个值（0 到 59）。但是，如果你需要显示的选择较粗粒度，你可以设置一个分钟轮显示的时间间隔为分钟，只要间隔均匀划分成 60 份。例如，你可能希望显示四分之一小时的间隔 0，15，30 和 45。

无论它的配置如何，日期和时间选择器的整体大小是固定的，键盘大小也是一样的。你可以选择在视图中把你的日期和时间选择器作为焦点元素，或者让它在需要时再显示。例如，在内置的时钟定时器模式的应用程序总是显示一个可见的日期和时间选择器，因为时间的选择是定时器至关重要的功能。另一方面，日期和时间偏好设置（在设置>综合>日期&时间中，当你关闭自动设置的情况下）显示临时的日期和时间选择器，这取决于用户是否需要设置日期或时间。

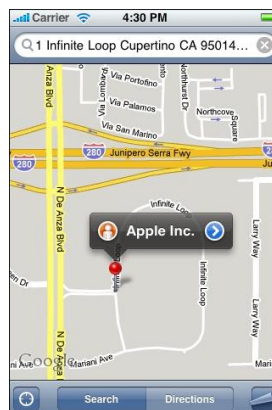
要了解有关在代码中如何使用日期和时间选择器的更多信息，请见 [UIDatePicker 类型参考](#)

### 2.5.3 详情显示按钮 Detail Disclosure Buttons

**详情显示按钮**揭示了额外的或更多的关于某事物的细节信息。通常用于表格视图中，他们为用户提供了解一个列表项的详细资料的途径（有关此用法的更多信息，请参见“表格视图元素”（第 92 页））。但是，你也可以在其它视图中使用此元素，用以提供揭示更多的信息和功能的一种途径。

例如，地图应用程序中显示了一个详情显示按钮，用户可以通过点击它访问更多的关于钉子所钉定位置的信息。图 9-3 显示了一个详情显示按钮的例子。

图 9-3 详情显示按钮，显示更多的细节和功能





要了解更多有关在代码中应用详情显示按钮的信息, 请参阅 [UIButton 类型参考](#)。

## 2.5.4 信息按钮 Info Buttons

一个信息按钮提供了一种用于揭示一个应用程序配置的详细信息的方法, often on the back of the screen。为此, 信息按钮特别适用于 utility applications。在天气应用程序的右下角你可以看到一个信息按钮的例子 (如图 9-4 所示)。

图 9-4 信息按钮的计时的信息, 往往是配置细节



信息按钮可与浅色背景和深色背景搭配。浅色背景风格 (见图 9-4) 用在深色背景的应用程序上较好。相反, 一个深色背景信息按钮更适合应用在浅色背景的应用程序上。

一个简单信息按钮在用户点击它时会发光。当你使用 iPhone 操作系统提供的信息按钮时, 此按下状态就会自动出现。

要了解有关在代码中使用信息按钮的更多信息, 请参阅 [UIButton 类型参考](#)。

## 2.5.5 标签 Labels

标签是指不同大小的静态文本。图 9-5 显示了一个标签的例子。

**图 9-5** 标签为用户提供信息

你可以使用标签来命名用户界面的某些部件或为用户提供有限的帮助。标签是最适合显示相对较少的文本。

你可以决定标签文本的各种属性，如字体，文本颜色和对齐方式，但首先，你应该使你的标签清晰。不要为了奇特的字体或艳丽的颜色而牺牲文字的清晰度。

当你撰写你的标签文本时，一定要使用用户的词汇。检查你那些以开发为中心的应用中的文本，并且用以用户为中心的应用中的文本取代他们。

要了解有关在代码中使用标签的更多信息，请参阅 [UILabel 类型参考](#)。

## 2.5.6 页面指示器 Page Indicators

页面指示器会显示为一个小点在一个应用中，从左到右，小点代表了被打开视图的顺序（最左边的小点代表第一个视图），用一个白色的亮点代表当前视图，用户点击亮点的左边或者右边，可以查看前一个或者后一个视图。图 9-6 展示了一个页面指示器的例子。

**图 9-6** 一个页面指示器



页面指示器可以使用户清楚的了解当前已打开视图的数量及其排列的顺序,但它不会记录视图的层次结构。因为 utility application 的视图通常对等出现,页面指示器便足以帮助用户浏览它们。另一方面,一个显示层级信息的 productivity application 应该通过导航栏中的元素为用户提供导航(可参考[“Navigation Bars”](#))。

通常情况下,页面指示器应该放在应用屏幕的底端,界面视图的下面。这样可以让用户很容易的看到屏幕上较重要的信息。页面指示应垂直居于视图底脚空白和屏幕底角空白之间。

虽然对页面指示器中指示点的数量没有明确的限制,但是当界面越来越多的时候,注意指示点之间不能过于密集。例如,在直式定向中,未剪辑前最多指示 20 个页面。因此,你的应用程序应该有一定的逻辑以免发生类似情况。

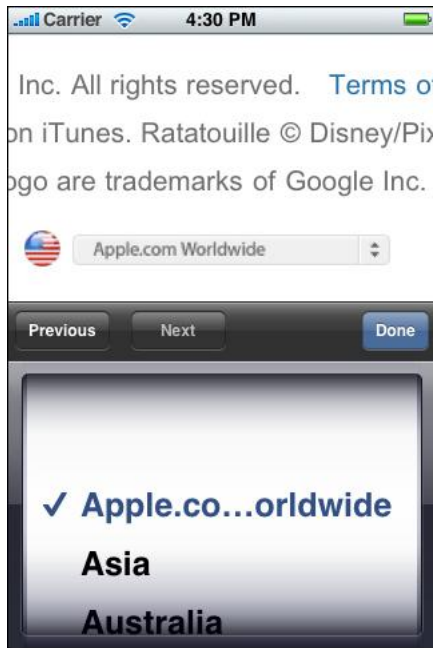
尽管在当前只有一页的时候,页面指示器可以隐藏,但默认状态是显示它。

了解更多关于在代码中使用页面指示器的内容,请参考 [UIPageControl Class Reference](#)。

### 2.5.7 选取器 Pickers

选择器是一个日期和时间选择器的一般版本(有关此控件的更多信息,见[“日期时间选择器”](#))。你可以使用一个选择器来显示任何一组数值。用日期和时间选择器,用户通过旋转选择器的轮子知道自己期望的数值出现。图 9-7 显示了一个单轮选择器。

图 9-7 在 iPhone 上,显示在 Safari 浏览器中的选择器



当一个轮子处于静止状态，它的大多数数值在用户默认状态下就是被隐藏的时候，你就应该考虑在你的应用中使用一个选择器了。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，在日期和时间选择器中，用户知道，隐藏在月份轮子下的数值只能是介于 1 和 12 之间的这些数字。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，这种情况下，选择器可能就未必是适当的控件。

如果你需要显示的值非常大，你可能更适合用一个表格视图，而不是选择器。这是因为一个更长的表格视图使滚动速度更快。

如果你需要在一个选择器里为一个值提供说明，如一个计量单位，那就显示在半透明的横跨控件中心的选项栏中。不要在选择器正上方或轮子里显示这种标签。关于正确显示标签的例子，参见内置时钟应用程序中的倒计时功能，在用户选择的值的旁边显示“时”和“分”。

同日期和时间选择器一样，一个通用选择器可以一直显示（作为用户界面的一个焦点），也可以只在需要时显示。一个选择器的总体大小，包括其背景，是固定的，与一个键盘的大小相同。

要了解有关在代码中使用选择器的更多信息，请参见 [UIPickerView 类型参考](#)。

### 2.5.8 进度视图 Progress Views

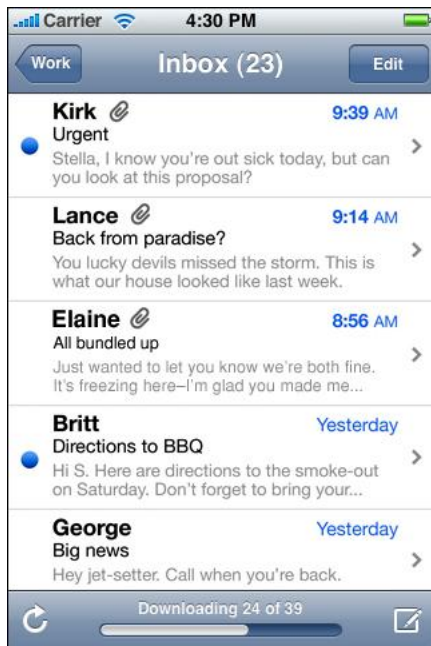
一个进度视图显示了一个任务的进展，或者有一个已知持续时间的过程的进度。如果你需要显示一个未知时间的任务进度，请使用活动指示器（有关此控件的详细信息见“活动指示器”（第 99 页））。

iPhone 操作系统提供了两种风格的进度视图，这是默认样式和栏样式。除了高度外，每个风格的外观是非常相似的：

- 默认风格是用于应用程序的主要内容区。
- 栏风格比默认样式瘦小，因此栏风格更适合于在工具栏中使用。例如，在邮件中，当用户下载新的信息或发送电子邮件时，栏风格的进度视图就会出现在工具栏中。

两种风格的进度视图在行为上是相同的。由于任务或进程的进行，进度视图的进度条从左到右逐渐填满。在任何特定时间，填充部分与未填充部分面积的比例给用户提供了一个要多长时间，任务或进程将结束的提示。图 9-8 显示了一个栏样式的进度视图。

图 9-8 工具栏中的一个栏样式进度视图



对于那些有明确持续时间的任务，进度视图是给用户反馈的一种很好的办法，特别是当很有必要告知用户任务所需要的时间时。当你采用一个进度视图时，你告诉了用户任务正在执行，你给他们足够的信息来让他们决定是否要等到任务完成或取消任务。

要了解更多有关在你的代码中应用进度视图的信息，请参阅 [UIProgressView 类型参考](#)。

## 2.5.9 圆角矩形按钮 Rounded Rectangle Buttons

**圆角矩形按钮**是一个多功能的按钮，你可以在一个视图中用它来执行的一个操作。在个人的联系人视图的底部，你可以看到很多这种按钮的例子：文字信息按钮，添加到收藏夹按钮都是圆角矩形按钮，如图 9-9 所示。

图 9-9 圆角矩形按钮，执行应用程序的具体行为



当你给一个圆角矩形按钮提供标题时，一定要：

- 使用标题样式首字母大写规则（即，大写每个单词的首字母，除了冠词，并列连词和四个或更少字母的介词）
- 避免创建一个太长的标题。超长文本被截断后，用户会很难理解它。

要了解有关在代码中使用圆角矩形按钮的信息，请参阅 [UIButton 类型参考](#)。

### 2.5.10 搜索栏 Search Bars

搜索栏是一个接受用户输入文本的域，你的应用程序可以使用这些文本来搜索。当用户点击一个搜索栏，键盘显示，当用户完成搜索条件的输入，输入以特定应用程序的方式处理。（关于在你的程序中应用搜索的指南，请参阅[“提供搜索和显示搜索结果”](#)）。

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元素：

- Placeholder text. 占位符文本。这些文字可能是强调控件的功能（例如，“搜索”），或提醒他们是在什么引擎中搜索（例如，“YouTube”或“Google”）。
- The Bookmarks button. 书签按钮。这个按钮可以提供一个快捷的信息便于用户再次找到它。例如，在地图搜索模式书签按钮提供了进入书签所定位的位置的入口，最近的搜索和联系人信息。
- The Clear button. 清除按钮。大多数搜索栏都包括一个清除按钮，允许用户通过点击清除按钮删除搜索栏里的内容。



- A descriptive title, called a prompt, that appears above the search bar. For example,

描述性标题，或称为提示，出现在搜索栏的上方。例如，提示可以是一个短语，为搜索栏提供前言的或者应用程序具体的介绍。

图 9-10 显示了一个搜索栏，其中包括自定义的占位符文本，书签按钮，默认搜索图标。

**图 9-10** 带有可选的占位符文本和书签按钮的搜索栏



默认情况下，书签和清除按钮通过以下方式交互：

- 当搜索栏包含任何非占位符文本时，清除按钮就会显示，用户可以删除文本。如果搜索栏里没有用户提供的文本或非占位符文本，清除按钮会被隐藏，因为没有必要抹去搜索栏的内容。
- 当搜索栏里没有用户提供的文本或没有非占位在文本时，书签按钮显示。这是因为在用户可能需要清除搜索栏里的文本时，清除按钮才显示。

你可以通过指定一个标准色背景样式来自定义搜索栏，比如：

- 蓝色（默认的渐变色，与工具栏和导航栏的外观相协调）。默认背景样式如图 9-10。
- 黑色

此外，你可以在搜索栏下面显示范围栏，其中包含那些用户点击之后可以选择一个搜索范围的按钮。范围栏采用了和你指定的搜索栏相同的外观，并为你提供自定义的范围按钮标题。

范围栏显示在搜索栏的下方，不分方向，除非你在代码中使用搜索显示控制器（更多信息，见 [UISearchDisplayController 类型参考](#)）。当你使用搜索显示控制器时，如果设备是横向显示的，范围栏处于搜索栏中，且在搜索输入区域的右边。

要了解更多有关在代码中使用搜索栏和范围栏的信息，请参阅 [UISearchBar 类型参考](#)。

### 2.5.11 分段控件 Segmented Controls

分段控件是一套线性 segment，每个 segment 起到按钮的作用，以显示不同的视图。当用户点击分段控件中的一个 segment 时，一个瞬时行动或有形的结果应该发生。例如，当用户使用分段控件选择一个电子邮件协议时，设置显示不同的信息，如图 9-11。

图 9-11 包含三个 segment 的分段控件



一个分段控件的长度取决于你显示的 segment 的数量，以及最大的那个 segment。一个分段控件的高度是固定的。你可以指定显示的段数，但是要注意，用户必须能够很舒适地点击而不用担心错误的点击了周边的 segment。由于被点击的区域应为 44 × 44 像素，建议分段控件所包含的 segments 不超过 5 个。

分段控件可以包含文字或图像，个体的 segment 也可以包含文字或图片，但不能同时包含两者。一般来说，最好避免在一个分段控件混合使用文本和图像。

分段控件能确保每个 segment 的宽度成比例，取决于总的 segments 数量。这意味着你需要确保你的内容和每一个 segment 的设计大小基本一致。

要了解更多有关在代码中使用分段控件的信息，请参阅 [UISegmentedControl 类型参考](#)。

## 2.5.12 滑块 Sliders

**滑块**允许用户在整个允许值的范围内调整一个数值或进程。当用户拖动一个滑块，数值或进程一会随之不断更新。图 9-12 显示了一个最大和最小的图像滑块的例子。

图 9-12 滑块

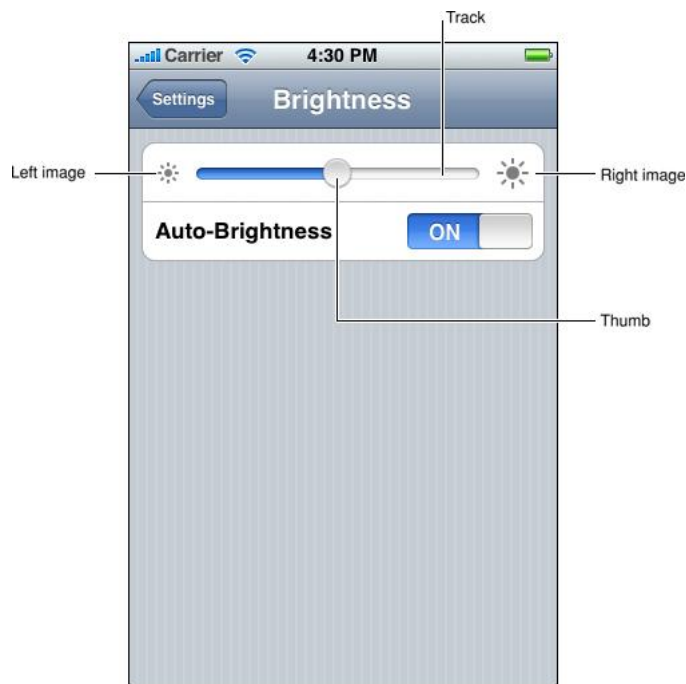


滑块主要在下面这两种情况使用：

- 你想让用户对他们选择的值进行精确的控制
- 你想让用户对当前过程进行精确的控制

滑块由一个轨道，thumb 和可选的左，右数值图像组成。图 9-13 显示了一个滑块这些部分。

图 9-13 滑块的四部分



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，你可以设定水平或垂直显示滑块。

There are several ways to customize a slider:

有几种方法定制一个滑块：

- 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 thumb 的外观，让使用者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提供图片（通常，这些对应的最小和最大值），以帮助用户了解滑块的作用。

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，可以在最小值的一端显示一个非常小的字体，在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.

你可以定义一个不同外观的轨道，这取决于它在 thumb 的那一端以及控件所处的状态。

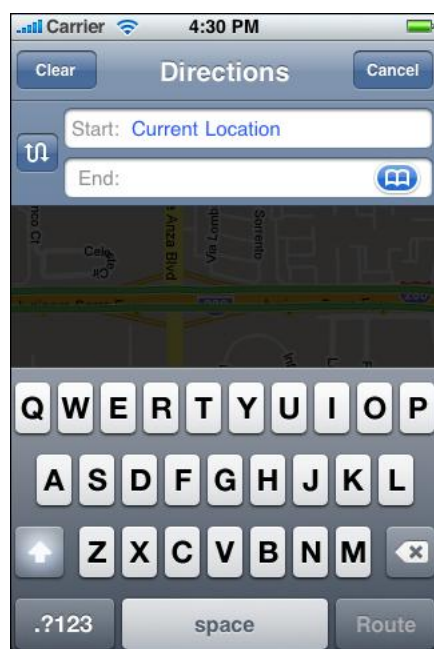
To learn more about using a slider in your code, see [UISlider Class Reference](#).

要了解更多有关在代码中使用滑块的信息，请参阅 [UISlider 类型参考](#)。

### 2.5.13 文本框 Text Fields

**文本框**是个圆角矩形域，用来接收用户的输入。当用户轻敲文本框的时候，键盘就会出现；当用户轻敲键盘上的返回键时，文本框用应用程序指定的方式来处理输入。文本框可以包含一行输入。图 9-14 显示了地图应用程序的两个文本框。

图 9-14 文本框可以接受用户输入



你可以自定义文本框，以帮助用户了解他们应该如何在使用程序中使用它。例如，你可以在文本框的左边或右边显示自定义图像，或一个系统提供的按钮，如在图 9-14 所示的书签按钮。一般而言，你应该使用文本框的左端显示其用途，右端显示存在的附加功能，比如书签。

你也可以让清除按钮出现在文本框的右端。当此清除按钮出现，点击它以清除文本框里的内容，不用考虑其他任何你可能会显示在这里的图像。

有时候，为了帮助用户了解一个文本框的目的，而显示提示，如“名称”。在文本框里没有文字时，文本框里会显示一个文本占位符。要了解更多关于使用文本框和个性化定制文本框来显示图像和按钮的信息，请参阅 [UITextField 类型参考](#)。

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内容。(注意你不能控制键盘的输入方式和布局,这是由用户的语言设置决定的。)举个例子,你可能想更方便用户输入 URL,密码或者电话号码。iPhone OS 提供多种不同的键盘样式,每种都设计成利于一种不同的输入。要了解可供使用的键盘类型,详见 [UIKeyboardType](#)。要了解更多如何在你的应用程序里管理键盘的信息,详见 [iPhone 应用程序编程向导](#) 中的 [“管理键盘”](#)。

## 2.6 系统自带的按钮与图标

### System-Provided Buttons and Icons

为了提高用户体验的一致性(并让你的工作轻松些),iPhone OS 自身提供了大量可用于导航栏和工具栏的标准按钮,以及用于标签栏的图标。

本章描述了那些可用的标准图标和按钮,以及关于如何恰当使用的准则。无论你开发的是何种应用程序,你都应当去熟悉对本章中的按钮和图标,以便:

- 正确使用系统自带的按钮与图标。
- 避免设计与系统提供的图标过于相似的自定义图标。
- [使用系统自带的按钮与图标 Using System-Provided Buttons and Icons](#)
- [用于工具栏与导航栏的标准按钮 Standard Buttons for Use in Toolbars and Navigation Bars](#)
- [用于标签栏的标准图标 Standard Icons for Use in Tab Bars](#)
- [用于表格行及其他用户界面元素中的标准按钮 Standard Buttons for Use in Table Rows and Other User Interface Elements](#)

#### 2.6.1 使用系统自带的按钮与图标 Using System-Provided Buttons and Icons

iPhone OS 在其内置的应用程序中贯穿使用了大量的标准工具栏和导航栏按钮、标签栏项及常规按钮等。你可以在 Mail 的工具栏上看到不少标准的工具栏按钮,如图 10-1 所示。

图 10-1 Mail 工具栏中的标准按钮





图 10-1 中如刷新、组织、删除、回复和撰写等按钮在许多内置应用程序中都被一致地使用，因此用户很熟悉其含义和用法。这意味着如果你的应用程序也支持这些功能的话，你便可以利用用户已经熟知这一点来优化应用程序的用户界面。相反，如果你将这些按钮用于其他的任务，你则可能会使用户感到糊涂甚至愤怒，因为你暗示的是某种他们期望得到的功能，而事实却相反。

除了可以充分利用用户先前经验的好处之外，使用系统自带的按钮和图标还将产生另外两个重要的影响，分别是：

- 减少开发时间，因为你无须为了展现标准功能而自行创建自定义的图形。
- 增加用户界面的稳定性，哪怕未来 iPhone OS 的更新版本更改了这些标准图标的外观。换句话说，你可以相信即使标准图标的外观变了，其语义仍将保持不变。

再次强调，为了能够真正得益于用户的熟悉性、较短的开发时间以及用户界面的语义一致性等优点，你必须恰当地使用这些按钮和图标。具体来说，你使用按钮或图标时应当依据其文档说明中的含义和推荐位置，而不要根据你自己对其外观的解释。关于这些系统自带按钮和图标的含义及位置信息，参见[“用于工具栏与导航栏的标准按钮”](#)、[“用于标签栏的标准图标”](#)和[“用于表格行及其他用户界面元素中的标准按钮”](#)。

Interface Builder 使得使用系统自带的按钮以及将系统自带的图标应用到你的控件上变得非常容易。相关指南，参见 [Interface Builder User Guide](#) 中的 [“iPhone OS Interface Objects”](#) 查阅与外观相关的信息。

如果你找不到与你应用程序中的特定功能含义相当的系统自带工具栏或导航栏按钮或者标签栏图标，你则应当设计自定义按钮或图标。[“用于导航栏、工具栏及标签栏的图标”](#) 提供了与此相关的设计准则。

## 2.6.2 用于工具栏与导航栏的标准按钮





iPhone OS 为你准备了用户能够在工具栏和导航栏上看到的大量标准按钮。表 10-1 所示的这些按钮具有两种样式，分别用于下列不同的情况：

- 带边框样式——例如，在电话联系人导航栏上的“添加”按钮。该样式同时适用于导航栏和工具栏。
- 平面样式——例如，Mail 工具栏上的“撰写”按钮。该样式仅适用于工具栏。事实上，如果你为导航栏上的按钮指定平面样式，它会转换成带边框样式。

对于所有系统自带的按钮，你应当避免将表 10-1 中的按钮用于表示其设计用途之外的操作。尤其要根据其文档中描述的含义来选择按钮，而非其外观。关于正确使用这些图标的原因的讨论，参见“[使用系统自带的按钮与图标](#)”。（关于这些按钮符号名称和可用状态可参阅 [UIBarButtonItem](#) 的文档。）





表 10-1 可用于工具栏和导航栏的标准按钮（以平面样式展示）

按钮	含义	名称
	Opens an action sheet that allows users to take an application-specific action 打开操作列表，使用户能够执行特定应用程序相关的操作	Action
	Opens an action sheet that displays a photo picker in camera mode 打开操作列表，以显示拍照模式下的照片选择器	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 显示与特定应用程序相关的书签	Bookmarks
	Display a search field 显示搜索框	Search
	Create a new item 创建新条目	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 将某项移动或传送至应用程序的目标位置，如某文件夹	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 停止当前操作或任务	Stop
	Refresh contents (use only when necessary; otherwise, refresh automatically) 刷新内容（仅当必要时使用，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 (这也暗示着要保留当前上下文位置)	Pause
	Move backwards through media playback or slides 在播放媒体或幻灯片快退	Rewind

除了表 10-1 中列出的按钮之外,你还可以使用系统自带的编辑、取消、保存和完成按钮以便在你的应用程序中支持编辑或其他类型的内容处理,如表 10-2 所示。(关于这些按钮符号名称和可用状态可参阅 [UIBarButtonItem](#) 的文档。)这些按钮在导航栏和工具栏上都可使用,但仅具有带边框样式。如果你为这些按钮指定平面样式,它会转换成带边框样式。

表 10-2 用于导航栏的带边框操作按钮

按钮	含义	名称
	Enter an editing or content-manipulation mode 进入编辑或内容处理模式	Edit
	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 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
	Exit the current mode and save changes, if any 退出当前模式,并保存所做的更改	Done

2.6.3 用于标签栏的标准图标 Standard Icons for Use in Tab Bars

iPhone OS 提供了用于标签栏的标准图标,如表 10-3 中所示。关于这些按钮符号名称和可用状态可参阅 [UITabBarSystemItem](#) 的文档。

对于所有标准按钮和图标,你必须依照其文档中所述的含义来使用。尤其要根据其文档中描述的含义来选择图标,而非其外观。这样一来,即使相关的图标的外观改变了,你应用程序的用户界面也依然合理。关于正确使用这些图标的原因的讨论,参见 [“使用系统自带的按钮与图标”](#)。

表 10-3 用于标签栏标签的标准图标

图标	含义	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 显示用户操作历史	History
	Show additional tab bar items 显示更多标签栏项	More
	Show the most recent item 显示最近一次使用的项	MostRecent
	Show items most popular with all users 显示所有用户最关注的项	MostViewed
	Show the items accessed by the user within an application-defined period 显示在应用程序定义的一段时间内用户访问过的项	Recents
	Enter a search mode 进入搜索模式	Search
	Show the highest-rated items, as determined by the user 显示用户评价最高的项	TopRated

## 2.6.4 用于表格行及其他用户界面元素中的标准按钮

iPhone OS 提供了一些用于表格行和其他元素中的按钮。与其他标准按钮和图标一样,这些按钮(表 10-4)应当根据其语义正确使用,尤其要避免仅仅根据其外观来选择按钮,而无视其文档中所描述的含义。关于正确使用这些图标的原因的讨论,参见[“使用系统自带的按钮与图标”](#)。

虽然详情显示按钮通常用在表格行中,但它也可以在其他任何地方使用。关于该按钮的更多信息,参见[“详情显示按钮”](#)。iPhone OS 也提供一些仅用于表格行的控件,有关其更多信息,参见[“表格视图元素”](#)。

关于这些按钮符号名称和可用状态可参阅 [UIButtonType](#) 的文档。(关于这些按钮符号名称和可用状态可参阅 [UITableViewCellAccessoryDetailDisclosureButton](#) 的文档。)

表 10-4 用于表格行及其他用户界面元素中的标准按钮

按钮	含义	名称
	Display a people picker to add a contact to an item 显示联系人选择器以将联系人加至某项	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（通常是在实用程序中）以显示配置选项或更多信息。	Info
	Note that the Info button is also available as a light-colored “i” in a dark circle. 注：也有深色圆圈浅色字母 “i” 的信息按钮。	

2.7 创建自定义图标和图像

Creating Custom Icons and Images

精美图像和丰富的颜色是 iPhone 应用程序的用户界面的特征。作为应用程序设计师，你应当提供赏心悦目的用户界面以融入这个环境。虽然 iPhone OS 提供了大量精美而引人注目的用户界面元素，但每个应用程序仍然需要两个自定义元素：应用程序图标和启动图像。另外，应用程序应当为 iPhone OS 提供用于显示在 Spotlight 搜索结果中的小图标。

有些应用程序需要自定义图标以在导航栏、工具栏和标签栏中表示特定的功能和模式。只要遵照创建这些图标的准则，你即可既与内置图标保持和谐，又能够适当强调你应用程序的风格。

**注：**图标和图像的标准位深是 24 位（红、绿、蓝各 8 位），加上 8 位的 Alpha 通道。推荐使用 PNG 格式，因为它可以保持色彩深度并内置支持 Alpha 通道。你可以使用预览程序来处理最终的 PNG 文件。

你无须将颜色限定在 Web 安全色内。虽然你可以为导航栏、工具栏和标签栏上的图标使用 Alpha 透明，但不要在应用程序图标中这么做。

2.7.1 应用程序图标 Application Icons

**应用程序图标**是用户会放在其主屏幕的图标，点击可启动应用程序。在这里，品牌宣传和大量的视觉设计将浓缩于一个精简、可立即识别且引人注目的整体。

用户选择希望显示在其主屏幕上的应用程序图标，因此你应当将图标设计得：

- 具有吸引力，这样用户觉得必须要将其留在主屏幕上
- 具有独特性，这样用户可以在其他图标中轻松将其找到

尽量在吸引眼球和含义清晰之间找到平衡，使你的图标在丰富多彩的同时能够表达应用程序的目的。而且，研究其他不同文化背景下的人们对你选择的图像和颜色的理解也是不错的办法。

当用户决定在主屏幕上显示你的应用程序图标时，iPhone OS 会自动添加一些视觉效果以与内置的图标保持协调。具体包括：

- 圆角 Rounded corners
- 投影 Drop shadow
- 光照反射 Reflective shine

例如，图 11-1 展示了一个应用程序提供的简单图标。

**图 11-1** 没有显示在主屏幕上的简单的应用程序图标



图 11-2 展示了同样的图标由 iPhone OS 显示在主屏幕上的样子

**图 11-2** 显示在主屏幕上的简单的应用程序图标



包含清晰背景的应用程序图标在主屏幕上看上去效果最佳。这主要是因为 iPhone OS 所添加的圆角：统一的圆角确保用户主屏幕上的所有图标都具有统一的外观。如果你创建了背景为黑色的图标，那么它的主屏幕上显示时就会消失，用户不会看到圆角。这种图标起来像不可点击的，尤其会打乱用户本主屏幕的次序结构

为了确保你的图标能够得益于 iPhone OS 提供的视觉增强，应当将图像处理成 PNG 格式并满足下列条件：



- 尺寸为 57 x 57 像素, 90 度直角 (如果图像尺寸不同, iPhone OS 将进行缩放)
- 未包含光泽或玻璃效果
- 未使用 Alpha 透明

将图标文件命名为 Icon.png 并放在你应用程序包的最顶层。要进一步了解关于应用程序包的内容, 参见 [iPhone Application Programming Guide](#) 中的 [“The Application Bundle”](#)。

**注:** 你也可以选择让 iPhone OS 不要给你的图标添加光泽效果。要实现这一点, 你需要在你应用程序的 Info.plist 文件 (要了解该文件, 参见 [iPhone Application Programming Guide](#) 中的 [“The Information Property List”](#)) 中添加 UIPrerenderedIcon 键。

你的图标尺寸应当仍然是 57 x 57 像素, 无论你是否打算使用光泽效果。

当提交应用程序时, 你必须包含你应用程序图标的 512 x 512 像素版本, 以用于显示在 App Store 中。尽管这个版本能够被立即识别为你的应用程序图标的确非常重要, 但它仍应当相比而言更加丰富细致。换句话说, 你不应当只是简单地将应用程序图标放大来创建 App Store 图标。

如果你开发应用程序是为了私下分发 (而不是通过 App Store) 的话, 你也必须为应用程序图标提供一个 512 x 512 像素的版本。在这种情况下, 将图标文件命名为 iTunesArtwork (没有文件扩展名) 并将其放在应用程序 bundler 的顶端。

## 2.7.2 小图标 Small Icons

每个应用程序都应当提供一个小图标, 这样 iPhone OS 才能在当应用程序名称符合 Spotlight 搜索条件时进行显示。

提供设置功能的应用程序也应当提供此图标, 以便在内置的设置应用程序中进行自我标识。

你的小图标应当能够清晰标识你的应用程序, 使用户能够在搜索结果列表中明确识别出来。要实现这一点, 应当创建一个简洁、吸引人的图标, 并符合下列条件:

- 使用 PNG 格式。 Uses the PNG format.
- 尺寸约为 29 x 29 像素。 Measures about 29 x 29 pixels.

将图标文件命名为 Icon-Small.png 并放在你应用程序包的最顶层。要进一步了解关于应用程序包的内容, 参见 [iPhone Application Programming Guide](#) 中的 [“The Application Bundle”](#)。

**注:** 如果你没有提供名为 Icon-Small.png 的图标, 且你的应用程序包中也不包含早期版本的名为 Icon-Settings.png 的小图标的话, iPhone OS 将会将你的应用程序图标缩小显示在搜索结果和设置中。

如果你的应用程序包中已经包含了名为 Icon-Settings.png 的小图标,但没有包含名为 Icon-Small.png 的图标的话,iPhone OS 会在搜索结果中显示设置图标。不过,你应当更新你的应用程序包,使其仅包含 Icon-Small.png 文件。

### 2.7.3 启动图像 Launch Images

为了提高应用程序启动时的用户体验,你应当提供一幅启动图像。**启动图像**通常与你应用程序要显示的第一屏非常相似。iPhone OS 会在用户在主屏幕上点击你的应用程序图标时立即显示该图像。一旦程序可以使用,你的应用程序将显示其第一屏,替换启动占位图像。

有必要强调一下提供启动图像的目的在于提高用户体验。而**不是**用于提供:

- “应用程序进入体验,如 Splash Screen
- 关于窗口
- 品牌宣传元素,除非它们本身就是你应用程序第一屏上的静态部分。

由于用户往往会在应用程序之间频繁快速地切换,你应当竭尽全力降低启动时间,而且你设计的启动图像应当能够减轻等待的体验,而不是吸引注意。

要实现这一点,你应当以 PNG 格式设计满足下列条件的图像:

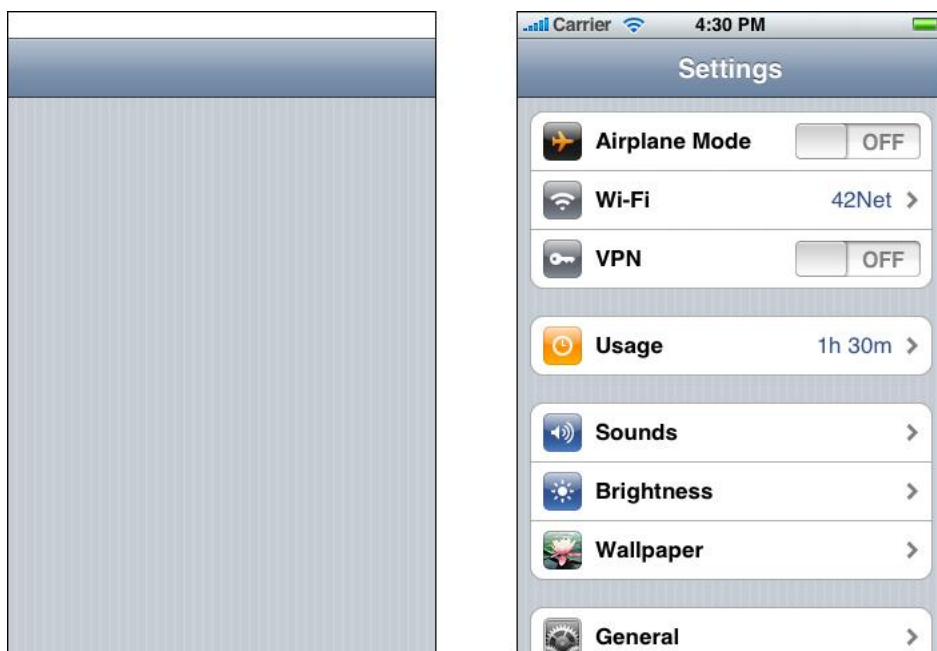
- 尺寸为 320 x 480 像素。将状态栏包括在内可使得立即显示你选择的状态栏颜色,而非等应用程序启动完成后才显示。
- 与应用程序的第一屏相同,除了:
  - 文本。启动图像是静态的,所以你要显示的任何文本都不会被文本化。
  - 可能变化的用户界面元素。避免包含那些在应用程序启动完成后可能看上去不同的元素,这样用户不会在启动图像和应用程序第一屏之间感到跳跃。

将启动图像文件命名为 Default.png 并放在你应用程序包的最顶层。要进一步了解关于应用程序包的内容,参见 [iPhone Application Programming Guide](#) 中的 [“The Application Bundle”](#)。

如果你觉得遵照这些准则做出来的启动图像非常平淡无趣,那就对了。记住,启动图像并不是用作艺术展示的,而只是纯粹用于让用户觉得你的应用程序启动迅速立即就绪。以下是一些启动图像的例子,看看能够无趣到什么程度。

第一个例子是内置的设置应用程序的启动图像,如图 11-3 所示。该设置启动图像仅包含了应用程序的背景,因为应用程序中的其他内容都不是静态的。

**图 11-3** 设置应用程序的启动图像



另一个启动图像的例子来自于内置的股票应用程序，如图 11-4 所示。注意启动图像中包含的唯一一幅图像是静态图像，

图 11-4 股票应用程序的启动图像



## 2.7.4 用于导航栏、工具栏和标签栏的图标

尽可能在导航栏、工具栏和标签栏中使用系统自带的按钮和图标。iPhone OS 提供了大量已在内置应用程序中表示用户熟知的标准任务和模式的标准按钮和图标。如果你的应用程序支持这些标准功能,比如刷新内容视图或删除项,或者是显示数据的具体部分,比如联系人或书签,则应当使用恰当的系统内置按钮或图标来表示。关于完整的标签按钮和图标列表及其使用准则,参见[“系统自带的按钮与图标”](#)。

当然,不可能你应用程序的每个功能都是标准的。如果你的应用程序提供了用户需要经常执行的特殊任务的话,你就需要创建一个自定义图标以便在工具栏或导航栏中代表那些任务。类似地,如果你的应用程序标签栏上提供了自定义的应用程序模式或者自定义的数据集,你也需要设计标签栏按钮以清晰描述这些模式或数据。本节提供了关于如何为导航栏、工具栏和标签栏设计合适的图标的一些准则。

在进行任何图标设计之前,你需要花点时间来思考它要表达的内容。既然是考虑设计,那么图标应当是:

- 简明扼要。太多细节会使图标看上去杂乱或者难以理解。
- 不易误认为是某个系统内置的图标。用户应当能够立即区别出你的自定义图标和标准图标。
- 能够迅速理解并得到广泛的认可。力求所创建的符号能够被大部分用户正确理解,且不会使用户感到厌恶。

当你决定了图标的外观后,应在制作时遵循下列准则:

- 使用 PNG 格式。
- 使用具有适当透明的纯白色。
- 不要使用投影。
- 使用消除边缘锯齿。
- 如果你希望添加斜边 (bevel) 效果,应当确保它是 90° 的 (想像光是从图标的正上方投射下来的)。
- 对于工具栏和导航栏图标,其尺寸应约为 20 x 20 像素。

对于标签栏图标,其尺寸应约为 30 x 30 像素。

**注:** 你为工具栏、导航栏和标签栏提供的图标将用于你最后在应用程序中看到的图标的蒙板 (mask)。没有必要创建全彩色的图标。

iPhone OS 自动为导航栏、工具栏和标签栏上的项提供按下和选中外观,因此你无须再单独提供。因为这些视觉效果是自动生成的,你无法更改其外观。