

iPhone 人機界面指南

介紹	7
本文件的組織結構	7
參考	8
設計您的iPhone應用程式	9
iPhone OS平台：豐富的可能性	10
牢記平台間的差異	10
緊湊的螢幕尺寸	10
記憶體是有限的	11
每次只能顯示一個視窗	11
每次只能運行一個程式	11
簡短的使用者幫助	12
您的選擇是甚麼？	12
iPhone應用程式	13
僅在網頁上顯示的內容	13
混合應用程式	14
三種應用程式風格	14
效率型應用程式	15
實用型應用程式	17
沈浸式應用程式	21
選擇一種應用程式風格	22
當您有個現成的電腦應用程式時	23
案例研究：將桌面應用程式移植到iPhone OS	24
郵件	24
iPhoto	26
人機介面準則：建立優秀的使用者介面	30

隱喻	30
直覺操作	30
即看即點	31
回饋	31
使用者控制	32
美學完整性	32
設計iPhone應用程式：從產品定義到品牌宣傳	33
給出產品定義說明	33
優秀iPhone應用程式的共性	35
簡單和易用性	35
顯而易見	36
由上而下的佈局	37
最大限度地減少所需的輸入	38
簡潔地表達資訊	38
為控制元件提供指尖大小的目標區域	38
將重點放在主要功能上	39
有效地溝通	41
適當的支援手勢	43
慎重地融合品牌元素	44
處理常見任務	46
啓動	46
停止	47
管理設置或配置選項	48
支援拷貝和貼上	49
支援還原和重做	50

啓用推送通知	51
保證應用程式的可用性	53
提供搜尋功能並顯示搜尋結果	54
使用使用者的位置資訊	55
處理方向的變化	56
使用聲音	57
震動/靜音切換—使用者的期望	57
音量按鈕—使用者的期望	58
耳機和耳麥—使用者的期望	59
無線音訊—使用者的期望	59
定義應用程式的音訊行為	60
摘要	62
提供選項	64
提供許可協議或免責聲明	64
設計您iPhone應用程式的使用者介面	66
使用者介面概述	67
應用程式的螢幕及其內容	67
導覽欄，標籤欄，工具欄和狀態欄	71
狀態欄	71
導覽欄	74
導覽欄的內容	75
導覽欄的尺寸和顏色	77
工具欄	78
工具欄的內容	79
工具欄的尺寸和顏色	80

標籤欄	81
提供附加標籤	83
為標籤欄上的標籤做標記	86
在應用程式螢幕中使用視圖和控制元件	87
用法和行為	89
使用警告	90
使用動作表單	91
使用模式視圖	92
設計警告	92
設計動作表單	95
設計模式視圖	99
表格視圖，文字視圖和Web視圖	103
表格視圖	103
用法和行為	103
表格視圖樣式	105
單元格樣式	107
表格視圖元素	115
開關控制元件	117
使用表格視圖實現常用的使用者操作	119
文字視圖	124
Web視圖	126
應用程式控制元件	128
活動指示器	128
日期時間選擇器	130
詳細資訊展開按鈕	132

資訊按鈕	133
標籤	135
頁面指示符號	137
選擇器	139
進度視圖	141
圓角矩形按鈕	143
搜尋欄	145
分段控制元件	147
滑塊	149
文字框	152
系統提供的按鈕和圖示	155
使用系統提供的按鈕和圖示	155
用於工具欄和導覽欄的標準按鈕	157
用於標籤欄的標準圖示	159
用於表格行和其它使用者介面元素的標準按鈕	161
創建自定義圖示和圖像	163
應用程式圖示	163
小圖示	165
啓動圖像	166
導覽欄，工具欄和標籤欄的圖示	169
文件修訂歷史	171

介紹

iPhone和iPod touch融合了革命性的多點觸摸技術和多種強大功能，如電子郵件，即時通訊，全功能網頁瀏覽器和iPod等，另外iPhone還俱有行動電話的功能。iPhone OS是在iPhone和iPod touch上運行的系統軟體。隨著iPhone SDK的發佈，這些強大的功能為開發人員提供了眾多的開發可能。開發者不但可以建立在iPhone OS裝置上使用的網頁內容，還可以使用iPhone SDK來建立在裝置中儲存和使用的本端程式。

本文件可以幫助您瞭解可以為iPhone OS建立何種類型的應用程式以及優秀iPhone程式基礎的人機介面設計準則。在本文件中，您可以瞭解到如何在設計最佳使用者介面和使用者體驗時遵循這些原則。不論您是經驗豐富的電腦程式開發員，或是一位熟練的行動裝置程式開發員，還是程式開發領域的新手，本文件中都能夠幫助您開發出使用者需要的iPhone程式。

注意：本文件簡要地介紹了如何為iPhone OS裝置進行基於網頁的開發。更多詳細資訊，請參考**iPhone Web應用程式人機介面指南**。

本文件的組織結構

iPhone人機介面指南分為兩部分，每一部分都包含幾個章節：

- 第一部分“設計您的iPhone應用程式”描述了iPhone OS環境和您可以為它開發的軟體類型。它也涵蓋了基本的人機介面設計原則，並描述了如何在您的iPhone應用程式中應用這些原則。
- 第二部分“設計您的iPhone應用程式的使用者介面”深入討論了那些用於建立iPhone應用程式的使用者介面的元素。它描述了各種可用的視圖和控制元件並提供了如何有效使用它們的說明。

參考

如果您需要瞭解編寫iPhone應用程式的更多資訊，請參考：

- iPhone應用程式程序指南

如果您需要瞭解為iPhone OS裝置設計Web應用程式的更多資訊，請參考：

- iPhone Web應用程式人機介面指南

設計您的iPhone應用程式

本部分描述了設計和開發iPhone OS應用程式的思想方法。通過閱讀本部分的章節，您可以瞭解iPhone OS上不同的程式類型以及您將在工作中用到的設計準則。您也可以瞭解如何在您的應用程式中應用這些準則，使您的程式俱有直覺而又引人注目的使用者介面。

iPhone OS平台：豐富的可能性

iPhone OS支援很多類型的軟體，從基於Safari的Web應用，到iPhone本端應用程式，都可以運行在iPhone OS裝置上。本章概要地介紹了為iPhone OS裝置建立不同類型軟體的解決方案。

如果您剛接觸該平台，您應先閱讀第一節中iPhone OS裝置和其他電腦的差異總結，“牢記平台間的差異”。雖然此節的資訊並不全面，但已經包含了您在設計iPhone應用程式時需要意識到的問題。

接下來，本章闡述了設計不同應用程式風格和特色時的思維方式，以幫助您設計iPhone應用程式。本章也闡述了如何將現有的Mac OS X應用程式轉換為iPhone OS版本的應用程式。

牢記平台間的差異

iPhone OS裝置不同於桌上型電腦或筆記型電腦，iPhone應用程式也與桌面應用程式不同。雖然這些似乎只是常識性陳述，但是您在為這些裝置開發軟體時意識到這些差異還是非常重要的。

為iPhone OS裝置開發軟體需要的思維方式可能是也可能不是您所習慣的。尤其是在您的經驗主要是開發桌面應用程式的情況下，您應該瞭解為行動平台設計軟體和為桌上型電腦設計軟體的差異。

本小節總結了這些具體的差異。這些差異對於您的設計選擇俱有潛在的影響。有關如何處理這些差異以及iPhone應用程式開發過程中的其他問題的具體資訊，請參考iPhone應用程式程序指南。

緊湊的螢幕尺寸

iPhone OS裝置俱有高解析度的小型螢幕，適合使用者放入口袋且俱有強大的顯示功能。然而這個對使用者來說龐大的優勢可能成為您的挑戰。因為這意味著您必須設計一個可能與您的設計習慣差異較大的使用者介面。

iPhone OS裝置螢幕的尺寸是480 x 320。考慮到這一點，使用者介面應該主要關注必不可少的部分。您沒有空間來放置那些非必要的設計元素。另外，擁擠的使用者介面將降低程式的吸引力並會提高使用的難度。

記憶體是有限的

記憶體是**iPhone OS**的重要資源，所以在應用程式中管理記憶體至關重要。因為**iPhone OS**的虛擬記憶體模型不包括磁碟交換空間，所以您必須注意避免分配的記憶體超出裝置的可用記憶體。發生記憶體過低的情況時，**iPhone OS**將警告運行中的應用程式，若問題仍然存在，應用程式可能被終止。確保您的應用程式對於記憶體使用警告作出回應，並及時清理記憶體。

在您設計應用程式時，可以通過避免記憶體洩露，減小資源文件大小以及延遲加載資源的方式，盡力地降低程式的記憶體佔用。關於如何在**iPhone**應用程式中合理處理記憶體的更多資訊，請參考**iPhone**應用程式程序指南。

每次只能顯示一個視窗

iPhone OS環境和電腦環境的最大區別之一就是視窗模式。除了一些模式視圖外，在**iPhone OS**裝置上，使用者每次在螢幕上只能看到應用程式的單個視窗。**iPhone**應用程式可以包含必要數量的不同視窗，但是使用者不能同時訪問，只能順序訪問和查看。

如果應用程式的桌上型電腦版本需要使用者同時看到幾個視窗，那麼您需要考慮是否存在其他的方式讓使用者能夠在單個視窗或一系列順序查看的視窗中完成同樣的任務。如果不存在這種方式，您的**iPhone**應用程式不應考慮拷貝電腦版本程式的完整功能，而應改為實現電腦版本的單個子任務。

每次只能運行一個程式

同一時刻只有一個iPhone應用程式可以運行，第三方應用程式不能在後台運行。這意味著當使用者切換到另一個應用程式，接電話或是查看郵件時，他們當前使用的應用程式將被關閉。重要的是，您應確保這一特色沒有對使用者產生負面影響。也就是說，在使用者離開您的iPhone應用程式並在稍後返回時，使用者應該不會感到比在電腦上的程式切換更加困難。

保證使用者享有良好的程式切換體驗的最有效方法就是將您的應用程式的啓動時間降到最低。關於如何盡可能地減少啓動時間的更多資訊，請參考iPhone應用程式指南。

一般說來，使用者通過切換至裝置上其他應用程式或服務的方式退出您的應用程式時，他們不會特地關閉您的應用程式。因此，不要期望使用者顯式地從選單中選擇退出或是點選關閉按鈕。這意味著您的程式可能在沒有太多通知的情況下退出，由此您應盡可能迅速地儲存使用者所做的修改。這樣就能在應用程式切換時形成快速而平穩的過渡，並且保證下一次您的應用程式啓動時能夠顯示使用者最近做的修改。

簡單程式模型的另一個重要方面是您處理程式特定配置的方式。在iPhone OS裝置上，使用者在設置程式中設定配置。您的iPhone應用程式可以提供這種配置，但是這意味著當使用者想在設置中訪問這些配置時，他們必須退出您的程式。如果您根據標準指南提供使用者僅需設定一次的設置，那麼您的程式的使用者體驗就會比較流暢。

簡短的使用者幫助

行動使用者在使用您的應用程式前沒有時間通讀大量的幫助內容。更重要的是，您也不想用寶貴的空間顯示或存儲幫助內容。iPhone OS裝置的設計特色是易用，所以滿足使用者的期望並使您的應用程式用法清晰明確是十分關鍵的。您可以採取如下措施：

- 正確地使用標準控制元件。因為使用者已經熟悉在內置應用程式中看到的標準控制元件，所以他們瞭解如何在您的應用程式中使用這些控制元件。
- 確保您的資訊中呈現的路徑俱有邏輯性，讓使用者能夠輕鬆作出預測。另外，確保提供標記，比如返回按鈕，這樣使用者就可以知道他們的位置並且瞭解如何重複原來的步驟。

您的選擇是甚麼？

在您決定如何將產品呈現給iPhone OS使用者前，您需要瞭解您可以選擇的範圍。根據產品的實現細節和目標使用者，某些類型的軟體可能更符合您的需求。

本節將為iPhone OS裝置開發的軟體根據實現方法分為三大類。粗略地說，您可以建立：

- iPhone應用程式，也就是您用iPhone SDK開發的，在iPhone OS裝置上本端運行的程式。
- 僅在網頁上顯示的內容，包括Web應用等，也就是像內置iPhone應用程式一樣運行的網站。
- 混合應用程式，在Web視圖區域中提供網頁內容的訪問，但是也包含了一些iPhone OS使用者介面元素。

iPhone應用程式

iPhone應用程式類似於Phone OS裝置上的內置應用程式，因為它們駐留在裝置本身並且利用了iPhone OS環境的特性。使用者在他們的裝置上安裝iPhone應用程式，並像使用內置應用程式一樣使用這些程式，例如股票，地圖，計算機和郵件。

iPhone應用程式啟動十分迅速且易於使用。不論應用程式的任務是發送郵件還是為使用者提供娛樂，它都應具備快速響應以及簡單的特色，另外還應有漂亮簡潔的使用者介面。

僅在網頁上顯示的內容

涉及到為iPhone OS使用者提供僅在網頁上顯示的內容時，您有一些不同的選擇：

- Web應用 提供了一個任務集中解決方案且遵從某種準則的網頁被稱為Web應用，因為它們的行為與iPhone OS內置的應用程式類似。Web應用與所有僅在網頁上顯示的內容相同，都在iPhone中的Safari上運行；使用者僅需跳轉到Web應用的URL，而不用在其裝置上安裝Web應用。
- 優化網頁 根據iPhone上的Safari特色進行顯示和操作優化的網頁（除了某些基於不支援的技術的元素，如外掛元件，Flash和Java）。另外，優化網頁的內容尺寸應適合裝置的螢幕。常常將其設計為在iPhone OS裝置上顯示時進行檢測，以對所提供的內容做相應調整。

- 兼容網頁顯示和操作與iPhone上的Safari兼容的網頁（除了某些基於不支援的技術的元素，如外掛元件，Flash和Java）。雖然兼容網頁不會採取額外的措施來優化其在iPhone OS裝置上的視覺體驗，但是一般均能成功地在裝置上進行顯示。

如果您已經擁有一個網站或Web應用，首先需要確保其能在iPhone OS裝置上正常工作。同時，您應考慮建立一個自定義圖示，使用者可以將該圖示放在主螢幕上並使用網頁剪輯功能。實際上，這樣使用者就能在他們的主螢幕上保留一個看起來像本端應用程式圖示的，指向您網站的書籤。瞭解有關建立自定義圖示以及如何讓iPhone OS裝置上的網頁內容俱有良好視覺效果的更多資訊，請參考iPhone人機介面指南之Web應用。

混合應用程式

在iPhone OS中您可以建立一個結合本端程式和網頁功能的程式。混合應用程式是一個本端iPhone應用程式，它通過Web視圖區域來展示其結構和功能，但是仍然包含標準的iPhone OS使用者介面元素。

混合應用程式讓使用者通過一個稱為Web視圖（如“Web視圖”所述）的元素訪問網頁內容。確切地說，如何在您的應用程式中使用Web視圖由您決定，但是應避免讓使用者覺得您的應用程式僅僅是一個小型的網頁瀏覽器。混合應用程式應像一個本端iPhone應用程式一樣運行；而不應凸顯依賴網路資源這一事實。

三種應用程式風格

根據視覺和行為特徵，數據模型以及使用者體驗，本文件定義了三種程式風格。在您深入閱讀之前，需要強調的是，這些分類的命名和描述是為了幫助您清楚地確認您的設計方案，並不表示所有的iPhone應用程式都必須嚴格地遵循這些分類。描述這些風格是為了幫助您瞭解不同風格的應用程式俱有不同的設計方案。

注意：應用程式的風格並沒有規定實現方法。雖然本文件主要闡述本端的iPhone應用程式設計，但是此處探討的程式風格同樣可以在iPhone OS裝置的網頁或混合應用程式中實現。

在您閱讀這三種應用程式風格時，思考一下每種風格的特色如何增強您的iPhone應用程式中的功能和的整體使用者體驗。在您瞭解iPhone應用程式的不同設計風格時，記住以下問題有助於發掘最適合您的應用程式的特徵組合：

- 您設想使用者是出於甚麼樣的動機去使用您的應用程式？
- 您計劃中的使用者會獲得怎樣的使用者體驗
- 您應用程式的目標或專注點是在哪兒？
- 您的應用程式如何組織，顯示人們關心的資訊？是否和應用程式的主要任務有天然的關係？

效率型應用程式

效率型應用程式俱有組織和操作具體資訊的功能。效率型應用程式通常用於完成比較重要的任務。郵件是效率型應用程式的典型例子。

目的的嚴肅性並不意味著效率型應用程式是乾癟無趣的，但是這種應用程式確實需要比較精簡的使用者體驗，從而不會阻礙使用者的工作。因此，成功的效率型應用程式將使用者體驗的重點放在任務上，使用者可以快速地找到需要的東西，輕鬆地完成必要的操作，以完成該任務繼而轉向其他事務。

效率型應用程式通常分層組織使用者數據。因為使用者可以逐步將選擇具體化，直到達到期望的具體程度，以找到所需的資訊。iPhone OS提供了表格視圖，保證了這一過程在iPhone OS裝置上良好的使用者體驗（更多有關使用者介面的資訊，請參考“表格視圖”）。圖1-1顯示了這種類型數據結構的示例。

圖 1-1 效率型應用程式通常分層組織資訊

Application



效率型應用程式中的典型的使用者交互模型通常包含：

- 組織列表
- 從列表中新增和刪除

- 逐步細化資訊直到達到目標的詳細級別，而後執行與該詳細級別對應的任務

效率型應用程式一般採用多個視圖，通常每一視圖顯示一個層次的資訊。使用者介面通常簡單，整潔且由標準視圖和控制元件組成。效率型應用程式通常沒有很多自定義介面。原因在於效率型應用程式專注於資訊和任務，而非環境或體驗。

在眾多類型的iPhone應用程式中，效率型應用程式最有可能在設置應用程式中提供使用者可操作的選項或者設置功能。這是因為效率型應用程式工作中涉及到大量資訊以及許多潛在的訪問和管理方法。但是，需要強調的是，使用者很少需要改變這些設置，所以這裡不應包含那些能夠在主介面中更改的配置資訊。

實用型應用程式

實用型應用程式完成的簡單任務對使用者輸入要求很低。使用者打開實用型應用程式，是為了快速查看資訊摘要或是在少數對象上執行簡單任務。天氣程式（如圖1-2所示）就是一個實用型應用程式的典型例子。它在一個易讀的摘要中顯示了重點明確的資訊。

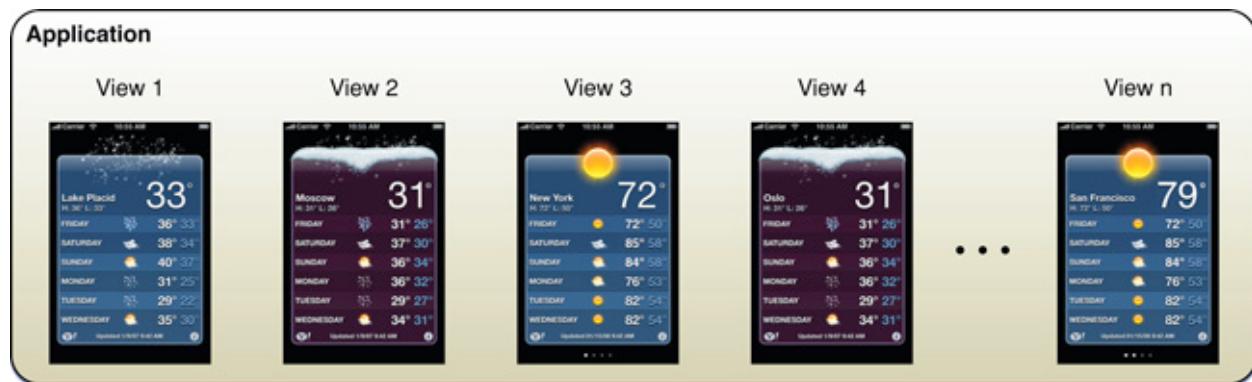
圖 1-2 天氣程式是實用型應用程式的例子



實用型應用程式俱有視覺上的吸引力，但是又能夠突出它們顯示的資訊。使用者使用實用型應用程式來查看一些事情的狀態或是查找一些東西，所以使用者希望能夠快速方便地看到他們感興趣的資訊。因此，實用型應用程式的使用者介面十分簡潔，並提供簡單的，標準的視圖和控制元件。

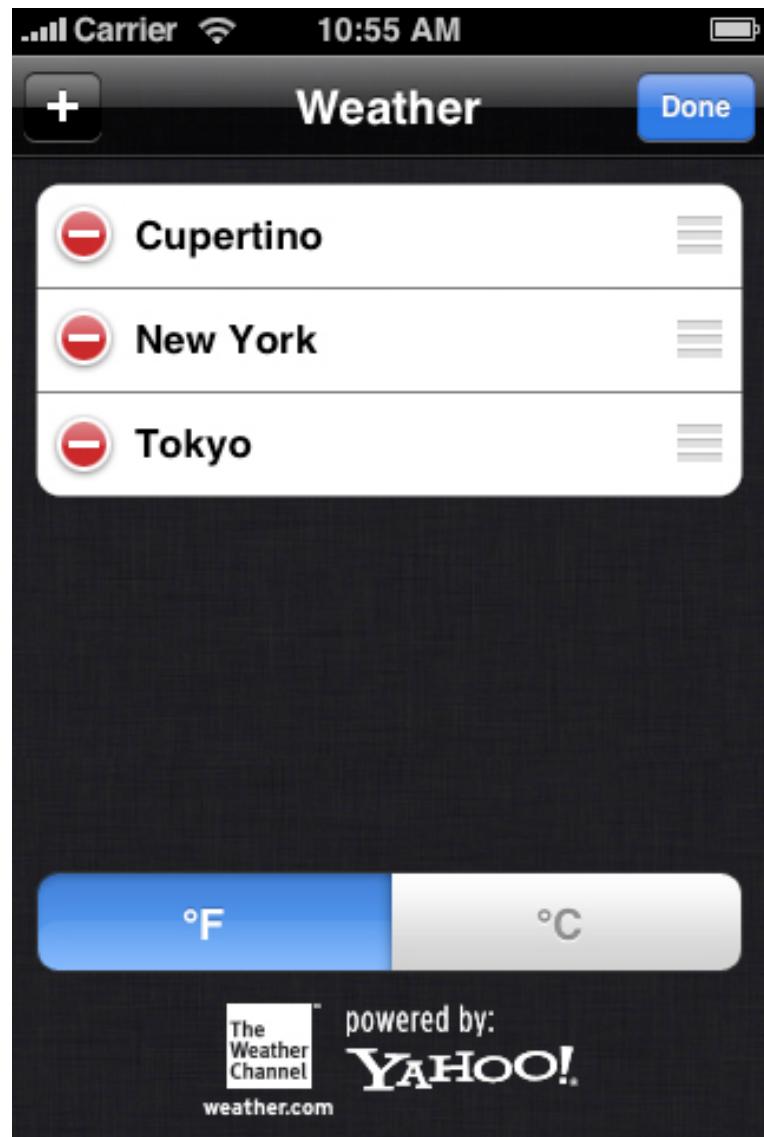
實用型應用程式一般以平面列表的方式顯示資訊；使用者通常不需要深入資訊層次結構。實用型應用程式中的每一個視圖通常都提供同樣的數據組織結構和細節深度，但是數據來源可以不同。由此，使用者可以打開單個實用型應用程式以對多個主題進行相似處理。一些實用型應用程式能顯示打開的視圖的數量；由此使用者可以按順序瀏覽，在一個視圖後選擇另一個視圖。圖1-3顯示了這種類型的數據組織結構。

圖 1-3 實用型應用程式通常在平面列表中顯示數據



實用型應用程式的使用者交互模型非常簡單：使用者打開程式瀏覽資訊摘要，然後能夠選擇改變資訊的設置或來源。實用型應用程式可能需要支援頻繁修改設置和資訊來源，所以它們通常在主視圖的背面提供一小部分這樣的選項。使用者可以點選位於主視圖右下角的，與資訊按鈕相似的按鈕來查看主視圖的背面。在調整完成之後，使用者點選完成按鈕即可返回主視圖的正面。在實用型應用程式中，主視圖背面的選項是應用程式功能的一部分，而不是使用者設置一次以後就極少再用的一組偏好設置。因此，實用型應用程式不應該把特定設置放在設置應用程式中。圖1-4顯示了天氣應用程式在主視圖背面提供的配置選項。

圖 1-4 使用者可以在天氣應用程式的背面進行調整



沈浸式應用程式

沈浸式應用程式提供俱有豐富視覺效果的全屏環境，專注於內容和使用者對內容的體驗。使用者通常使用沈浸式應用程式進行娛樂，不論是玩遊戲，觀看多媒體內容還是執行簡單的任務。

雖然很容易發現遊戲非常適合這種iPhone應用程式風格，但是您可以想象一下，沈浸式應用程式的特色也可以增加其他類型任務的吸引力。呈現一個獨立環境，不顯示大量的文字資訊，需要使用者注意力的任務就十分適合採用沈浸式應用程式。例如，一個應用程式模擬了使用氣泡水平儀的體驗，它在全屏環境下執行的很好，即使它不符合遊戲的定義。在這類程式中，也像遊戲中一樣，使用者的焦點在於視覺上的內容和體驗，而非體驗背後的數據。圖1-5顯示了沈浸式應用程式的例子，它模擬了實際的使用者體驗並執行了簡單的任務。

圖 1-5 沈浸式應用程式不一定是遊戲



注意：雖然橫向顯示的程式啓動時主頁按鈕應處於螢幕的右邊，但是圖1-5中顯示的水準儀程式啓動時的方向恰恰相反。這保證了裝置邊緣的物理按鈕不會干擾測量。關於啓動的更多資訊，請參考“啓動”一節。

沈浸式應用程式通常隱藏裝置的使用者介面，代之以自定義使用者介面，由此增強使用者融入該應用程式的感覺。使用者期望尋找和探索成為沈浸式應用程式體驗的一部分，所以非標準控制元件顯得更為適合。

雖然沈浸式應用程式運行時可能涉及到大量數據的處理，但是通常不顯示這些數據，不讓使用者順序查看或探究明細。而是將資訊呈現在遊戲內容中，如播放，故事或是體驗。同樣由於這個原因，沈浸式應用程式通常顯示自定義導覽方法以配合環境，而標準的，數據驅動的方法則用於實用型應用程式或效率型應用程式中。

沈浸式應用程式的使用者交互模型由應用程式提供的體驗決定。雖然它不像遊戲一樣需要在設置中提供程式特定配置，但是其他類型的沈浸式應用程式可能需要提供。沈浸式應用程式也可能在主視圖背後提供配置選項。

選擇一種應用程式風格

在瞭解效率型應用程式，實用型應用程式和沈浸式應用程式風格以後，考慮一下您的應用程式顯示的資訊類型和所能完成的任務。理論上來說，您應建立的應用程式類型已經十分明顯，您可以開始動手開發了；但實際上並不總是那麼簡單。這裡有一個假設的情景來幫助您做出決定。

如果您有一個想要關注的主題，考慮一下與其相關的對象和任務。想像一下人們對這一主題的不同理解。例如，拿棒球作為例子。棒球在眾多事物中會讓您想起，團隊，遊戲，統計，歷史和球員等。因為棒球對於一個應用程式主題來說可能過於廣泛，所以僅考慮球員。現在，想像一下如何建立一個與球員有關的應用程式，例如，採用他們棒球卡上的肖像。

您可以開發一個效率型應用程式來幫助認真的收藏家管理他們的棒球卡收藏。使用列表的格式，您可以在團隊，球員和賽季的層次結構中顯示卡片。在詳細資料視圖中，使用者能夠記錄他們在哪裡獲得卡片，為卡片花費了多少錢，卡片當前的市場價值和卡片的副本數量。由於此應用程式的重點在於定義收藏的數據，所以使用者介面簡化了資訊收集和新增的任務。

您也可以開發一個實用型應用程式來顯示特定棒球卡片的當前市場價值。每一視圖都看起來像一張棒球卡片，上面新增了它的當前價值，視圖背面可以允許使用者選擇特定的卡片來追蹤和顯示。因為該應用程式的重點在於獨立卡片上，所以使用者介面強調卡片的外觀並提供一個或兩個控制元件以允許使用者查找新卡片。

當然，您也可以開發一款遊戲。遊戲的重點可能在於使用者對個別棒球卡片的某些統計知識或是識別知名卡片的能力。又或者，僅用棒球卡片作為另一類型遊戲中的圖示，比如滑動拼圖遊戲。在這些情況中，應用程式的重點均為棒球卡片的圖像和遊戲的玩法。使用者介面通過顯示某些棒球主題控制元件，隱藏iPhone OS使用者介面的方式來滿足這一點。

需要重申的是，不要局限於單個程式風格。您可能會發現不同程式風格的特色組合起來最符合您的程式目標。

如果有疑問，就簡化問題。將功能列表減少到最小，並建立一個僅執行單個簡單任務的程式（更多建議請參考“給出產品定義說明”）。當您看見人們如何使用並對應用程式作出響應時，您可以選擇對應用程式重點做輕微修改或是更改呈現方式，由此建立應用程式的另一版本。或者，您可能發現對於同一概念有更多或更少的細節要求的版本需求。

當您有個現成的電腦應用程式時

如果您已有一個電腦應用程式，不要只是將其移植到iPhone OS上。iPhone OS裝置的使用方法與台式電腦和筆記型電腦大不相同，使用者對於使用者體驗的期望也不同。

記住，使用者使用iPhone OS裝置時通常處於行動，紛雜的環境中。這意味著使用者希望打開您的程式時，能夠輕鬆簡單地使用，進而可以進行別的活動。如果您的應用程式要求使用者在很長一段時間內集中注意力，那麼如果您在將該應用程式轉換為iPhone OS程式時，您需要重新考慮其結構和目的。

如果您的桌面應用程式能夠執行複雜的任務或是一系列任務，那麼您應該仔細觀察使用者是如何使用該應用程式的，以便找到當使用者處於行動狀態時希望應用程式能夠完成的子任務。例如，支援項目調度，結算和費用報告的，面向企業的應用程式可以轉換為顯示項目的進度摘要的iPhone實用型應用程式，或是允許行動使用者跟蹤其業務相關開支的效率型應用程式。

當您考慮將您的桌面應用程式轉換為iPhone應用程式時，應在設計應用程式時遵循80-20規則。最大比例使用者（至少80%）將使用應用程式中非常有限的特性，僅一小部分使用者（不大於20%）將使用所有的特性。因此，您應仔細考慮，您是否想讓您的iPhone應用程式承載僅有少部分使用者使用的強大特性。您應意識到桌面應用程式可能是能夠提供那些特性的較好環境，同時，將您的iPhone應用程式的重點放在滿足絕大部分使用者需求的特性上通常來說是明智的選擇。

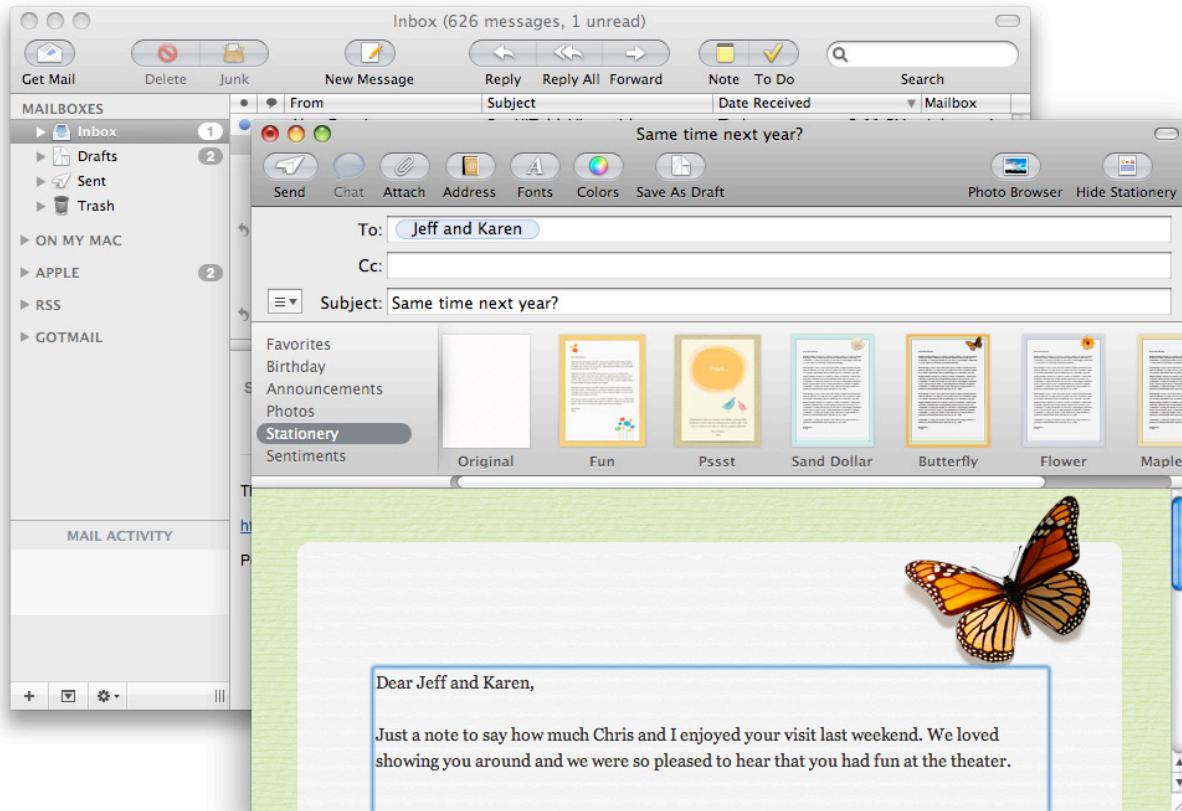
案例研究：將桌面應用程式移植到iPhone OS

為了幫助您理解如何建立桌面應用程式的iPhone OS版本，本小節介紹了一些常見的Mac OS X應用程式和其對應的iPhone OS版本的一些設計差異。當您瞭解了每個應用程式中哪些特色和功能適合iPhone OS時候，那麼，當您需要開發iPhone應用程式時，您就可以清楚明白所需要的設計方案。

郵件

郵件是Mac OS X中最明顯的，廣泛使用，且廣受好評的應用程式。同時它也是功能非常強大的應用程式，允許使用者編寫，收取，分類，存儲郵件，跟蹤行動項目和事件，建立筆記和邀請。郵件在一個多面板的視窗中提供大部分功能。這對於使用桌上型電腦電腦的使用者來說十分方便，因為他們可以將一直郵件視窗留在顯示屏上（或最小化到Dock）並在他們選擇的任意時刻轉到郵件視窗。圖1-6顯示了郵件查看視窗和編寫視窗提供的許多功能。

圖 1-6 桌上型電腦上的郵件程式在兩個視窗中提供了很多的強大功能

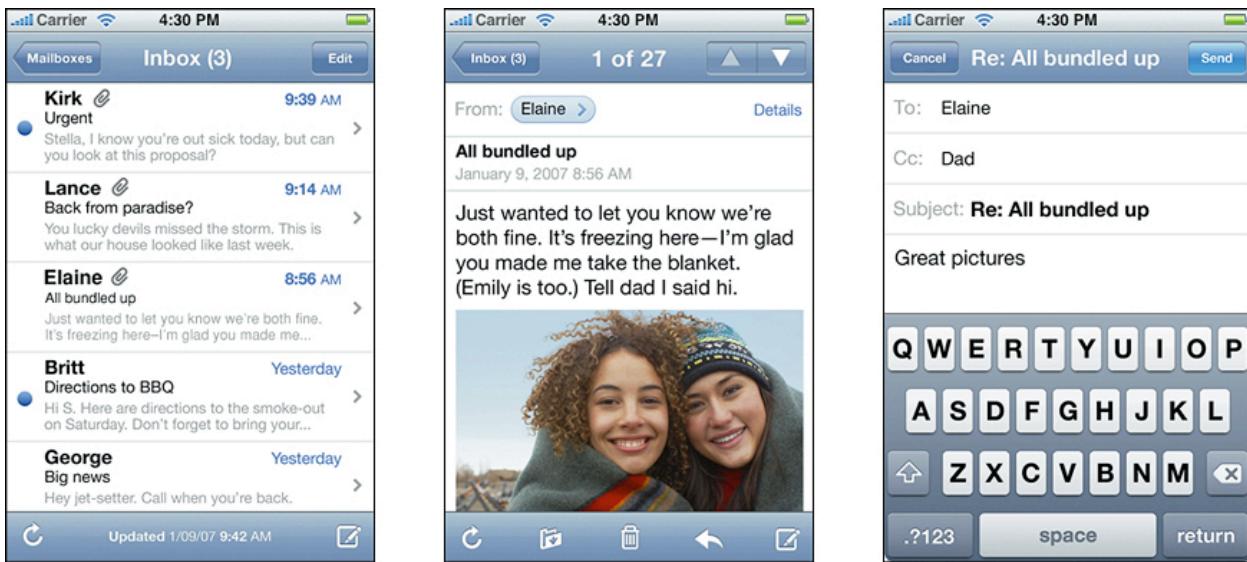


但是在人們處於行動狀態時，他們需要一個更加簡單，且能夠快速完成其核心功能的郵件程式。因此，在iPhone OS的裝置上的郵件應用程式致力於實現使用者處理郵件的最關鍵事項：收取，編寫，發送和管理郵件。為此，使用者介面被精簡，使得使用者的帳戶和郵箱的結構更為清晰，並將使用者的注意力集中在郵件上。

iPhone OS中的郵件應用程式是一個效率型應用程式的完美例子。為了便於內容導覽，iPhone OS中的郵件應用程式利用了人們郵件中自然的層次結構，使用一連串的頁面顯示帳號，郵箱，郵件列表以及郵件內容。使用者可以在列表中選擇某一項，從一般（帳號列表）至特殊（一條資訊）地進行深入瞭解，並查看與該項相關的事物。瞭解更多效率型應用程式風格的iPhone應用程式，請參考“效率型應用程式”。

另外，iPhone OS中的郵件應用程式允許多種操作，比如建立和發送。這是通過顯示幾個可以被點選且使用者熟悉的控制元件來實現的。圖1-7顯示了郵件應用程式如何使使用者在iPhone OS中查看和發送郵件變得簡單。注意每一屏上方的元素，它們讓使用者更輕鬆地瞭解到他們過去和當前在程式中所處的位置。

圖 1-7 iPhone OS中的郵件應用程式使得查看和發送郵件變得簡單



iPhoto

另一個從Mac OS X應用程式重構為iPhone OS程式的例子是iPhoto。在桌上型電腦上，iPhoto支援全面搜尋和管理，俱有強大的編輯能力以及富有創意的打印選項。當人們在他們的桌上型電腦或筆記型電腦上使用iPhoto時，他們希望能夠查看和管理整個收藏集，對照片進行調整並且以各種方式來操縱照片。雖然iPhoto的重點是使用者的內容，但是程式也在其視窗中提供了豐富的功能。圖1-8顯示了桌上型電腦上iPhoto的使用者介面。

圖 1-8 iPhoto使用者介面



但是在使用者處於行動狀態時，他們沒有時間來編輯圖片（同時他們也不期望能夠打印圖片）；而是希望能夠快速瀏覽並分享他們的圖片。

為了在iPhone OS裝置上滿足這一要求，蘋果提供了照片應用程式。該程式能夠顯示圖片並與其他使用者分享圖片。照片應用程式的使用者介面完全以照片為核心；以至於實際裝置的使用者介面被部分隱藏。當使用者選擇以幻燈片形式查看圖片時，照片應用程式將隱藏導覽欄，工具欄甚至狀態欄。當使用者需要查看這些工具欄時，這些元素又以半透明狀態顯示出來。

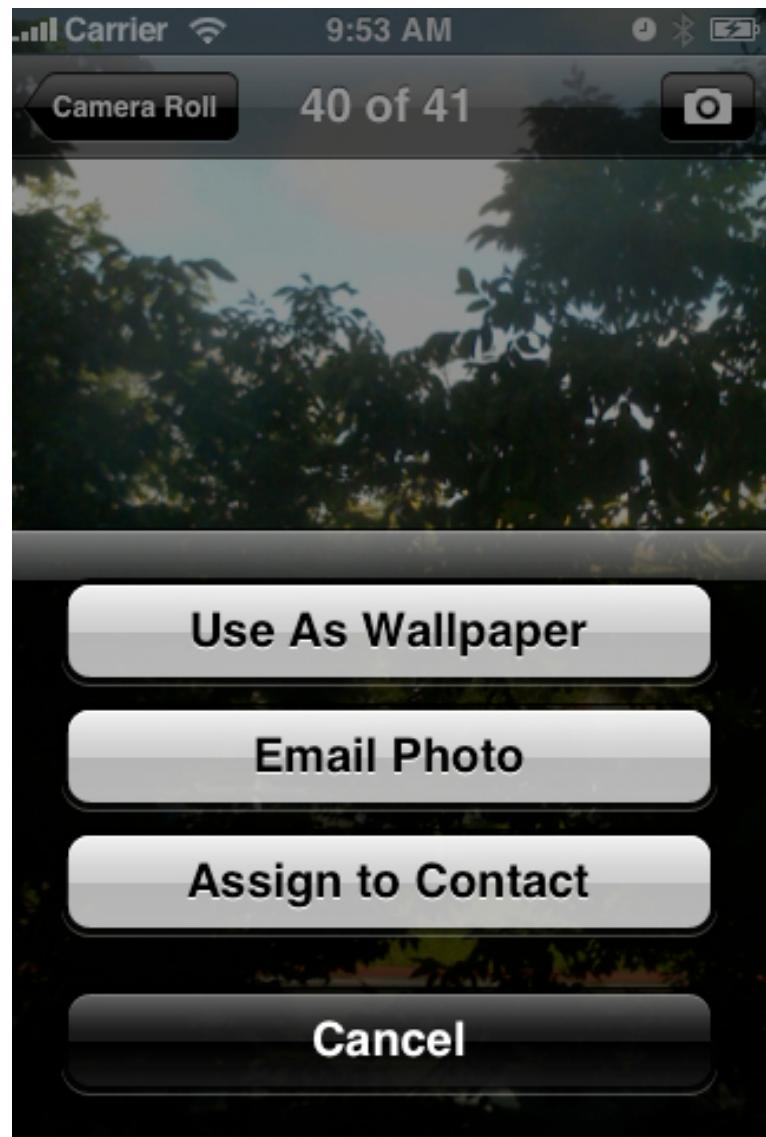
照片應用程式使用層級結構使得使用者可以輕鬆地整理，查找照片：使用者可以選擇一個包含了一系列照片的相冊，而後從相冊中選擇單張照片。按照這種方式，照片應用程式就是一個結合了效率型應用程式風格和沈浸式應用程式風格特性（瞭解更多有關這些風格的資訊，見“三種應用程式風格”）的應用程式。圖1-9顯示了使用者如何在照片應用程式中查看圖片。

圖 1-9 照片應用程式中的三個介面



另外，照片應用程式提供臨時視圖，稱為動作表單（如“警告，動作表單以及模式視圖”所述），從而讓使用者能夠在不退出圖片查看視圖的情況下使用其他的功能。圖1-10顯示了照片應用程式如何為單張圖片提供動作表單。

圖 1-10 照片應用程式的動作表單



人機介面準則：建立優秀的使用者介面

優秀的使用者介面應該遵循一些人機介面設計準則，這些準則來源於人—也即使用者—思考和工作的方式，而不是裝置的能力。沒有吸引力的，令人難以理解的或是不合邏輯的使用者介面會讓一個原本優秀的程式成為垃圾，而漂亮，直覺而又吸引人的使用者介面不僅可以改善程式的功能，還能激發使用者的正面依賴感。

即使您已經對基本的設計準則有所瞭解，這一章也值得您讀一下，因為本章著重講述了這些準則是如何應用在iPhone應用程式中的。

隱喻

盡可能按照現實世界中的對象和操作來對您程式中的對象和操作建模。這能夠幫助使用者，尤其是新手，迅速地理解您的應用程式的是如何工作的。資料夾就是一個典型的軟體隱喻。因為在現實世界中，人們將文件放在資料夾裡，所以人們就能夠迅速地理解在電腦上將數據放入資料夾的概念。

iPhone OS中的隱喻包括iPod播放控制元件，點選控制元件來觸發事件，滑動開關，以及調色板上的數據。

雖然隱喻暗示了iPhone OS使用者介面中對象和操作的含義，但是並沒有限制其軟體實現。回到資料夾的例子，軟體實現的資料夾對象所俱有的容量與現實世界中所對應的資料夾的物理容量毫無關係。

在您設計程式時，需要注意iPhone OS中已存在的隱喻，避免重複定義。同時，檢查您的程式功能是否有對應的可用的自然隱喻。記住，即使如此，相較於為了適應您的應用程式使用者介面而勉強使用現實世界的對象或操作的隱喻而言，使用標準控制元件和操作更為妥善。除非您選擇的隱喻能得到大部分使用者的認可，包括那些會增加而不是減少使用者困惑的隱喻。

直覺操作

直覺操作意為人們感到他們在物理上，而非抽象地控制某物。其優點在於，使用者更容易理解他們的行為對於操作對象所產生的結果。

得益於多點觸摸的使用者介面，iPhone OS使用者享有高度直覺的操控感。手勢的使用讓人們感到螢幕上顯示的對象俱有更大的親和力和更強的操控感，因為他們並不需要用任何中間裝置（如鼠標）就操控它們。

為了提高您的iPhone應用程式的操控感，您應確保：

- 當使用者操作螢幕上的對象時，這些對象保持可見
- 使用者操作的結果是一目瞭然的

即看即點

iPhone應用程式比人更善於記住一些選項，命令，數據等。通過列表的形式來提供選擇或選項，使用者可以輕鬆地瀏覽它們並作出選擇。盡量減少文字輸入的需要，使得使用者就不必花費大量時間輸入命令，您的程式也不需要執行大量的錯誤檢查。

讓使用者做出選擇，而不是要求他們進行開放式的輸入，這樣能夠讓使用者能夠集中精力通過您的程式完成任務，而不是疲於記憶如何操作。

回饋

使用者除了需要看到操作的結果外，還需要得到操作及時的回饋，以及在長時間操作時的狀態報告。您的程式應通過一些視覺上的變化來回應使用者的每一個操作。例如，當使用者點選列表項時，確保其高亮顯示。聲音提示同樣有效，但是它不能作為主要或唯一的回饋機制。因為人們可能在無法聽見聲音或是必須關閉聲音的環境中使用iPhone OS的裝置。另外，您不能使用iPhone OS的系統聲音，因為他們已經和系統警告關聯。

iPhone OS繁忙時會顯示活動指示符以自動提供回饋。在持續幾秒以上的操作過程中，您的應用程式應顯示完成進度，在適當的情況下，還應顯示解釋性消息。

精細且有意義的動畫是向使用者提供回饋的良好方式。動畫貫穿於整個iPhone OS，即使是在非沈浸式的程式中。作為一種提供回饋的方式，動畫只是用於增強使用者的體驗，但不是使用者體驗的焦點。

使用者控制

讓使用者來觸發和控制操作，而不是您的應用程式。保持操作簡單直接，使得使用者可以輕鬆地理解和並記住他們。盡可能使用使用者已經熟悉的標準控制元件和行為。

在操作開始前應提供充分的取消操作的機會，並且確保在使用者觸發潛在的破壞性操作時取得操作確認。盡可能使使用者能夠正常終止正在進行的操作。

美學完整性

雖然一個程式的首要目的是為了完成某個任務，即使該任務僅僅是玩遊戲，我們也不應低估程式外觀的重要性。這是因為外觀對功能有深遠的影響：一個看起來零亂或者缺乏邏輯的程式是很難被理解和使用的。

美學上的完整性並不是用來衡量您的程式有多漂亮，而是衡量您程式的外觀和程式功能結合得有多好。例如，一個效率型應用程式應盡量精簡裝飾性的元素並置於背景中，同時使用標準控制元件和行為以突出任務。

沈浸式程式是另一極端，使用者期待有趣且需要一些探索的漂亮外觀。雖然程式趨向於提供消遣，但是其外觀仍需和其任務結合。仔細設計這類程式的使用者介面，從而為使用者提供內部一致的使用者體驗。

設計iPhone應用程式： 從產品定義到品牌宣傳

當您開發iPhone應用程式時，您需要瞭解iPhone OS以及行動裝置環境的各個方面給您的設計帶來的影響。本章涵蓋了從產品定義到品牌宣傳的一系列程式設計問題的指導方針，並介紹了在iPhone應用程式中是如何解決這些問題的。

給出產品定義說明

在開始設計您的應用程式之前，精確定義您的應用程式是用來做甚麼的非常重要。一個比較好的方法是編寫一份產品定義說明—用於描述您的應用程式的主要功能及目標使用者的文件。建立產品定義說明不僅僅是一種練習，相反地，它是將一系列產品功能轉化成一個清晰完整的產品的最佳方法之一。

首先，花些時間定義您的目標使用者：他們是專家還是新手？是嚴謹的還是隨和的？是帶著特定任務來尋求幫助還是為了娛樂消遣的？瞭解這些可以幫助您根據使用者的特殊需求來定制使用者體驗和使用者介面。

由於您是在設計一個iPhone應用程式，有一些使用者需求是已知的。例如：

- 他們是行動的。
- 他們希望能夠盡可能快地打開您的應用程式，並立即看到有用的資訊。
- 他們應只需要簡單幾步就能夠完成任務

然後思考一下，甚麼樣的特性使得您的使用者和iPhone OS的其他使用者區分開。他們是商人，青少年還是退休人員？他們是否會在每一天的結束時，或是當他們查收電子郵件時，亦或是當他們有一些額外時間時，使用您的應用程式呢？您對目標使用者定義得越準確，您對使用者介面的外觀，體驗以及功能的設計也就越準確。

例如，如果您的應用程式是幫助商務人士記錄他們的開支，那麼您的使用者介面應該側重於提供正確的分類，並且易於輸入費用，而不是詢問許多無關核心功能的細節。另外，您最好為介面選擇一種比較專業的顏色，並且不會讓使用者產生視覺疲勞。

或者，如果您的應用程式是一個以青少年為目標受眾的遊戲，那麼您可能需要一個活潑一點的使用者介面，煽動性的語言，時尚一點的色彩。

最後，研究一下您打算提供的產品功能。根據您對目標使用者的想象，嘗試將一系列產品功能提煉成一份產品定義說明文件，這份文件描述了您的產品所提供的解決方案以及您的目標使用者。例如，iPhoto桌面應用程式允許使用者整理，編輯，共享，打印和查看照片，但一份優秀的產品定義說明不僅僅包括產品的功能，同時還描述了其目標使用者。因此，一份完整的iPhoto產品定義說明可能是“一種為業餘攝影愛好者設計的，簡單易用的照片管理應用程式。”注意，在產品定義說明中包含您的目標使用者定義是多麼重要：想象一下，如果iPhoto被定義為成“一種為專業攝影師提供的，簡單易用的照片管理應用程式”，那將有多麼不同。

一份優秀的產品定義說明就像一個工具，您應該在整個開發過程中使用它來確定功能，工具和術語的是否合適。尤其重要的是清除那些不符合產品定義說明的元素，因為iPhone應用程式沒有多餘空間去支援非核心功能。

例如，想象一下，您正在考慮開發一個人們在購買生活用品時可以使用的iPhone應用程式。在計劃階段，您可能需要考慮使用者可能喜歡的許多種不同的活動，例如：

- 獲取關於特定食品的營養資訊
- 尋找打折和優惠資訊
- 建立和使用購物清單
- 定位商店
- 查閱食譜
- 比較價格
- 查看購物總額

然而，您認為您的使用者最關心的是記住他們需要購買的所有東西，盡可能地省錢，並且很可能急於帶著採購品回家。根據這些對使用者的定義，您可以為您的應用程式製作一份產品定義說明，比如“為忙碌的人群提供建立購物清單，查找折扣的工具。”通過這份產品定義說明過濾掉產品的潛在功能，您決定主要側重於如何使購物清單易於建立，存儲和使用。您也為使用者提供了查找他們購物清單上所列商品的打折資訊的功能。即使其他功能

也是有用的（並且可能成為其他應用的主要功能），但它們不符合這個應用程式的產品定義說明。

當您確定了一個可靠的產品定義說明，並且開始使用它來過濾提出的那些功能，您可能還希望用它來確保最初關於應用程式類型的決定是正確的。如果您在頭腦中帶著一個具體的應用程式類型開始了您的開發過程，您可能會發現定義產品定義說明的過程已經改變了其面貌。（關於您所能開發的不同類型的應用程式，請參考“三種應用程式風格”）

優秀iPhone應用程式的共性

優秀的iPhone應用程式精確地滿足了使用者需求，同時提供了使用者想要的體驗。為了幫助您在應用程式中平衡這兩點，本小節歸納了優秀iPhone應用程式的一些共有特色，並就如何將這些特色融合到您的產品中給予了建議。

簡單和易用性

簡單和易用是所有軟體的基本原則，但在iPhone應用程式中它們尤其重要。iPhone OS使用者很可能在使用您的應用程式的同時，還做著其他事情。如果使用者無法快速明白如何使用您的應用程式，他們很可能會轉而使用其他競爭者的應用程式，而且不會再嘗試您的應用程式。

當您設計應用程式流程以及使用者介面時，請遵循如下準則以保證應用程式的簡單且易於使用：

- 如何使用您的應用程式應顯而易見。
- 常見的功能和資訊集中於螢幕的頂部。
- 最大限度地減少文字輸入。
- 簡潔地表達基本資訊。
- 為所有可點選元素提供一個指尖大小的目標區域。

以下各小節就簡單和易用性的準則進行更詳細的說明。

顯而易見

您不能假設使用者有時間（或能夠分出注意力）來研究您的應用程式是如何運作。因此，您必須努力讓應用程式能夠立刻被使用者理解。

您的應用程式的主要功能應該一目瞭然。您可以最大限度地減少使用者必須選擇的控制元件數量，並清晰地標識這些控制元件以便使用者明白它們的功能。例如，在內置的碼錶功能中（時鐘應用程式的一部分），如圖3-1所示，使用者一眼就可以看出哪個是停止按鈕，哪個是啓動按鈕，哪個是用來記錄每圈時間。

圖 3-1 內置碼錶功能的用法顯而易見



由上而下的佈局

使用者可以用手指或大拇指點選一個iPhone OS的裝置的螢幕。使用者傾向於用他們的非慣用手握住裝置（或擋在手心），用慣用手的手指點選裝置。當使用大拇指時，人們要麼將裝置一手握住，然後用大拇指點選，要麼將裝置握於雙手間，同時用兩根大拇指點選。無論用哪一種方式，對於使用者來說螢幕最上方都是最顯眼的。

由於這些使用模式，您在設計應用程式的使用者介面時，應將最常使用的（通常也是重要的）資訊置於最明顯且最方便的頂部。當使用者從螢幕頂部瀏覽到底部的時候，呈現的資訊的過程應該是由一般到特殊，從高級到低級的過程。

最大限度地減少所需的輸入

無論使用者是點選控制元件還是使用鍵盤，輸入資訊都會佔用他們的時間和精力。如果您的應用程式需要使用者在獲取有用資訊前進行大量輸入，這將降低使用者效率，並且會妨礙使用者繼續使用您的程式。

當然，您通常需要一些來自使用者的資訊。但您應該在使用者提供的資訊和您回饋給使用者的資訊之間做出權衡。換句話說，努力為使用者提供的每一條資訊回饋盡可能多的資訊或功能。這樣，當使用者使用您的應用程式時他們會感到正在取得進展，並沒有被阻塞。

當您向使用者請求輸入時，考慮使用表格視圖（或選擇器）來代替文字輸入框。對使用者來說，從一個列表中選擇一項通常比輸入單詞容易得多。有關表格視圖和選擇器的細節，請分別參考“表格視圖”和“選擇器”小節。

簡潔地表達資訊

當使用者介面上的文字簡短而又直接的話，使用者就能夠快速地理解它。因此，簡明扼要地表達最重要的資訊，並顯著地突出該資訊，這樣使用者不必為了所尋找的資訊或為了弄清下一步該怎麼做而閱讀大段的文字。

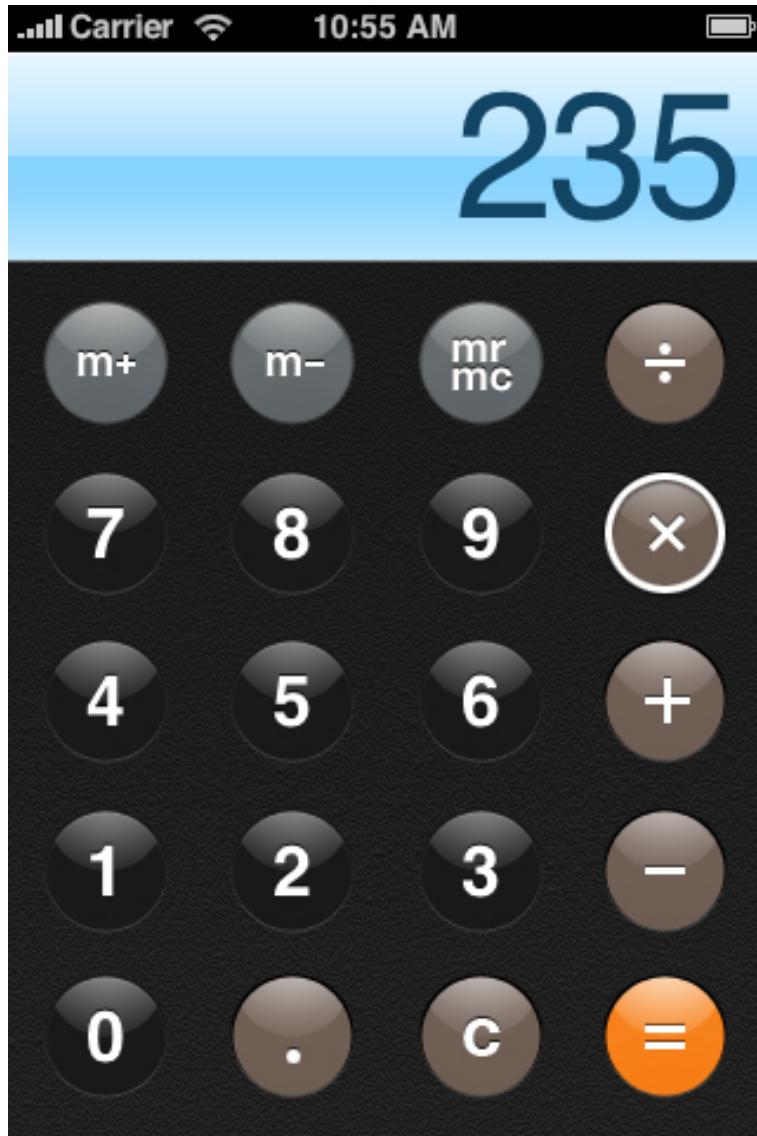
為了幫助您做到這一點，您可以像一位報紙編輯一樣地思考，努力用壓縮標題的風格傳遞資訊。為控制元件加上簡短的標籤（或易於理解的符號），以便使用者一眼就能瞭解如何使用它們。

為控制元件提供指尖大小的目標區域

如果您的佈局將控制元件放置得過於靠近，使用者需要花費更多的時間和精力來確定他們點選的區域，並且很容易點錯控制元件。一個簡單易用的使用者介面應該合理地分配控制元件和其他使用者介面元素，使得使用者能夠更方便地點中控制元件。

例如，內置的計算機應用程式顯示很大的，易於點選的控制元件，每個控制元件的目標區域大約44x44像素。圖3-2 展示了計算機應用程式。

圖 3-2 內置計算機應用程式的控制元件



將重點放在主要功能上

一個令使用者滿意且愉快iPhone應用程式應將重點放在它的主要功能上，因此，當您設計應用程式時，要始終將重點放在您的產品定義說明上，並確保每個產品功能和使用者介面元素支援這個說明。關於如何建立產品定義說明的相關建議，請參考“給出產品定義說明”。

一個比較好的方法是確定在每個上下文中甚麼是最重要。當您決定要在每個螢幕中顯示甚麼時不斷問問自己，這是使用者現在需要的關鍵資訊或功能嗎？或者，想得更加具體些，當使用者在商場購物或在會議間步行時，這個資訊或功能是不是他們需要的？如果不是，那它在另一個不同上下文中是不是重要？或者它終究不是那麼重要？例如，一個幫助使用者記錄汽車里程數的應用程式不應關心汽車經銷商的位置。

當您遵照這些準則時，您的解決方案變得更為清晰。特別是，您的應用程式使用起來更為顯而易見，且最小限度地減少了使用者輸入。集中關注您的解決方案的重點，從而使得使用者能更加方便，快速地進入到應用程式中最重要的部分（關於這些準則的具體內容，請參考“簡單和易用性”）。

例如，內置的日曆應用程式（如圖3-3 所示）重點在於日期及當天發生的事件。使用者可以使用俱有明確標籤的按鈕顯示當前日期，選擇查看選項，新增事件等。最重要的資訊是日期以及與之相關的事件，也是最顯眼的。使用者輸入也很簡單，它允許使用者從列表中選擇時間，重複間隔以及提醒選項，而不是要求按鍵輸入。

圖 3-3 內置的日曆應用程式重點是日期和事件



有效地溝通

溝通和回饋，在iPhone應用程式中的重要性和在桌面應用程式一樣。使用者需要知道他們的請求是否正在處理中，以及他們的行為是否會導致數據丟失或其他問題。儘管如此，避免矯枉過正的溝通也很重要，例如在並非真正嚴重的情況下提醒使用者或過於頻繁地請求確認。

動畫是一種進行有效溝通的好方法，前提是它不會妨礙使用者的任務或影響他們的效率。精細適當的動畫可以傳遞資訊，提供有用的回饋，以可視化的方式向使用者展示他們行為的結果。但是過多或無意義的動畫會阻礙應用程式的流程，影響其性能，並困擾使用者。

在所有與使用者的基於文字的溝通中，一定要使用以使用者為中心的術語，特別是要避免在使用者介面中使用技術術語。例如，在Wi-Fi無線網路偏好設置介面使用清楚的，非技術性的語言來描述如何將裝置連接到網路，如圖3-4所示。

圖 3-4 在應用程式的使用者介面中使用以使用者為中心的術語



適當的支援手勢

人們習慣用手指來操作iPhone OS裝置獨特的多點觸摸介面，使用如輕敲，搖動，夾等手勢來選擇，操作，閱讀網頁內容或使用應用程式。使用手指操作裝置其真正的優勢在於：手指總是可用的，它們有能力完成許多不同的動作，並且它們給使用者一種直覺的，和裝置關聯的感覺，這是諸如鼠標等外部輸入裝置無法實現的。

然而，手指有一個主要缺點，和鼠標指針相比，無論是在大小，形狀還是靈敏度方面都要差很多。在螢幕的上下文中，手指永遠不可能像鼠標指針一樣精確。

所幸的是，您可以用一個良好的使用者介面設計來彌補手指輸入裝置帶來的挑戰。在大多數情況下這意味著您需要確保您的佈局可容納一個指尖的平均大小，同樣也意味著您需要響應手指的動作以讓使用者獲得其期望的行為。

使用者進行一些特定的動作，從而獲得特殊的結果，這種動作稱之為手勢。例如，使用者點選一個按鈕來選中它，並搖動，或拖拽以滾動一個列表。因為內置的應用程式一致使用，iPhone使用者已經習慣了這些手勢。因此，您應該在您的應用程式中恰當地使用這些手勢，避免混淆。

還有一些更加複雜的手勢，如swipe或pinch open，一樣在內置的應用程式中被使用，但它們並不常見。一般來說，這些手勢被用作快捷方式來加快完成任務，而不是執行任務的唯一方法。例如，當查看郵箱中郵件列表時，使用者顯示消息預覽行中的刪除按鈕並點選它來刪除一條正在查看的消息。使用者可以用兩種不同的方式顯示刪除按鈕：

- 點選導覽欄中的編輯按鈕，這個按鈕將在每個預覽行裡顯示一個刪除控制元件。然後點選該控制元件就會顯示刪除按鈕。
- 在特定的預覽行使用swipe手勢，就會顯示這條消息的刪除按鈕。

第一種方法需要一個額外的步驟，但容易被發現，因為它僅僅需要點選已經清楚標識的編輯按鈕。第二種方法顯然更快，但它需要使用者學習並記住特定swipe手勢。

因此，為確保您的應用程式易於使用，盡量使用常見的那些手勢中，如點選，拖拽等。您應該避免將諸如swipe或pinch open等不常見的手勢作為執行一個動作的唯一方法。您的應用程式應該有一個簡單直接的方法來執行一個動作，即使它意味著多點選幾次。

在大多數應用程式中，避免定義新的手勢同樣很重要，特別是這些新手勢所執行的動作已經和標準手勢想關聯。一個例外是沈浸式的應用程式，在這種應用中可以適當地加入自定義的手勢。例如，一個效率型應用程式要求使用者作出畫圈手勢來在一個表格行中顯示刪

除按鈕，這樣的應用程式將是混亂的且難於使用的。但另一方面，一個遊戲要求做出畫圈的手勢來轉動遊戲的某部分就顯得比較合理。

表3-1列出了使用者可以使用的標準手勢。不要試圖重新定義這些手勢的含義；相反地，如果您的應用程式中支援這些行為，請確保也能夠支援與之對應的手勢。瞭解更多有關如何處理由手勢產生的事件，請參考iPhone應用程式程序指南。

表3-1 用於和iPhone OS裝置交互的手勢

手勢	操作
Tap	按或選擇一個控制元件或項目（類似於單擊一次鼠標）。
Drag	拖動
Flick	快速滾動或搖晃
Swipe	在一個表視圖行中，顯示刪除按鈕。
Double tap	放大並將內容或圖片部分居中。 縮小（如果已經被放大的話）。
Pinch open	放大
Pinch close	縮小
Touch and hold	在可編輯文字中，顯示一個放大的光標視圖。

慎重地融合品牌元素

當品牌的設計巧妙且簡約時，它是最有效的。使用者使用您的iPhone應用程式來完成某項任務或娛樂時，他們不希望產生被強迫觀看廣告的感覺。因此，您應該以一種低調的方

式引入您的品牌的顏色或圖片。例如，您可以在視圖和控制元件中使用一個自定義的配色方案。

應用程式圖示是例外情況，它應該突出顯示您的品牌（應用程式圖示是使用者在安裝完應用程式後，在主螢幕中可以看到的圖示），使用者會經常看到這個圖示，因此，花費點時間讓您的品牌更吸引眼球是有必要的。關於如何設計一個應用程式圖示，請參考“[應用程式圖示](#)”。

處理常見任務

如果您俱有使用桌面電腦或便攜式電腦應用程式的經驗，那麼您會發現iPhone應用程式處理很多常見任務的方式與它們不同。本節從人機介面的角度描述了這些常見任務；關於程式碼實現的技術細節，請參考iPhone應用程式程序指南。

啓動

iPhone應用程式應能夠迅速啓動，從而使用者可以立即開始使用它們。啓動時，iPhone應用程式應該：

- 指定適當的狀態欄樣式（關於可用樣式的資訊請參考“狀態欄”一節）。
- 顯示一幅酷似應用程式初始螢幕的啓動圖像。這減少了使用者感知到的應用程式的啓動時間。要瞭解更多有關的資訊，請參考“啓動圖像”。
- 避免顯示“關於”視窗，啓動畫面，也不要提供任何其他類型的妨礙使用者立即使用應用程式的體驗。
- 預設情況下，在螢幕上縱向啓動應用程式。如果您打算讓應用程式只能在螢幕橫向時使用，則無論裝置當前的方向如何，都在螢幕上橫向啓動應用程式。如果必要的話，要允許使用者將裝置旋轉為橫向。僅橫向顯示的應用程式應該支援兩種“橫向”—不論“主頁”按鈕在螢幕右側還是左側，都能夠正常顯示應用程式。如果裝置本身已經被橫向放置，則這種僅橫向顯示的應用程式就按照裝置的方向啓動。否則，在預設情況下，僅橫向顯示的應用程式在啓動時，“主頁”按鈕只能在應用程式的右方。
- 將您的應用程式恢復到最後一次運行時的狀態。

重要：不要在您的應用程式安裝完成後提示使用者重新引導或重新啓動他們的裝置。如果您的應用程式存在記憶體使用或其他方面的問題，導致除非系統是剛剛啓動過的，否則您的應用程式難以運行，那麼您需要解決這些問題。例如，您可以閱讀iPhone應用程式程序指南中的“高效地使用記憶體”一節，瞭解如何開發運轉良好的應用程式。

停止

當使用者打開另一個應用程式或使用裝置自身功能（比如電話）時，iPhone應用程式必需終止。要特別注意的是，應用程式的終止不需要使用者點選應用程式關閉按鈕或在選單上選擇“退出”操作。iPhone應用程式應該：

- 隨時準備接收退出或終止通知。因此，要盡可能快並且在合理情況下經常儲存使用者數據。
- 當應用程式終止時，盡可能詳細地儲存它的當前狀態。例如，如果您的應用程式能夠顯示滾動的數據，則應該儲存當前的滾動位置。

iPhone應用程式不應該直接通過程式碼退出，因為這樣做會使使用者覺得應用程式崩潰了。但是，可能有些時候，外部環境會阻止您的應用程式正常運行。處理這種情況的最好方法是讓螢幕顯示醒目的內容，在螢幕上描述該問題並向使用者提出解決問題的建議。這樣做會在以下兩方面對使用者有所幫助：

- 它提供了回饋資訊，告訴使用者您的應用程式沒有出現錯誤，使使用者放心。
- 它能夠控制使用者行為，讓他們決定是採取一些糾正的行動來繼續使用您的應用程式，還是按下“主頁”按鈕來打開另一個應用程式。

如果在一些特定的環境中，您的應用程式只是部分功能無法正常工作，那麼在使用者想要激活該功能時，您可以在螢幕上顯示資訊或警告。雖然警告在設計上不具備太大的靈活性，但如果您可以保證以下幾點，它也可以成為一種很好的選擇：

- 非常簡潔地描述當前情況
- 提供一個執行糾正動作的按鈕
- 僅當使用者試圖訪問無法正常工作的功能時顯示警告

和所有的警告一樣，使用者越少看到它們，就說明它們越有效。關於建立警告的更多資訊請參考“使用警告”。

管理設置或配置選項

iPhone應用程式可以向使用者提供設置功能，使使用者可以根據自己的喜好來設置應用程式行為或配置選項，從而改變應用程式的一些功能。可設置的應該是一些使用者設置一次後很少（如果有的話）改變的資訊，比如帳戶名稱。使用者可以在內置的“設置”程式中查看特定應用程式的設置。配置選項是一些使用者可能想要經常改變的值，比如在列表中顯示的分類的類型；配置選項應該由應用程式本身提供。

您應該把設置和配置選項看作是互斥的。也就是說，您不應該在您的應用程式中同時提供設置和配置選項。

iPhone應用程式最好不要求使用者指定任何設置。這樣使用者無需提供設置資訊就可以立即使用這些應用程式。為了在您的應用程式中實現這一點，您可以採用以下這些設計決策：

- 使您的解決方案滿足80%使用者的需求。當您採用這樣的設計時，大多數使用者不需要提供設置資訊，因為您的應用程式已經按照大部分使用者所預期的行為進行了設置。可以不考慮那些只有少數使用者需要的功能和大部分使用者僅需使用一次的功能。
- 從其他途徑獲取盡可能多的資訊。如果您可以用到任何使用者在內置應用程式或裝置設置中提供的資訊，那麼您可以向操作系統查詢這些資訊的值；不要讓使用者再次輸入它們。
- 如果您必須向使用者請求設置資訊，請在應用程式內部提示使用者輸入資訊。然後立即將這一資訊存儲在您的應用程式的設置中。這樣，使用者在開始使用您的應用程式之前，就不會被迫退出應用程式而先打開“設置”介面。如果使用者稍後需要更改這些資訊，他們可以隨時在您的應用程式的設置中進行更改。

如果使用者想要打開“設置”程式，就必須首先退出您的應用程式，而您應該不鼓勵使用者採取這樣的行為。系統並沒有提供支援這一行為的圖示或控制元件，而且建議您也不要為此建立自定義的圖示或控制元件。如果您決定一定要在您的iPhone應用程式中提供設置，請參考iPhone應用程式程序指南中的“設置程式包”一節，瞭解如何在您的程式碼中實現對這些功能的支援。

注意：應用程式特定的設置不應該包括使用者幫助的內容。

與設置不同，由於使用者會選擇從新資源或以不同佈局來查看資訊，因此配置選項很可能經常發生改變。對於這些選項所做的更改，您可以動態地作出響應，因為使用者不需要離開您的應用程式來訪問它們。

您可以在主使用者介面或螢幕的背面提供配置選項。具體使用哪一種技術更合理取決於該選項代表的是不是主要功能以及使用者對其進行更改的頻率。

例如，“日曆”程式允許使用者以日，星期和月為單位查看他們的時間表。這些選項本來可以在螢幕的背面提供給使用者，但是查看日曆的不同部分是程式的主要功能，並且使用者可能會頻繁改變所關注的焦點。

再舉一個例子，“天氣”程式的主要功能是顯示一個城市的當前天氣狀況和6天之內的天氣預報。雖然讓使用者能夠選擇是以攝氏還是華氏為單位顯示溫度也很重要，但是使用者不太可能經常改變這個選項，因此，將它放在主使用者介面中並不合理。在“天氣”螢幕的背面提供溫標選項，就顯得既方便又不突兀。

支援拷貝和貼上

iPhone OS提供了編輯（或剪貼簿）選單，它支援在文字視圖，**Web**視圖和圖像視圖中的“剪切”，“拷貝”，“貼上”，“選擇”和“全選”操作。一種向使用者顯示該選單的方法是，首先使用者觸摸並按住裝置螢幕直到出現放大的視圖（該視圖允許使用者將插入點或選擇點行動到正確的位置），然後放開。如果當前的上下文支援這種選單，則當使用者抬起手指時它就會出現。選單中的“選擇”操作可以選擇視圖中的單詞或應用程式定義的項目。使用者可以通過拖動當前所選區域的把手來擴大他們選擇的內容。在內容被選中之後，選單就會根據情況顯示“剪切”，“拷貝”或“貼上”。

您可以調整編輯選單的某些行為以適應您的應用程式。（要瞭解更多關於如何用編碼實現這些行為的資訊，請參考**iPhone**應用程式程序指南中的“拷貝和貼上操作”一節。）例如，您可以指定選單中顯示的操作，您可以改變選單出現的位置。但是，您不能控制選單本身的顏色和形狀。

編輯選單中可見的操作在當前上下文都是有意義的。例如，如果沒有任何內容被選中，選單中不會包含“拷貝”或“剪切”操作，因為這些操作要作用於被選中的內容。同樣，如果有內容被選中，選單則不包含“選擇”操作。如果您要在一個自定義視圖中支援編輯選單，您應該確保選單所顯示的操作適用於當前上下文。請注意，您不能在選單中顯示自定義的操作。

UIKit會根據可用空間的大小，在插入點或被選中內容的上方或下方顯示編輯選單，並放置選單指針，以便使用者可以看到選單操作是如何與當前內容相關聯的。您可以通過程序的方式，在選單出現之前決定它的位置，因此，在必要的情況下，您可以防止應用程式的使用者介面中的重要部分被選單遮住。

請注意，雖然“觸摸並按住”這個操作是向使用者顯示編輯選單的主要方式，但是使用者也可以通過雙擊文字視圖中的一個單詞來選中它，同時顯示出選單。如果您要在自定義視圖中支援選單，那麼您應該對這兩種操作都做出響應。此外，您可以定義在使用者雙擊時預設被選中的對象。

如果一個按鈕執行的是編輯選單中已有的操作，則要避免建立這樣的按鈕。例如，要讓使用者執行拷貝操作的話，使用編輯選單比向使用者提供“拷貝”按鈕更好，因為使用者會很困惑，為甚麼在您的應用程式中有兩種方式來完成同樣的事情。

您可以啓用對靜態文字的選擇，但是應該僅當靜態文字顯示對使用者有用內容時才這樣做。例如，使用者有可能想要拷貝一幅圖像的標題，但是他們可能不想拷貝一個標籤項或螢幕標題的標籤，比如“帳戶”。在文字視圖中，預設應該是按單詞選取的。

按鈕的標題應該是不可以被選中的，因為很難不觸發按鈕的點選事件並顯示出編輯選單。一般來說，俱有按鈕行為的元素不需要被選中。

如果您在自己的應用程式中支援“剪切”，“拷貝”和“貼上”操作，您也應該支援還原和重做（在“支援還原和重做”介紹）。這是因為編輯選單不需要在動作執行前進行確認，而且如果使用者改變了主意，他們總是希望能夠還原最近的操作。

支援還原和重做

iPhone OS在文字視圖中為使用者提供了還原和重做的能力。使用者通過搖晃裝置發起還原動作，裝置會顯示一個警告，允許使用者還原他們剛才的輸入，重做先前未完成的輸入或取消還原操作。

UIKit允許您在自己的應用程式中以一種更通用的方式支援還原（有關如何在程式碼中實現這一行為的資訊，請參考還原架構）。您可以指定：

- 使用者可以還原和重做的操作
- 您的應用程式應該何時將一個搖晃事件理解為觸發還原
- 支援多少層的還原

為了給您的應用程式中的還原和重做功能提供出色的使用者體驗，您應該：

- 提供簡短的描述性的語言精確地向使用者描述他們正在還原或重做的內容。UIKit自動採用字符串“還原”和“重做”作為還原警告按鈕的標題，但是您需要提供一至兩個單詞來描述使用者可以還原和重做的動作。（請注意，“取消”按鈕不能被改變。）例如，您可以提供文字“刪除姓名”或“地址變更”來建立如“還原刪除姓名”或“重做地址變更”這樣的按鈕標題。一定要避免提供過長的文字：過長的按鈕標題會被截斷，並且使用者很難解讀。此外，由於文字段位在按鈕的標題中，因此要使用標題式的大寫方式，而且不要新增標點。（簡單地說，標題式的大寫方式是指每個單詞都大寫，但冠詞，並列連詞和四個或四個字母以下的介詞除外。）
- 避免重載搖晃操作。即使您可以通過程序來制定您的應用程式何時將一個搖晃事件理解為觸發還原，但是如果搖晃可以用來執行另一個動作，使用者會感到迷惑。搖晃操作是使用者期望的發起還原和重做的主要方式，但是在適當情況下，您也可以在導覽欄中包含系統提供的“還原”和“重做”按鈕。如果在您的應用程式環境中，明確地顯示一個專用按鈕來執行這些功能非常重要，則您可以採取這種做法，但這種情況很不常見。
- 考慮您允許還原和重做的動作所處的環境。一般來說，使用者希望他們所做的更改和執行的動作能夠立即生效。還原和重做功能應該盡可能明確地關聯到使用者當前所處的環境，而非先前的環境。

啓用推送通知

當您的應用程式註冊了“蘋果推送通知服務”時，您可以在有新數據到來時向使用者發出警告，即使您的應用程式沒有運行。當裝置收到的消息是發給一個沒有運行的應用程式時，它可以通過以下方式通知使用者：

- 在應用程式的主螢幕圖示上更新一個 標記
- 播放警告聲音
- 顯示一條警報消息

或者您可以組合使用以上方式。使用者的反應可能是啓動應用程式來處理新數據，或者僅僅是注意到有新數據到來就可以了。（要瞭解如何在程式碼中處理推送通知，請閱讀蘋果推送通知服務程序指南。）

注意：推送通知的投遞是無保證的。此外，使用者也可以拒絕接收系統範圍內的通知。推送通知的目的是提醒使用者有新數據到達，而不是向您的應用程式傳遞關鍵的數據。

內置的設置程式中的“通知”部分為每一個註冊了“蘋果推送通知服務”的應用程式提供推送通知的設置。針對每一個應用程式，iPhone OS都可以讓向使用者設置是否允許標記，聲音和警報消息。

您應該花一些時間來思考哪種類型的事件更能讓通知引起使用者的注意。通知應該向使用者提供有用的，可操作的資訊，這些資訊是使用者即使在沒有使用您的應用程式時也想要得到的。

當您確定了使用者可能關心的事件之後，您還應該讓使用者決定每種事件應該產生甚麼類型的通知（如果有通知的話）。如果使用者無法定制您的應用程式的推送通知，那麼使用者可能會被他們不感興趣的通知所打擾。

使用者可以選擇他們想要接收的通知的類型，因此以下三種類型您應該全部支援：

- 標記。 標記是一種對使用者打擾最小的方式，它告訴使用者有新的他們可能感興趣的內容出現。標記是一個紅色的小橢圓形，出現在主螢幕圖示的右上角。您對於標記的外觀沒有任何控制權，它僅包含數字，不包含字母和標點符號。標記適用於告訴使用者有多少項有待他們查閱。例如，標記中的數字表示的可能是未讀的消息數，新分配的任務數，或當前有多少個遠程玩家正在進行遊戲。
- 聲音。 您可以提供自定義的警報聲音，也可以使用內置的警報聲音。如果您建立了自定義的聲音，一定要保證它簡短，獨特並且製作專業。（要瞭解有關自定義聲音的技術要求，請參考蘋果推送通知服務程序指南中的“準備自定義警報聲音”一節。）請注意，當有通知被投遞時，您不能強制使裝置振動；使用者能夠控制收到警報時是否伴有振動。如果通知到達本身就為使用者提供了足以採取行動的資訊，在這種情勢下，採用一種容易辨識的聲音是非常適合的。例如，一個協同任務管理

系統在成員的任務完成時可能會伴隨著一段獨特的聲音。僅僅是聽到這種聲音，使用者就知道任務已經完成了。

- 警告。警告是一種通知使用者有新內容時最打擾使用者的一種方式。在警告的頂端顯示您的應用程式的名稱，在它下面是您發送的消息，在警告底部有一至兩個按鈕。如果您指定了兩個按鈕，則警告會在左側顯示“關閉”按鈕，右側顯示“查看”按鈕（使用者點選“查看”按鈕可以在解除警告的同時啓動您的應用程式）。如果您只指定了一個按鈕，則警告只顯示一個“確定”按鈕。“關閉”按鈕和“確定”按鈕都會關閉警告而不會打開您的應用程式。警告會打斷使用者的工作流程，因此最好謹慎地使用它，並且只用它來投遞有關某事件的簡短的，重要的消息。特別地，一定要避免在您的警告消息中包含任何廣告內容。

保證應用程式的可用性

一個易於使用的應用程式應該允許有障礙的使用者在輔助程式或裝置的幫助下可以成功使用。iPhone OS裝置包含許多功能，使所有使用者都可以更加方便地使用該裝置，比如可視化語音郵件，縮放以及語音控制功能。您無需在應用程式中採取任何動作，使用者可以直接獲益於這些功能。

有了VoiceOver，事情就變得不一樣了。VoiceOver是蘋果公司一項創新的螢幕閱讀技術，它讓使用者無需看到螢幕，就可以控制他們的裝置。為了確保VoiceOver使用者可以充分地使用您的應用程式，您可能需要提供一些關於使用者介面中視圖和控制元件的自定義資訊。

幸運的是，在預設情況下，UIKit控制元件和視圖是易於訪問的，因此，當您以完全標準的方式使用這些標準元素時，您只有很少的額外工作要做（如果有的話）。使用者介面的自定義程度越高，您需要提供的自定義資訊就越多，以便VoiceOver可以正確地向俱有視覺障礙的使用者描述您的應用程式。

重要：為了使您的應用程式易於訪問，您要做的工作包括為VoiceOver提供它所需的資訊來幫助使用者使用您的應用程式。您不需要為了適應VoiceOver而改變使用者介面的視覺設計。

讓您的iPhone應用程式易於被VoiceOver使用者訪問是非常正確的做法。這種做法還可以增加您的使用者群，並有可能幫助您滿足由各主管機構建立的可用性準則。

提供搜尋功能並顯示搜尋結果

UIKit提供了搜尋欄控制元件，您可以使用它顯示一致的啓動搜尋的介面，但要您需要在您的應用程式中實現搜尋功能。（要瞭解有關搜尋欄的更多資訊，請參考“搜尋欄”；要瞭解有關在程式碼中處理搜尋結果的更多資訊，請參考**UISearchDisplayController**類參考。）為了確保搜尋擁有實用而方便的使用者體驗，請花一些時間考慮如何實現搜尋過程以及如何顯示其結果。

一般來說，您應該：

- 為您的數據建立索引，以便隨時進行搜尋。
- 即時過濾本端的數據，一旦使用者開始輸入，您就顯示結果，並且隨著使用者繼續輸入而逐步縮摘要果範圍。
- 如果可能的話，在使用者輸入時也同時過濾遠程數據，但是，如果這部分的響應時間有可能將搜尋結果的計算推遲1-2秒鐘以上，一定要經過使用者的允許。
- 在列表上面顯示搜尋欄或者在列表內顯示索引。
- 避免為搜尋打開一個標籤頁，除非它是您應用程式中的主要功能，應該被標識為一個不同的模式。

雖然即時過濾數據通常能夠產生出色的使用者體驗，但這並不總是可行的。如果無法即時過濾數據，您可以在使用者在鍵盤上點選“搜尋”按鈕之後再開始搜尋過程。如果您要這樣做，一定要提供有關搜尋進度的回饋資訊，以便讓使用者知道搜尋過程沒有停止。一種方法就是盡快顯示文字結果，並為那些可能需要更長時間檢索的數據顯示佔位符內容。

例如，在YouTube中，使用者點選“搜尋”按鈕發起視訊的搜尋。如果網路連接速度很慢，YouTube會先顯示“載入中……”消息和旋轉的活動指示符，讓使用者知道搜尋正在進行。然後，YouTube會顯示一個結果列表，其中，每一行填寫搜尋的文字結果（比如視訊的標題和收視率），以及帶有虛線輪廓的立方體自定義圖像。隨著使用者瀏覽視訊標題的列

表，下載完的視訊縮略圖會逐步替換掉原來的虛線立方體。像這樣，在更多的數據仍在下載時向使用者顯示部分搜尋結果，能夠及時地為使用者提供有用的資訊。

如果您處理的數據可以歸類於多個不同的類別，您可以提供一個範圍欄。範圍欄包含至多4個範圍按鈕，每個按鈕代表一種分類。例如，“郵件”程式提供了一個範圍欄，允許使用者將他們的搜尋集中在郵件的發件人，收件人或主題字段，或者將搜尋範圍擴大至包含所有的字段。如果範圍欄能夠幫助使用者集中他們的搜尋，或者能夠大大減小搜尋結果的數量，請您考慮使用範圍欄控制元件。（要瞭解如何在您的程式碼中實現範圍欄，請參考[UISearchBar類參考](#)。）

使用使用者的位置資訊

使用者喜歡能夠自動使用他們的物理位置對內容進行標記的功能，或者查找當前在附近的朋友。使用者同時也希望當他們不想與他人分享自己的位置時能夠禁用這些功能。使用者可以通過“設置”>“一般”中的“位置服務”設置來選擇同意（或拒絕）系統範圍內對他們的物理位置的訪問。

如果使用者關閉了位置服務，而隨後使用的應用程式功能需要獲知他們的位置，則使用者會看到一個警告，此警告告訴他們必須改變他們的首選項設置才能使用此功能。該警告不允許使用者在應用程式的內部做此更改；相反，他們必須進入設置應用程式改變他們的首選項設置。這樣可以確保使用者充分意識到他們正在授予整個系統使用他們位置資訊的權限。

為了讓使用者知道他們為什麼要打開位置服務，您最好只在使用者試圖使用一項顯然需要獲知他們當前位置的功能時，才顯示警告。例如，當位置服務關閉時，使用者仍然可以使用地圖應用程式，但是，當他們訪問發現並跟蹤其當前位置的功能時，會看到警告。

如果位置服務處於關閉狀態，iPhone OS會在您的應用程式第一次試圖訪問位置資訊時顯示警告。[Core Location框架](#)為您提供了一種獲取使用者偏好設置的方法，使您避免不必要的或不適當地觸發警告。（要瞭解關於這個程序接口的更多資訊，請參考[Core Location框架參考](#)。）

知道了使用者的偏好設置資訊，您就可以盡可能準確地為需要位置資訊的功能觸發警告，或是完全地消除警告。

- 如果您的應用程式在沒有這些資訊的情況下無法執行它的主要功能，您最好在使用者啓動應用程式時盡快讓他們看到警告。使用者不會為此感到困擾，因為他們明白應用程式的主要功能依賴於知曉他們的位置。
- 如果使用者的位置不是您應用程式基本功能的一部分，您可以選擇簡單地限制那些用到位置資訊的功能。例如，當位置服務關閉時，相機應用程式會自動關閉將使用者的位置新增到他們所拍攝的照片的功能。但應用程式並不會阻止使用者拍照，除非他們改變偏好設置的選項，這是因為“將位置資訊新增到照片”只是一個附加功能，而不是基本功能。
- 如果某項功能需要位置資訊才能工作，一定要避免在使用者實際選擇該功能之前執行任何程序調用觸發警告。（獲取使用者偏好設置資訊的調用不會觸發警告。）這樣，您就可以避免讓使用者感到奇怪，為甚麼您的應用程式在使用者做一些看似不需要位置資訊的事情時，想要得到他們的位置資訊。

處理方向的變化

使用者可以隨時旋轉iPhone OS裝置，並且他們期望正在瀏覽的內容做出適當的調整。在您的iPhone應用程式中，請務必：

- 注意加速度表的值（關於加速度表和加速度表程序接口參考的更多資訊，請閱讀iPhone應用程式程序指南）。如果合適的話，您的應用程式應該對所有的裝置方向變化做出響應。
- 如果您的應用程式使用者介面的某一部分只在一個方向上顯示內容，則該區域只適於在此方向上出現，而且不需要對裝置方向的變化做出響應。例如，當使用者選擇一個iPod視訊來觀看時，無論當前裝置方向如何，該視訊都橫向顯示。這向使用者表明，要旋轉裝置以便更好地觀看該視訊。該例中最重要的一點是，iPod沒有提供“旋轉”按鈕；相反，使用者知道要旋轉裝置，因為視訊是橫向顯示的。讓使用者旋轉裝置來正確地瀏覽應用程式使用者介面中需要特定方向的部分。避免建立新的控制元件或定義新的操作，告訴使用者旋轉裝置。
- 利用一步操作就能改變方向的過程，完成更順暢且往往更快的旋轉。但是，如果您的螢幕佈局非常複雜，當發生方向變化時，您可以選擇執行一種淡入淡出的轉換。要瞭解如何在您的程式碼中支援一步操作過程，請參考UIViewController類參考。

- 使用者經常因為想要“看到更多”而將他們的裝置旋轉為橫向。如果您只是按比例放大螢幕的內容，則無法滿足使用者的期望。相反，您應該重新打包文字行，而且如果必要的話，重新安排使用者介面的佈局，以便更多的內容填充到螢幕當中。

使用聲音

使用者期望iPhone OS裝置俱有非常美妙的聲音，無論是操作系統的聲音（比如鈴聲和警告聲），還是應用程式的聲音（比如媒體播放，環境聲音和配樂）。此外，使用者還希望裝置發出的聲音能夠遵從他們的偏好和目的。

使用者決定聲音的音量，以及他們是否想要聽到這些聲音。但是有些時候，即使當前的設置表明使用者更傾向於靜音，他們還是希望聽到某些聲音。例如，使用者總是期望聽到他們設置的警告聲。從本質上講，使用者想要聽到他們期待聽到的聲音，而不願聽到他們不期待的聲音。

為了幫助您順應這樣的需求，iPhone OS提供了一些程序接口，您可以用來：

- 描述您應用程式的聲音應該如何與裝置上的其他聲音保持一致。
- 確保應用程式的聲音能夠按照使用者的期望進行播放。

在您決定如何處理應用程式中的聲音之前，您需要瞭解，當使用者調整裝置控制元件和使用外部裝置（如耳機和耳麥）時，他們期望應用程式和裝置如何運作。

震動/靜音切換—使用者的期望

如果使用者希望做到以下幾點，他們可以使用“震動/靜音”切換將他們的裝置靜音：

- 避免被意外的聲音打擾，比如電話鈴聲和來信提示音。
- 避免聽到使用者操作的附帶聲音，比如鍵盤或其它回饋的聲音，偶然的聲音或應用程式啓動的聲音。
- 避免聽到遊戲的聲音，包括附帶的聲音和配樂，它們並不是使用遊戲程式所必需的。

例如，在劇場中，使用者會將他們的裝置切換至靜音狀態，以免打擾到劇場中的其他人。在這種情況下，使用者仍然想要使用他們裝置上的應用程式，但他們不想被不期望的或沒有明確要求的聲音嚇到，比如鈴聲或新消息提示音。

但是，對於旨在產生聲音的使用者動作，“震動/靜音”切換不會消除它們產生的聲音。例如：

- 媒體應用程式中的媒體播放不會被“震動/靜音”切換靜音，因為媒體播放是使用者明確請求的。
- 時鐘應用程式的警告不會被“震動/靜音”切換靜音，因為此警告是使用者明確設置的。
- 語言學習程式中的音效素材不會被“震動/靜音”切換靜音，因為使用者採取明確行動想要聽到它。
- 語音聊天程式中的會話不會被“震動/靜音”切換靜音，因為使用者啓動此類應用程式的唯一目的就是進行語音聊天。

這種行為遵循使用者控制的原則，因為是由使用者（而不是裝置）來決定聽到使用者明確請求的聲音是否合適。

音量按鈕—使用者的期望

使用者使用裝置的音量按鈕來調節裝置播放的所有聲音的音量，包括歌曲，應用程式的聲音和裝置的聲音。這意味著使用者可以隨時使用音量按鈕關閉任何聲音，無論“震動/靜音”切換的當前狀態如何。

在某些情況下，應用程式適宜在其介面為使用者提供音量設置功能。例如，YouTube顯示了一個音量滑動器，使用者可以用它調整正在觀看的視訊的音量。儘管YouTube正在運行，使用者可以交替使用此滑動器和音量按鈕來調整視訊的音量。這是因為在應用程式運行時，滑動器起到音量按鈕代理的作用：滑動器同時作用於應用程式的音量和整個系統的音量（鈴聲音量除外）。

如果您需要顯示音量滑動器，當您使用MPVolumeView類時一定要使用系統提供的滑動器。請注意，如果當前激活的音訊輸出裝置不支援音量控制（比如A2DP裝置），音量滑動器將被相應的裝置名稱取代。

使用音量按鈕調整應用程式當前播放的音訊，也會同時調整整個系統的音量（鈴聲音量除外）。（在當前沒有播放任何音訊時，使用音量按鈕調整鈴聲的音量。）

這種行為遵循使用者控制的原則，因為使用者可以隨時決定裝置發出的聲音應該有多大。

有時候，應用程式可能需要調整相對和絕對音量級，以便在其音訊輸出中產生最佳的混合。但是，最終的音訊輸出的音量應該始終受到系統音量的控制，無論它是通過音量按鈕還是音量滑動器進行調整的。這就意味著，對應用程式音訊輸出的控制仍然掌握在它所歸屬的使用者的手中。

耳機和耳麥—使用者的期望

使用者插入耳機和耳麥就可以獲得私人的聲音體驗並且解放他們的雙手。在使用和不使用附件的情況下，使用者對應用程式的行為有著不同的期望。

當使用者插入耳機和耳麥時，他們是打算繼續聽當前的聲音，只是轉為私下收聽。因此，他們希望當前正在播放音訊的應用程式繼續播放此音訊。

當使用者拔下耳機和耳麥時，他們不想自動將正在收聽的內容分享給他人。因此，他們希望當前正在播放音訊的應用程式暫停播放，讓他們準備好之後顯式地重新開始播放。

無線音訊—使用者的期望

使用者非常喜歡無線耳機的便捷，比如藍牙A2DP裝置。人們使用無線耳機和耳麥的理由與使用有線耳機和耳麥的理由是一樣的：他們想要私下聽到聲音，並希望解放他們的雙手。

使用者對無線耳機的使用者體驗也有著非常類似的期望：

- 當使用者連接到無線音訊裝置時，他們打算繼續聽到當前的聲音，只是轉為私下收聽。在這種情況下，他們希望音訊能夠繼續播放。
- 當使用者斷開無線裝置時（或者當裝置超出作用範圍或關閉時），他們不想自動將正在收聽的內容分享給他人。在這種情況下，他們希望暫停正在播放的音訊，讓他們準備好之後顯式地重新開始播放。

即使使用者沒有實際地插入或拔出無線音訊裝置，他們仍然希望能夠選擇一個不同的音訊通道。為瞭解決這個問題，iPhone OS自動顯示了一個控制元件，讓使用者選擇音訊輸出路線。由於選擇不同的音訊通道是使用者發起的動作，所以使用者希望正在播放的音訊繼續播放。

定義應用程式的音訊行為

如果聲音能夠增強使用者體驗或應用程式的功能，或者是使用者體驗或應用程式功能必不可少的一部分，您需要決定您的音訊應該如何與裝置的音訊環境保持一致，以及應該如何響應使用者的動作。例如，你需要決定：

- 當裝置鎖定或切換至靜音時，您的音訊是否應該繼續播放。
- 您的音訊是否應該與當前正在播放的其他音訊混合在一起（比如iPod中的歌曲）。
- 您的應用程式是否需要順序或並行地同時處理音訊輸入和輸出。
- 您的音訊是否應該在中斷後自動恢復播放。

要控制應用程式的音訊在這些情況下應該如何表現，請使用“音訊會話服務”或 AVAudioSession類。這些程序接口不能產生聲音；它們可以幫助您說明您的音訊應該如何與裝置上的音訊進行交互，以及如何響應中斷和裝置配置中的變化。音訊會話服務管理採用AV基礎框架，音訊隊列服務，OpenAL和I/O音訊單元等技術產生的聲音。

注意：如果您的應用程式僅需要產生功能附帶的使用者介面音效，您可以使用“系統聲音服務”。系統聲音服務是iPhone OS技術，用於產生警告聲音和使用者介面音效，以及振動；它不適用於任何其他目的，而且它產生的聲音不由“音訊會話服務”管理。使用此技術的示例請參考**SysSound**示例項目。

重要：無論您使用何種技術產生音訊，無論您如何定義它的行為，電話可以隨時中斷當前正在運行的應用程式。這是因為任何應用程式都不應該阻止使用者接聽來電。

音訊會話是您應用程式和系統之間的音訊中介。從使用者體驗的角度來看，音訊會話最重要的一個方面就是定義應用程式的音訊行為的類別。

為了提供良好的音訊使用者體驗，應選擇能最好地描述應用程式音訊的類別。一定要基於類別的語義做出選擇，而不是其行為的確切集合。這將確保您的應用程式能夠按照使用者

的期望運轉。此外，如果日後該類別的行為集合被重新修訂，它也能最大限度地保證您的應用程式正常工作。

在極少數情況下，您可能需要通過為音訊會話新增屬性，來增強或改進某個類別的標準行為。例如，您可以新增kAudioSessionProperty_OtherMixableAudioShouldDuck屬性，以確保您應用程式的音訊比所有其他音訊（電話音訊除外）更響亮。如果能夠在其他音訊播放的同時聽到您應用程式的音訊對使用者來說很重要的話，您可以這樣做。但是，您應該注意，一個類別的標準行為代表了大多數使用者的期望，所以您應該在新增屬性完善此行為之前，認真仔細地考慮一下。要瞭解有關音訊會話屬性的更多內容，請參考音訊會話程序指南中的“微調類別”一節。

您可以根據裝置當前的音訊環境選擇您的類別。舉個例子，如果使用者可以在收聽其他音訊（不是您提供的配樂）的同時使用您的應用程式，您可能想要這樣做。如果這對於您的應用程式來說行得通，一定要避免強迫使用者停止收聽他們的音樂，或是在您的應用程式啓動時強迫使用者做出明確的配樂選擇。要瞭解如何做到這一點，參考“摘要”中的場景2。

當應用程式正在運行時，您也可以改變音訊會話的類別，雖然很少有必要這樣做。這樣做的主要原因是，應用程式需要在不同的時刻支援錄音和播放。在這樣的應用程式中，更好的做法是根據需要在Record類別和Playback類別之間進行切換，而不是選擇Play和Record類別。這是因為選擇Record類別，會使警告（比如來信警告）在錄音正在進行時沒有聲音。

表4-1列出了您可以使用的音訊會話類別。iPhone OS預設為音訊會話分配了Solo Ambient類別。

注意：由於空間有限，表4-1只顯示了每個類別名稱的最後一部分。每個類別的實際符號名稱均以AVAudioSessionCategory開始。例如，MixWithOthers屬性的實際符號名稱是kAudioSessionProperty_OverrideCategoryMixWithOthers。

表 4-1 您可以用來在應用程式中定義聲音行為的音訊會話類別

類別	含義	通過“震動/靜音”切換置為無聲並鎖定	與其他音訊混合

SoloAmbient	增強應用程式功能的聲音，應該將其他音訊靜音	是	否
Ambient	增強應用程式功能的聲音，但不應將其他音訊靜音	是	是
Playback	對應用程式功能來說必不可少的聲音，可以與其他音訊混合	否	否（預設） 是（當新增Mix-WithOthers屬性時）
Record	使用者錄制的音訊	否	否
PlayAndRecord	代表音訊輸入和輸出的聲音，順序地或並行地	否	否（預設） 是（當新增Mix-WithOthers屬性時）
Audio-Processing	執行輔助硬體的音訊編碼的應用程式（它不播放或錄音）	-	否

摘要

下面是一些場景，它們說明瞭如何選擇音訊會話類別，以提供使用者期望的音訊體驗。

場景 1. 假設您正在開發一個教育應用程式，幫助人們學習一門新的語言。您需要提供在使用者點選特定控制元件時播放的回饋聲音；並提供在使用者想要聽到正確的發音示例時播放的單詞和短語的錄音。

在這個應用中，聲音對於應用程式的主要功能來說是必不可少的。人們使用該應用程式，收聽他們所學語言中單詞和短語的發音，所以，即使當“震動/靜音”切換設置為靜音或裝置鎖定時，也應該播放應用程式的聲音。由於使用者需要清楚地聽到發音，因此，他們希望其他正在播放的音訊被靜音。

為了產生使用者期望的音訊體驗，您應該使用Playback類別。雖然您可以改進這一類別，以便與其他音訊（如表4-1中所述）進行混合，但是這個應用程式應該使用預設的行為，以確保其他音訊不會與使用者明確選擇要收聽的學習內容發生競爭。

場景 2. 假設您正在開發一個遊戲，讓使用者控制螢幕上的人物完成許多不同的任務。您需要提供各種各樣的遊戲音效和一段遊戲配樂。

在這個應用中，聲音會大大提升使用者的體驗，但它並不是主要任務必不可少的一部分。此外，使用者很可能希望能夠在靜音狀態下玩遊戲，或者一邊聽音樂庫中的歌曲（而不是遊戲的配樂）一邊玩遊戲。

最好的策略是，要瞭解當您的應用程式啓動時使用者是否正在收聽其他音訊。不要讓使用者選擇是否想要聽其他的音訊或您應用程式的配樂。相反，要使用“音訊會話服務”的AudioSessionGetProperty功能，查詢kAudioSessionProperty_OtherAudioIsPlaying屬性的狀態。根據查詢的結果，您可以選擇Ambient類別或Solo Ambient類別（這兩個類別都允許使用者在靜音狀態下玩遊戲）：

如果使用者正在收聽其他音訊，您應該假設他們想要繼續收聽，而不想被迫收聽遊戲的配樂。在這種情況下，您應該選擇Ambient類別。

如果當您的應用程式啓動時，使用者沒有收聽任何其他音訊，應選擇Solo Ambient類別。

場景 3. 假設您正在開發一個應用程式，為使用者提供準確，即時的到達所選目的地的導覽指示。您需要為行程中的每一步提供語音指導，以及一些回饋聲音。此外，您認為使用者希望在使用應用程式的同時，能夠聽到他們自己的音訊。

在這個應用中，語音導覽指示代表了程式的主要任務。基於這個原因，您應該使用Playback類別，它讓您的音訊在裝置鎖定或“震動/靜音”切換設置為靜音時仍能播放。

為了讓人們在使用您應用程式的同時，收聽其他的音訊，您可以新增kAudioSessionProperty_OverrideCategoryMixWithOthers屬性。但是，您也想要確保使用者可以在當前正在播放的音訊之上，聽到應用程式的語音指令。要做到這一點，您可以將kAudioSessionProperty_OtherMixableAudioShouldDuck屬性應用到音訊會話。這可以確保您的音訊比目前播放的所有音訊（除了電話音訊）更加響亮。

場景 4. 假設您正在開發一個博客應用程式，允許使用者向中心網站上傳他們的文字和圖片。您可能有一個簡短的啓動聲音文件，各種各樣簡短的聲音效果（比如當使用者完成上載時播放的聲音），以及當上載失敗時播放的警告聲音。

在這個應用中，聲音會提升使用者的體驗，但它只是附加的。程式的主要任務與音訊無關，使用者不需要聽到任何聲音，也能成功使用該應用程式。在這種情況下，您可以使用“系統聲音服務”產生聲音。這是因為應用程式中所有聲音的音訊上下文都符合這一技術的目的，也就是要產生使用者期望的，遵從裝置鎖定和“震動/靜音”切換的使用者介面音效和警告聲音。

提供選項

iPhone OS包含一些幫助使用者做出選擇的元素。當您需要在應用程式中提供選項時，您應該使用這些選擇方法，因為使用者已經熟悉了它們的行為。一般來說，您不應該試圖拷貝在桌面電腦應用程式中看到的選擇控制元件的外觀和行為，比如應用程式選單或一組單選按鈕。iPhone OS提供了以下元素，您可以用來向使用者提供選項：

- 列表（即表格視圖）。使用者點選列表中的某一行選擇一項。列表幾乎適合於顯示任何數量的選項。有關在應用程式中使用表格視圖的方法的詳細資訊，請參考“表格視圖”。
- 選擇器，包括日期和時間選擇器。使用者轉動選擇器的轉輪，直到每個轉輪顯示出值的相應部分，比如包含年，月，日的日曆日期。要瞭解有關在您的iPhone應用程式中使用選擇器的更多資訊，請參考“日期和時間選擇器”和“選擇器”。
- 開關控制元件。使用者將開關控制元件從一側滑動至另一側，顯示出兩個值之一。開關控制的設計意圖是在列表的內部提供一個簡單的選項。有關開關控制元件的更多資訊，請參考“開關控制元件”。

提供許可協議或免責聲明

如果您隨同iPhone應用程式提供了終端使用者的許可協議（或EULA），App Store會顯示該協議，以便使用者在使用您的應用程式之前可以閱讀它。

如果可能的話，盡量避免要求使用者在第一次啓動您的應用程式時，表示他們同意您的終端使用者許可協議。這樣使用者能夠立即享用您的應用程式。但是，即使這是首選的使用者體驗，它可能無法在所有情況下都行得通。如果您必須在您的應用程式中顯示許可協議，請嘗試採用一種與您的使用者介面相一致的方式，這樣可以將給使用者造成的不便降到最低。

同樣，如果您需要提供免責聲明，一定要平衡好業務需求與良好的使用者體驗。如果可以的話，在您的應用程式描述或EULA中提供您的免責聲明，以便它可以用在App Store中。

設計您iPhone應用程式的使用者介面

iPhone OS中的使用者介面元素包括視圖和控制元件。視圖是提供了良好定義的功能集合的內容區域。控制元件則是能夠觸發即時動作或可視化結果的圖形對象。雖然一個應用程式的所有視圖和控制元件都包含在同一個應用程式視窗中，使用者卻是根據它們不同的可見狀態在螢幕中與之交互的。

iPhone OS定義了這些使用者介面元素的標準外觀，並提供了使用者所期望的行為。第二部分中的章節能夠幫助您瞭解可用的使用者介面元素的類型以及如何在您的應用程式的使用者介面中使用它們。

使用者介面概述

在深入研究視圖和控制元件的具體細節之前，先對這些元素在一起工作的方式以及使用者期望的行為有一個較高層次的理解，對您來說大有裨益。本章將介紹構成大多數應用程式的各種視圖，幫助您瞭解它們的具體資訊以及如何使用它們。

想要詳細瞭解每一個使用者介面元素的外觀，行為和使用指南，請務必閱讀本章後面的章節。理解每個使用者介面元素是如何針對它的用法進行設計的，能夠幫助您在應用程式中正確地使用它，並在適當的情況下根據您的需要定制使用者介面元素。

應用程式的螢幕及其內容

無論是甚麼類型的應用程式，都有一個應用程式視窗，該視窗為您提供了一個能夠呈現應用程式的所有資訊的背景。但是使用者對這個視窗沒有概念，他們對應用程式的體驗來自螢幕上的畫面，並且使用者通過螢幕對應用程式進行操作。

雖然不是一個程式中的結構，但是您仍可以認為每個螢幕對應於應用程式的不同的可視化狀態或模式。當使用者瀏覽資訊，切換標籤頁，或者點選“資訊”按鈕查看側邊彈出的配置資訊時，他們看到的是各個獨立的螢幕。

取決於程式的類型，您的應用程式的螢幕可能有多有少。例如，郵件應用程式可以顯示一個帳戶螢幕，各個帳戶中的郵箱列表螢幕，各個郵箱內容的螢幕和顯示一條消息的螢幕，還有一個寫郵件的螢幕。然而，股票應用程式只顯示兩個螢幕：一個螢幕顯示公司列表和股票走勢圖，另一個螢幕顯示應用程式的配置資訊。

通常，使用者會把應用程式螢幕和裝置螢幕當成是一回事。然而，應用程式螢幕的內容可能會超出裝置螢幕的邊界，這就需要使用者滾動螢幕。例如，電話應用程式中的聯絡人只有一個單獨螢幕，即使它的內容是裝置螢幕的好幾倍。

應用程式的螢幕可以包含各種各樣的視圖和控制元件的組合。某些視圖會包含一些特定的控制元件，而有些控制元件可以用於很多不同的視圖當中。

告警，動作表單和模式視圖和普通視圖不同，普通視圖一般存在於應用程式的螢幕中，而它們浮動於應用程式的螢幕及其視圖之上。更多有關這些視圖的資訊請參考“告警，動作表單和模式視圖”。

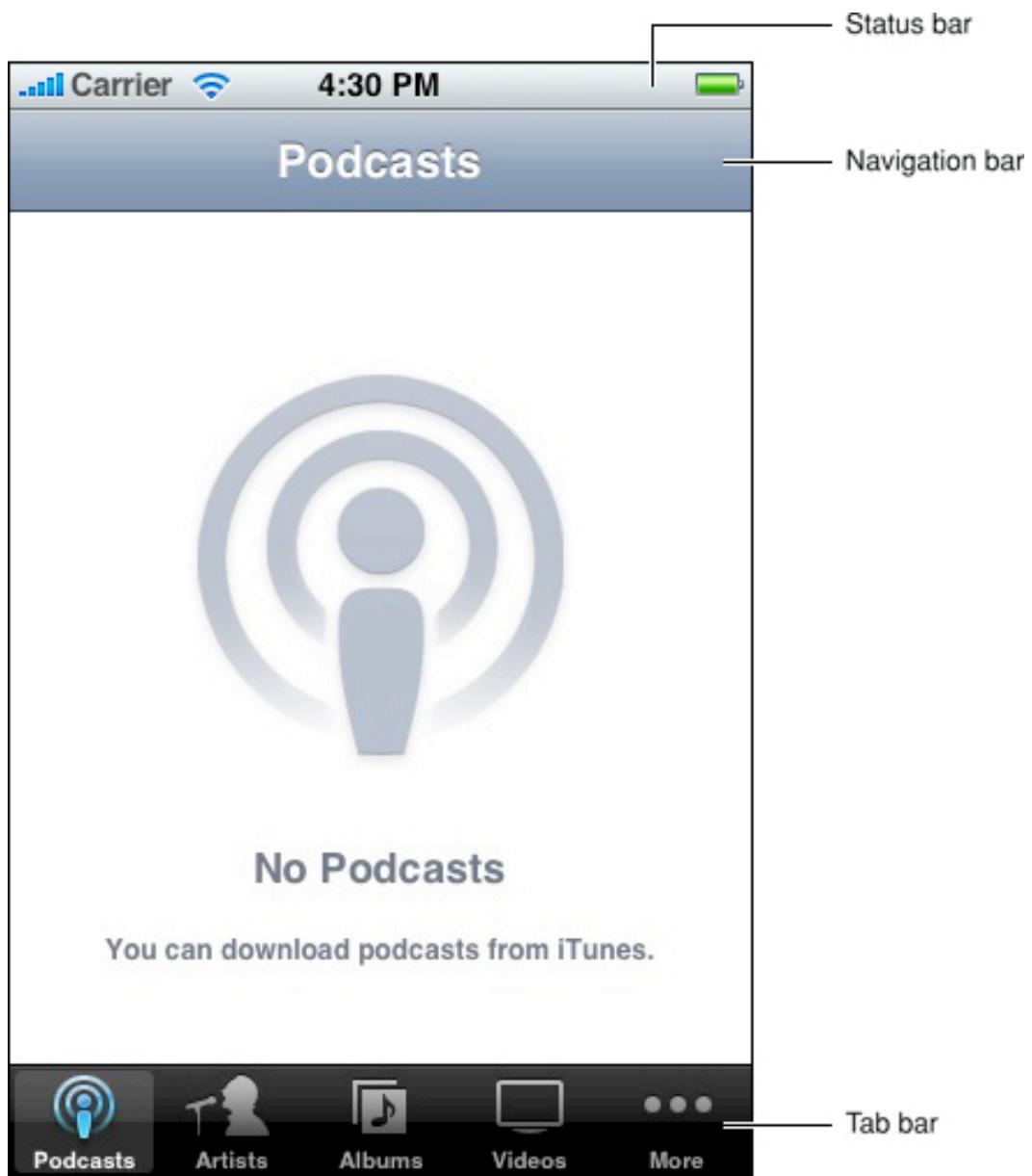
下面四種類型的視圖在應用程式的使用者介面中俱有特殊的地位，但是它們並不需要在每個應用程式中被包含或總是可見。

- 狀態欄。這是一個獨一無二的視圖，雖然應用程式可以在一定程度上自定義狀態欄的外觀，但從技術角度講，它並不屬於應用程式視窗的一部分。更多資訊請參考“狀態欄”。
- 導覽欄。這是個可選的視圖，它出現於狀態欄的下方，可以包括標題，按鈕和分段控制元件。更多資訊請參考“導覽欄”。
- 標籤欄。這是個可選的視圖，它位於螢幕底部的邊緣，能夠切換應用程式中不同的模式。更多資訊請參考“標籤欄”。
- 工具欄。這也是個可選的試圖，它同樣位於螢幕底部的邊緣，包含在應用程式的當前上下文中完成特殊動作的控制元件。更多資訊請參考“工具欄”。

圖5-

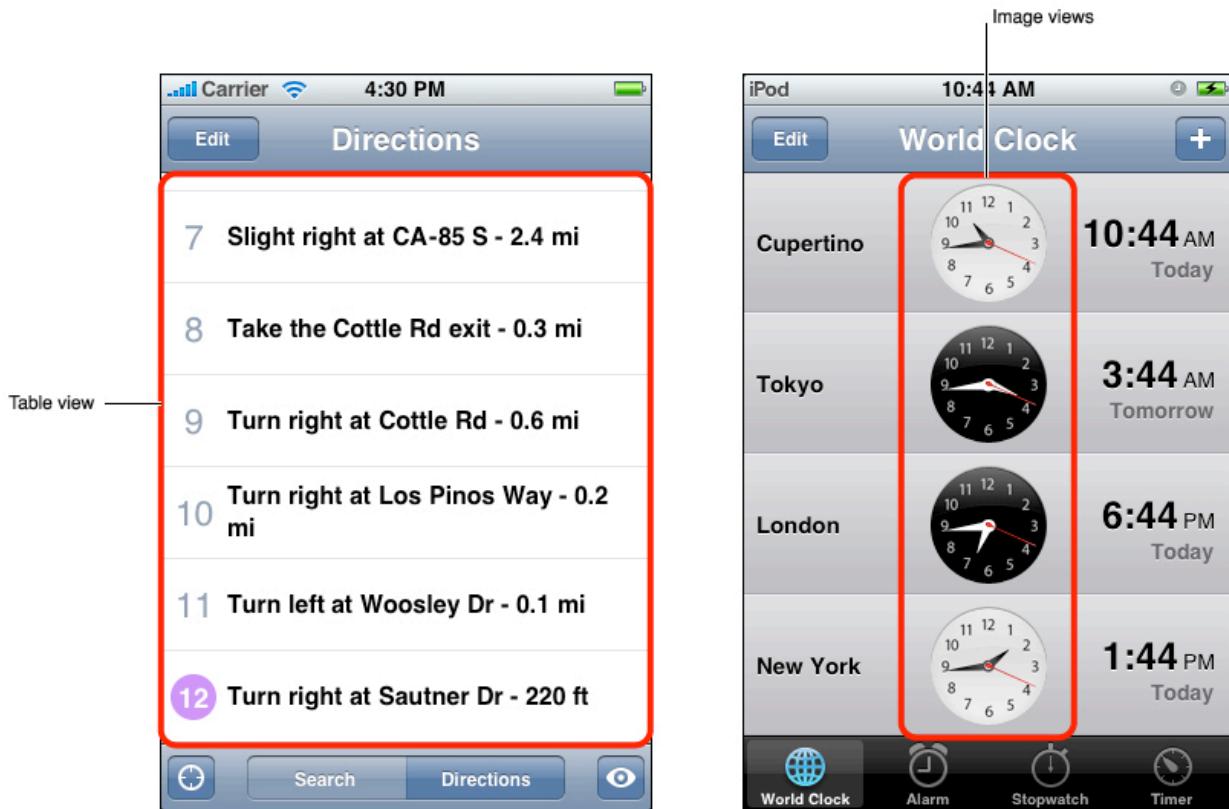
1在一個應用程式螢幕中顯示了這些視圖中的三種。請注意，如果該應用程式使用工具欄，它將出現在圖中的標籤欄處。

圖 5-1 一個包含狀態欄，導覽欄和標籤欄的應用程式螢幕



在一個能夠顯示這四種視圖的組合的應用程式中，您可以將導覽欄底部到工具欄頂部之間的區域視為內容區。在該區域中，應用程式螢幕可以包含任意視圖來顯示內容，如表視圖，web視圖和圖像視圖。圖5-2顯示了兩個在iPhone操作系統中可用的內容區視圖：表視圖的一種和圖像視圖。請參閱“表視圖，文字視圖和Web視圖”以瞭解更多有關這些視圖的行為和外觀，以及與其相關的控制元件。

圖 5-2 兩種類型的內容區視圖



正如上文說過的那樣，有一些控制元件只能在特定的視圖中使用。顯示提示符控制元件就是這樣的一個例子，它在表視圖中俱有特殊用法。您可以在圖8-1中的左邊的列表中看到顯示提示符（它看上去像是一個大於號>）。在介紹其相應視圖的小節對這些控制元件有具體描述，但是，需要補充一點，也有少量控制元件（如詳細顯示提示符）俱有更廣泛的用途。請參閱“應用程式控制元件”來獲得更多控制元件相關的資訊。

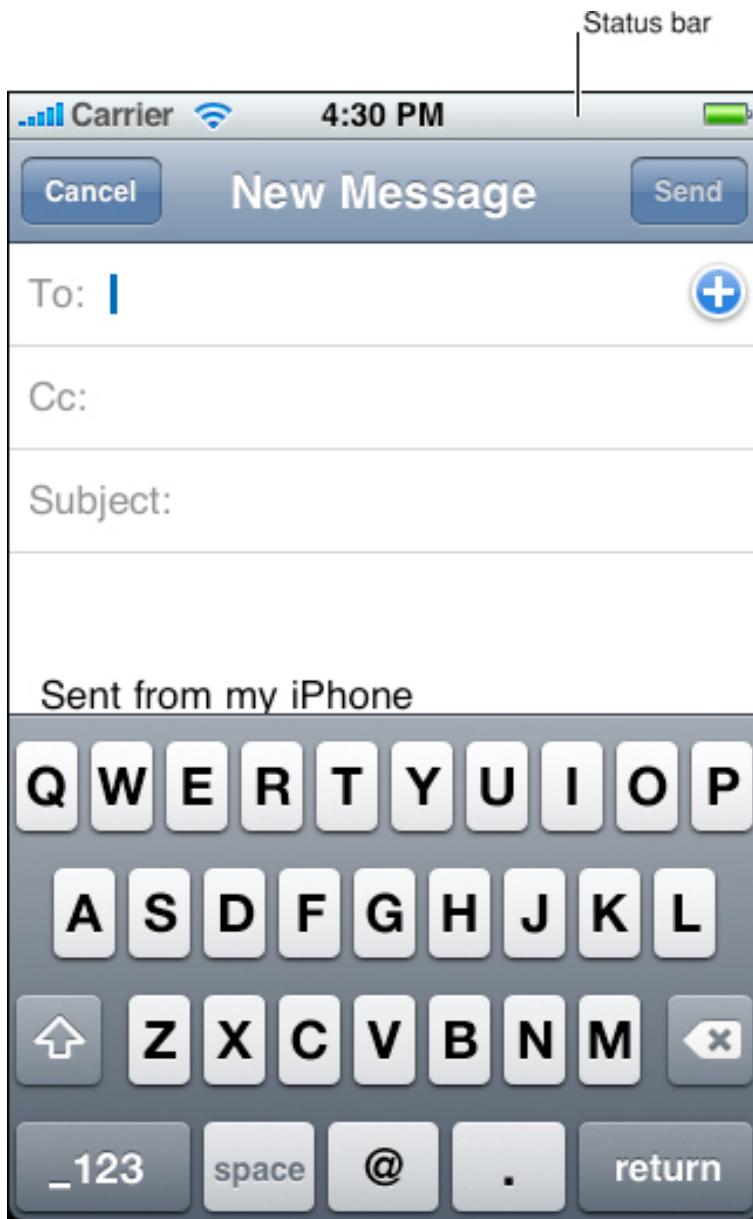
導覽欄，標籤欄，工具欄和狀態欄

狀態欄，導覽欄，標籤欄和工具欄在iPhone應用程式中俱有特別定義的外觀和行為的視圖。它們不需要在每個應用程式中都出現（在那些令人驚嘆的應用程式中經常是一個也不出現），不過一旦決定讓它們出現在應用程式中，就一定要正確地使用這些視圖。原因在於使用iPhone OS裝置的使用者都熟悉它們所提供的功能，對它們顯示的資訊和執行的功能類型早已習慣。

狀態欄

狀態欄顯示與使用者裝置相關的重要資訊，包括電話信號強度，當前網路連接和電量資訊。圖6-1顯示了一個狀態欄的例子。

圖 6-1 一個包含使用者所需的重要資訊的狀態欄



儘管一個全屏的應用程式可以隱藏狀態欄，但是您應該仔細考慮這一設計方案。人們總是希望可以看到當前裝置的電量；隱藏這一資訊，會迫使使用者需要退出您的應用程式才能獲取電量資訊，這可不是一種理想的使用者體驗。

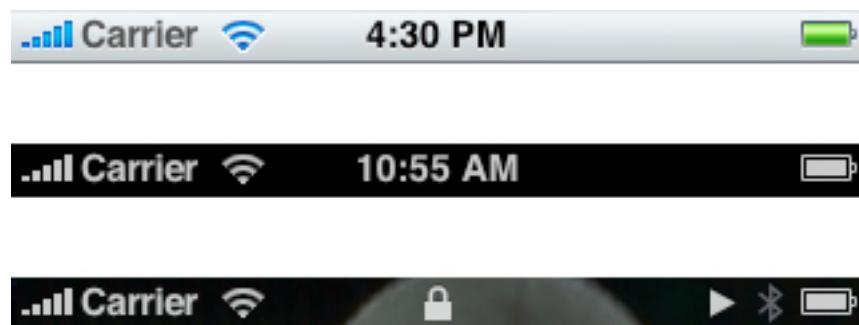
例如，照片應用程式以全螢幕視圖顯示相機中的各個照片時，會在幾秒鐘之後漸漸隱去狀態欄，導覽欄和工具欄。這種做法是很恰當的，因為在照片應用程式中，使用者僅僅關注照片的內容，而並不與應用程式進行交互。但是，使用者可以通過輕輕點選螢幕，讓狀態欄，導覽欄和工具欄重新顯示出來。

如果您想要不時地在應用程式中隱藏狀態欄，那麼您應該使得使用者能夠通過單擊螢幕重新獲得狀態欄。除非您有非常令人信服的理由，否則最好避免用自定義的方式來重新顯示狀態欄，因為使用者未必會發現或記住這樣的方式。

雖然您難以控制狀態欄的內容，但是您可以定制它的外觀，並在一定程度上定制它的行為。具體來說，您可以：

- 設定網路活動指示器是否應該可見。如果您的應用程式正在執行的一項網路操作會持續幾秒鐘以上，則應該顯示網路活動指示器。如果該網路操作將在幾秒鐘之內完成，則不必顯示網路活動指示器，因為它很可能在使用者意識到它的存在之前便消失了。（在您的程式碼中，您可以使用UIApplication的方法networkActivityIndicatorVisible來控制指示器的可見性。）
- 指定狀態欄的顏色。您可以選擇灰色（預設的顏色），不透明的黑色或半透明的黑色（也就是 α 值為0.5的黑色）。圖6-2顯示了這些樣式。（請注意，您應該在Info.plist文件中設置一個值來指定狀態欄的樣式；更多相關的資訊請參考iPhone應用程式程序指南。）
- 設置是否應該採用動畫方式顯示狀態欄顏色的改變。（請注意，動畫效果會使舊的狀態欄一直滑出螢幕，最後消失，同時新的狀態欄會滑動到相應的位置上。）

圖 6-2 狀態欄的三種樣式



請確保狀態欄的外觀與您的應用程式的其餘部分能夠協調搭配。例如，如果導覽欄是不透明的，就要避免使用半透明的狀態欄。

導覽欄

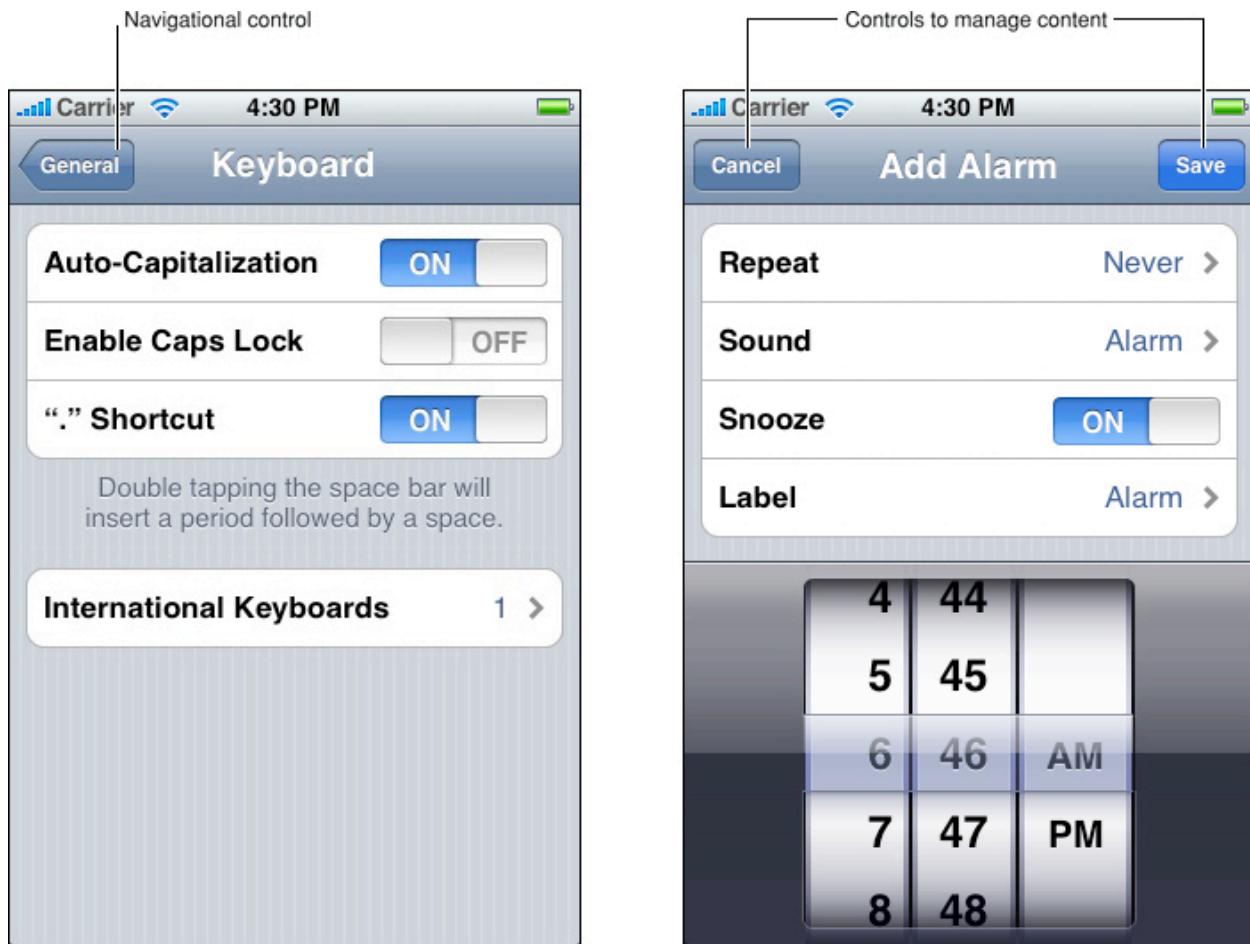
導覽欄位於應用程式螢幕的上邊緣，在狀態欄之下。導覽欄通常會顯示當前視圖的標題，包含導覽控制元件，並在適當的情況下也可以包含作用於視圖內容的控制元件。在效率型應用程式（詳見“效率型應用程式”）中，導覽欄非常有用，因為這些應用程式通常會按照層次結構顯示資訊。

導覽欄有兩種用途：

- 激活應用程式中不同視圖之間的導覽
- 提供一些控制元件，用於管理視圖中的條目

圖 6-3 顯示了這兩種用途的示例。

圖 6-3 包含導覽控制元件和內容管理控制元件的導覽欄



導覽欄的內容

導覽欄可以根據當前視圖示題的寬度居中顯示標題，如圖6-4所示。效率型應用程式中的初始視圖應該包含一條導覽欄，上面只需顯示第一個視圖的標題，這是因為使用者還沒有切換到其它位置。

圖 6-4 顯示當前視圖示題的導覽欄



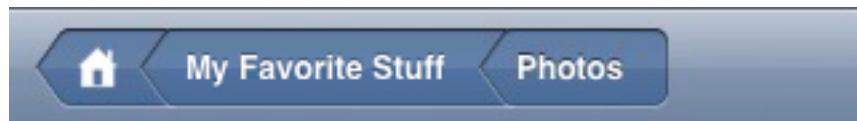
一旦使用者切換到另一個視圖，導覽欄就應該將它的標題更改為新視圖的標題，並且應該提供一個返回按鈕，標記為先前視圖的標題。例如，圖6-5顯示了“日期和時間”設置中的導覽欄，該設置屬於“常規”設置視圖中的一部分。

圖 6-5 包含導覽控制元件的導覽欄



標準的返回按鈕為使用者提供了一種返回到先前螢幕的可靠方式，所以一定不要改變該按鈕的行為。特別地，您應該避免建立多段的返回按鈕，比如圖6-6中所示的按鈕。

圖 6-6 不推薦多段的返回按鈕



使用多段的返回按鈕會引起以下列問題：

- 多段返回按鈕持續增長的寬度會佔用顯示當前視圖示題的空間。
- 無法表明單獨某一段的選定狀態。
- 段越多，每一段的可點選區域就越小，這使得使用者很難準確點選到特定的某一段。
- 當使用者瀏覽層次比較深時，無法確定顯示哪些層的內容。

如果您認為沒有多段返回按鈕的幫助來顯示路徑，使用者可能會感到迷惑，那麼這也就意味著使用者必須瀏覽很深的資訊層次才能找到他們所需要的內容。為瞭解決這個問題，您應該盡量將資訊層次扁平化。

除了返回按鈕之外，導覽欄還可以在標題的右側包含另一個按鈕。如果您不需要顯示返回按鈕（由於您的應用程式不支援分層導覽），您也可以選擇在標題的左側顯示一個作用於視圖內容的按鈕，比如“編輯”按鈕。圖6-7顯示了這樣的一個例子。

圖 6-7 包含用於管理視圖內容的控制元件的導覽欄



要瞭解如何在您的應用程式中實現導覽欄，請參考“導覽控制器”。

正如您在上述示例中所見，導覽欄上的按鈕周圍有一圈帶邊角的框。在iPhone OS中，這種樣式被稱為邊框樣式。導覽欄的所有控制元件都要使用邊框樣式。事實上，即使您在導覽欄上放置一個無格式（沒有邊框）的控制元件，它也會自動轉換為有邊框的樣式。

您可以設計自己的圖示用作導覽欄的按鈕，或者您也可以利用iPhone OS提供的預定義按鈕。關於您可以使用的按鈕相關的更多資訊，請參考“用於工具欄和導覽欄的標準按鈕”。

雖然您可以為所有在導覽欄中顯示的文字指定字體，但為了可讀性，推薦您使用系統字體。當您使用適當的UIKit程序接口建立導覽欄時，其標題將自動採用系統字體來顯示。

導覽欄的尺寸和顏色

當裝置的方向由縱向變為橫向時，導覽欄的高度也會自動隨之改變（您不應該通過程序的方式指定它的高度）。裝置處於橫向時，窄一點的導覽欄可以為螢幕上的內容留出更大的空間。您在設計導覽欄控制元件的圖示和設計螢幕上的佈局時，一定要將這種高度上的差異計算在內。

您可以指定導覽欄的顏色和半透明效果，以和您的應用程式的整體外觀以及其它欄目（即工具欄，標籤欄和狀態欄）一致。您可以使用自定義顏色或選擇下列標準顏色中的一種：

- 藍色（預設顏色）
- 黑色

如果能對您的應用程式的外觀有所幫助，您還可以為導覽欄新增半透明效果。當您使用半透明的導覽欄時，螢幕會給人感覺俱有更大的可視區域，這在裝置橫向放置時效果非常理想。一定要避免將半透明的導覽欄和黑色不透明的狀態欄混合使用（然而半透明的導覽欄可以同灰色不透明的狀態欄一起顯示）。

導覽欄的外觀要力爭與應用程式中的其它欄目的外觀保持一致。例如，如果您使用半透明的導覽欄，就不要將它與不透明的工具欄一起使用。此外，請避免在同一方向的不同螢幕中改變導覽欄的顏色或半透明效果。

工具欄

如果您的應用程式為使用者提供了一系列可在當前上下文中執行的動作，那麼您可以為使用者提供一個工具欄。工具欄位於螢幕的下邊緣，它包含若干按鈕，這些按鈕用於執行與當前視圖中的對象相關的動作。工具欄不是用來切換應用程式的不同模式的。如果您需要這樣做，您應該使用標籤欄而非工具欄（更多資訊請參考“標籤欄”）。

例如，當使用者在郵件應用程式中查看消息時，應用程式提供了一個工具欄，除了查收新郵件和編寫新郵件之外，工具欄中還包含了用於刪除，回復和行動消息的項目。通過這樣的方式，使用者即使停留在消息查看的上下文環境中，也仍然可以使用他們所需的命令來管理他們的電子郵件。圖6-8顯示了這個場景。

圖 6-8 當前上下文下提供功能的工具欄



工具欄的內容

工具欄在其寬度範圍內等距離地顯示工具欄上的項目。對於工具欄上顯示的項目，最好對其數目加以限制，這樣使用者可以輕鬆地點選到他們所需的項目。我們向您推薦的使用者介面元素的點選區域為 44×44 像素，因此，為使用者提供五個以下的工具欄項是比較合理的。圖6-9中的例子向您展示了一個各項間距適中的工具欄示例。

圖 6-9 間距適當的工具欄



圖6-8和圖6-9中的項目都沒有邊框。在iPhone OS中，這種樣式叫做無格式樣式。（關於邊框樣式的例子請參考圖6-7。）雖然您既可以在工具欄中使用帶邊框樣式的按鈕，也可以使用無格式按鈕，但是您不要將這兩種樣式在同一工具欄中混合使用。

您可以為工具欄按鈕自行設計圖示，或者您也可以充分利用iPhone OS提供的預定義按鈕。（關於可以使用的按鈕的更多資訊，請參考“用於工具欄和導覽欄的標準按鈕”。）如果您選擇建立自定義的工具欄按鈕，請盡可能保持它們的尺寸相近，以獲得均勻而又不失吸引力的外觀。

工具欄的尺寸和顏色

當裝置的方向由縱向變為橫向時，工具欄的高度也會自動隨之改變（您不應該在程式中指定它的高度）。裝置處於橫向時，窄一點的工具欄可以為螢幕上的內容留出更大的空間。您在設計工具欄按鈕的圖示和設計螢幕上的佈局時，一定要將這種高度上的差異計算在內。

您可以指定工具欄的顏色和半透明效果，以和應用程式的整體外觀以及其它欄目（即導覽欄，標籤欄和狀態欄）一致。您可以使用自定義顏色或選擇下列標準顏色中的一種：

- 藍色（預設顏色）
- 黑色

如果能對您的應用程式的外觀有所幫助，您還可以為工具欄新增半透明效果。當您使用半透明的工具欄時，螢幕會給人感覺具有更大的可視區域，這在裝置橫向放置時效果非常理想。

工具欄的外觀要力爭與應用程式中的其它欄目的外觀保持一致。例如，如果您使用半透明的工具欄，就不要將它與不透明的導覽欄一起使用。此外，請避免在同一方向的不同螢幕中改變工具欄的顏色或半透明效果。

標籤欄

如果您的應用程式為同一組數據提供了不同的視圖，或者與應用程式的整體功能相關的不同子任務，您可能會想要使用標籤欄。標籤欄出現在螢幕的下邊緣。

標籤欄使使用者能夠在應用程式的不同模式或不同視圖之間進行轉換，並且使用者應該能夠從應用程式的任何地方進入這些模式。但是，標籤欄絕不應該被當做工具欄使用，區別在於工具欄上的按鈕是對當前模式下的元素起作用的（有關工具欄的更多資訊請參考“工具欄”）。

例如，在iPhone中，iPod使用了播客，藝術家，視訊和播放列表等標籤欄，使得使用者能夠在他們的媒體收藏中選擇他們所關注的部分。另外，時鐘應用程式也使用了標籤欄，讓使用者訪問該程式的四項功能，即“世界時鐘”，“鬧鐘”，“碼錶”和“計時器”。圖6-10顯示了在標籤欄中選擇標籤是如何改變時鐘程式的視圖的。請注意圖6-10中的標籤欄，在時鐘應用程式的不同模式中是如何保持始終可見的。這使得使用者能夠很容易地看到他們正處於哪種模式，並且不論當前處於哪種模式都能訪問時鐘應用程式的所有模式。

圖 6-10 使用標籤欄切換應用程式視圖



標籤欄使用標籤顯示圖示和文字，這些標籤的寬度一致，並且都以黑色為背景。當一個標籤被選中時，它的背景變亮，標籤中的圖像也會高亮顯示。圖6-11顯示了這一場景。

圖 6-11 標籤欄中被選中的標籤



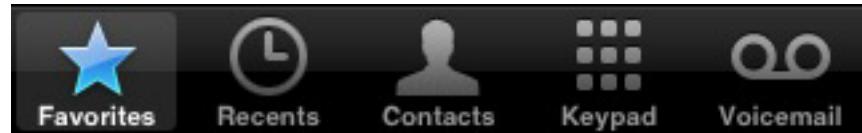
注意：不管裝置的方向如何變化，標籤欄不會改變它的不透明性和高度。

iPhone OS為標籤提供了一系列圖示，比如像圖6-11中“功能”和“書籤”標籤項那樣的圖示。如果您選擇使用這些圖示，請確保對它們的使用符合圖示所表達的意思。關於您可以使用的標籤欄圖示的更多資訊，請參考“用於標籤欄的標準圖示”。

提供附加標籤

如果您的應用程式的標籤欄只包含五個或五個以下的標籤，那麼iPhone OS可以在標籤欄上等距離地將它們全部顯示出來，如圖6-12所示。

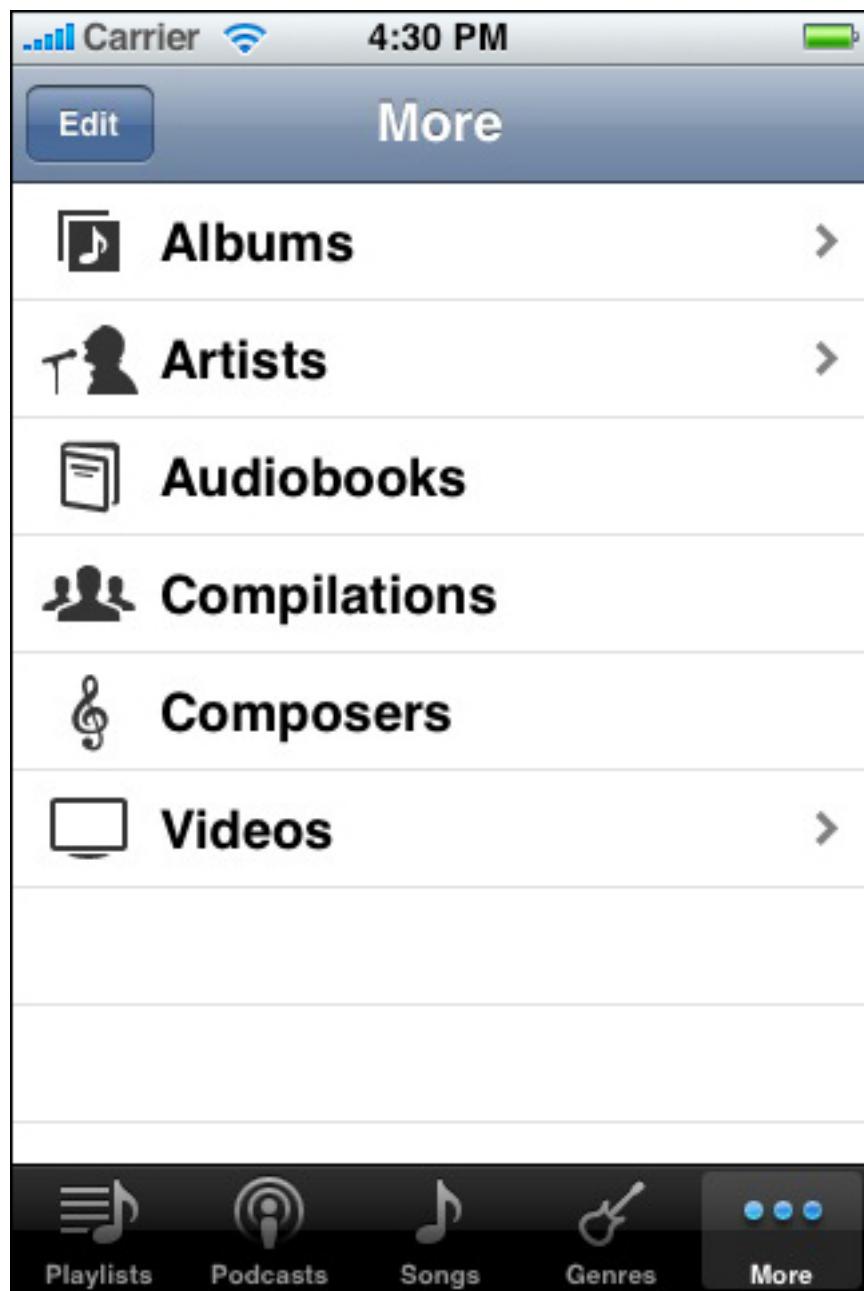
圖 6-12 iPhone OS的標籤欄最多可顯示五個標籤



如果您的應用程式的標籤欄包含的標籤數目超過五個，iPhone OS會在標籤欄上顯示四個標籤，並新增一個“更多”標籤，如圖6-11所示。

使用者點選“更多”標籤後，能夠在一個單獨的螢幕中查看附加標籤的列表，如圖6-13所示。

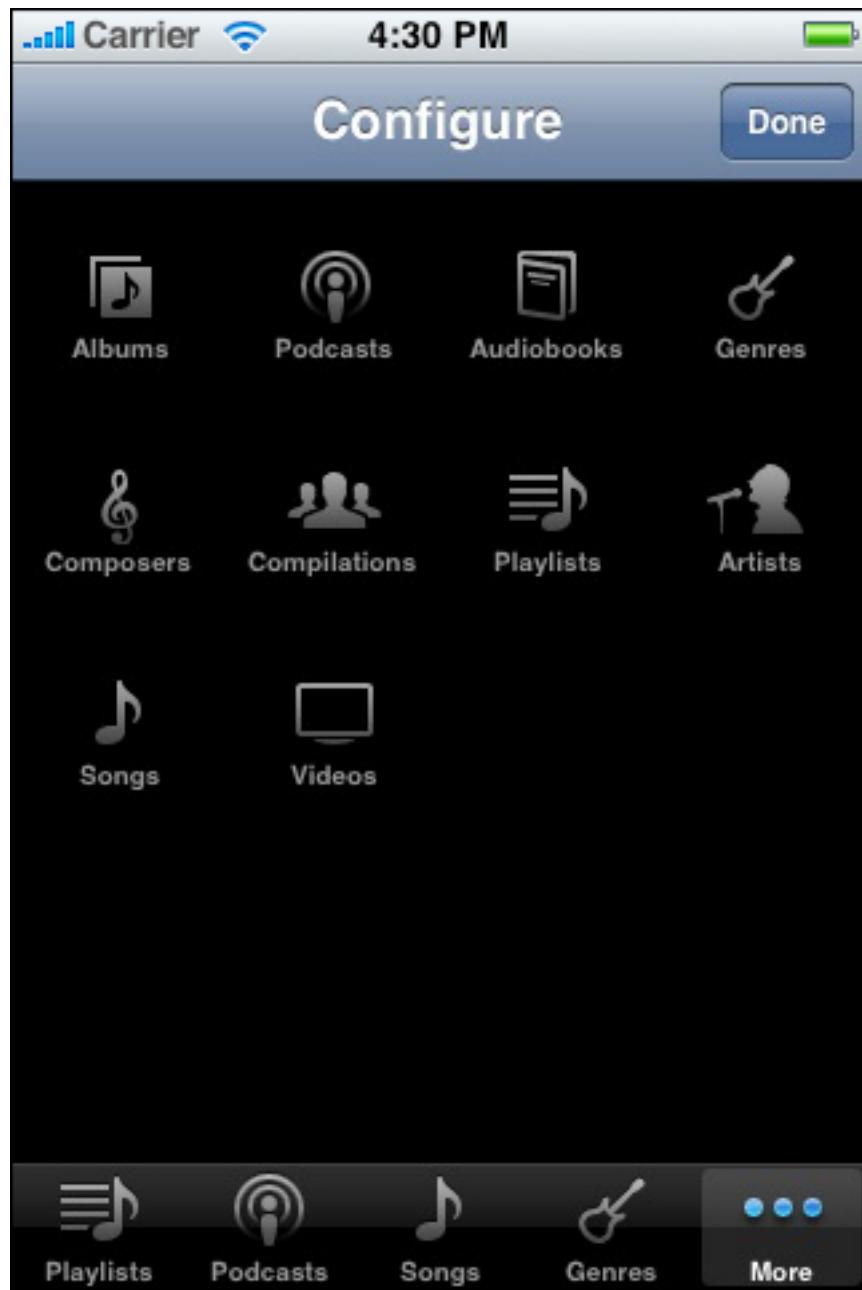
圖 6-13 當使用者點選“更多”標籤時顯示附加標籤



在“更多”螢幕中同樣能夠包含一個“編輯”按鈕，使用者可以通過點選該按鈕來配置標籤欄，使其顯示使用者最常用的標籤。例如，當使用者在iPod程式的“更多”螢幕上點選“編輯”按鈕後，他們會看到如圖6-14 所示的“配置”螢幕。

圖 6-

14 當應用程式俱有五個以上的標籤時，使用者可選擇在標籤欄上顯示他們最喜愛的標籤



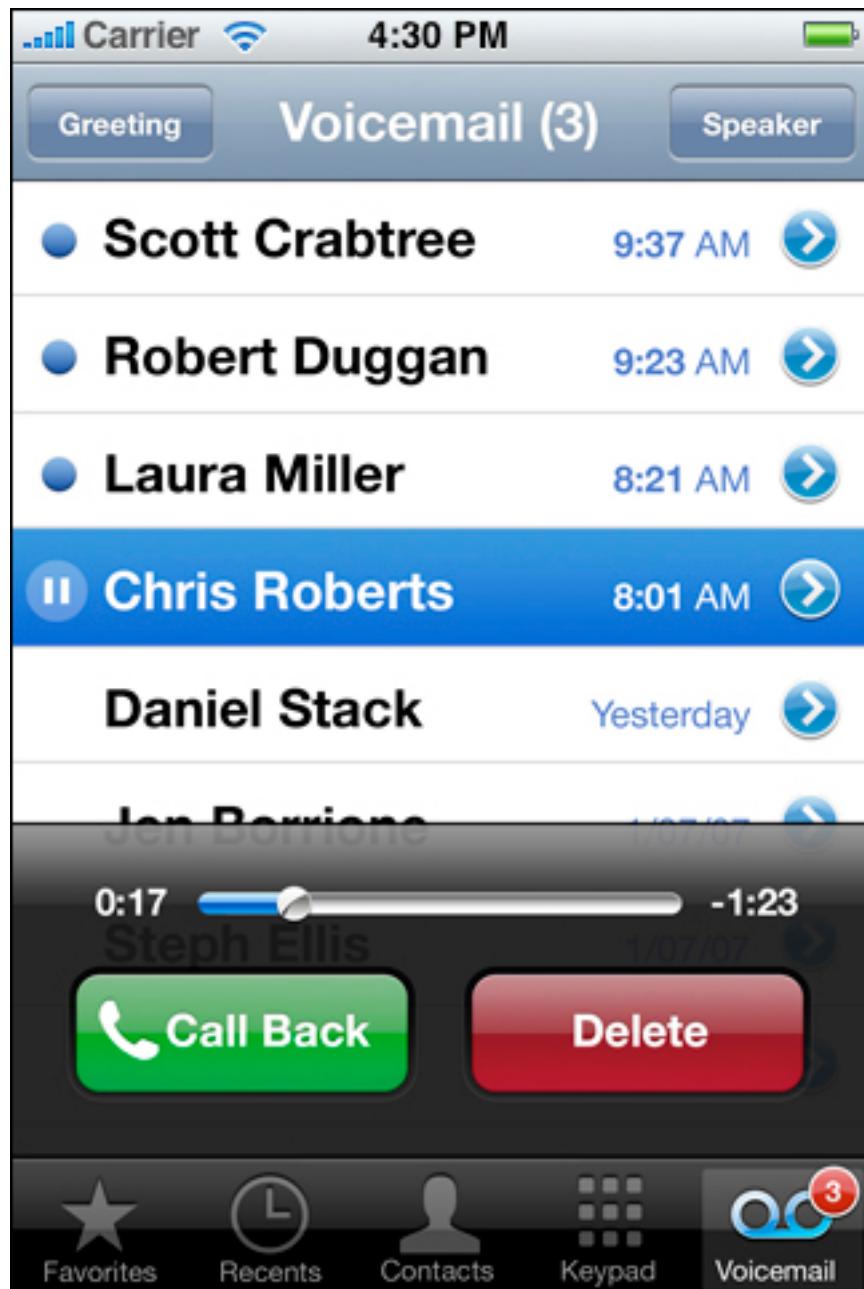
請注意iPod在以上三處（標籤欄，“更多”螢幕和“配置”螢幕）都使用了相同的標籤圖示。這有助於使使用者確定不管該圖示出現在哪裡，它都代表相同的意思。

為標籤欄上的標籤做標記

您可以在標籤上顯示一種標記，通過這種低調的，不十分顯眼的方式與使用者進行通訊。在使用者的任務或上下文環境中可能有些資訊並非不可或缺，但它們對使用者很有幫助，因此做標記的回饋方式就很適合於傳遞這些資訊。

通過在一個特定的標籤上做標記，您可以把標記所表達的資訊和您的應用程式中的一種特定模式—甚至是當前模式—聯繫在一起。圖6-15顯示了一個在標籤上做標記的例子。

圖 6-15 標籤欄上傳達資訊的標記



請注意，如果您註冊了蘋果通知推送服務並且使用者允許顯示標記，那麼您也可以在應用程式的主螢幕上顯示標記。請參考“啓用推送通知”來詳細瞭解這些內容是如何實現的。

在應用程式螢幕中使用視圖和控制元件

在iPhone OS中，UIKit決定了視圖以及控制元件的行為和預設外觀。您應該盡可能地使用UIKit提供的標準使用者介面元素，並遵循它們的使用建議。這樣做對您最主要的好處有兩點：

- 使用者已經習慣了標準視圖和控制元件的外觀和行為。當您使用相似的使用者介面元素時，使用者之前的經驗對於他們學習使用您的應用程式有所幫助。
- 即使iPhone OS改變了標準視圖或控制元件的外觀或行為，您的應用程式也能夠繼續工作，並且只需要少許改動（如果有的話）就能自動更新。

許多控制元件支援某種程度上的使用者自定義，通常是顏色或內容（比如新增一個文字標籤或一幅圖像）方面。如果您想要開發一個能使使用者沈醉其中的應用程式，一個好的做法是，您所建立的控制元件完全不同於預設控制元件。這是因為您創造了一個獨特的環境，而去探索如何操控這個環境正是使用者在優秀的應用程式中所期待的體驗。

然而，一般來講您應該避免徹底改變標準行為控制元件的外觀。如果您使用使用者不熟悉的控制元件執行標準行為，則使用者不得不花時間熟悉如何使用它們，並且好奇這些控制元件俱有哪些（哪怕只有一丁點）標準控制元件不具備的功能。

警告，動作表單和模式視圖

警告，動作表單和模式視圖是視圖的幾種類型，當需要引起使用者的注意，或者需要提供額外的選項或功能時，這些視圖就會出現。圖7-1顯示了這幾種視圖的例子。

圖 7-1 動作表單，模式視圖和警告



要瞭解這幾種類型視圖的程序實現，請參考“模式視圖控制器”。

用法和行為

警告，動作表單和模式視圖都是模式的，這意味著使用者必須通過點選按鈕顯式地關閉它們，然後才可以繼續使用應用程式。雖然有些時候您需要警告使用者潛在的危險動作，或是向他們提供額外的選項，但您一定要避免過度使用這些視圖，這是因為：

- 所有類型的模式視圖都會中斷使用者的工作流程。
- 過度頻繁地出現請求確認的視圖很可能會適得其反，它們想要帶給使用者的幫助反而變成了打擾。

特別地，應該盡量少使用警告。如果警告出現得過於頻繁，使用者很可能為使自己免受它們的打擾，不經閱讀就關閉這些警告。

警告，動作表單和模式視圖的設計目的在於傳遞不同的資訊：

- 警告為使用者提供影響他們使用應用程式（或裝置）的重要資訊。警告通常是不可預料的，因為它們一般會向使用者報告某個問題或當前環境的變化，這可能需要使用者採取相應的行動。
- 動作表單為使用者提供與他們當前執行的動作相關的額外選項。當使用者點選一個工具欄按鈕，開始執行一個俱有潛在破壞性的動作（比如刪除所有最近的通話）或是一個可以以不同的方式完成的動作（比如使用者可以從多個目的地中指定其中一個作為發送動作的目的地）時，他們會期待動作表單的出現。
- 模式視圖能夠在當前任務的環境下提供更廣泛的功能。對於那些與使用者的工作流程直接相關的子任務，模式視圖還能夠提供執行這些子任務的方法。

這幾種類型的視圖在外觀和行為上也各不相同，這也強調了它們向使用者傳遞的消息不同。由於使用者已經習慣了這些視圖的外觀和行為，所以您一定要在應用程式中正確地，一致地使用這些視圖。請閱讀下面的章節來瞭解更多有關警告，動作表單和模式視圖的資訊。

使用警告

警告會在應用程式螢幕的中間彈出並浮在應用程式的視圖之上，以俱有高可視性的方式向使用者提供關鍵資訊。其特別的外觀強調的是，警告的出現是由於應用程式或裝置中的一些變化，而未必是使用者最近執行的動作所產生的結果。警告應該顯示文字資訊來說明當前的情況，在理想情況下，還應該允許使用者為下面的操作選擇適當的動作。

使用者對於裝置和後台運行的內置應用程式所發出的警告已經習以為常，比如消息應用程式，但是您還是應該少在應用程式中使用這樣的警告。例如，您可能會使用警告告訴使用者，他們啓動的任務被阻塞了。採用警告顯示這類消息是有道理的，因為告訴使用者問題所在並讓他們來選擇如何處理是非常重要的。

您也可以使用警告讓使用者有機會選擇接受或拒絕一種俱有潛在危險的結果。當出現這種情況時，警告應該顯示兩個按鈕：一個用於關閉該警告並執行動作，另一個用於關閉該警告但不執行動作。通常，應該為關閉警告但不執行動作的按鈕使用“取消”作為標籤。請注

意，如果當顯示這樣的警告時，使用者按下Home鍵，那麼除了退出應用程式之外，其結果應該與點選“取消”按鈕相同：即關閉警告但不執行動作。

那些很少出現的警愾能夠讓使用者認真地對待它們。因此，一定要盡量減少應用程式中顯示的警愾的數量，要確保每一個警愾都提供了關鍵資訊和有用的選項。一般情況下，應該盡量避免建立這些警愾：

- 將正常進行中的任務的最新消息通知使用者。相反，您應該考慮使用進度視圖或活動指示器，向使用者提供與進度相關的回饋資訊（“進度視圖”和“活動指示器”介紹了這些控制元件）。
- 請求確認使用者發起的動作。如果要確認使用者發起的動作，即使是刪除聯絡人這種俱有潛在危險性的動作，您應該使用的是動作表單（在下面的“使用動作表單”中介紹）。
- 通知使用者令他們無法操作的錯誤或問題。即使有必要使用警愾來告訴使用者出現了他們無法修復的關鍵問題，但如果可能的話，更好的做法是將這些資訊融入到使用者介面中。例如，與其在每次連接服務器失敗時都通知使用者，不如在使用者介面上顯示上一次成功連接服務器的時間。

使用動作表單

當使用者通過點選應用程式的工具欄上的按鈕發起了一項任務，動作表單顯示一組與該務相關的可選方案。動作表單適合於：

- 選擇完成任務的方法。例如，在照片應用程式中，使用者可以在瀏覽單獨的一張照片時點選“發送”按鈕。這時會出現一個動作表單，有三個發送目的地供使用者選擇（還有一個“取消”按鈕，用於取消發送）。在類似這樣的情況下，顯示動作表單十分有用，因為您可以提供一系列在當前任務環境下有意義的選項，而無需在使用者介面中為這些選項保留一片固定的區域。
- 在完成一項俱有潛在危險性的任務之前獲得使用者的確認。例如，根據郵件應用程式的設置，當使用者點選“郵件”工具欄上的“垃圾郵件”按鈕時，會顯示一個動作表單，允許他們選擇繼續進行刪除還是取消該動作。當您在類似的情況下顯示動作表單時，要確保使用者瞭解他們將要執行的步驟能夠帶來的危險後果，並且您可以提供一些可選的方案。在iPhone OS裝置上，這種類型的交互是非常重要的，因為使用者有時會無意中點選控制元件。

動作表單總是從應用程式螢幕的底部出現，覆蓋應用程式的視圖（如圖7-1中的左圖所示）。然而，與警告不同的是，動作表單的側邊緣固定在螢幕的兩側，增強了它與應用程式和使用者最近動作的聯繫。

動作表單包含一些按鈕，讓使用者選擇如何完成他們的任務。您不應該向動作表單中新增消息，因為按鈕標籤—再聯繫當前正在執行的任務—應該已經為使用者提供了足夠多的環境資訊，幫助他們理解這些選項的含義。當使用者點選其中一個按鈕時，動作表單就會消失。由於動作表單為使用者提供動作的選項，因此它一定會提供不止一個按鈕。

使用模式視圖

預設情況下，模式視圖會從螢幕的下邊緣滑出，並且始終覆蓋在整個應用程式的螢幕之上（如圖7-1中位於中間的圖片所示）。由於模式視圖隱藏了當前的應用程式螢幕，因此它使使用者明確感知到自己進入了一種不同的，臨時的模式，他們可以在這種模式中完成一些任務。

模式視圖可以在適當的情況下顯示文字，並且可以包含執行任務所必需的控制元件。此外，模式視圖通常會顯示一個用於完成任務並關閉視圖的按鈕，以及一個用於放棄任務的“取消”按鈕。

相比於動作表單，模式視圖支援更廣泛的使用者交互。不像動作表單只能接受單一的選項，模式視圖支援多重步驟的使用者交互，比如多項選擇或資訊輸入。

當您需要實現一項與應用程式的主功能相關的獨立任務時，請使用模式視圖。有的使用者介面元素無法始終屬於應用程式主使用者介面，模式視圖尤其適合於需要這種使用者介面元素的多步驟子任務。郵件應用程式中的編寫郵件視圖就是一個使用模式視圖的好例子。當使用者點選“編寫郵件”按鈕時，會出現一個模式視圖，包含填寫地址和消息的文字區，用於輸入的鍵盤，一個“取消”按鈕和一個“發送”按鈕。

設計警告

您可以指定警告中的文字，按鈕的數量和按鈕的內容，但是您不能定制警告本身的寬度，背景外觀以及文字的對齊方式（它是中心對齊的）。

注意：當您閱讀這些準則時，應該瞭解以下定義：

- 標題大寫 指每個單詞都大寫，除了冠詞，並列連詞和四個或更少字母的介詞。
- 句子大寫 指第一個單詞大寫，其餘的單詞都小寫，除非它們是專有名詞或專有形容詞。

警告的標題（以及可選的消息）應該簡明扼要地說明相關的情況，並且告訴使用者可以做些甚麼。理想情況下，您寫的文字應該向使用者提供足夠的背景資訊，使他們明白為甚麼會出現警告並決定點選哪個按鈕。

當您編寫警告標題時：

- 請保持標題足夠簡短，如果可能的話，盡量顯示在一行之內。長的警告標題對於人們來說難以快速閱讀，並且可能會導致警告消息滾動顯示。
- 避免建立僅包含一個單詞的標題，它無法提供任何有用的資訊，比如“錯誤”或“警告”。
- 推薦使用句子片段。一個簡短且內容豐富的說明往往比一個完整的句子更容易理解。
- 不要害怕負面消息。人們能夠理解大部分警告是在告訴他們出現的問題，或是在向他們發出有關危險情況的警告。因此，直接的負面消息遠比拐彎抹角的正面消息要好得多。
- 盡量不要使用“你”，“你們”或“我”。有時候，直接指明人物對象的文字可能會引起歧義，甚至可能會被理解為一種侮辱。
- 使用標題式的大寫方式且不要使用結束標點，如果：
 - 標題是一個句子片段
 - 標題由一個非疑問句的單句組成
- 如果標題由一個疑問句單句組成，請使用句子式的大寫方式和結束問號。在一般情況下，請考慮使用疑問句作為警告的標題，如果這樣可以讓您避免增加一條額外的消息。
- 如果標題由兩個或更多的句子組成，請使用句子式的大寫方式，並且為每個句子新增適當的結束標點。實際上，很少有必要使用包含兩個句子的警告標題，雖然您可能認為它可以讓您避免一條額外的警告消息。

如果您選擇提供警告消息：

- 請保持消息足夠簡短，如果可能的話，盡量顯示在1-2行之內。如果消息太長，將需要滾動顯示。
- 始終使用句子式的大寫方式和適當的結束標點。警告消息應該是一個完整的句子。

避免在您的警告文字避免描述點選哪個按鈕，比如“點選‘視圖’查看資訊”。理想情況下，清晰的警告文字和合乎邏輯的按鈕標籤向使用者提供了足夠的資訊來瞭解當前的情況和他們的選擇。當然，如果您必須提供詳細的指南，請遵循以下原則：

- 一定要使用單詞“tap”（而不是“touch”，“click”或“choose”）來描述選擇動作。
- 不要將按鈕標題括在引號之內，但要保留其大寫方式。

一定要測試您的警告在兩個方向上的外觀。在橫向上，由於警告的高度受到限制，它可能看起來和在縱向時不太一樣。推薦您優化警告文字的長度，以便它在兩個方向上的效果看起來都不錯（避免滾動顯示）。

推薦使用俱有兩個按鈕的警告。通常俱有兩個按鈕的警告是最實用的，因為在兩者之間做出選擇對使用者來說是最容易的。顯示只有單個按鈕的警告並不是一個好主意，因為這樣的警告無法讓使用者控制當前的局勢；相反，它只能顯示資訊並提供一個關閉按鈕。與俱有兩個按鈕的警告相比，包含三個或以上按鈕的警告明顯更加複雜，因而應該盡量避免使用這樣的警告。事實上，如果您需要向使用者提供兩種以上的選擇，您應該考慮使用動作表單來代替警告（更多與這種類型的視圖相關的資訊請參考“使用動作表單”和“設計動作表單”）。

適當地使用警告按鈕的顏色。警告按鈕或是暗色的，或是亮色的。在俱有兩個按鈕的警告中，左邊的按鈕總是暗色的，而右邊的按鈕總是亮色的。在俱有單個按鈕的警告中，按鈕總是亮色的。

- 在指出潛在危險動作的雙按鈕警告中，取消動作的按鈕應該位於右邊，而且是亮色的。
- 在指出人們所期望的安全動作的雙按鈕警告中，取消動作的按鈕應該位於左邊（因此它應該是暗色的）。

注意：“取消”按鈕既可能是亮色的也可能是暗色的，既可能在右邊也可能在左邊，這取決於該選擇是否具有破壞性。一定要在您的程式碼中正確識別哪個按鈕是“取消”按鈕。

為警告按鈕提供簡短並合乎邏輯的標題。最好的標題應該由1-2個單詞組成，並且這些單詞在警告文字的上下文環境中是合理，有意義的。當您為警告按鈕建立標題時，請遵循如下準則：

- 和所有的按鈕標題一樣，請使用標題式的大寫方式，且不要使用結束標點。
- 推薦使用動詞和動詞短語，比如與警告文字直接相關的“取消”，“允許”，“回復”和“忽略”。
- 如果沒有更好的選擇，推薦使用“確定”作為簡單的接受選項。避免使用“是”或“否”。
- 盡量不要使用“你”，“你們”和“我”。使用這些單詞的按鈕標題往往既容易產生歧義又顯得缺乏尊重。

設計動作表單

您可以選擇動作表單的背景，使其與您的應用程式的外觀是一致的，您也可以指定按鈕的數量和按鈕上的內容。

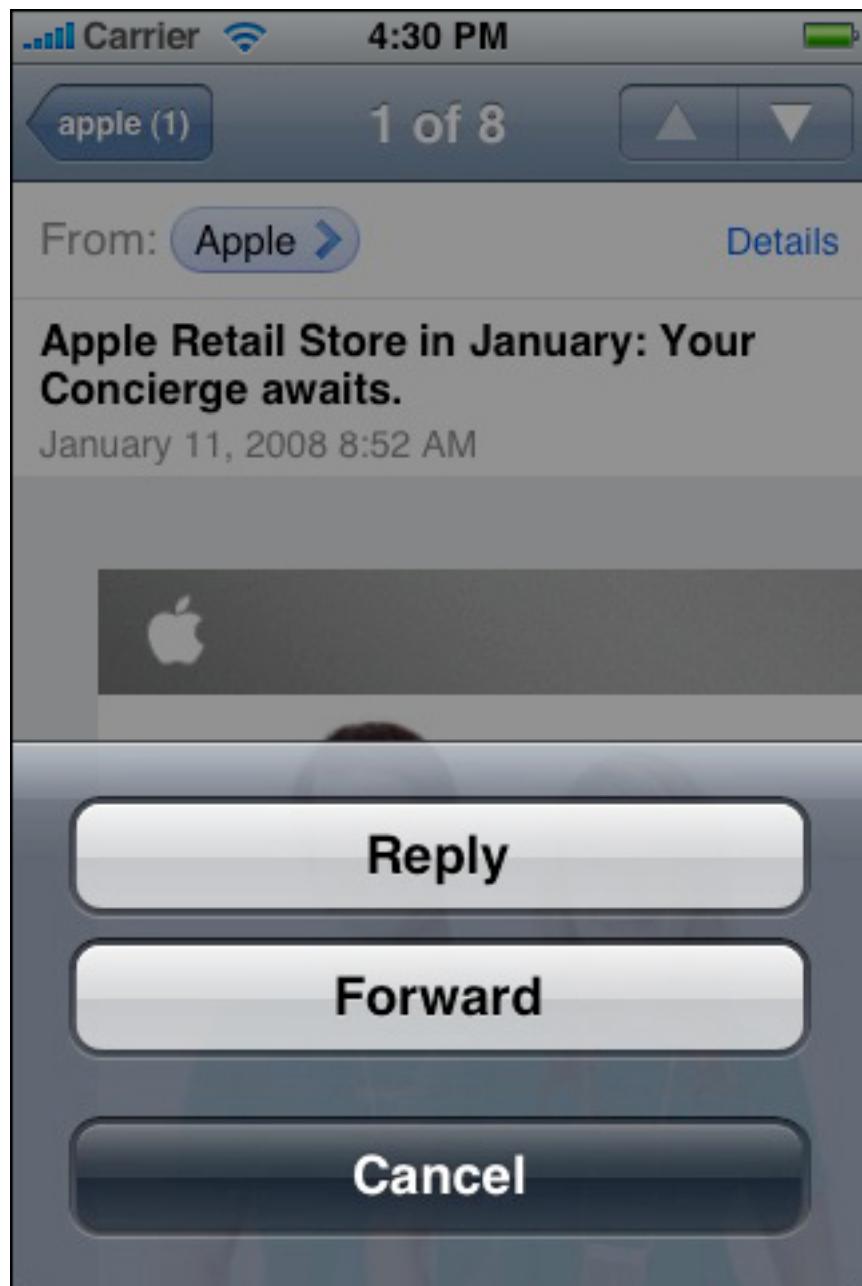
與警告不同的是，動作表單不應該顯示文字資訊。這是因為動作表單是作為使用者動作的結果出現的，比如點選“刪除”或“發送”按鈕，因此沒有必要解釋它為什麼出現。

動作表單可以有兩種不同背景的外觀。您需要確保動作表單的背景與應用程式的工具欄或導覽欄的外觀是一致的。例如，如果您的應用程式使用黑色的導覽欄和工具欄，則動作表單的背景應該是半透明黑色的。預設情況下，iPhone OS在顯示動作表單時採用標準的藍色背景，這與標準的藍色工具欄和導覽欄是一致的。您的應用程式中的所有動作表單應俱有相同的背景顏色，並且應該與導覽欄和工具欄的顏色一致。

一定要將“取消”按鈕顯示在動作表單的底部。這樣能夠促使使用者在看到“取消”選項之前通讀所有的候選項。

圖7-2中所示的動作表單俱有預設的背景外觀，而且“取消”按鈕被放置在推薦的位置上。

圖 7-2 一個典型的動作表單



如果您需要提供一個按鈕來執行具有潛在破壞性的動作，比如刪除使用者購物清單上的所有條目，那麼按鈕的顏色應該採用紅色。這樣的破壞性按鈕一定要顯示在動作表單的頂部，這一點至關重要，原因有兩個：

- 一個按鈕越是靠近動作表單的頂部，它的可視性就越強。
- 當使用者想要點選Home按鈕時，有時他們會誤點選裝置螢幕的底部。所以不要將俱有破壞性的按鈕放置在動作表單的底部，這樣因使用者的誤操作引發不良後果的可能性就會比較小。

圖 7-

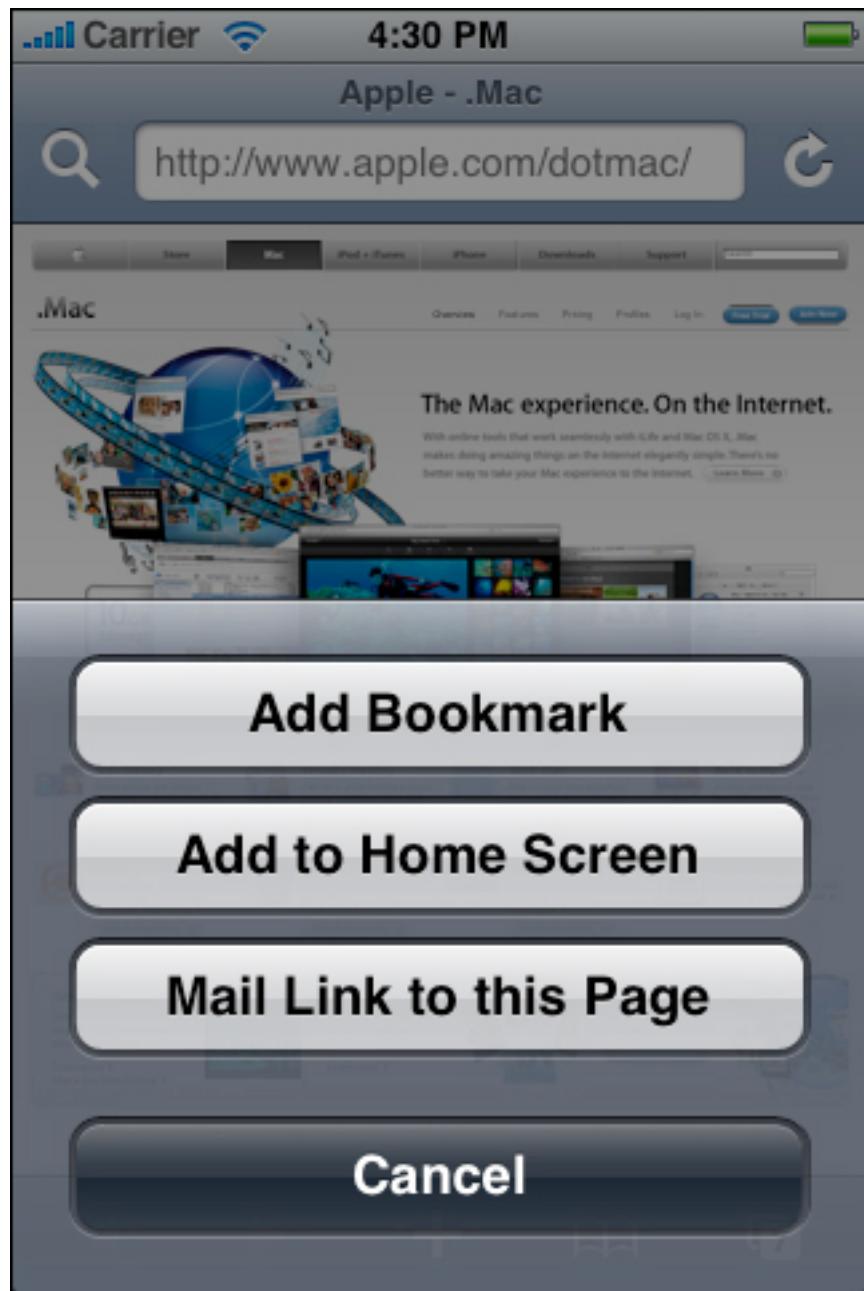
3中所示的動作表單俱有半透明的黑色背景外觀，而且“取消”按鈕和破壞性按鈕都被放置在了推薦的位置上。

圖 7-3 執行破壞性動作的按鈕應該是紅色的，並且要位於動作表單的頂部



您可以在動作表單中顯示多個按鈕，只要確保每個按鈕都能很容易地與其它按鈕區分開來。圖7-4顯示了一個動作表單，其背景與標準的藍色工具欄相匹配，並且除“取消”按鈕之外，它還提供了三個可選的按鈕。

圖 7-4 傑有四個按鈕的動作表單



設計模式視圖

模式視圖的整體外觀應該與顯示它的應用程式是一致的。例如，模式視圖通常包含一個導覽欄，導覽欄俱有標題和用於取消或完成模式視圖任務的按鈕。該導覽欄應該與應用程式的導覽欄俱有相同的背景外觀。

模式視圖通常應該顯示一個以某種方式標識任務的標題。如果合適的話，您還可以在視圖的其它區域顯示文字，來更充分地描述任務資訊或提供一些指導。例如，消息應用程式在使用者想要編寫文字消息時提供了一個模式視圖。如圖7-5所示，該模式視圖顯示了一個導覽欄，它與應用程式的導覽欄背景相同，標題為“新消息”。

圖 7-5 模式視圖應該與應用程式螢幕保持一致



在模式視圖中，您可以使用完成任務所需的任何控制元件。例如，您可以在視圖中包含文字域，按鈕和表視圖。

您可以根據您的應用程式選擇顯示模式視圖的方式，增強使用者對該視圖所表現的臨時環境轉換的感知。為了做到這一點，您可以指定以下切換風格中的一種：

- 垂直 模式視圖從螢幕的下邊緣滑出，視圖關閉時則向下滑動直到消失。（這是預設的切換風格。）
- 翻轉 當前視圖從右向左水平翻轉來顯示模式視圖。在視覺上，模式視圖看起來好像是當前視圖的背面。當模式視圖關閉時，它會從左向右水平翻轉來顯示先前的視圖。

如果您決定更改應用程式中模式視圖的切換風格，請不要僅僅為了追求多樣化而這樣做。您要意識到使用者會認為這種差異意味著別的含義。因此，最好建立一個俱有邏輯性和一致性的模式，使使用者可以輕鬆地覺察到並記住，避免使切換風格發生不必要的改變。

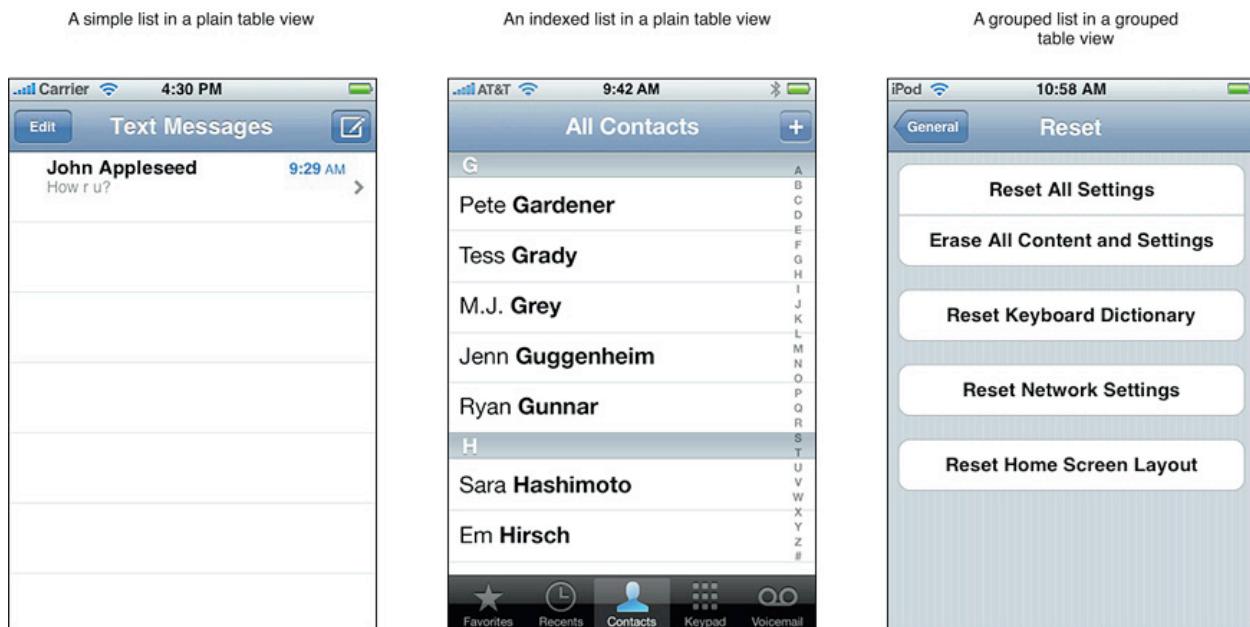
表格視圖，文字視圖和Web視圖

表格視圖，文字視圖和web視圖是俱有多種功能的介面元素，它們在iPhone應用程式中有著不同的用途。例如，表格視圖可以用來顯示簡短的選項列表，詳細資訊的分組列表或者長的項目索引列表。文字視圖和web視圖相對來說可以不受約束地接受和顯示內容。

表格視圖

表格視圖通過多行單列的表格呈現數據。行可以被分為不同的段或組，每一行可以包含文字，圖像和控制元件的組合。使用者可以以輕擊或拖動的方式來滾動行或行的分組。圖8-1顯示了不同樣式的表格視圖，它們以不同的方式顯示列表。

圖 8-1 使用表格視圖顯示列表的三種方式



用法和行為

表格視圖在iPhone應用程式中非常有用，因為無論是資訊量的多少，它們都提供了組織資訊的有效方式。儘管實用型應用程式也可以小範圍的使用表格視圖，但表格視圖通常在需要處理大量使用者數據的效率型應用程式中最為有用。沈浸式應用程式一般不會使用表格視圖來顯示資訊，但是它可能會使用表格視圖來顯示簡短的選項列表。

表格視圖提供了允許使用者瀏覽和操作資訊的內置介面元素。此外，表格視圖還支援：

- 顯示頁眉和頁腳資訊。您可以在列表中每一段或一組的上方或下方顯示描述文字，也可以在整個列表的上方或下方顯示描述文字。
- 編輯列表。您可以允許使用者以統一的方式對列表項進行新增，刪除或排序等操作。表格視圖還支援對多個列表項的選擇和操作，您可以利用這一點向使用者提供一種能夠同時刪除多個列表項的便捷方法。

當使用者選擇一個列表項時，表格視圖應該提供回饋資訊。如果一個列表項可以被選中，則當使用者選擇該項時，包含該項的行應該暫時高亮顯示，以此向使用者回饋已收到使用者的選擇。然後，立即執行動作：要麼顯示一個新的視圖，要麼在這一行顯示一個復選標記來表明該項已被選中或啓用。

在極少數情況下，當與該行中的項目相關的附加詳細資訊或控制元件顯示在同一屏上時，該行可能會繼續高亮顯示。但是，並不鼓勵這樣做，因為要在螢幕上同時顯示選項列表，被選中的項目和相關的詳細資訊或控制元件，很容易造成佈局擁擠，令使用者感到不適。

如果選中某一行的結果是導覽到新的螢幕上，那麼被選中的行會在新的螢幕滑動到相應位置之前高亮顯示。當使用者返回到先前的螢幕時，原來被選中的行會再次高亮顯示一下，提示使用者他們之前的選擇。

請注意，您也可以以動畫的方式展現使用者對列表項的更改。這樣做有利於向使用者提供回饋資訊和增強使用者的直接操作體驗。例如，在設置應用程式中，當您關閉自動設置日期和時間時（在“日期和時間”>“自動設置”中選擇關閉），列表分組會平滑地展開，顯示兩個新的列表項，“時區”以及“設置日期和時間”。

表格視圖應該立即顯示其內容。如果表格的內容繁多或者十分複雜，請不要一直等到所有的數據都可用時才顯示它們。相反，應該立即使用文字數據填充螢幕上可見的行，當更複雜的數據（比如圖像）可用時再顯示它們。這種技術能夠馬上為使用者提供有用的資訊，並且提高了您的應用程式的響應能力。

如果您的應用程式要顯示的數據很少發生變化，您可以考慮在新的數據可用之前顯示那些實際上已經“過期的”數據。這種技術也能夠讓使用者馬上看到有用的資訊，但不推薦那些數據頻繁變化的應用程式也這樣做。在您決定這樣處理之前，一定要估計數據變化的頻率和使用者對迅速看到新數據的期待程度。

如果很難立即顯示有用的資訊，也一定要避免顯示空行，因為這會給使用者造成“應用程式已經出現問題而終止”的錯誤暗示。相反，表應該在螢幕中心顯示一個旋轉的活動指示符和一條資訊標籤，比如“加載中...”。如果您可以顯示舊的數據，您就不必擔心空行的問題，但是您應該在獲得新數據後立即更新螢幕上的資訊。這兩種技術都為使用者提供了回饋資訊，讓他們知道應用程式的處理正在進行當中。

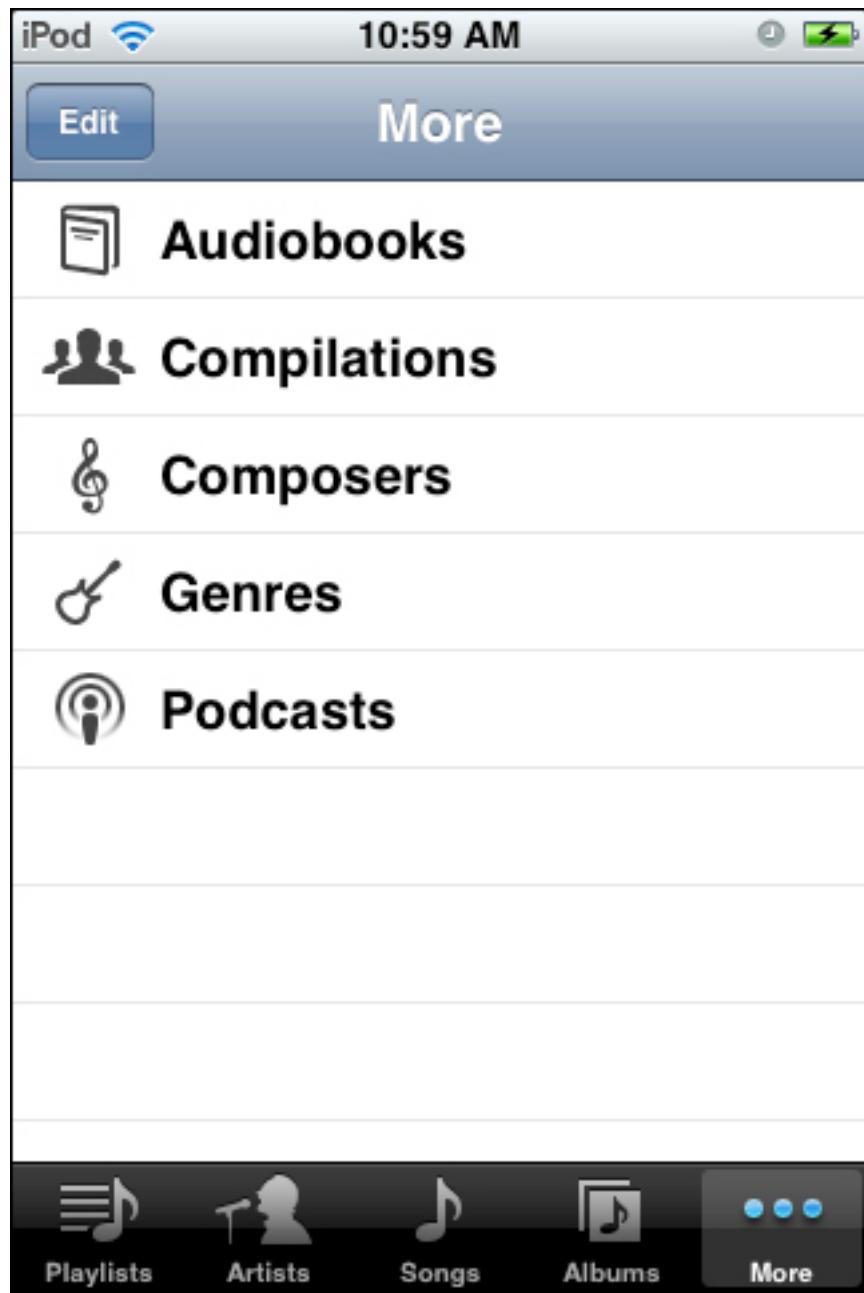
表格視圖樣式

iPhone OS定義了兩種樣式的表格視圖，它們的區別主要在於外觀：

無格式樣式（UITableViewStylePlain）這種視圖樣式所顯示的行從螢幕一側的邊緣延伸至另一側的邊緣。行的背景是白色的。每一行可以被分成帶標籤的段，並且可以在視圖的右邊緣垂直地顯示一個可選的索引。

圖8-2顯示了iPod應用程式中的一個無格式樣式的列表（沒有頁眉，頁腳和索引）。

Figure 8-2 一個簡單的無格式樣式的列表



分組樣式 (UITableViewStyleGrouped) 這種視圖樣式顯示行的分組，每組表格行會相對螢幕兩側縮進。這些分組顯示在獨特的垂直條紋背景下，而分組內部的背景則是白色的。分組樣式的表格可以包含任意數量的分組，每一個分組可以包含任意數量的行。在每一組之前可以加入頁眉文字，其後可以加入頁腳文字。這種樣式的表格視圖不提供索引。

圖8-

3顯示了一個分組樣式的列表，其中每個分組均包含一行。該列表出現在設置應用程式中，不包含頁眉或頁腳文字。

圖 8-3 一個包含四個分組的，俱有分組樣式的列表



單元格樣式

iPhone OS 3.0及後續版本中包含了四種預定義的單元格樣式，您可以使用它們快速，方便地為無格式樣式和分組樣式的表格行建立常用的佈局。請注意，從程序角度而言，這些樣式是應用在表格視圖的單元格上的，單元格規定了表格應該如何繪制各行。

當您使用標準單元格樣式時，您的應用程式的風格會與內置應用程式保持一致，這將在以下幾個方面使您獲益：

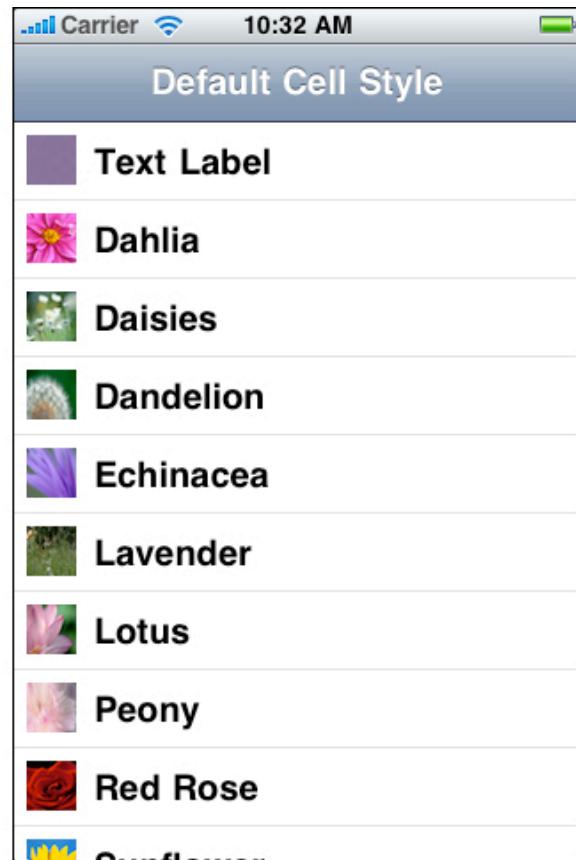
- 使用者可以更快地瞭解您的應用程式如何工作
- 如果今後標準單元格樣式有所改進，您的應用程式可以輕鬆地保持一致，而無需付出大量的額外工作

如果您想要以非標準的方式佈置您的表格行，您最好建立一種自定義的單元格樣式，而不是大刀闊斧地修改已有的標準樣式。**iPhone OS**表格視圖程序指南中的“自定義單元格”一節可以幫助您瞭解如何建立自己的單元格。

請注意，在所有的單元格樣式中，文字都會被自動截斷。一般來說，您應該確保您的文字盡可能簡潔，避免顯示出使用者難以理解的被截斷的單詞或詞組。具體來講，文字截斷是個可大可小的問題，這取決於您使用的單元格樣式和截斷發生的位置。

iPhone OS提供下列標準單元格樣式：

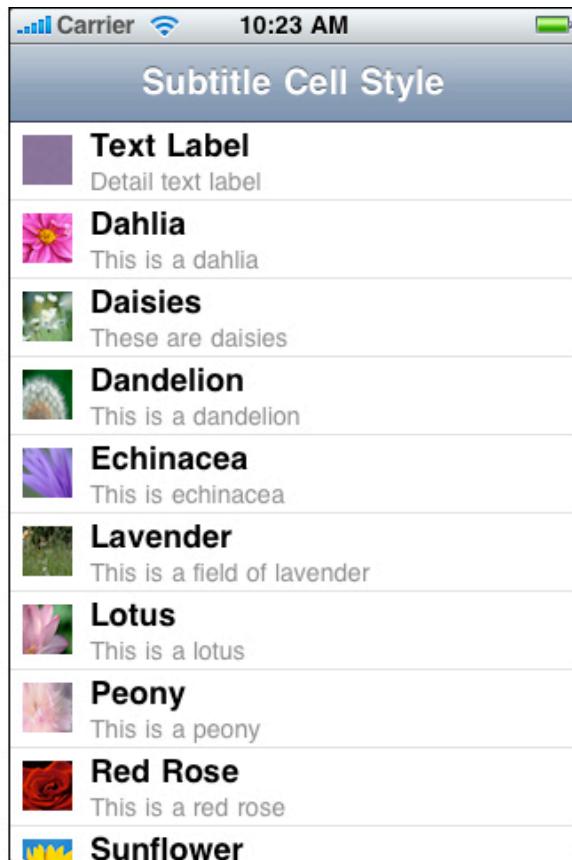
- 預設單元格樣式 (`UITableViewCellCellStyleDefault`) 包含位於左側的可選圖像，後面緊跟左對齊的黑色文字標籤。 圖 8-4 分組表格（左）和無格式表格（右）中的預設單元格樣式



文字標籤代表一個表格項的名稱或標題，而且其左對齊的格式使得列表很易於瀏覽。因此，預設樣式非常適合於顯示不需要根據附加資訊加以區分的表格項列表。

最好使用簡短的文字標籤，但是如果文字截斷無法避免，要盡量確保最重要的資訊包含在前幾個單詞中。

- 副標題單元格樣式 (UITableViewCellStyleSubtitle) 包含位於左側的可選圖像，後面緊跟一行左對齊的文字標籤，以及另一行位於文字標籤下方的左對齊的詳細資訊文字標籤。文字標籤採用黑色字體，詳細資訊文字標籤則採用較小的灰色字體。圖 8-5 分組表格（左）和無格式表格（右）中的副標題單元格樣式



- 文字標籤代表一個表格項的名稱或標題，而詳細資訊文字標籤則表明它包含與該表格項相關的附加資訊。文字標籤的左對齊格式使得列表很易於瀏覽。在列表項看起來比較相似的情況下，這種單元格樣式很適用，因為使用者可以參考詳細資訊文字標籤中的附加資訊，以便區分文字標籤中列舉的表格項。

文字標籤應該盡量簡短，以避免發生文字截斷。如果文字截斷無法避免，請注意將最重要的資訊放在前幾個單詞中。如果詳細資訊文字標籤發生截斷，使用者則不會太介意，因為他們認為它只是用於加強或補充文字標籤中列舉的表格項。

- value 1單元格樣式 (UITableViewCellStyleValue1) 將文字標籤和詳細資訊文字標籤顯示在同一行，文字標籤採用左對齊格式和黑色字體，而詳細資訊文字標籤採用右對齊格式和較小的藍色字體。這種單元格樣式不宜使用圖像。 圖 8-6 分組表格（左）和無格式表格（右）中的value 1單元格樣式

The image displays two side-by-side screenshots of an iPhone application interface. Both screens show a grouped table with a title bar at the top. The title bar contains the text "Value 1 Cell Style" and the time "10:35 AM". The first screenshot shows a single column of data with a "Text Label" header. The second screenshot shows a two-column layout where the first column is labeled "Text Label".

Value 1 Cell Style	
Text Label	Detail text label
Dahlia	This is a dahlia
Daisies	These are daisies
Dandelion	This is a dandelion
Echinacea	This is echinacea
Lavender	This is a field of lavender
Lotus	This is a lotus
Peony	This is a peony
Red Rose	This is a red rose

Value 1 Cell Style	
Text Label	Detail text label
Dahlia	This is a dahlia
Daisies	These are daisies
Dandelion	This is a dandelion
Echinacea	This is echinacea
Lavender	This is a field of lavender
Lotus	This is a lotus
Peony	This is a peony
Red Rose	This is a red rose

- 文字標籤代表一個表格項的名稱或標題，而詳細資訊文字標籤提供了與該表格項緊密相關的重要資訊。

文字標籤的左對齊格式和字體樣式有助於使用者瀏覽他們所需的項目列表，而詳細資訊文字標籤的右對齊格式可以引起使用者對表格項相關資訊的關注。這種單元格樣式適合於顯示表格項的當前值，而該值可能選自一個子列表。

在這種佈局中，可能很難避免文字截斷的發生（因為兩個標籤位於同一行），但是仍值得您努力嘗試。否則，兩組標籤之間沒有留白，使用者很難瞭解這兩部分資訊之間的關係。

雖然您可以在無格式表格或分組表格中使用 value 1 單元格樣式，但是其外觀更適合於分組表格。例如，在設置應用程式的“用法”螢幕中，分組表格使用了 value 1 樣式：

圖 8-7 在分組表格中俱有最佳視覺效果的value 1單元格樣式

Usage	30 Minutes
Standby	1 Day, 17 Hours
Call Time	
Current Period	1 Day, 12 Hours
Lifetime	1 Day, 12 Hours
Cellular Network Data	
Sent	53.1 MB
Received	530 MB

- value 2單元格樣式（UITableViewCellStyleValue2）同樣將文字標籤和詳細資訊文字標籤顯示在同一行，文字標籤採用右對齊格式和較小的藍色字體，而詳細資訊文字標籤採用左對齊格式和較大的黑色字體。這種單元格樣式不宜使用圖像。

圖 8-8 分組表格（左）和無格式表格（右）中的value 2單元格樣式

Value 2 Cell Style	
Text Label	Detail text label
Dahlia	This is a dahlia
Daisies	These are daisies
Dandelion	This is a dandelion
Echinacea	This is echinacea
Lavender	This is a field of lavender
Lotus	This is a lotus
Peony	This is a peony
Red Rose	This is a red rose

Value 2 Cell Style	
Text Label	Detail text label
Dahlia	This is a dahlia
Daisies	These are daisies
Dandelion	This is a dandelion
Echinacea	This is echinacea
Lavender	This is a field of lavender
Lotus	This is a lotus
Peony	This is a peony
Red Rose	This is a red rose

文字標籤的右對齊格式，受限寬度及其字體樣式意味著它僅起到一標題的作用，左對齊詳細資訊文字標籤則包含更重要的資訊。

在這種佈局中，標籤是每一行中的相同位置相向排列的。這樣，就在列表中的文字標籤和詳細資訊文字標籤之間形成了一條清晰的垂直邊緣，有助於使用者一眼就能看到詳細資訊文字標籤的首單詞。如果您允許文字標籤被截斷，該垂直邊緣的清晰程度則會有所消減，這會造成使用者瀏覽詳細資訊文字標籤中的資訊變得更加困難。

雖然您可以在無格式表格或分組表格中使用 value 2 單元格樣式，但是其外觀更適合於分組表格。例如，在聯絡人應用程式的“資訊”螢幕中，分組表格使用了 value 2 樣式：

圖 8-9 在分組表格中俱有最佳視覺效果的 value 2 單元格樣式



注意：所有的標準單元格樣式都允許新增表格視圖元素，比如復選標記或展開指示符。但是一定要注意，新增這些元素會減少單元格中可用於標題和副標題的空間寬度。

您也許能夠通過增加表格行的高度，以支援文字換行，從而避免文字截斷，但這可能會產生一些問題：

- 您必須以程序的方式檢查文字的長度，並判斷是否可能發生文字換行。您必須針對縱向和橫向這兩種情況都作出相應的判斷，因為表格的寬度會對文字換行產生影響。
- 您應該避免在裝置處於一個方向時顯示換行的文字，而在處於另一個方向時不顯示。
- 無論表格視圖的樣式如何，可變的行高都會對表格視圖在應用程式中的整體性能產生負面影響。

最後要強調的是，雖然在分組表格中可變的行高是可以接受的，但是它們會使無格式表格顯得混亂不均。

表格視圖元素

iPhone OS包含一些可以擴展表格視圖的表格視圖元素。除非特別注明，否則這些元素都只適用於表格視圖。在您的應用程式中，一定要正確地使用這些元素，因為使用者已經習慣了它們在內置應用程式中的外觀和行為。

注意：從程序角度而言，表格視圖元素是以不同的方式實現的。有些是單元格（控制表格如何繪制各行的對象）的附屬視圖，其他的可能在表格視圖進入編輯模式時顯示。要瞭解管理這些元素的不同方式，請參考**iPhone OS表格視圖程序指南**。

- **展開指示符** 當這個元素出現時，使用者知道他們可以點選這一行中的任意位置來查看下一級的資訊，或是與當前列表項相關的選項。當選中一行後會顯示另一個列表時，應使用展開指示符。不要使用展開指示符來顯示某個列表項的詳細資訊；這種情況下應使用詳細資訊展開按鈕。
- **詳細資訊展開按鈕** 使用者可以通過點選該元素來查看某個列表項的詳細資訊。（請注意，您可以在表格視圖之外的其他視圖中使用這個元素，以顯示指定對象的詳細資訊；更多資訊請參考“詳細資訊展開按鈕”）。在表格視圖中，可以在某一行中使用詳細資訊展開按鈕，來顯示與該列表項相關的詳細資訊。請注意，與展開指示符不同，詳細資訊展開按鈕所執行的動作可以獨立於行的選擇動作。例如，在電話收

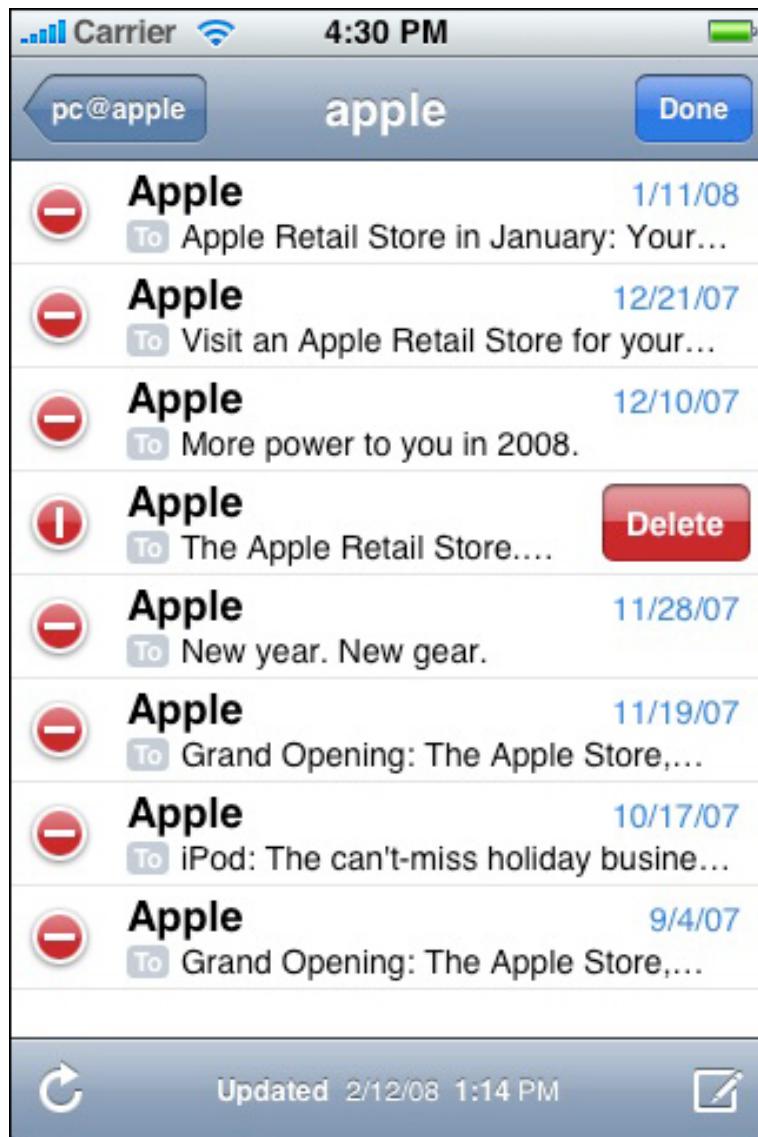
藏櫃應用程式中，點選某一行會呼叫在該行位置上顯示的聯絡人；而點選該行中的詳細資訊展開按鈕會顯示該聯絡人的詳細資訊。

- 刪除按鈕 使用者通過點選該元素可以刪除對應的列表項。當使用者用力點選某一行，或者在編輯環境下點選刪除控制按鈕時，該元素會出現在列表項的右邊。（該元素的例子請參考圖8-10）。
- 刪除控制按鈕 使用者通過點選該元素可以顯示或隱藏每個列表項的“刪除”按鈕。為了向使用者提供更多的回饋資訊，當使用者點選這個按鈕時，該按鈕內的減號標誌由水平方向變為垂直方向，並顯示“刪除”按鈕。該元素的例子請參考圖8-10。在支援臨時編輯模式的分組表格中，刪除控制按鈕出現在表格視圖外部的左側。例如，您可以在編輯聯絡人應用程式中的個人資訊時看到這種情景。在處於固定編輯模式的分組表格中（比如“股票”和“天氣”應用程式中的分組表格），刪除控制按鈕出現在表格內部的左側。

在無格式樣式的表格中，刪除控制按鈕總是出現在表格內部的左側，如圖8-10所示。

- 行插入按鈕 使用者通過點選該元素可以向列表中新增一行。
- 行排序控制元件 當該元素出現時，使用者可以拖動一行到列表中的另一位置上。
- 選定符號 該元素出現在列表項的右側，表明該項當前被選中。
-

圖 8-10 表格視圖可以顯示“刪除”按鈕和刪除控制按鈕



開關控制元件

開關控制元件呈現給使用者兩種互斥的選項或狀態，比如“是/否”或“開/關”。開關控制元件每次只能顯示兩個選項中的一個；使用者滑動該控制元件可以顯示隱藏的選項或狀態。圖 8-11 顯示了開關控制元件的例子。

圖 8-11 表格視圖中的開關控制元件



當您需要向使用者提供兩個簡單的，完全對立的選項時，您可以在分組表格視圖中使用開關控制元件。由於其中一個選項始終是隱藏的，因此最好在使用者已經知曉這兩個選項是甚麼的情況下才使用開關控制元件。換句話說，不要讓使用者僅僅為了弄清楚另一個選項是甚麼而滑動開關控制元件。

您可以使用開關控制元件來改變視圖中的其他使用者介面元素的狀態。根據使用者做出的選擇，可能會有新的列表項出現，或者有列表項消失，或者有的列表項變為有效或無效的狀態。

使用表格視圖實現常用的使用者操作

表格視圖是非常通用的使用者介面元素，因為它們可以通過不同的配置方式來支援不同的使用者操作，比如：

- 選項列表。在iPhone OS中沒有類似於選單或彈出式選單那樣的多項選擇控制元件，但是表格視圖能夠以一種簡單的，整潔的方式顯示選項列表，供使用者進行選擇。此外，表格視圖可以提供選定標記圖像，向使用者標識列表中的哪一項（或哪些項）處於被選中的狀態，如圖8-12所示。

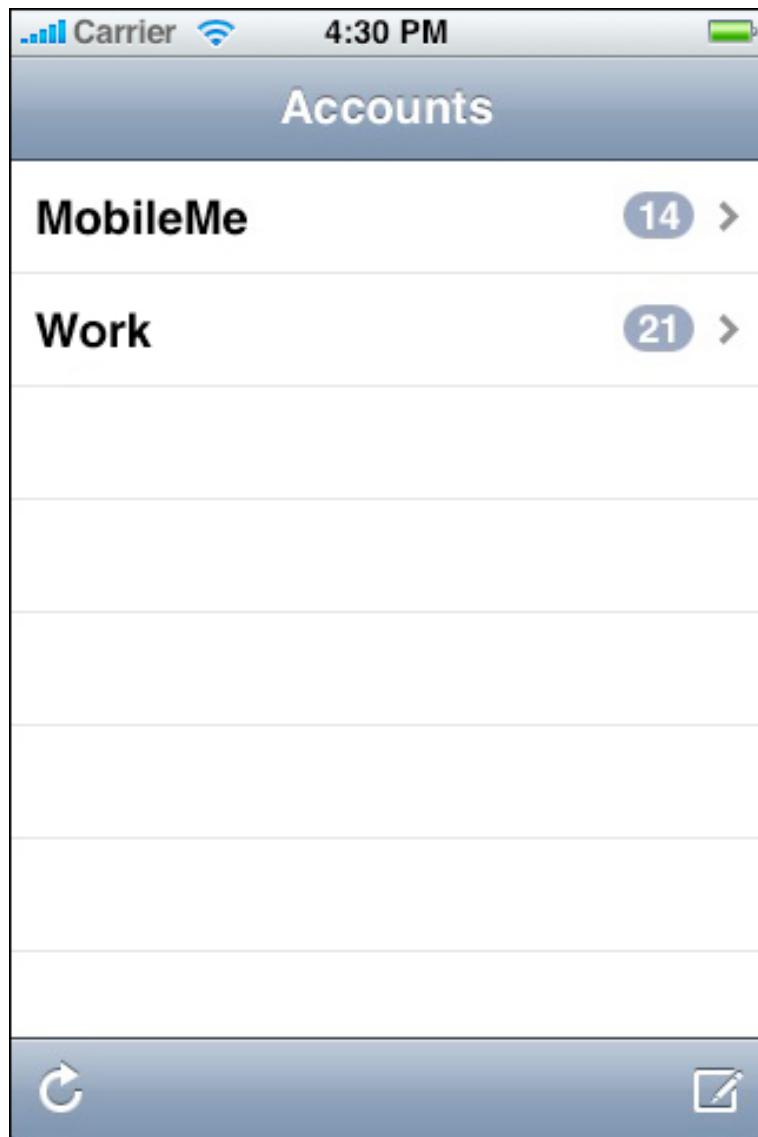
圖 8-12 選項列表中的選定標記



如果您需要在使用者點選表格視圖中的某行時向使用者顯示選項列表，您可以使用任意一種樣式的表格視圖。但是，如果您需要在使用者點選按鈕或其它位於表格行之外的使用者介面元素時向使用者顯示選項列表，則您應該選用無格式樣式的表格視圖。

- 導覽層次資訊。表格視圖適用於顯示有層次結構的資訊，也就是說每個節點（即列表項）可以包含它自己的子集，因為其子集可以顯示在一個單獨的列表中。使用者可以很輕鬆地沿著層次結構中的路徑，從每層列表中選擇一項。展開指示符表示使用者點選該行中的任意位置都可以在一個新列表中查看其子集，如圖8-13所示。

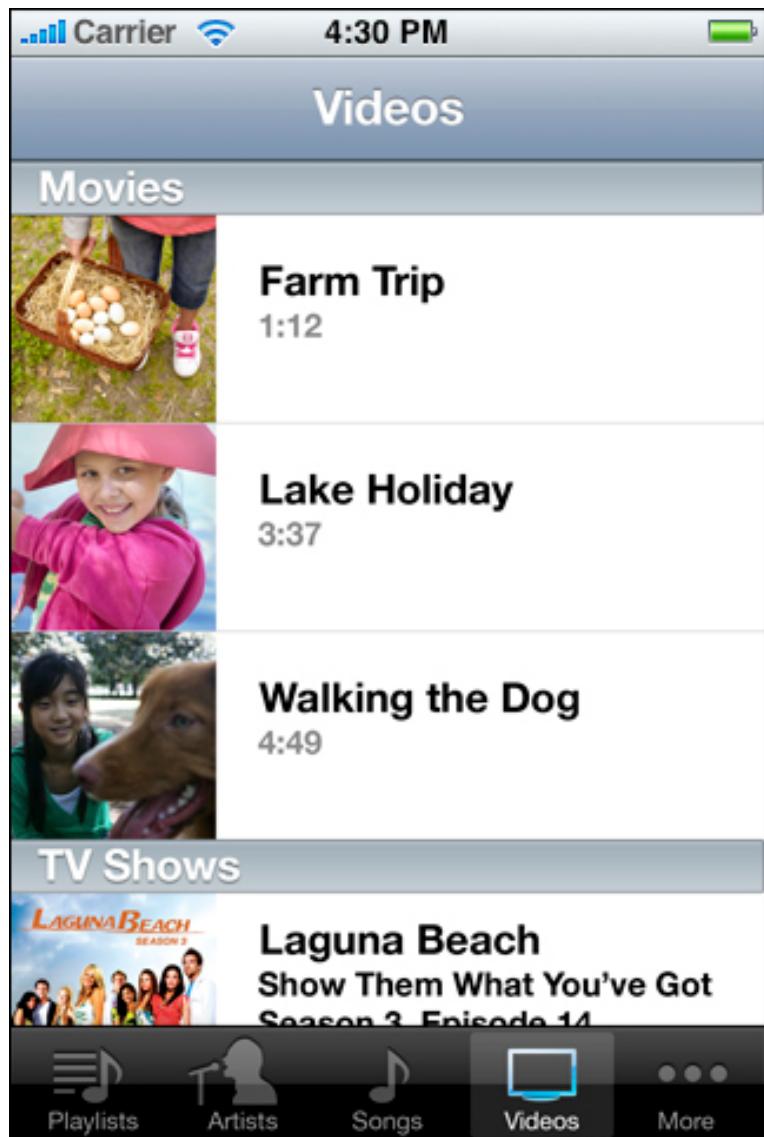
圖 8-13 展開指示符表示下一屏將顯示該行的資訊子集



如果一個表格視圖用於導覽時，當使用者按照層次結構回溯他們瀏覽過的內容時，先前被選中過的表格行不再保持高亮顯示。

- 查看按概念分組的資訊。您可以使用任意一種表格視圖樣式，將資訊歸類為不同的邏輯分組，比如工作，家庭或學校。無格式樣式表格和分組樣式表格都允許您通過提供頁眉和頁腳文字來為每一段提供上下文資訊，如圖8-14所示。

圖 8-14 無樣式樣式的表格視圖中通過頁眉來分組



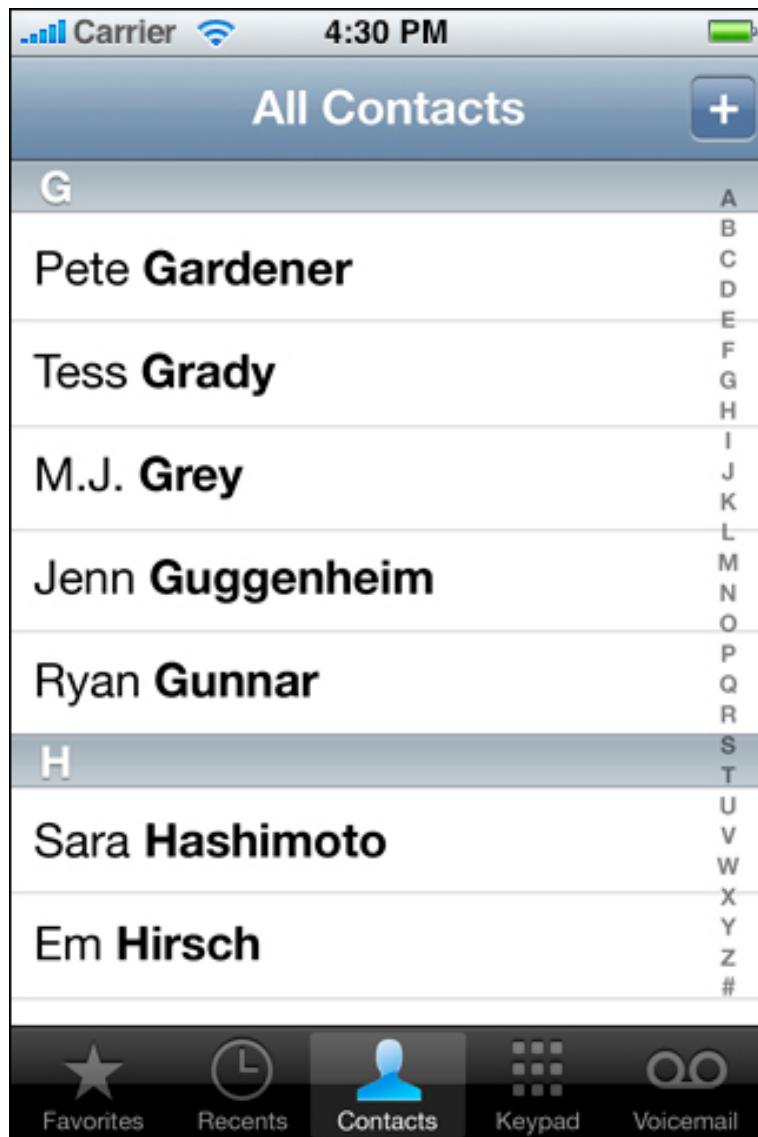
- 一般來說，分組樣式表格提供了更清晰的，可視化的分組表示，因為即使在快速滾動的情況下，使用者也很容易認出分組所俱有的圓形的四角。圖8-15顯示了iPod設置中一些概念相近的分組。

圖 8-15 分組樣式表格視圖可以包含多個分組



- 查看索引過的資訊。如果您正在使用無格式樣式的表格視圖，則您可以顯示索引來幫助使用者迅速找到他們需要的東西。索引由一列懸浮於螢幕右邊緣的條目（通常是字母表中的字母）組成，如圖8-16所示。使用者點選（或拖移至）一條索引項可以看到在列表中與之對應的區域。當列表的長度跨越數個螢幕時索引最為有用。

圖 8-16 包含索引的無格式樣式的表格視圖



如果您在無格式樣式表格中包含了索引，您就應該避免使用那些需要在表格的右邊緣顯示的表格視圖元素（比如展開指示符），因為這些元素會影響索引的使用。

文字視圖

文字視圖是一片能夠顯示多行文字的區域，並在內容過長超出其邊界範圍時可以滾動。郵件應用程式就使用了文字視圖讓使用者建立簽名，簽名會出現在使用者所編寫的每一封郵件的正文末尾處，如圖8-17所示。

圖 8-17 顯示多行文字的文字視圖



您不但可以使用文字視圖來顯示多行文字，如一份大文字文件的內容，而且您還可以使用文字視圖來支援使用者編輯。如果您提供的文字視圖是可編輯的，那麼當使用者在文字視圖內點選時會自動顯示鍵盤。鍵盤的輸入法和佈局取決於使用者的語言設置。當使用者點

選“.?123”鍵（如圖8-17所示）時，鍵盤會切換到數字鍵盤，使得數字和標點符號的輸入更加方便。您也可以指定不同的鍵盤樣式，這取決於您希望使用者所輸入的文字類型。關於可使用的鍵盤樣式請參考“文字框”。

您可以在文字視圖中控制文字的字體，顏色和對齊方式，但這些屬性會應用於整個文字。也就是說，您無法只對部分文字做出上述屬性的改變。正如您所期望的那樣，預設的字體和顏色採用系統字體和黑色，因為其可讀性最強。預設的對齊屬性為左對齊（您可以將其改為居中對齊或右對齊）。

如果您一定要在同一個文字視圖中使用不同的字體，顏色和對齊方式，您可以使用web視圖代替文字視圖，使用HTML來格式化文字。

Web視圖

Web視圖是應用程式螢幕上可以顯示豐富的HTML內容的一片區域。例如，郵件應用程式使用web視圖顯示郵件的內容，因為這種視圖可以包含比純文字更豐富的內容（圖8-18顯示了這樣的一個例子）。

圖 8-18 Web視圖可以顯示基於web的內容



除了顯示web內容之外，web視圖還提供了一些元素來支援使用者瀏覽開放的網頁。雖然您可以選擇向使用者提供網頁瀏覽功能，但是最好避免讓您建立的應用程式看起來像是一個小型web瀏覽器。

如果您擁有一個網頁或web應用程式，您可以選擇使用web視圖對其進行包裝，實現一個簡單的iPhone應用程式。如果您打算訪問由您操控的web內容，請首先閱讀Safari Web內容指南，確保您已經知道對於iPhone OS裝置，如何建立與之兼容並針對其顯示進行過優化的web內容。

應用程式控制元件

iPhone OS為您提供了若干控制元件，您可以在自己的應用程式中使用它們，並且這些控制元件中的絕大部分已經被iPhone OS裝置的使用者所熟知。這些控制元件有很多只能用於特定位置，比如表格視圖，但也有一些控制元件具備一定的通用性。本章將向您介紹可以在應用程式的任意視圖中使用的控制元件。

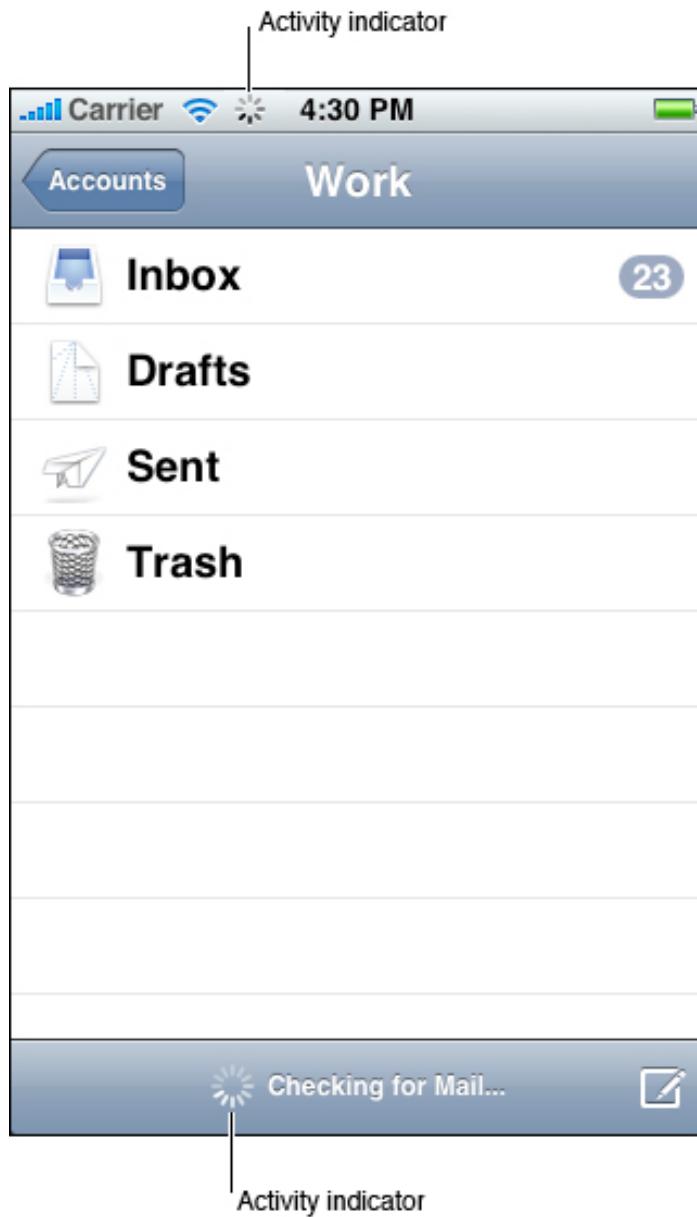
當您設計應用程式的使用者介面時，永遠要牢記一點：對於熟悉的控制元件，使用者期望它們俱有與內置應用程式中一樣的行為。只要您在應用程式中適當地使用這些控制元件，一定會為自己的程式增色不少。

活動指示器

活動指示器表示有一項持續時間未知的任務或過程正在進行當中。如果已知一項任務的持續時間，需要顯示該任務的進度，請使用進度視圖（更多關於該控制元件的資訊請參考“進度視圖”）。活動指示器看起來像一個“旋轉的齒輪”，告訴使用者處理正在進行中，但不知何時能夠完成。

圖9-1顯示了兩種活動指示器。狀態欄上的是網路活動指示器；如果您的應用程式訪問網路時超過了一定的秒數，狀態欄就會顯示該指示符。工具欄上較大一點的活動指示器在您的應用程式執行當前任務的時間多於1秒或2秒時顯示。

圖 9-1 兩種活動指示器



活動指示器是一種很好的回饋機制，它告訴使用者他們的任務或過程並沒有終止，而不是說明任務何時結束。

您可以選擇活動指示器的大小和顏色來配合它所在的視圖的背景。預設情況下，活動指示器是白色的。

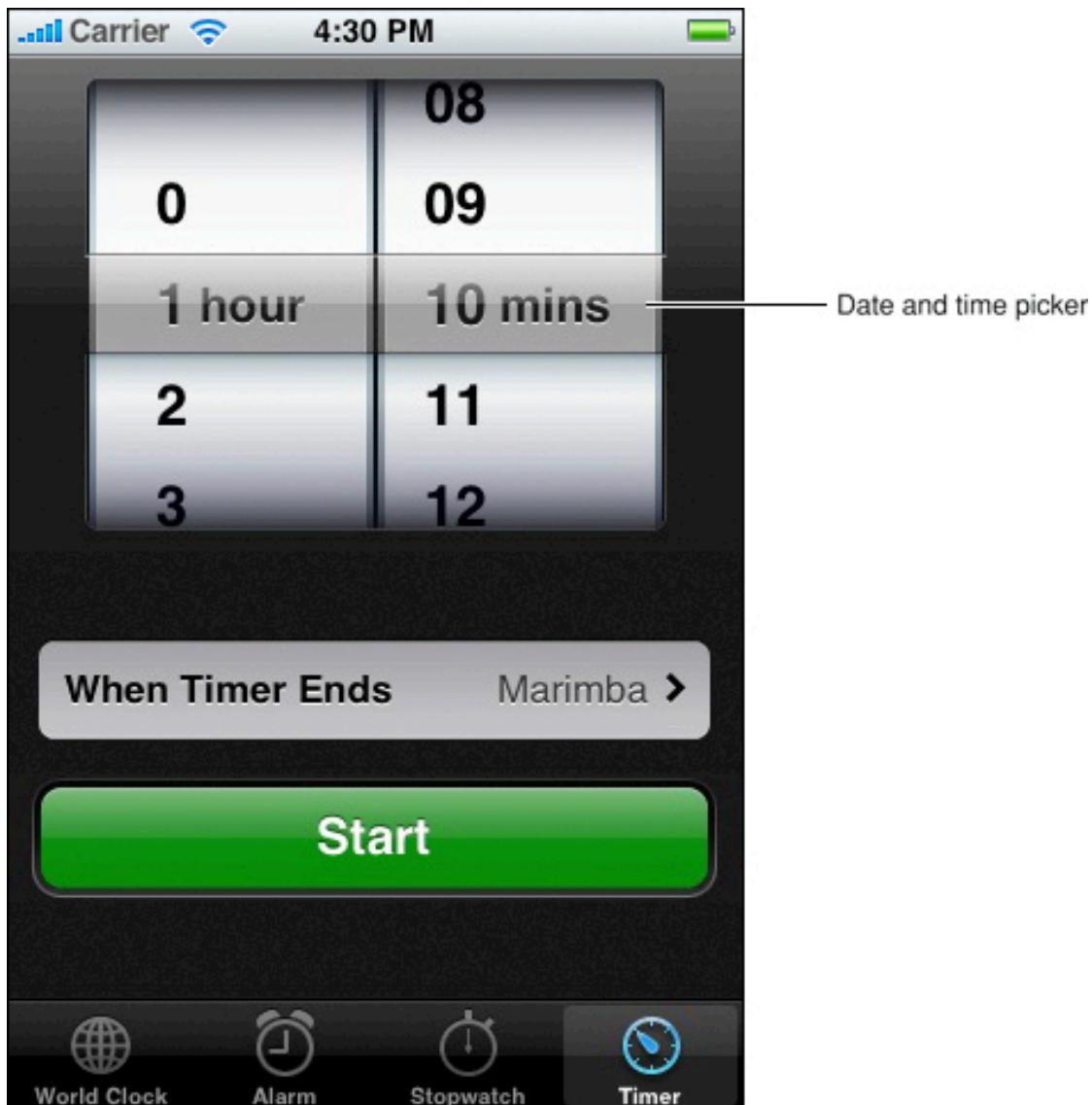
當任務或過程已經完成時，活動指示器就會消失。推薦您使用這種預設行為，因為使用者期望在有動作發生時看到活動指示器，而且他們會將靜止不動的活動指示器與停滯的過程聯想到一起。

要瞭解如何顯示網路活動指示器，請參考UIApplication類參考中的networkActivityIndicatorVisible方法。要瞭解如何在您的程式碼中顯示一個較大的非網路活動指示器，請參考UIActivityIndicatorView類參考。

日期時間選擇器

日期時間選擇器為使用者提供了一種簡單的選擇特定日期或時間的方式。日期時間選擇器最多可以有4個獨立的轉輪，每個輪只顯示一種類型的值，比如月份或小時。使用者通過點選或拖拽使轉輪轉動，直到它將使用者想要的值顯示在選擇器中間的一條突出顯示的選擇欄上。最終的值由各個輪上的值組合而成。圖9-2顯示了一個日期時間選擇器的例子。

圖 9-2 日期時間選擇器



使用日期時間選擇器可以避免使用者輸入由多段組成的值，比如一個日期的年，月，日。日期時間選擇器使用起來很方便，因為每一部分的值都有相對小的範圍，並且使用者知道這些值應該是多少。

日期時間選擇器會根據您指定的不同模式顯示出不同數量的轉輪，每個轉輪都有一組不同的值。日期時間選擇器定義了如下模式：

- 時間 時間模式下顯示小時和分鐘的轉輪，以及一個指定上午/下午的可選輪。
- 日期 日期模式下顯示月，日和年的轉輪。

- **日期和時間** 日期和時間模式下顯示日曆日期，小時和分鐘的輪，以及一個指定上午/下午的可選輪。這也是預設的模式。
- **倒數計時器** 倒數計時器模式下顯示小時和分鐘的轉輪。您可以指定倒計時的總時長，最長為23小時59分鐘。

預設情況下，分鐘輪顯示60個值（0到59）。但是，如果您想要比較粗略地顯示時間，您可以設置分鐘輪，使其每隔幾分鐘顯示一個值，只要時間間隔的值是60的因數就可以。例如，您可以以一刻鐘為間隔，顯示0, 15, 30和45。

不管如何設置，日期時間選擇器的整體大小是固定的，與鍵盤的尺寸相同。您可以選擇將日期時間選擇器設為視圖中的焦點元素，或者讓它只在需要時出現。例如，內置的時鐘應用程式的計時器模式會顯示一個總是可見的日期時間選擇器，這是因為時間的選擇對“定時器”的功能起至關重要的作用。另一方面，“設置日期和時間”首選項（在您關閉“自動設置”的情況下，可在“設置”>“一般”>“日期和時間”找到該選項）只顯示臨時的日期時間選擇器，顯示與否取決於使用者是否想要設置日期或時間。

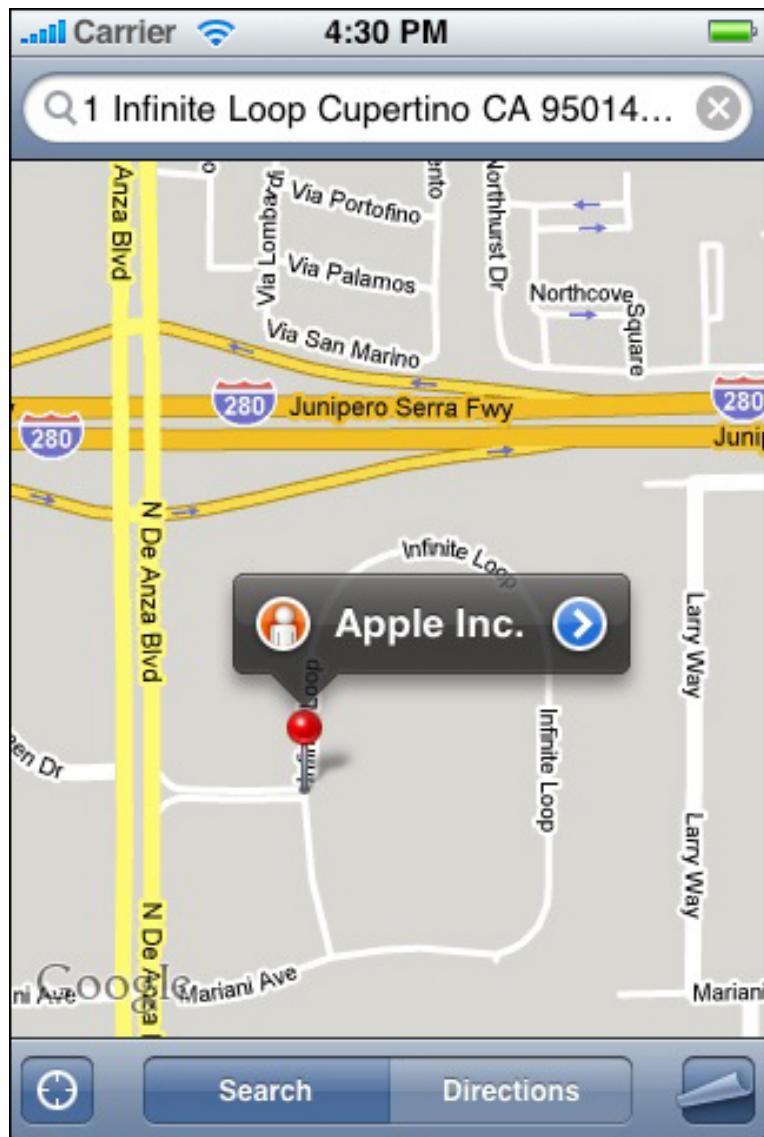
要瞭解更多有關在您的程式碼中使用日期時間選擇器的內容，請參考UIDatePicker類參考。

詳細資訊展開按鈕

詳細資訊展開按鈕提供了關於某一項的額外資訊或更詳細的資訊。通常用於表格視圖中，向使用者提供查看某個列表項（請參考“表格視圖元素”查看更多關於該用法的資訊）的詳細資訊的方法。然而，您也可以在其他類型的視圖中使用該元素，來提供一種顯示更多資訊或功能的方法。

例如，地圖應用程式顯示了一個詳細資訊展開按鈕，使用者可以點選該按鈕來訪問和大頭針所釘位置相關的更多功能。圖9-3顯示了一個詳細資訊展開按鈕的例子。

圖 9-3 顯示額外的詳細資訊或功能的詳細資訊展開按鈕



要瞭解更多有關在您的程式碼中使用詳細資訊展開按鈕的內容，請參考UIButton類參考。

資訊按鈕

資訊按鈕提供了一種顯示應用程式配置資訊詳情的方法，它通常位於螢幕的背面。因此，資訊按鈕特別適合於工具應用程式。您可以在天氣應用程式（圖9-4所示）的右下角看到一個資訊按鈕的例子。

圖 9-4 資訊按鈕用於顯示資訊，通常是配置資訊詳情



資訊按鈕提供了亮色和暗色兩種背景。亮色背景樣式（如圖9-4所示）在暗色背景的視圖中效果比較好。相反，暗色背景的資訊按鈕在亮色背景的視圖中效果更佳。

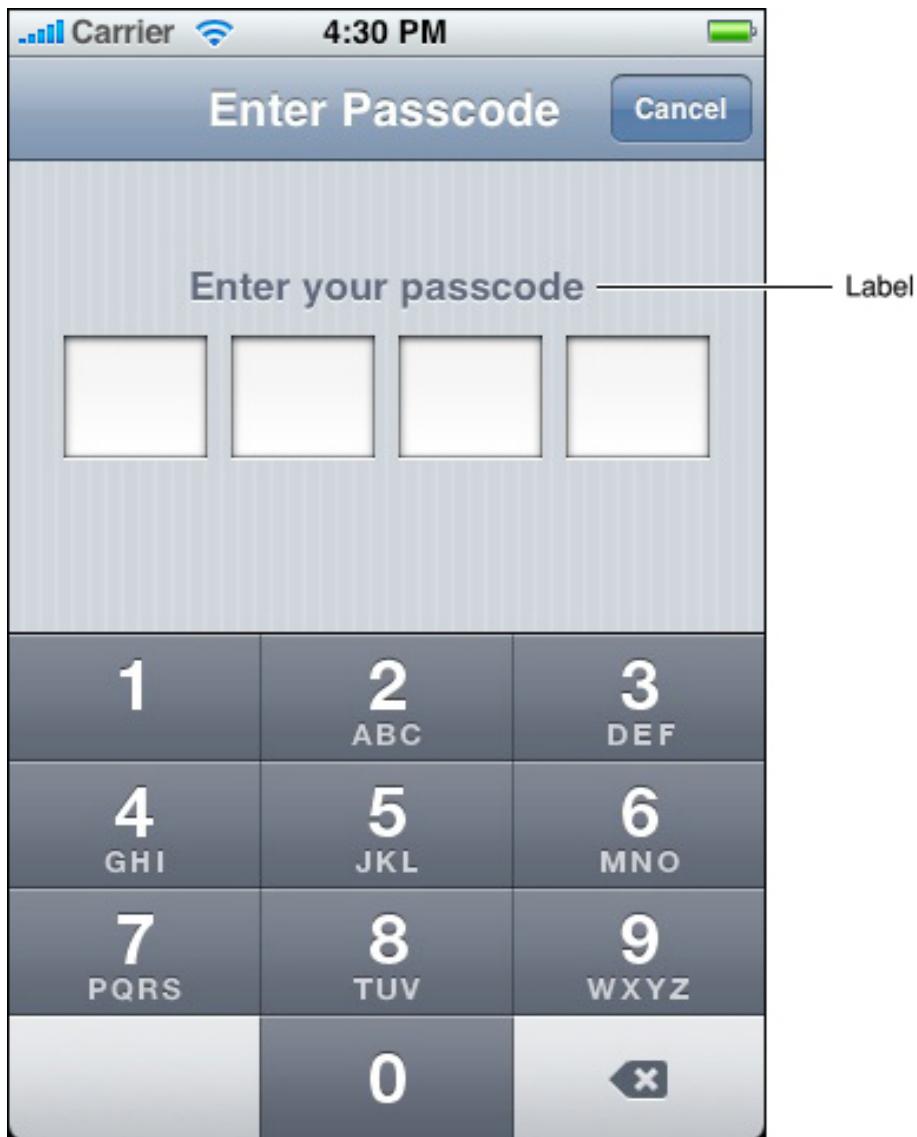
資訊按鈕會在使用者點選它時短暫發光。當您使用iPhone OS提供的資訊按鈕時，您將自動獲得這種處於被按下狀態下的外觀。

要瞭解有關在您的程式碼中使用資訊按鈕的更多內容，請參考UIButton類參考。

標籤

標籤是一種大小可變的靜態文字。圖9-5顯示了一個標籤的例子。

圖 9-5 為使用者提供資訊的標籤



您可以使用標籤為使用者介面的各部分命名，或者向使用者提供一定的幫助資訊。標籤最適合顯示數量相對較少的文字。

您可以設定標籤文字的各種屬性，比如字體，文字顏色和對齊方式，但最重要的是，您要確保您的標籤清晰易讀。不要為了華麗的字體或鮮艷的顏色而犧牲文字的清晰度。

當您編寫標籤的文字時，一定要使用使用者的詞彙。請檢查您的應用程式中的文字，如果存在以開發人員為中心的詞語，請使用以使用者為中心的詞語替換它們。

要瞭解有關在您的程式碼中使用標籤的更多內容，請參考UILabel類參考。

頁面指示符號

頁面指示符號為當前在應用程式中打開的每一個視圖顯示一個小圓點。這些小圓點從左到右的順序表示了視圖被打開的順序（最左邊的小圓點代表第一個視圖）。對於當前螢幕上顯示的視圖，用發光的小圓點加以表示。使用者點選發光小圓點的左右兩側可以查看上一個或下一個打開的視圖。圖9-6顯示了一個頁面指示符號的例子。

圖 9-6 頁面指示符號



頁面指示符號為使用者提供了一種快捷的方法來查看當前有多少打開的視圖，同時也能表示出這些視圖被打開的先後順序；它並不能幫助使用者追蹤他們在視圖層次中所處的位置。因為實用型應用程式中的視圖彼此之間往往是對等的，所以頁面指示符號能夠充分幫助使用者在各視圖之間進行導覽。另一方面，能夠顯示層次資訊的效率型應用程式應該通過導覽欄（請參考“導覽欄”獲取更多資訊）上的元素提供導覽資訊。

典型情況下，頁面指示符號最好出現在應用程式螢幕的下邊緣，位於它所包含的視圖的下方。這樣就可以把螢幕的上面部分留給更重要的資訊（視圖本身），使使用者很容易看到。一定要確保頁面指示符號垂直居中於視圖的下邊緣和螢幕的下邊緣之間。

雖然對於在頁面指示符號中可以顯示的小圓點的數量沒有規劃性的限制，但是當頁面越來越多時，指點之間不能過於密集。例如，當裝置處於縱向時，您只能在一個頁面指示符號上顯示至多20個小圓點。因此，您應該在應用程式中避免發生這種情況。

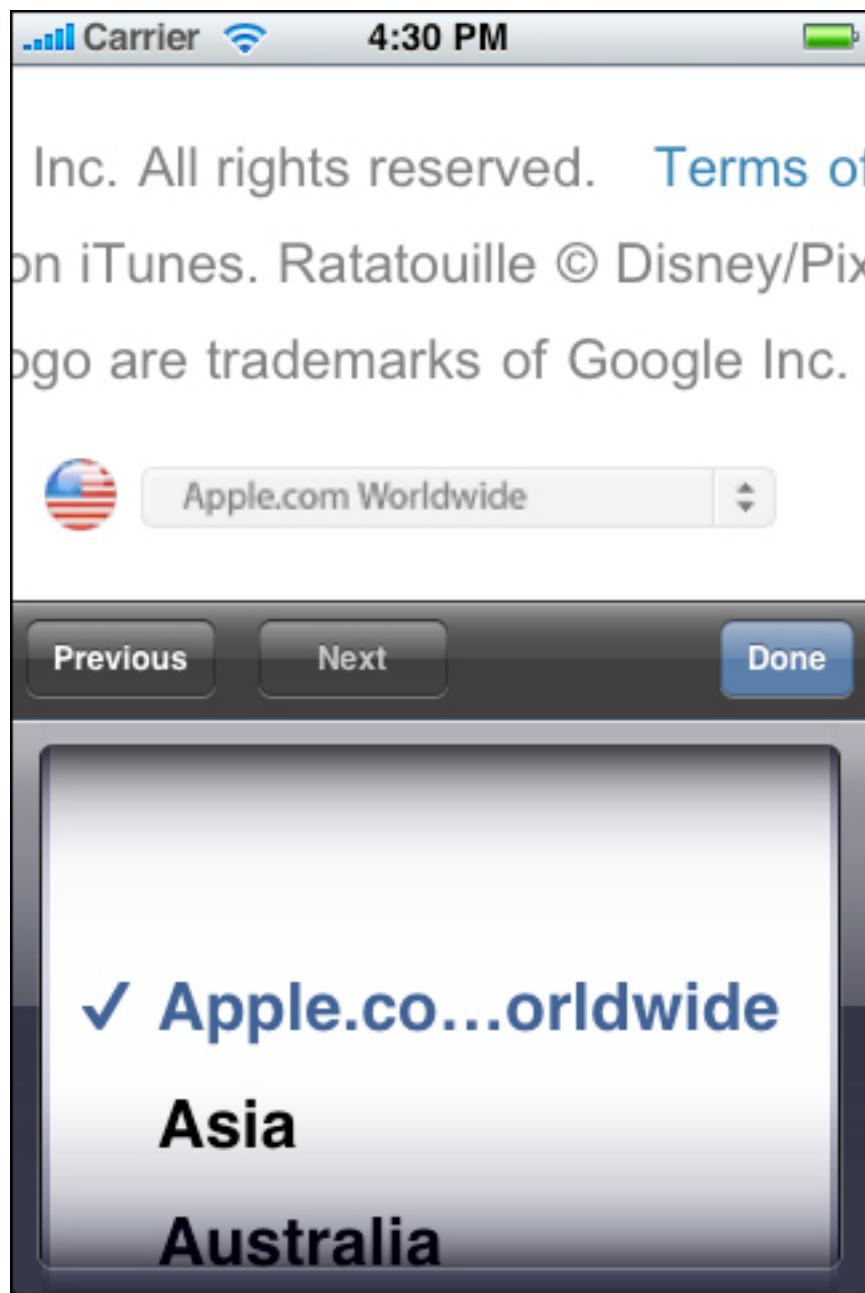
雖然您可以在只有一個打開的視圖時隱藏頁面指示符號，但預設的行為是顯示它。

要瞭解有關在您的程式碼中使用頁面指示符號的更多內容，請參考UIPageControl類參考。

選擇器

選擇器是日期時間選擇器（關於該控制元件的更多資訊請參考“日期時間選擇器”）的通用版本。您可以使用選擇器來顯示任意一組值。與日期時間選擇器一樣，使用者旋轉選擇器的單個或多個轉輪直到出現他們想要的值。圖9-7顯示了一個單輪的選擇器。

圖 9-7 iPhone的Safari中出現的選擇器



當您決定是否在應用程式中使用選擇器時，要考慮到當轉輪靜止時，輪上的許多甚至絕大多數數值對使用者來說是隱藏的。但這不一定是個問題，特別是在使用者已知這些值是多少的情況下。例如，在日期時間選擇器中，使用者瞭解在月份輪上隱藏的值只能是1到12之間的數字。然而，如果您要提供的選項不是像這樣的一組眾所周知的值，那麼使用選擇器控制元件可能並不合適。

如果您需要顯示特別多數量的值，則您應該用表格視圖將這些值列出，而不是用選擇器。這是因為表格視圖俱有更高的高度，能夠更快速地進行滾動。

如果您需要為選擇器中的值提供上下文資訊，比如計量單位，請將它顯示在水平地穿過控制元件中心的半透明選擇欄中。不要在選擇器上方或轉輪本身上顯示這樣的標籤。有關正確顯示標籤的例子，請參考內置的時鐘應用程式的“定時器”功能，可以看到在使用者選擇的值旁邊顯示了單位“小時”和“分鐘”。

與日期時間選擇器一樣，通用的選擇器可以一直可見（作為您的使用者介面的焦點）或者只在需要時出現。選擇器的整體大小，包括它的背景，都是固定的，與鍵盤的尺寸相同。

要瞭解更多有關在您的程式碼中使用選擇器的內容，請參考UIPickerView類參考。

進度視圖

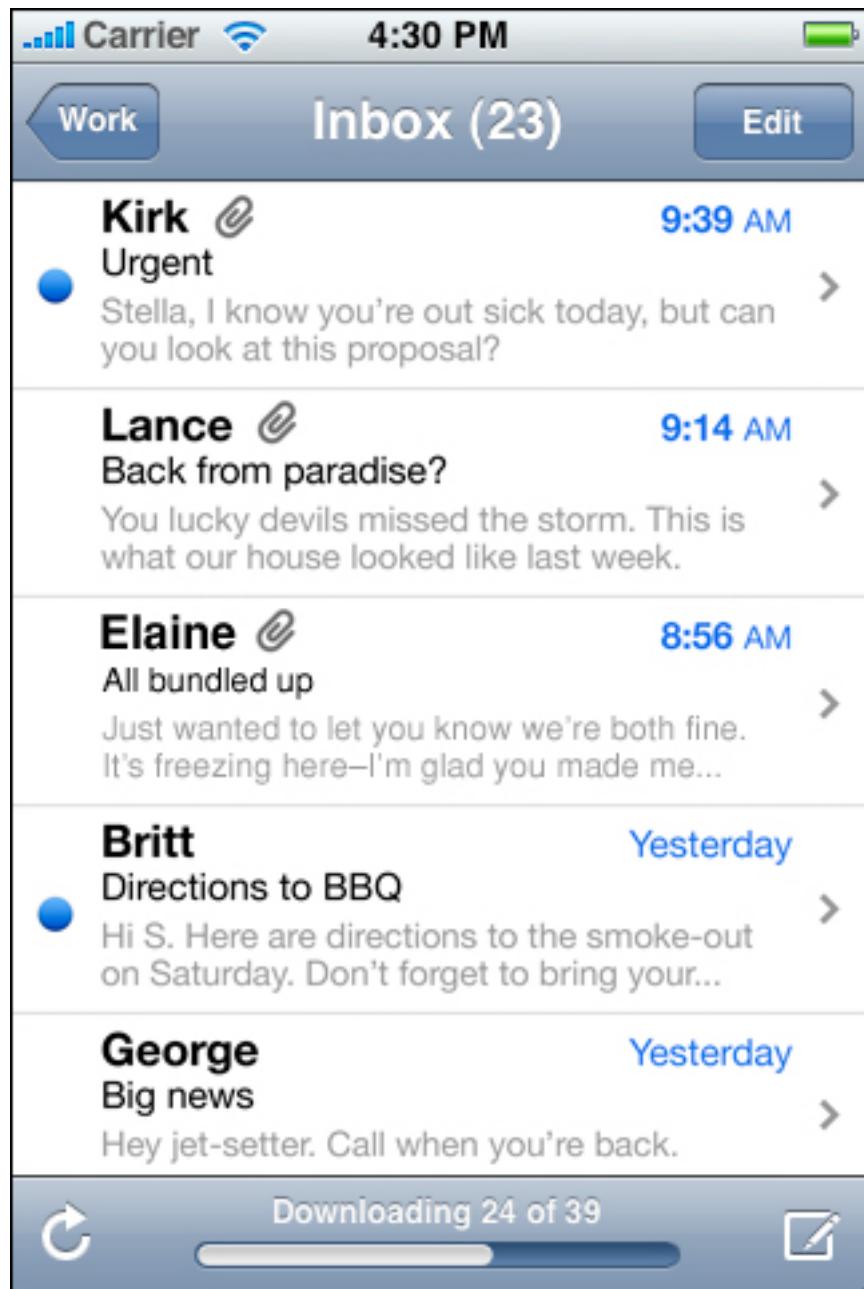
進度視圖顯示俱有確定持續時間的任務或過程的進度。如果您需要顯示未知持續時間的任務的進度，請使用活動指示器（關於該控制元件的更多資訊請參考“活動指示器”）。

iPhone OS提供了兩種樣式的進度視圖，分別是預設樣式和欄樣式。這兩種樣式的外觀非常相似，區別在於高度不同：

- 預設樣式用於應用程式的主內容區。
- 欄樣式比預設樣式薄一些，非常適合在工具欄中使用。例如，在郵件應用程式中，當使用者下載新郵件或發送電子郵件時，工具欄上就會出現欄樣式的進度視圖。

兩種樣式進度視圖的行為是相同的。隨著任務或過程的執行，進度視圖會從左到右被填滿。在任意一刻，視圖中已填充和未填充區域的比例都向使用者表明該任務或過程還有多久能夠完成。圖9-8 顯示了一個欄樣式進度欄的例子。

圖 9-8 工具欄上的欄樣式進度視圖



進度視圖是那些俱有明確持續時間的任務向使用者提供回饋資訊的好方法，尤其是在向使用者顯示一項任務大約會執行多久十分重要的情況下。當您顯示進度視圖時，使用者會得知他們的任務正在執行，並且使用者獲得了足夠的資訊，他們可以決定是否等待任務完成，或者取消該任務。

要瞭解更多有關在您的程式碼中使用進度視圖的內容，請參考UIProgressView類參考。

圓角矩形按鈕

圓角矩形按鈕是一種多功能的按鈕，您可以在視圖中使用它來執行一個動作。您可以在每一個聯絡人視圖的底部看到這種按鈕的例子：文字消息和新增到收藏櫃按鈕就是圓角矩形按鈕，如圖9-9所示。

圖 9-9 執行應用程式特定動作的圓角矩形按鈕



當您為圓角矩形按鈕建立標題時，請務必：

- 使用標題式的大寫方式（即每個單詞都大寫，但冠詞，並列連詞和五個字母以下的介詞除外）
- 避免建立過長的標題。長度過長的文字會被截斷，這樣會使使用者很難理解。

要瞭解更多有關在您的程式碼中使用圓角矩形按鈕的內容，請參考UIButton類參考。

搜尋欄

搜尋欄是一片接收使用者輸入的文字的區域，您的應用程式可以以搜尋欄中的文字為輸入進行搜尋。當使用者點選搜尋欄時，會出現一個鍵盤；當使用者鍵入要搜尋的條目後，輸入的資訊將按照應用程式指定的方式處理。（關於在應用程式中處理搜尋，請參考“提供搜尋功能並顯示搜尋結果”。）

預設情況下，搜尋圖示顯示在搜尋欄的左側。此外，搜尋欄還可以顯示一些可選的元素：

- 佔位符文字。該文字可以用來顯示控制元件的功能（例如“搜尋”），或者用來提醒使用者他們所處的搜尋環境（例如“YouTube”或“Google”）。
- 書籤按鈕。該按鈕可以提供資訊的快捷方式，使得下一次使用者也能夠輕鬆地找到這些資訊。例如，通過地圖應用程式搜尋模式中的書籤按鈕，使用者可以訪問已儲存在書籤的地點，最近的搜尋和聯絡人。
- 清除按鈕。大多數搜尋欄都包含一個“清除”按鈕，允許使用者通過點選該按鈕清除搜尋欄中的內容。
- 一個可以被稱作提示的描述性標題，位於搜尋欄的上方。舉例來說，這種提示可以是一個短語，為搜尋欄提供簡要介紹或應用程式特定的環境資訊。

圖9-10顯示了一個包含自定義佔位符文字，書籤按鈕和預設搜尋圖示的搜尋欄。

圖 9-10 帶有可選的佔位符文字和書籤按鈕的搜尋欄



預設情況下，書籤按鈕和清除按鈕會按下列方式對彼此產生影響：

- 當搜尋欄包含非佔位符文字時，清除按鈕是可見的，使用者可以使用它來清除文字。如果搜尋欄中不存在使用者提供的文字或非佔位符文字，清除按鈕是隱藏的，因為沒必要清除搜尋欄的內容。
- 只有當搜尋欄中不存在使用者提供的文字或非佔位符文字時，書籤按鈕才是可見的。這是因為當搜尋欄中存在使用者可能想要清除的文字時，清除按鈕是可見的。

您可以通過指定一種標準顏色的背景樣式來自定義搜尋欄，比如：

- 藍色（與工具欄和導覽欄的預設外觀一致的預設漸變顏色）。預設的背景樣式如圖9-10所示。
- 黑色

此外，您可以在搜尋欄的下方顯示一個範圍欄，其中包含一些按鈕，使用者可以點選這些按鈕來選擇搜尋的範圍。範圍欄的外觀與您所指定的搜尋欄外觀相同，並且由您為範圍欄提供自定義的標題。

不論裝置的方向如何，範圍欄總是顯示在搜尋欄的下方，除非您在程式碼中使用搜尋顯示控制器（詳細資訊請參考[UISearchDisplayController類參考](#)）。當您使用搜尋顯示控制器時，範圍欄會顯示在搜尋欄的內部，如果裝置橫向放置，則範圍欄位於搜尋字段的右側（如果裝置縱向放置，則範圍欄位於搜尋欄的下方）。

要瞭解更多有關在您的程式碼中使用搜尋欄和範圍欄的內容，請參考[UISearchBar類參考](#)。

分段控制元件

分段控制元件是處在一條直線上的各段的集合，其中的每一段都相當於一個按鈕，可以顯示不同的視圖。當使用者點選分段控制元件中的一段時，應該產生一個瞬時的動作或可見的結果。例如，當使用者使用分段控制元件選擇一種郵件協議時，設置應用程式將顯示不同的資訊，如圖9-11所示。

圖 9-11 分為三個段的分段控制元件



分段控制元件的長度取決於您所顯示的段的數量以及最長一段的尺寸。分段控制元件的高度是固定的。雖然您可以指定要顯示的段的數目，但要注意，使用者必須能夠舒服地點選到任何一段，而無需擔心觸碰到相鄰的段。由於點選區域應該是 44×44 像素，因此建議將分段控制元件分為5段或5段以下。

分段控制元件可以包含文字或圖像；每一個單獨的分段要麼包含文字，要麼包含圖像，但不能同時包含兩者。一般來說，最好避免在同一個分段控制元件中混合使用文字和圖像。

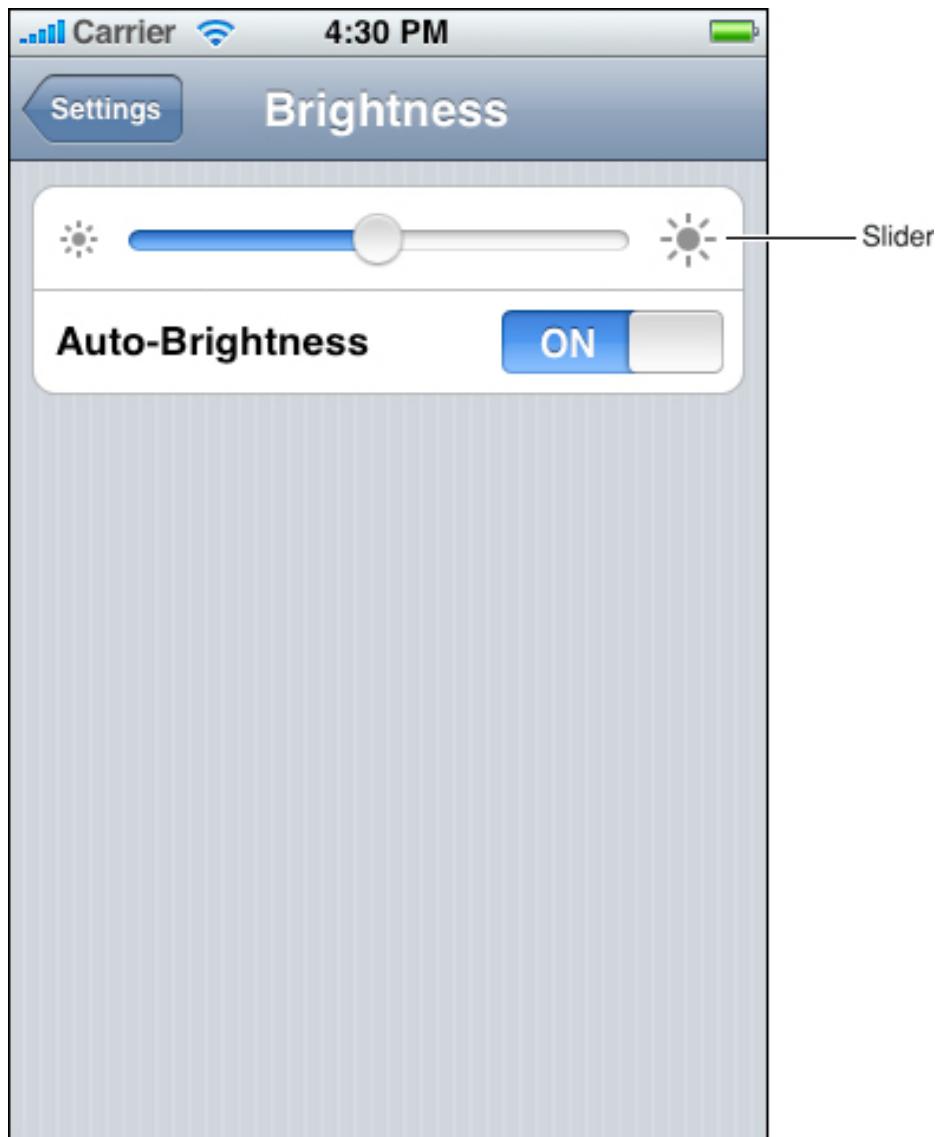
分段控制元件要保證每一段的寬度與段的總數相稱。這意味著您要確保為每一段設計的內容在尺寸上要大致相等。

要瞭解有關在您的程式碼中使用分段控制元件的更多內容，請參考[UISegmentedControl](#)類參考。

滑塊

滑塊允許使用者在允許的值的範圍內對一個值或一個過程進行調整。當使用者拖拉滑塊時，相應的值或過程也不斷被更新。圖9-12顯示了一個帶有最小值和最大值的圖像滑塊的例子。

圖 9-12 滑塊



滑塊主要在下面兩種情況下很有用：

- 當您希望使用者能夠對他們選擇的值進行較高精度的控制時
- 當您希望使用者能夠對當前的過程進行較高精度的控制時

滑塊由軌道，滑塊以及可選的左右兩邊值的圖像組成。圖9-13顯示了滑塊的各個部分。

圖 9-13 滑塊的四部分



您可以設定滑塊的寬度，使其適合於您的應用程式的使用者介面。此外，您可以選擇水平地或垂直地顯示滑塊。

有以下幾種方法自定義滑塊：

- 您可以定義滑塊的外觀，以便使用者可以一眼辨認出滑塊是否處於活動狀態。
- 您可以提供在滑塊兩端出現的圖像（通常，它們分別對應最小值和最大值），以此幫助使用者理解滑塊的作用。例如，控制字體大小的滑塊可以在最小值一端顯示一個非常小的字符，而在最大值一端顯示一個特別大的字符。

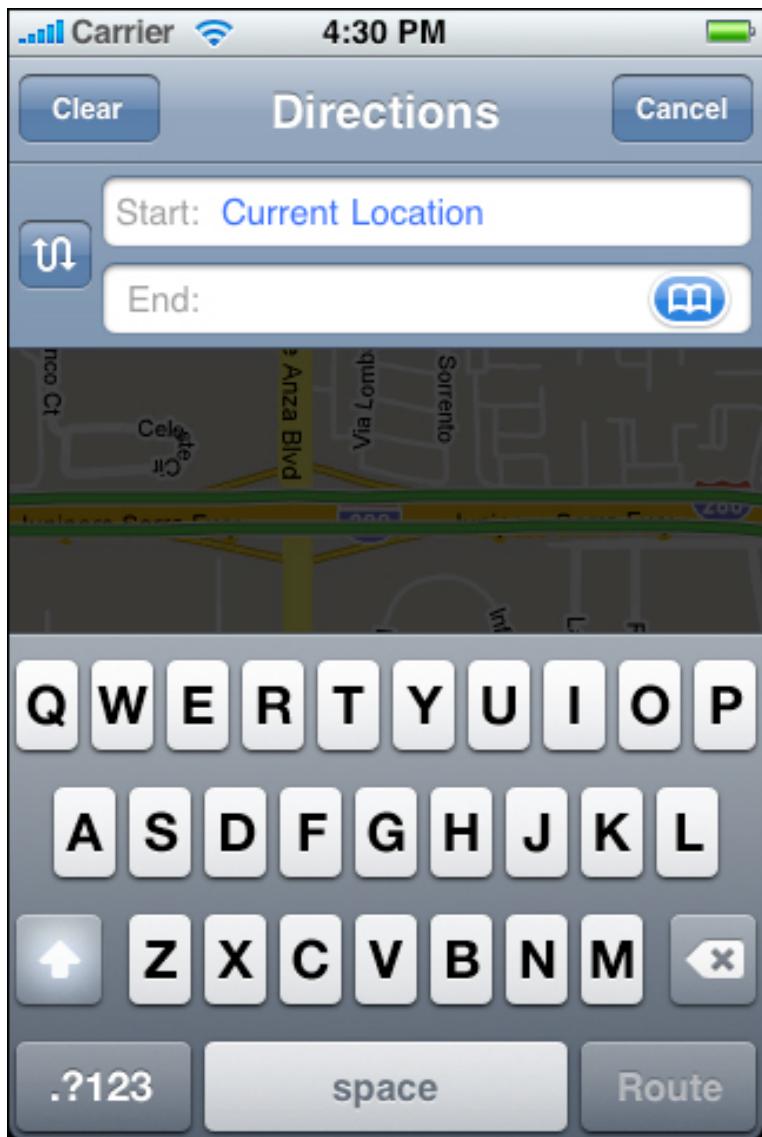
- 您可以根據滑塊所在的位置和控制元件所處的狀態，為軌道定義不同的外觀。

要瞭解有關在您的程式碼中使用滑塊的更多內容，請參考[UISlider類參考](#)。

文字框

文字框是一個接收使用者輸入的圓角矩形的區域。當使用者點選文字框時，會出現一個鍵盤；當使用者點選鍵盤上的Return鍵時，文字框會以應用程式指定的方式處理使用者的輸入資訊。文字框可容納單行的輸入。圖9-14顯示了地圖應用程式中的兩個文字框。

圖 9-14 可接收使用者輸入的文字框



您可以使用自定義文字框來幫助使用者在您的應用程式中進行輸入。例如，您可以在文字框的左側或右側顯示自定義的圖像，或者顯示一個系統提供的按鈕，比如圖9-14中的“書籤”按鈕。在一般情況下，您應該在文字框的左端指明該文字框的用途，而在右端顯示一些附加的功能，比如書籤。

您也可以在文字框的右端顯示清除按鈕。當該按鈕存在時，不論您在它的上面顯示甚麼圖像，點選該按鈕都會清除文字框的內容。

有時候，如果文字框顯示一些提示資訊，比如“姓名”，則有助於使用者理解這個文字框的用途。文字框支援顯示佔位符文字，它們可以在文字框中沒有其他文字時顯示。要瞭解與使用文字框，以及自定義顯示圖像和按鈕的文字框的詳情，請參考UITextField類參考。

您可以指定不同的鍵盤樣式，來迎合使用者輸入的內容的不同類型。（注意，您不能控制鍵盤的輸入法和佈局，這些是由使用者的語言設置決定的。）例如，你可能希望使用者能夠方便地輸入URL地址，PIN密碼或電話號碼。**iPhone OS**提供了幾種不同的鍵盤類型，這樣設計是為了方便不同類型的輸入。要瞭解可供您使用的鍵盤類型，請參考**UIKeyboardType**。要瞭解更多有關在您的應用程式中管理鍵盤的內容，請閱讀**iPhone**應用程式程序指南中的“鍵盤管理”一節。

系統提供的按鈕和圖示

為了打造一致的使用者體驗（也為了讓您的工作更加輕鬆），iPhone OS提供了許多用於導覽欄和工具欄的標準按鈕，以及用於標籤欄的圖示。

本章介紹了可供您使用的標準圖示和按鈕，以及如何正確地使用它們。不論您正在開發哪種類型的應用程式，您都應該熟悉本章中所介紹的按鈕和圖示，這有助於您：

- 正確地使用系統提供的元素
- 避免設計出與系統提供的圖示過於相似的自定義圖示

使用系統提供的按鈕和圖示

在iPhone OS中，許多標準的工具欄和導覽欄按鈕，標籤欄上的項目以及在內置應用程式中常見的通用按鈕都是可用的。在圖10-1中，您可以在郵件應用程式的工具欄上看到一些標準工具欄按鈕。

圖 10-1 郵件應用程式工具欄上的標準按鈕



像圖10-1所示的這幾個按鈕—“更新”，“組織”，“刪除”，“回復”和“編寫”，早已被許多內置的應用程式所使用，因此，使用者非常熟悉它們的含義和用法。這意味著，如果您的應用程式支援這些功能，您就可以利用使用者對它們的熟悉，使得應用程式的使用者介面更加簡潔而高效。反之，這也意味著，如果您將這些按鈕與其他任務聯繫起來，很可能會使使用者覺得迷惑和不適應，因為使用者以為這些按鈕仍俱有他們所熟知的功能，但實際上它們會執行另外的功能。

除了可以充分利用使用者以往的使用經驗之外，使用系統提供的按鈕和圖示還能夠帶來另外兩大好處，具體為：

- 縮短開發時間，這是因為您不需要為標準功能建立自定義的圖形元素。

- 增強使用者介面的穩定性，即使日後iPhone OS的更新版本改變了標準圖示的外觀也沒有問題。換句話說，您盡可信賴於標準圖示，即使外觀有所變化，它們代表的意思始終保持不變。

值得重申的是，為了利用使用者熟悉的優勢，使用更短的開發時間並保持使用者介面的語義一致性，您必須恰當地使用這些按鈕和圖示。具體來說，這意味著您應該按照按鈕或圖示規定的含義和推薦位置來使用它們，而不是依靠您對其外觀的臆測。請參考“用於工具欄和導覽欄的標準按鈕”，“用於標籤欄的標準圖示”和“用於表格行和其他使用者介面元素的標準按鈕”等小節，來瞭解關於系統提供的按鈕和圖示的含義和位置資訊。

Interface Builder可以很方便地使用系統提供的按鈕，並將系統提供的圖示應用到您的控制元件中。您可以參考Interface Builder使用者指南中的“iPhone OS使用者介面對象”部分，來瞭解關於外觀的資訊。

如果任何系統提供的工具欄按鈕，導覽欄按鈕或標籤欄項目圖示都無法恰當地反映您的應用程式中的某種特定功能，那麼您應該設計自定義的按鈕或圖示。“導覽欄，工具欄和標籤欄的圖示”小節為您提供了一些指導，能夠幫助您完成這項工作。

用於工具欄和導覽欄的標準按鈕

iPhone OS為您準備了在工具欄和導覽欄中常見的標準按鈕。表10-1所示的這些按鈕俱有兩種樣式，分別用於下列不同情況：

- 邊框樣式—例如，電話聯絡人應用程式導覽欄中的“新增”按鈕。這種樣式對於導覽欄和工具欄都適用。
- 無格式樣式—例如，郵件應用程式工具欄中的“編寫”按鈕。這種樣式只適用於工具欄。事實上，即使您指定一個導覽欄中的按鈕為無格式樣式，它也會被轉換為邊框樣式。

如同所有系統提供的按鈕一樣，請不要將表10-1中的按鈕用於其設計用途之外的操作。尤其要避免只根據外觀選擇按鈕，而不考慮這個按鈕的文件描述。請參考“使用系統提供的按鈕和圖示”一節，這部分內容討論了正確使用圖示的重要意義。（有關這些按鈕的符號名稱和可用性的資訊請參考UIBarButtonItem的文件。）

表 10-1 可用於工具欄和導覽欄的標準按鈕（以無格式樣式顯示）

按鈕	含義	名稱
	打開動作表單，允許使用者執行應用程式特定的動作	動作
	打開動作表單，顯示相機模式下的照片選擇器	相機
	打開新消息的編輯模式視圖	編寫
	顯示應用程式特定的書籤	書籤
	顯示搜尋框	搜尋
	建立一個新項目	新增
	刪除當前項目	垃圾桶
	在應用程式中將某一項行動或路由至指定目的地，如資料夾	組織
	將某一項發送或路由至另一位置	回復
	停止當前過程或任務	停止
	更新內容（只在必要時使用；其他情況下自動更新）	更新
	開始播放媒體或幻燈片	播放
	快轉當前播放的媒體或幻燈片	快轉
	暫停當前播放的媒體或幻燈片（請注意，這意味著上下文儲存）	暫停

	倒轉當前播放的媒體或幻燈片	倒轉
---	---------------	----

除了表10-1所示的按鈕之外，在您的應用程式中，您也可以使用表10-2中由系統提供的“編輯”，“取消”，“儲存”和“完成”按鈕，以支援編輯或其他類型的內容操作。（有關這些按鈕的符號名稱和可用性的資訊請參考介紹UIBarButtonItem的文件。）這些按鈕對於導覽欄和工具欄都適用，但只能以有邊框樣式提供給使用者。如果您使用上述按鈕，即使您將其指定為無格式樣式，它也會被轉換為有邊框樣式。

表 10-2 用於導覽欄的有邊框樣式的動作按鈕

按鈕	含義	名稱
	進入編輯或內容操作模式	編輯
	退出編輯或內容操作模式而不儲存更改	取消
	儲存更改，並在合適的情況下退出編輯或內容操作模式	儲存
	退出當前模式並儲存修改（如果有的話）	完成

用於標籤欄的標準圖示

iPhone OS提供了用於標籤欄的標準圖示，如表10-3所示。有關這些圖示的符號名稱和可用性的資訊請參考介紹UITabBarSystemItem的文件。

與所有的標準按鈕和圖示一樣，您必須根據其規定含義使用這些圖示。尤其要注意，要基於圖示代表的含義來使用圖示，而不是它的外觀。這樣即使圖示的外觀發生改變，它所代表的特定含義不變，從而使應用程式的使用者介面看起來仍然合理，請參考“使用系統提供的按鈕和圖示”一節，進一步瞭解為甚麼正確使用這些圖示如此重要。

表 10-3 用於標籤欄的標準圖示

圖示	含義	名稱
	顯示應用程式特定的書籤	書籤
	顯示聯絡人	聯絡人
	顯示下載	下載
	顯示當前使用者的收藏櫃	收藏櫃
	顯示應用程式的特色內容	特色
	顯示使用者操作的歷史	歷史
	顯示標籤項上的更多項	更多
	顯示最近一次使用的項目	最近一次使用
	顯示所有使用者最關注的項目	最多關注
	顯示由應用程式定義的一段時間以來使用者訪問過的項目	最近使用

	進入搜尋模式	搜尋
	顯示使用者評級最高的項目	最高級

用於表格行和其它使用者介面元素的標準按鈕

iPhone OS提供了少量用於表格行和其他元素的按鈕。表10-4列出了這些按鈕，與所有的標準按鈕和圖示一樣，它們應該按照其含義被正確地使用。特別是要避免只根據外觀選擇按鈕，而不是這個按鈕的文件描述。請參考“使用系統提供的按鈕和圖示”一節，這部分內容討論了正確使用這些圖示的重要意義。

雖然“詳細資訊展開”按鈕通常在表格行中使用，但它也可以用在別處。有關該按鈕的更多資訊請參考“詳細資訊展開按鈕”。iPhone OS還提供了一組只能在表格行中使用的控制元件；有關這些控制元件的具體資訊請參考“表格視圖元素”一節。

有關這些按鈕的符號名稱和可用性的資訊請參考介紹UIButtonType的文件。（表格視圖元素“詳細資訊展開”的符號名稱和可用性的資訊請參考介紹UITableViewControllerAccessoryDetailDisclosureButton的文件。）

表 10-4 用於表格行和使用者介面元素的標準按鈕

Button	Meaning	Name
	顯示聯絡人選擇器，向某一項中新增聯絡人	新增聯絡人
	顯示一個新視圖，其中包含當前項目的詳細資訊	詳細資訊展開



翻轉至視圖的背面（通常出現在工具程式中）以顯示配置選項或其他資訊。

請注意，“資訊”按鈕也可以是暗色的圓圈當中有一個亮色字母“i”。

資訊

創建自定義圖示和圖像

iPhone應用程式的使用者介面向來以其精美的圖像和豐富的色彩著稱。作為應用程式的設計者，您需要提供一個美觀的使用者介面以融入這樣的整體風格。雖然iPhone OS提供了大量精美的使用者介面元素，但是每個應用程式仍然需要兩個自定義元素：應用程式圖示和啓動圖像。此外，應用程式還應該為iPhone OS提供用於顯示在Spot-light搜尋結果中的小圖示。

有些應用程式需要自定義圖示，用於在導覽欄，工具欄和標籤欄中表示應用程式的特定功能和模式。遵循建立這些圖示的準則，您就可以設計出既與內置圖示相稱，又能巧妙地突顯您應用程式風格的圖示和圖像。

注意： 圖示和圖像的標準比特深度是**24位**（其中，紅色，綠色和藍色各佔**8比特**），外加**8位的alpha通道**。建議您使用**PNG**格式，因為它保留了顏色深度，並且內置支援**a**通道。您可以使用預覽應用程式產生最終的**PNG**文件。

您無需將顏色限定在Web安全色內。雖然您可以在為導覽欄，工具欄和標籤欄建立的圖示中使用alpha透明度，但不要在應用程式圖示中這麼做。

應用程式圖示

應用程式圖示是使用者會放置在其主螢幕上的圖示，點選該圖示可以啓動應用程式。在這裡您可以將品牌宣傳和強烈的視覺效果結合起來，使其簡潔，易辨認且引人注目。

使用者可以選擇想要在主螢幕上顯示的應用程式圖示，因此您應該設計一個這樣的圖示：

- 俱有吸引力的，以至於使用者覺得應該將它留在主螢幕上
- 獨特的，以便使用者可以很容易地在眾多的圖示中找到它

盡量使您的圖示在吸引眼球和表意清晰之間找到平衡點，這樣可以使圖示既豐富，漂亮，又能傳達應用程式的本意。此外，調查一下具有不同文化背景的人們會對您選擇的圖像和顏色產生哪些不同的理解也是不錯的辦法。

當使用者決定在主螢幕上顯示您的應用程式圖示時，iPhone OS會自動為其新增一些視覺效果，以便該圖示與內置圖示保持一致。具體包括：

- 圓角
- 陰影
- 反光

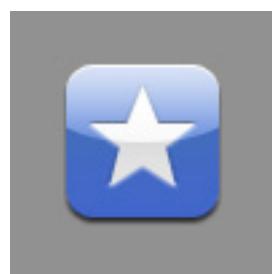
例如，圖11-1顯示了一個由應用程式提供的簡單圖示。

圖 11-1 一個簡單的應用程式圖示，未顯示在主螢幕之前的樣子



圖11-2顯示了同一個圖示被iPhone OS顯示在主螢幕時的樣子。

圖 11-2 一個簡單的應用程式圖示，顯示在主螢幕時的樣子



在主螢幕上，俱有清晰背景的應用程式圖示看起來效果最佳。這主要是因為iPhone OS新增了圓角效果：統一的圓角效果使使用者主螢幕上的所有圖示都俱有一致的可點選外觀。如果您建立的圖示俱有黑色背景，當顯示在主螢幕上時，它的背景會消失，使用者就無法看到它的圓角效果。這樣的圖示往往讓人覺得是無法點選的，會干擾到主螢幕的次序結構。

為了確保您的圖示可以充分利用iPhone OS所提供的視覺增強功能，建議您建立這樣的PNG格式圖像：

- 尺寸為57x57像素，90度直角（如果圖像使用了其它尺寸，iPhone OS會對其進行調整）
- 不要使用光暈或者修飾效果
- 不要使用alpha透明度

將您的圖示文件命名為Icon.png，並將它放在應用程式包的頂層。要瞭解更多與應用程式包的內容相關的資訊，請參考iPhone應用程式程序指南中的“應用程式包”一節。

注意：您也可以選擇讓iPhone OS不要為您的圖示新增光暈效果。要做到這一點，您需要在應用程式的Info.plist文件（要瞭解該文件，請閱讀iPhone應用程式程序指南中的“資訊屬性列表”一節）中新增UIPrerenderedIcon鍵。

無論您是否使用光暈效果，您的圖示都應該採用57 x 57像素。

當您提交您的應用程式時，必須包含一個512 x 512像素版本的應用程式圖示，用於在App Store中顯示。雖然，讓使用者一眼就能認出這個版本的圖示就是您的應用程式的圖示非常重要，但這個版本的圖示應該更華麗，更精細。換句話說，您不應該只是簡單地放大應用程式圖示，直接作為您為App Store建立的圖示。

如果您正在開發一個私下發佈的應用程式（即不通過App Store發佈），您也必須提供一個512 x 512像素版本的應用程式圖示。在這種情況下，將圖示文件命名為iTunesArtwork（沒有文件擴展名），並把它放在應用程式包的頂層。這個圖示用於在iTunes中標識您的應用程式。

小圖示

每個應用程式都應該提供一個小圖示，當應用程式的名稱與Spotlight搜尋欄中的詞條相匹配時，iPhone OS可以顯示這個小圖示。

俱有設置功能的應用程式也應該提供這種小圖示，用於在內置的設置應用程式中標識應用程式。

您的小圖示應該能清晰地標識您的應用程式，以便使用者可以在搜尋結果列表中明確地識別出來。要做到這一點，您應該建立這樣精簡而吸引人的圖示：

- 使用PNG格式。
- 尺寸約為29 x 29像素。

將您的圖示文件命名為Icon-Small.png，並將它放在應用程式包的頂層。要瞭解更多與應用程式包的內容相關的資訊，請參考iPhone應用程式程序指南中的“應用程式包”一節。

注意：如果您沒有提供名為Icon-Small.png的圖示，並且您的應用程式包中不包含早期版本的名為Icon-Settings.png的小圖示的話，那麼iPhone OS會將您的應用程式圖示進行縮放，用於在搜尋結果和設置應用程式中進行顯示。

如果您的應用程式包中包含名為Icon-Settings.png的小圖示，但是不包含名為Icon-Small.png的圖示，那麼iPhone OS會在搜尋結果中顯示設置圖示。不過，您應該更新應用程式包，使其只包含Icon-Small.png文件。

啓動圖像

為了增強應用程式啓動時的使用者體驗，您應該提供一個啓動圖像。啓動圖像與應用程式的首螢幕看起來非常相似。當使用者在主螢幕上點選您的應用程式圖示時，iPhone OS會立即顯示這個啓動圖像。一旦準備就緒，您的應用程式就會顯示它的首螢幕，來替換掉這個啓動佔位圖像。

一定要強調的是，之所以提供啓動圖像，是為了改善使用者體驗，並不是為了提供：

- “應用程式進入體驗”，比如啓動動畫

- “關於”視窗
- 品牌宣傳元素，除非它們是您應用程式首螢幕的靜態組成部分

由於使用者可能會頻繁，快速地在應用程式之間進行切換，因此，您應該盡一切努力將啓動時間降到最短，而且您設計的啓動圖像應該盡量減輕使用者等待的體驗，而不是吸引使用者的注意力。

要做到這一點，您應該設計這樣的PNG格式的圖像：

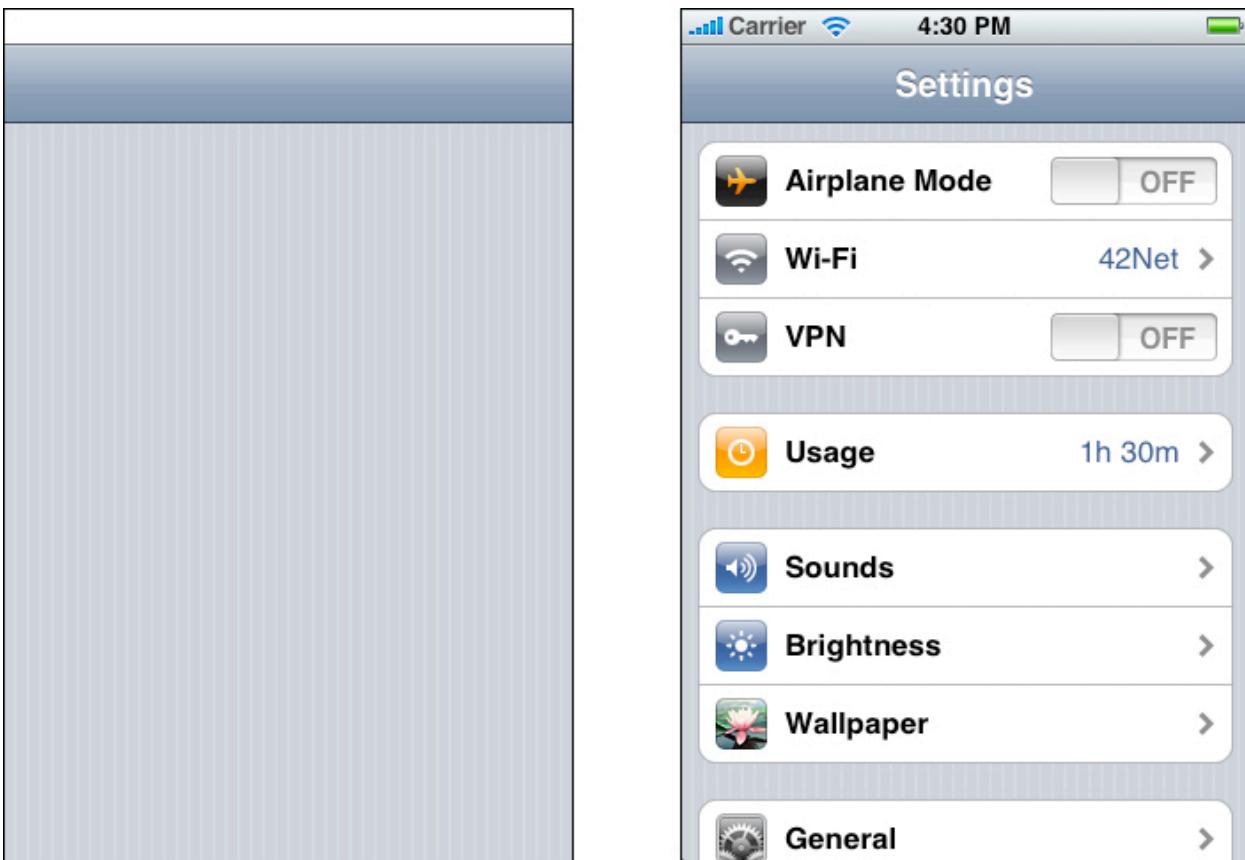
- 尺寸為320 x 480像素。其中包含的狀態欄區可以立即顯示您選擇的狀態欄顏色，而不是等到您的應用程式啓動完成後才顯示。
- 與應用程式的首螢幕相同，除了：
- 文字。啓動圖像是靜態的，因此您在其中顯示的任何文字都不會本端化。
- 可能動態改變的使用者介面元素。不要包含在應用程式啓動完成後看上去不同的元素，防止使用者感覺到啓動圖像和應用程式首螢幕之間的閃動。

將您的啓動圖像命名為Default.png，並把它放在您的應用程式包的最頂層。要瞭解更多與應用程式包的內容相關的資訊，請參考iPhone應用程式程序指南。

您可能覺得遵循這些準則會導致啓動圖像顯得平淡無趣，然而，實際上正是這樣。請記住，啓動圖像並不是為您提供機會進行藝術展示；它完全是為了增強使用者對應用程式能夠快速啓動並立即投入使用的感知度。下面的例子向您展示了啓動圖像有多麼簡單。

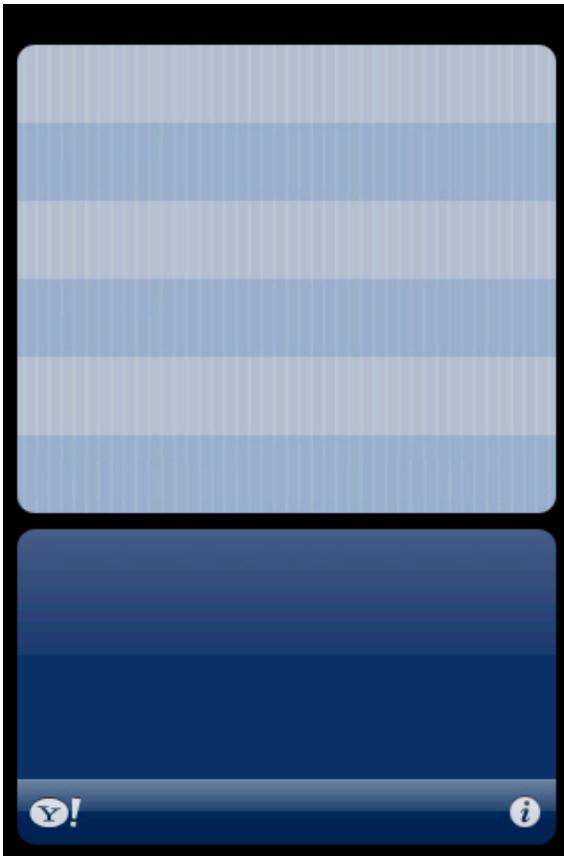
第一個例子是內置設置應用程式的啓動圖像，如圖11-3所示。設置應用程式的啓動圖像只顯示了應用程式的背景，因為應用程式的其他內容都無法保證是靜態的。

圖 11-3 設置應用程式的啓動圖像



另一個啓動圖像的例子源自內置股票應用程式，如圖11-4所示。請注意，只有包含在啓動圖像中的圖像才是靜態的圖像，它們在股票應用程式的正面視圖中總是可見的。

圖 11-4 股票應用程式的啓動圖像



導覽欄，工具欄和標籤欄的圖示

如果可能的話，您應該在導覽欄，工具欄和標籤欄中使用系統提供的按鈕和圖示。iPhone OS提供了各式各樣的標準按鈕和圖示，使用者看到它們就會聯想到內置應用程式所支援的標準任務和模式。如果您的應用程式支援一些標準功能（比如更新內容區視圖或刪除一項），或者會顯示不同的數據子集（比如聯絡人或書籤），一定要使用系統提供的相應按鈕或圖示來表示它們。關於標準按鈕和圖示的完整列表，以及如何使用它們的指南，請參考“系統提供的按鈕和圖示”。

當然，並不是應用程式所執行的每個任務都是一個標準功能。如果您的應用程式支援一些使用者需要頻繁執行的自定義任務，您需要建立自定義的圖示，用於在工具欄或導覽欄中代表這些任務。同樣，如果您的應用程式顯示了標籤欄，允許使用者在自定義的應用程式模式之間，或者自定義的數據子集之間進行切換，你需要設計自定義的標籤欄圖示，用於

清晰地描述這些模式和子集。本節向您介紹如何設計出能夠很好地用於導覽欄，工具欄和標籤欄的圖示。

在您為圖示建立美術圖案之前，您需要花些時間想想該圖示應該傳達些甚麼資訊。當您考慮設計的時候，請以這樣的圖示為目標：

- 簡明扼要。太多的細節會使圖示顯得雜亂或是難以理解。
- 不會被誤認為是系統提供的圖示。使用者應該能夠一眼就將您的自定義圖示與標準圖示區分開。
- 易於理解且廣為接受。請盡力建立一個能夠被大多數使用者正確理解，不會令任何使用者反感的符號。

在您確定了圖示的外觀之後，請在建立時遵循以下準則：

- 使用PNG格式。
- 使用帶有適當alpha透明度的純白色。
- 不要新增陰影效果。
- 使用抗鋸齒效果。
- 如果您要新增斜邊效果，請確保角度為90°（為了便於理解和操作，您可以想象在圖示的頂部有一處光源）。
- 對於工具欄和導覽欄圖示，尺寸應為20 x 20像素。對於標籤欄圖示，尺寸應為30 x 30像素的圖示。

注意：您為工具欄，導覽欄和標籤欄提供的圖示在您的應用程式中將用做建立圖示的蒙板。沒必要建立全彩色的圖示。

iPhone OS自動為導覽欄，工具欄和標籤欄中的項目提供了被按下或被選中時的外觀，因此您只需要提供圖示的單個版本。由於這些視覺效果是自動呈現的，您無法改變它們的外觀。

文件修訂歷史

下表是對文件iPhone人機介面指南文件的更新記錄。

日期	修改
2010-03-24	改進設計警告視圖的介紹。
2010-02-19	新增單元格樣式的介紹。
2009-11-20	修改少許錯誤並重新整理“表格視圖”部分的內容。
2009-09-09	更新“使用聲音”部分的介紹；補充一些細小修正。
2009-06-04	新增“使用使用者的位置”和“使應用程式可訪問”部分的介紹；更新“設置”、“搜尋”和“欄外觀”部分的介紹。
2009-03-27	進行少許修正。
2009-03-12	新增“處理編輯和還原功能”，“搜尋”，“推送通知”和“模式視圖轉換”部分的介紹。
2009-02-04	改進“標籤欄中的標籤”部分的介紹。
2008-11-21	擴展“在iPhone操作系統中使用聲音”部分的介紹並進行少許修正。
2008-09-09	將“Web介紹”部分的內容轉移到“Web應用程式的iPhone人機介面介紹”中。
2008-06-27	建立文件，介紹如何設計iPhone應用程式的使用者介面，為建立基於iPhone操作系統的裝置上的web內容提供介紹。