

使用 Packager for iPhone[®] 建立 ADOBE[®] AIR[®] 應用程式

上次更新 2017/6/1

法律聲明

如需法律聲明，請參閱 http://help.adobe.com/zh_TW/legalnotices/index.html。

內容

第 1 章 開始為 iPhone 建立 AIR 應用程式

重要概念	1
從 Adobe 取得開發人員工具	3
從 Apple 取得開發人員檔案	3
使用 Flash Professional CS5 建立 Hello World iPhone 應用程式	7

第 2 章 對 iPhone 應用程式進行編譯及除錯

iPhone 圖示與起始螢幕影像	11
iPhone 應用程式設定	12
編譯 iPhone 應用程式安裝程式 (IPA) 檔案	17
安裝 iPhone 應用程式	19
iPhone 應用程式除錯	20
將您的 iPhone 應用程式送出至 App Store	22

第 3 章 行動裝置的 ActionScript 3.0 API 支援

行動裝置不支援 ActionScript 3.0 API	23
行動 AIR 應用程式專屬的 ActionScript API	25
行動應用程式開發人員特別感興趣的 ActionScript 3.0 API	28

第 4 章 iPhone 應用程式設計注意事項

硬體加速	30
其他改善顯示物件效能的方法	31
資訊密度	32
字體及文字輸入	32
儲存應用程式狀態	33
螢幕方向變更	33
感應區目標	33
記憶體配置	34
繪圖 API	34
事件反昇	34
最佳化視訊效能	34
Flex 與 Flash 組件	34
縮小應用程式檔案	35

第 1 章 開始為 iPhone 建立 AIR 應用程式

您可以使用 Adobe® Flash® Platform 工具和 ActionScript® 3.0 程式碼，建立 iPhone 和 iPod Touch 的 Adobe® AIR® 應用程式。散發、安裝以及執行這些應用程式的方式，與其他 iPhone 應用程式完全相同。

備註：本文件將 iPhone 和 iPod Touch 統稱為「iPhone」。

Packager for iPhone® 隨附在 Adobe® Flash® Professional CS5 中。Packager for iPhone 可將 ActionScript 3.0 位元組碼編譯成原生的 iPhone 應用程式程式碼。iPhone 應用程式將透過 iTunes Store，以 iPhone 應用程式安裝程式檔案 (.ipa 檔案) 的形式散發。

您可以使用 Flash Professional CS5 或 Adobe® Flash® Builder™ 4，編輯應用程式的原始 ActionScript 3.0 內容。

若要開發 iPhone 應用程式，請使用 Flash Professional CS5。

您也需要從 Apple 取得 iPhone 開發人員憑證。

重要事項：開發 iPhone 應用程式之前，請詳讀有關設計 iPhone 應用程式的資訊。請參閱第 30 頁「[iPhone 應用程式設計注意事項](#)」。同時，也請瞭解建立 iPhone 應用程式所需的開發人員檔案。請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

重要概念

使用 ActionScript 3.0 開發 iPhone 應用程式之前，請務必瞭解其中的概念和工作流程。

詞彙

建立 iPhone 應用程式時，請務必瞭解以下詞彙。

iPhone Dev Center 網站 您可以從此 Apple Computer 網站 (<http://developer.apple.com/iphone/>) 執行下列工作：

- 申請成為 iPhone 開發人員。
- 管理和建立 iPhone 開發憑證、佈建描述檔和應用程式 ID (將於下方定義)。
- 將應用程式送出至 App Store。

iPhone 開發憑證 用於識別開發應用程式的開發人員。

從 Apple 取得此檔案。您可以將此憑證轉換成 P12 憑證檔案，這樣就可以簽署使用 ActionScript 3.0 建立的 iPhone 應用程式。請參閱 P12 憑證檔案。

若只要在開發電腦上進行 Flash Professional CS5 應用程式的除錯和測試，您並不需要 iPhone 開發憑證。不過，您必須取得開發憑證才能在 iPhone 上安裝並測試應用程式。

開發憑證與散發憑證並不相同，散發憑證是用來建立應用程式的最終版本。當您建立應用程式的最終版本時，請向 Apple 取得散發憑證。

憑證簽名要求 包含用來產生開發憑證之個人資訊的檔案。也稱為 CSR 檔。

佈建描述檔 可讓您測試或散發 iPhone 應用程式的檔案。您可以從 Apple 取得佈建描述檔。佈建描述檔會指派給特定的開發憑證、應用程式 ID 及一或多個裝置 ID。目前有不同類型的佈建描述檔：

- 開發佈建描述檔：用來將測試版的應用程式安裝至開發人員的 iPhone。
- 測試佈建描述檔：又稱為臨時佈建描述檔。用來將測試版的應用程式散發給多個使用者 (以及 iPhone 手機)。有了此佈建描述檔與測試版的應用程式，使用者便可以測試您的應用程式，您不需將應用程式送出至 App Store。注意：您也可以使用開發佈建描述檔，將測試版的應用程式散發給多個裝置。

- 散佈建描述檔：可建立一個 iPhone 應用程式，以便將您的應用程式送出至 App Store。

應用程式 ID 可針對特定開發人員識別其 iPhone 應用程式 (或多個應用程式) 的專屬字串。您會在 iPhone Dev Center 網站建立應用程式 ID。每個佈建描述檔都有相關的應用程式 ID 或應用程式 ID 樣式。您在開發應用程式時，會使用此應用程式 ID (或樣式)。您會在 Flash Professional CS5 「iPhone 設定」對話方塊 (或應用程式描述器檔案) 中使用應用程式 ID。

iPhone Dev Center 的應用程式 ID 格式為組合種子 ID 後接組合識別名稱。組合種子 ID 是 Apple 指派給應用程式 ID 的一個字元字串，例如 5RM86Z4DJM。組合識別名稱是您所選擇的反向網域名稱。組合識別名稱可能以星號 (*) 結尾，這表示萬用字元應用程式 ID。例如：

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.* (萬用字元應用程式 ID)

iPhone Dev Center 中有兩種類型的應用程式 ID：

- 萬用字元應用程式 ID：在 iPhone Dev Center 中，這些應用程式 ID 以星號 (*) 結尾，例如 96LPVWEASL.com.myDomain.* 或 96LPVWEASL.*。當佈建描述檔使用這種應用程式 ID 時，您可以產生使用符合該應用程式 ID 樣式的測試應用程式。如果是應用程式的應用程式 ID，您可以使用任何有效的字元字串來取代星號。例如，如果 iPhone Dev Center 網站指定應用程式 ID 為 96LPVWEASL.com.example.*，則您可以使用 com.example.foo 或 com.example.bar 做為應用程式的應用程式 ID。
- 特定應用程式 ID：這種 ID 會定義應用程式使用的唯一應用程式 ID。在 iPhone Dev Center 中，這些應用程式 ID 不會以星號 (*) 結尾。例如，96LPVWEASL.com.myDomain.myApp。當佈建描述檔使用這種應用程式 ID 時，應用程式必須完全符合該應用程式 ID。例如，如果 iPhone Dev Center 網站指定 96LPVWEASL.com.example.helloWorld 做為應用程式 ID，則您必須使用 com.example.foo 做為應用程式的應用程式 ID。

開發應用程式時，您會在 Flash Professional CS5 「iPhone 設定」對話方塊或應用程式描述器檔案中指定應用程式 ID。如需應用程式 ID 的詳細資訊，請參閱第 13 頁「在 Flash Professional CS5 中設定 iPhone 應用程式屬性」的「部署標籤」一節，或參閱第 14 頁「在應用程式描述器檔案中設定 iPhone 應用程式屬性」。

重要事項：指定應用程式 ID 時，請忽略應用程式 ID 的組合種子 ID 部分。例如，如果 Apple 列出您的應用程式 ID 為 96LPVWEASL.com.example.bob.myApp，則請忽略 96LPVWEASL，使用 com.example.bob.myApp 做為您的應用程式 ID。如果 Apple 列出您的應用程式 ID 為 5RM86Z4DJM.*，則請忽略 5RM86Z4DJM。這是萬用字元應用程式 ID。

您可以在 iPhone Dev Center (<http://developer.apple.com/iphone>) 找到與佈建描述檔相關的應用程式 ID (或萬用字元應用程式 ID 樣式)。移至 iPhone Developer Program Portal，然後移至「Provisioning」區段。

P12 憑證檔案 P12 檔案 (副檔名為 .p12 的檔案) 是一種憑證檔案 (個人資訊交換檔案)。Packager for iPhone 會使用此類型的憑證來建立 iPhone 應用程式。您要將從 Apple 收到的開發人員憑證轉換成此憑證格式。

唯一裝置 ID 用於識別特定 iPhone 的唯一代碼。又稱為 UDID 或裝置 ID。

開發工作流程的概觀

開發 iPhone 的應用程式時，請按照以下步驟執行：

- 1 安裝 Adobe 的 Flash Professional CS5。
- 2 安裝 iTunes。
- 3 從 Apple 取得開發人員檔案。這些檔案包含開發人員憑證和佈建描述檔。請參閱第 3 頁「從 Apple 取得開發人員檔案」。
- 4 將開發憑證轉換成 P12 憑證檔案。Flash CS5 規定憑證必須是 P12 憑證。請參閱第 3 頁「從 Apple 取得開發人員檔案」。
- 5 使用 iTunes 來建立佈建描述檔與 iPhone 的關聯。
- 6 使用 Flash Professional CS5 撰寫應用程式。

您必須瞭解 iPhone 應用程式設計及最佳化程式碼的最佳作法。請參閱第 30 頁「iPhone 應用程式設計注意事項」。

此外，部分 ActionScript 3.0 API 只能用於 iPhone，而部分則不受 iPhone 支援。請參閱第 23 頁「[行動裝置的 ActionScript 3.0 API 支援](#)」。

您也可以使用 Flash Builder 4.0 編輯應用程式的 ActionScript 3.0 程式碼。

您可以使用 Flash Professional CS5，在開發電腦上測試您的應用程式。

7 建立應用程式的圖示圖案及起始螢幕圖案。每個 iPhone 應用程式都包括一組圖示，使用者可藉由這些圖示識別該應用程式。iPhone 會在載入程式時顯示起始螢幕影像。請參閱第 11 頁「[iPhone 圖示與起始螢幕影像](#)」。

8 編輯 iPhone 設定。這些設定包括下列項目：

- 應用程式的識別 (包括檔案名稱、應用程式名稱、版本號碼以及應用程式 ID)。
- 應用程式之來源圖示圖案的位置
- 指派給應用程式的 P12 憑證及佈建描述檔
- 應用程式的起始外觀比例

在 Flash Professional CS5 中，您可以在「iPhone 設定」對話方塊中編輯這些設定。如需詳細資訊，請參閱第 13 頁「[在 Flash Professional CS5 中設定 iPhone 應用程式屬性](#)」。

您也可以直接在應用程式描述器檔案中編輯這些設定。如需詳細資訊，請參閱第 14 頁「[在應用程式描述器檔案中設定 iPhone 應用程式屬性](#)」。

9 使用 Packager for iPhone 編譯 IPA 檔。請參閱第 17 頁「[編譯 iPhone 應用程式安裝程式 \(IPA\) 檔案](#)」。

10 在 iPhone 安裝及測試應用程式。使用 iTunes 安裝 IPA 檔。

若要進行臨時散發，請重複此共通程序，但請使用測試佈建描述檔，而不是開發佈建描述檔。正式散發應用程式時，請使用散發佈建描述檔重複此程序 (如需不同佈建描述檔類型的相關資訊，請參閱第 1 頁「[詞彙](#)」)。

建立應用程式的散發版本時，請參閱第 22 頁「[將您的 iPhone 應用程式送出至 App Store](#)」。

如需建立基本 iPhone 應用程式的快速教學課程，請參閱第 7 頁「[使用 Flash Professional CS5 建立 Hello World iPhone 應用程式](#)」。

從 Adobe 取得開發人員工具

若要使用 ActionScript 3.0 開發 iPhone 應用程式，您需要 Flash Professional CS5。

重要：您應該更新 Flash Professional CS5 隨附的 Packager for iPhone Preview 版本。在 Flash Professional CS5 中，選取「說明 > 更新」。

您也可以使用 Flash Builder 4 編輯 ActionScript 程式碼。請從 <http://www.adobe.com/tw/products/flashbuilder/> 取得 Flash Builder 4。

從 Apple 取得開發人員檔案

不論開發何種 iPhone 應用程式，您都必須先從 Apple 取得 iPhone 開發人員檔案。您必須取得 iPhone 開發人員憑證及行動佈建描述檔。您也必須取得其他佈建描述檔。請參閱第 1 頁「[詞彙](#)」，瞭解這些檔案的定義。

備註：在開發應用程式的過程中，取得這些檔案是很重要的一環。開發您的應用程式之前，請務必先完成此程序。取得開發人員檔案並不是項簡單的工作。請詳細閱讀這些指示以及 Apple iPhone Dev Center 網站的指示。

取得及使用 iPhone 開發人員檔案

您必須從 Apple 取得 iPhone 開發人員憑證及佈建描述檔。您也需要將憑證轉換成 P12 憑證。

安裝 iTunes

您必須有 iTunes 才能在 iPhone 上安裝應用程式。此外，您也需要使用 iTunes 來判斷 iPhone 的裝置 ID。因為申請 iPhone 開發人員憑證時，您將需要知道裝置 ID。

申請 iPhone 開發人員憑證並建立佈建描述檔

如果您尚未申請開發人員憑證，請前往 Apple iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>) 申請成為註冊的 iPhone 開發人員。

備註：為 iPhone 開發 AIR 應用程式時，並不需要 iPhone SDK 或 XCode。不過您必須是註冊的 iPhone 開發人員。而且您需要取得開發人員憑證和佈建描述檔。

- 1 使用您的 iPhone 開發人員帳戶 ID 登入 iPhone Dev Center。
- 2 在 iPhone Dev Center 申請 (以及購買) iPhone 開發人員憑證。
您將收到一封來自 Apple 的電子郵件訊息，內容包含您的 iPhone Developer Program 啟用碼。
- 3 返回 iPhone Dev Center。遵從指示啟用您的開發人員計劃 (並在提示時輸入您的啟用碼)。
- 4 當系統接受您的啟用碼後，請移至 iPhone Dev Center 的「iPhone Developer Program Portal」區段。
- 5 建立憑證簽名要求檔。您將使用此檔案來取得 iPhone 開發憑證。如需指示，請參閱第 5 頁「[產生憑證簽名要求](#)」。
- 6 在下個步驟中，系統將要求您提供 iPhone 的「裝置 ID」(或「唯一裝置 ID」)。您可以從 iTunes 取得 UDID：
 - a 使用 USB 纜線連接 iPhone。然後，在 iTunes 中選取 iPhone 的摘要標籤。
 - b 從 iPhone 開發人員中心網站下載佈建描述檔之後，請將佈建描述檔加到 iTunes 中。
 - c 接著按一下顯示的「序號」。即會顯示 UDID。按一下 Command-C (Mac) 或 Control-C (Windows)，將 UDID 複製到剪貼簿。
- 7 建立並安裝「佈建描述檔」及「iPhone 開發憑證」。
請遵從 iPhone Dev Center 上的指示執行。請至「iPhone Developer Program Portal」區段查看指示。您可以使用 Development Provisioning Assistant 來取得您的開發憑證，並建立佈建描述檔。
略過與 XCode 相關的步驟。不需使用 XCode。只要使用 Flash Professional CS5 即可開發 iPhone 應用程式。
- 8 在 iTunes 中，選取「檔案 > 加到資料庫」。然後選取佈建描述檔 (副檔名為 mobileprovision)。接下來使用 iTunes 同步您的 iPhone。
這可讓您在 iPhone 上測試與這個佈建描述檔有關的應用程式。
若要確認特定的佈建描述檔是否已新增至 iTunes，您可以嘗試將該佈建描述檔加到資料庫。如果 iTunes 詢問您是否要取代現有的佈建描述檔，您可以按「取消」按鈕 (這表示此描述檔已安裝)。此外，您可以檢查 iPhone 上安裝的佈建描述檔：
 - a 在 iPhone 開啟「設定」應用程式。
 - b 開啟「一般」類別。
 - c 點選「描述檔」。「描述檔」頁面會列示您已安裝的佈建描述檔。
- 9 如果您尚未這樣做，請下載 iPhone 開發憑證檔案 (.cer 檔案)。
Development Provisioning Assistant 可能會提供您下載此檔案的連結。您也可以到 Apple iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>) Provisioning Portal 的「Certificates」區段中找到這個檔案。
- 10 接著，將 iPhone 開發人員憑證轉換成 P12 檔案。如需指示，請參閱第 5 頁「[將開發人員憑證轉換成 P12 檔案](#)」。

現在，您可以開始建立簡單的 Hello World 應用程式了。請參閱第 7 頁「[使用 Flash Professional CS5 建立 Hello World iPhone 應用程式](#)」。

產生憑證簽名要求

若要取得開發人員憑證，您必須先產生憑證簽名要求檔，您將在 Apple iPhone Dev Center 網站中提交此檔案。

在 **Mac OS** 產生憑證簽名要求

您可以在 Mac OS 利用「鑰匙圈存取」應用程式產生代碼簽名要求。「鑰匙圈存取」應用程式位於 Applications 目錄下的 Utilities 子目錄。在「鑰匙圈存取」選單中，選取「憑證輔助程式 > 從憑證授權要求憑證」。

- 1 開啟「鑰匙圈存取」。
- 2 在「鑰匙圈存取」選單中，選取「偏好設定」。
- 3 在「偏好設定」對話方塊中，按一下「憑證」。然後將「線上憑證狀態協定」和「憑證撤銷列表」設成「關閉」。關閉對話方塊。
- 4 在「鑰匙圈存取」選單中，選取「憑證輔助程式 > 從憑證授權要求憑證」。
- 5 輸入與您的 iPhone 開發人員帳戶 ID 相應的電子郵件地址和名稱。不要輸入 CA 電子郵件地址。選取「要求儲存到磁碟」，然後按一下「繼續」按鈕。
- 6 儲存檔案 (CertificateSigningRequest.certSigningRequest)。
- 7 將 CSR 檔上傳至 Apple 的 [iPhone 開發人員網站](#) (請參閱「[申請 iPhone 開發人員憑證並建立佈建描述檔](#)」)。

在 **Windows** 產生憑證簽名要求

對 Windows 開發人員而言，最簡單的方法是取得 Mac 電腦上的 iPhone 開發人員憑證。不過，他們也可以在 Windows 電腦上取得憑證。首先，使用 OpenSSL 建立憑證簽名要求 (CSR 檔)：

- 1 在 Windows 電腦上安裝 OpenSSL (移至 <http://www.openssl.org/related/binaries.html>)。

您可能也需要安裝「Open SSL」下載頁面中所列出的 Visual C++ 2008 可轉散發套件檔案 (您不用在電腦安裝 Visual C++)。

- 2 開啟 Windows 命令工作階段，然後使用 CD 命令切換至 OpenSSL bin 目錄 (例如 c:\OpenSSL\bin\)
- 3 在命令列輸入以下命令以建立專用密鑰：

```
openssl genrsa -out mykey.key 2048
```

儲存此專用密鑰。您稍後將會用到它。

使用 OpenSSL 時，請勿忽略錯誤訊息。OpenSSL 即使產生錯誤訊息，可能仍會輸出檔案。但這些檔案可能無法使用。如果發生錯誤，請檢查您的語法並重新執行命令。

- 4 在命令列輸入以下命令以建立 CSR 檔：

```
openssl req -new -key mykey.key -out CertificateSigningRequest.certSigningRequest -subj  
"/emailAddress=yourAddress@example.com, CN=John Doe, C=US"
```

以您自己的值取代電子郵件地址、CN (憑證名稱) 及 C (國家 / 地區) 值。

- 5 將 CSR 檔上傳至 Apple 的 [iPhone 開發人員網站](#) (請參閱「[申請 iPhone 開發人員憑證並建立佈建描述檔](#)」)。

將開發人員憑證轉換成 P12 檔案

若要使用 Flash Professional CS5 開發 iPhone 應用程式，您必須使用 P12 憑證檔案。您將根據從 Apple 收到的 Apple iPhone 開發人員憑證檔案來產生此憑證。

在 **Mac OS** 將 iPhone 開發人員憑證轉換成 **P12** 檔案

從 Apple 下載 Apple iPhone 憑證之後，請將該憑證輸出為 P12 憑證格式。在 Mac® OS 執行此動作：

- 1 開啟「鑰匙圈存取」應用程式（位於 Applications/Utilities 資料夾）。
- 2 如果尚未將憑證新增至鑰匙圈，請選取「檔案 > 輸入」。然後瀏覽至您向 Apple 取得的憑證檔案 (.cer 檔)。
- 3 在「鑰匙圈存取」中選取「鑰匙」類別。
- 4 選取與「iPhone 開發憑證」相關的專用密鑰。
專用密鑰由 iPhone 開發人員所指定：< 名字 > < 姓氏 > 與其配對的公用憑證。
- 5 選取「檔案 > 輸出項目」。
- 6 將您的鑰匙儲存為個人資訊交換 (.p12) 檔案格式。
- 7 電腦會提示您建立一個當您嘗試在其他電腦輸入此鑰匙時需使用的密碼。

在 **Windows** 將 Apple 開發人員憑證轉換成 **P12** 憑證

若要使用 Flash CS5 開發 iPhone 應用程式，您必須使用 P12 憑證檔案。您將根據從 Apple 收到的 Apple iPhone 開發人員憑證檔案來產生此憑證。

- 1 將您從 Apple 收到的開發人員憑證檔案轉換成 PEM 憑證檔案。從 OpenSSL bin 目錄執行以下命令列陳述式：

```
openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
```

- 2 如果您使用的是 Mac 電腦上的鑰匙圈專用密鑰，請將它轉換成 PEM 密鑰：

```
openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
```

- 3 現在您可以根據 iPhone 開發人員憑證的密鑰及 PEM 密鑰，產生有效的 P12 檔案：

```
openssl pkcs12 -export -inkey mykey.key -in developer_identity.pem -out iphone_dev.p12
```

如果您使用的是 Mac OS 鑰匙圈的密鑰，請使用上一個步驟所產生的 PEM 密鑰。否則，請使用先前產生的 OpenSSL 金鑰 (Windows)。

管理憑證、裝置 ID、應用程式 ID 以及佈建描述檔

您可以在 Apple iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>) 管理憑證、裝置 ID、應用程式 ID 以及佈建描述檔。移至網站的「iPhone Developer Program Portal」區段。

- 按一下「Certificates」連結，管理您的開發憑證。您可以建立憑證、下載憑證或撤銷憑證。若要建立憑證，您必須先建立憑證簽名要求。請參閱第 5 頁「產生憑證簽名要求」。
- 按一下「Devices」連結，管理可安裝開發應用程式的裝置清單。
- 按一下「App IDs」連結，管理您的應用程式 ID。建立佈建描述檔時，該檔案會與一個應用程式 ID 連結。
- 按一下「Provisioning」連結，管理您的佈建描述檔。您也可以使用 Development Provisioning Assistant 建立開發佈建描述檔。
- 當您要將應用程式送出至 App Store，或建立臨時版本的應用程式時，請按一下「Distribution」連結。本節包含可以連結至 iTunes Connect 網站的連結，您可以藉此將應用程式送出至 App Store。

使用 Flash Professional CS5 建立 Hello World iPhone 應用程式

重要事項：建立應用程式之前，請下載必要的開發人員應用程式和檔案。請參閱第 3 頁「[從 Adobe 取得開發人員工具](#)」和第 3 頁「[從 Apple 取得開發人員檔案](#)」。

建立 Flash Professional CS5 專案

您可以直接在 Flash Professional CS5 產生 iPhone 應用程式：

- 1 開啟 Flash CS5。
- 2 選取「檔案 > 開新檔案」。
- 3 選取「iPhone OS」。
- 4 按一下「確定」按鈕。

將內容加到應用程式

接著輸入“Hello world!”這兩個字到應用程式：

- 1 選取「文字工具」，然後按一下舞台。
- 2 在文字欄位的「屬性」設定中，選取「傳統文字」（不是「TLF 文字」）。
這是一個基礎應用程式，使用「傳統文字」便足以正確顯示。若要使用「TLF 文字」，您需要套用其他設定。請參閱第 32 頁「[字體及文字輸入](#)」。
- 3 在下一個文字欄位中，輸入“Hello world!”
- 4 使用「選取工具」來選取文字欄位。
- 5 然後開啟「屬性」檢測器以進行以下設定：
 - 字元 > 系列：_sans
 - 字元 > 大小：50
 - 位置 > X：20
 - 位置 > Y：20
- 6 儲存檔案。
- 7 選取「控制 > 測試影片 > 在 AIR Debug Launcher (行動裝置) 中」。

Flash Professional CS5 會編譯 SWF 內容，並在 AIR Debug Launcher (ADL) 中顯示應用程式的版本。這可讓您取得應用程式的快速預覽。

建立應用程式的圖示圖案及起始螢幕圖案

所有 iPhone 應用程式在 iTunes 應用程式的使用者介面及 iPhone 螢幕上都會有自己的圖示。

- 1 在專案目錄中建立目錄，並命名為 icons。
- 2 在 icons 目錄中建立 3 個 PNG 檔。將這 3 個檔案命名為 Icon29.png、Icon57.png 及 Icon512.png。
- 3 編輯 PNG 檔，為您的應用程式建立適當的圖案。檔案必須是 29 x 29 像素、57 x 57 像素及 512 x 512 像素。在此測試中，您可以使用實色的矩形做為圖案。

所有 iPhone 應用程式在載入 iPhone 時都會顯示其起始影像。您會在 PNG 檔定義起始影像：

- 1 在主要開發目錄中，建立名為 Default.png 的 PNG 檔。(請勿將此檔案放入 icons 子目錄中，務必將檔案命名為大寫 D 的 Default.png)。
- 2 編輯檔案，讓檔案的寬度為 320 像素，而高度為 480 像素。就現在而言，內容可以是簡單的白色矩形 (我們將在稍後予以變更)。

備註：將應用程式送出至 Apple App Store 時，您將使用 512 像素的 JPG 檔 (不是 PNG 檔)。測試應用程式的開發版本時，您便可以使用 PNG 檔。

如需這些圖形的詳細資訊，請參閱第 11 頁「[iPhone 圖示與起始螢幕影像](#)」。

編輯應用程式設定

重要事項：如果您尚未編輯應用程式設定，請下載所需的開發人員應用程式和檔案以開發 iPhone。請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

在 Flash Professional CS5 的「iPhone 設定」對話方塊中，定義 iPhone 應用程式的多種基本屬性。

- 1 選擇「檔案 > iPhone OS 設定」。
- 2 在「一般」標籤中，進行以下設定：

- 輸出檔案：HelloWorld.ipa

這是產生的 iPhone 安裝程式檔案名稱。

- 應用程式名稱：Hello World

顯示於 iPhone 應用程式圖示下方的應用程式名稱。

- 版本：1.0

應用程式的版本。

- 外觀比例：直向

- 全螢幕：已選取。

- 自動方向：已取消選取。

- 顯示：CPU

其他選項 (GPU 和自動) 使用硬體加速功能來進行顯示。當圖形密集應用程式 (例如遊戲) 設計為可利用硬體加速時，此功能可協助提昇該應用程式的效能。如需詳細資訊，請參閱第 30 頁「[硬體加速](#)」。

- 包含的檔案：將起始螢幕圖案檔案 (Default.png) 加入「包含的檔案」清單中。

備註：在此 Hello World 範例中，請勿修改指示中提供的各項設定。部分設定 (例如，版本設定) 涉及特定的限制。這些限制將在第 12 頁「[iPhone 應用程式設定](#)」中加以描述。

- 3 在「部署」標籤中，進行以下設定：

- 憑證：瀏覽並選取您根據從 Apple 取得之開發人員憑證所轉換的 .p12 憑證。

此憑證將用來簽署檔案。您必須將 Apple iPhone 憑證轉換成 .p12 格式。如需詳細資訊，請參閱第 3 頁「[從 Adobe 取得開發人員工具](#)」。

- 密碼：輸入憑證的密碼。

- 佈建檔案：瀏覽並選取您從 Apple 取得的開發人員佈建檔案。請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

- 應用程式 ID：如果此欄位可供選取，則您可以輸入與您提供給 Apple 之應用程式 ID 相符的應用程式 ID (例如 com.example.as3.HelloWorld)。

應用程式 ID 可識別唯一的應用程式。

如果無法選取欄位，則佈建描述檔會與特定應用程式 ID 連結。該應用程式 ID 便會顯示在欄位中。

如需指定應用程式 ID 的詳細資訊，請參閱第 13 頁「在 [Flash Professional CS5 中設定 iPhone 應用程式屬性](#)」的「部署標籤」一節。

- 4 在「圖示」標籤中，按一下「圖示」清單中的「圖示 29 x 29」。然後指定先前建立之 29 x 29 像素 PNG 檔的位置（請參閱第 7 頁「[建立應用程式的圖示圖案及起始螢幕圖案](#)」）。然後指定 57 x 57 像素圖示及 512 x 512 像素圖示的 PNG 檔案位置。
- 5 按一下「確定」按鈕。
- 6 儲存檔案。

如需應用程式設定的詳細資訊，請參閱第 12 頁「[iPhone 應用程式設定](#)」。

編譯 IPA 檔

您現在即可編譯 IPA 安裝程式檔案：

- 1 選取「檔案」>「發佈」。
- 2 在「iPhone 設定」對話方塊中，按一下「確定」按鈕。

Packager for iPhone 會在專案目錄中產生 iPhone 應用程式安裝程式檔案 (HelloWorld.ipa)。編譯 IPA 檔需要花費數分鐘的時間。

在 iPhone 上安裝應用程式

若要安裝 iPhone 應用程式以在 iPhone 上進行測試：

- 1 開啟 iTunes 應用程式。
- 2 如果您尚未這樣做，請將此應用程式的佈建描述檔加到 iTunes 中。在 iTunes 中，選取「檔案 > 將檔案加到資料庫」。選取佈建描述檔（檔案類型為 mobileprovision）。

現在，請使用開發佈建描述檔在開發人員 iPhone 上測試應用程式。

稍後要將應用程式散發至 iTunes Store 時，請使用散發描述檔。若要散發臨時應用程式（至多個裝置，但不經過 iTunes Store），請使用臨時佈建描述檔。

如需瞭解佈建描述檔的詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

- 3 在部分 iTunes 版本中，即使您已安裝相同版本的應用程式，新安裝的應用程式並不會取代已安裝的應用程式。在這種情況下，請從裝置及 iTunes 的應用程式清單刪除應用程式。
- 4 按兩下應用程式的 IPA 檔。該檔案應該會顯示在應用程式清單中。
- 5 將 iPhone 連接至電腦的 USB 連接埠。
- 6 在 iTunes 中，選取裝置的「應用程式」標籤，並確認在要安裝的應用程式清單中已選取該應用程式。
- 7 在 iTunes 應用程式的左側清單選取裝置。然後按一下「同步」按鈕。完成同步之後，Hello World 應用程式就會顯示在您的 iPhone 中。

如果還是未安裝新的版本，請從 iPhone 及 iTunes 的應用程式清單中刪除該應用程式，然後重做此程序。發生這種情況的原因可能是目前安裝的版本與新安裝的版本使用相同的應用程式 ID 和版本號碼。

如果 iTunes 在您嘗試安裝應用程式時顯示錯誤，請參閱第 19 頁「[安裝 iPhone 應用程式](#)」中的「疑難排解應用程式安裝問題」。

編輯起始螢幕圖形

編譯應用程式之前，您可以建立一個 `Default.png` 檔案 (請參閱第 7 頁「[建立應用程式的圖示圖案及起始螢幕圖案](#)」)。此 PNG 檔會在載入應用程式時，做為啟動影像使用。在 iPhone 測試應用程式時，您會在啟動時看到此空白螢幕。

您應該將此影像變更為與應用程式相符的啟動螢幕 ("Hello World!")：

- 1 在裝置上啟動應用程式。首次顯示 "Hello World" 文字時，請按住「主畫面」按鈕 (螢幕下方)。按住「主畫面」按鈕的同時，請按住「電源 / 睡眠」按鈕 (在 iPhone 的最上方)。這樣會擷取一張螢幕快照並傳送至「相機膠卷」。
- 2 藉由從 iPhoto 或其他照片傳送應用程式傳送照片，您可以將影像傳送至您的開發電腦 (在 Mac OS，您也可以使用影像擷取應用程式)。

您也可以將照片透過電子郵件傳送至開發電腦：

- 開啟「照片」應用程式。
- 開啟「相機膠卷」。
- 開啟您擷取的螢幕快照影像。
- 點選影像，然後點選左下角的「下一頁」(箭號)按鈕。然後按一下「透過電子郵件傳送照片」按鈕，將影像傳送給自己。

- 3 以螢幕擷取影像的 PNG 版本取代 `Default.png` 檔案 (位於開發目錄中)。
- 4 重新編譯應用程式 (請參閱第 9 頁「[編譯 IPA 檔](#)」)，並將該應用程式重新安裝到您的 iPhone。

從現在起，應用程式會於載入時使用新的啟動螢幕。

備註：您可以為 `Default.png` 檔案建立任何圖案，但外觀比例必須正確 (320 x 480 像素)。不過，最好還是讓 `Default.png` 影像與應用程式的起始狀態相符。

第 2 章 對 iPhone 應用程式進行編譯及除錯

您可以使用 Packager for iPhone 編譯 iPhone 應用程式。Packager for iPhone 隨附在 Flash Professional CS5 中。

您可以在開發電腦上對應用程式進行除錯。您也可以安裝除錯版本，然後在 Flash Professional CS5 接收 trace() 輸出。

如需建立 iPhone 應用程式的完整教學課程，請參閱第 7 頁「[使用 Flash Professional CS5 建立 Hello World iPhone 應用程式](#)」。

iPhone 圖示與起始螢幕影像

所有 iPhone 應用程式在 iTunes 應用程式的使用者介面及 iPhone 上都會有自己的圖示。

iPhone 應用程式圖示

iPhone 應用程式需定義下列圖示：

- 29 x 29 像素圖示：iPhone 與 iPod touch 上的 Spotlight 搜尋結果使用此圖示。
- 48 x 48 像素圖示：iPad 上的 Spotlight 搜尋結果使用此圖示。
- 57 x 57 像素圖示：iPhone 與 iPod touch 主畫面顯示此圖示。
- 72 x 72 像素圖示 (選擇性)：iPad 主畫面顯示此圖示。
- 512 x 512 像素圖示：iTunes 會顯示此圖示。512 像素的 PNG 檔案僅用於測試應用程式的開發版本。當您將最終的應用程式送出至 Apple App Store 時，請以 JPG 檔案個別送出 512 影像。它不會包含在 IPA 之中。

在 Flash Professional CS5 中，於「iPhone 設定」對話方塊的「圖示」標籤中新增這些圖示。請參閱第 13 頁「[在 Flash Professional CS5 中設定 iPhone 應用程式屬性](#)」。

您也可以將圖示的位置新增至應用程式描述器檔案：

```
<icon>
  <image29x29>icons/icon29.png</image29x29>
  <image57x57>icons/icon57.png</image57x57>
  <image72x72>icons/icon72.png</image72x72>
  <image512x512>icons/icon512.png</image512x512>
</icon>
```

iPhone 會對圖示添加閃光效果。您不需要在來源影像中加入這種效果。若要移除此預設閃光效果，請將以下項目新增至應用程式描述器檔案的 InfoAdditions 元素中：

```
<InfoAdditions>
  <![CDATA[
    <key>UIPrerenderedIcon</key>
    <true/>
  ]]>
</InfoAdditions>
```

請參閱第 14 頁「[在應用程式描述器檔案中設定 iPhone 應用程式屬性](#)」。

起始螢幕圖案 (Default.png)

所有 iPhone 應用程式在載入 iPhone 時都會顯示其起始影像。您將在名為 Default.png 的 PNG 檔中定義起始影像。在主要開發目錄中，建立名為 Default.png 的 PNG 檔。(請勿將此檔案放入子目錄。務必將檔案命名為大寫 D 的 Default.png)。

不管應用程式的起始方向為何，或應用程式是否設定為全螢幕，Default.png 檔的大小為寬 320 像素 x 高 480 像素。

如果應用程式的起始方向是橫向，請使用與直向應用程式相同的尺寸：寬 320 像素 x 高 480 像素。但您可以在 PNG 檔中逆時針旋轉圖案 90 度。PNG 圖案的左側會成為 iPhone 橫向模式的螢幕頂端（如需設定起始應用程式方向的資訊，請參閱第 12 頁「iPhone 應用程式設定」。）

而全螢幕的應用程式則會忽略預設影像圖案頂端的 20 像素。iPhone 狀態列會在預設影像的頂端以寬 20 像素的矩形顯示。在橫向的應用程式中，此區域為 Default.png 檔左側寬 20 像素的矩形（在橫向模式中顯示於頂端）。在直向應用程式中，此區域為 Default.png 檔頂端寬 20 像素的矩形。

在大部分的應用程式中，Default.png 影像應與應用程式的啟動螢幕相同。擷取應用程式的啟動螢幕快照：

- 1 在 iPhone 開啟應用程式。顯示第一個使用者介面螢幕時，按住「主畫面」按鈕（螢幕下方）。按住「主畫面」按鈕的同時，請按住「電源 / 睡眠」按鈕（在裝置的最上方）。這樣會擷取一張螢幕快照並傳送至「相機膠卷」。
- 2 藉由從 iPhoto 或其他照片傳送應用程式傳送照片，您可以將影像傳送至您的開發電腦（在 Mac OS，您也可以使用影像擷取應用程式）。

您也可以將照片透過電子郵件傳送至開發電腦：

- 開啟「照片」應用程式。
- 開啟「相機膠卷」。
- 開啟您擷取的螢幕快照影像。
- 點選影像，然後點選左下角的「下一頁」（箭號）按鈕。然後按一下「透過電子郵件傳送照片」按鈕，將影像傳送給自己。

備註：您可以為 Default.png 檔案建立任何圖案，但外觀比例必須正確。不過，最好還是讓 Default.png 影像與應用程式的起始狀態相符。

如果您的應用程式翻譯成多國語言，則請勿在 Default.png 影像中加入任何文字。因為 Default.png 是靜態的，所以影像中的文字將不適用於其他語言。

在 Flash Professional CS5 中，請務必將 Default.png 檔加入「iPhone 設定」對話方塊的「包含的檔案」清單。請參閱第 13 頁「在 Flash Professional CS5 中設定 iPhone 應用程式屬性」。

在命令列上使用 PFI 應用程式編譯時，請確定已參考位於包含的資源清單中的此檔案。請參閱第 18 頁「從命令列建立 iPhone 應用程式安裝程式檔案」。

iPhone 應用程式設定

應用程式設定包括以下各項：

- 應用程式名稱
- IPA 檔案名稱
- 應用程式版本
- 應用程式的起始螢幕方向，以及當 iPhone 旋轉時，螢幕方向是否自動隨著轉動。
- 起始螢幕是否為全螢幕
- 應用程式圖示相關資訊
- 硬體加速相關資訊

您可以在 Flash Professional CS5 中編輯應用程式的各項設定。

您也可以在應用程式描述器檔案中編輯這些設定。應用程式描述器檔案是內含應用程式各項設定的 XML 檔案。

在 Flash Professional CS5 中設定 iPhone 應用程式屬性

Flash Professional CS5 的「iPhone 設定」對話方塊可讓您定義 iPhone 應用程式的各種基本屬性。

開啟「iPhone 設定」對話方塊：

- ❖ 選擇「檔案 > iPhone 設定」。

一般標籤

「一般」標籤包含以下 iPhone 相關設定：

- 輸出檔案：顯示於 iPhone 應用程式圖示下方的應用程式名稱。請勿在輸出檔案名稱中使用加號 (+) 字元。
- 應用程式名稱：顯示於 iPhone 應用程式圖示下方的應用程式名稱。請勿在應用程式名稱中使用加號 (+) 字元。
- 版本：協助使用者判斷所安裝的應用程式版本。該版本將做為 iPhone 應用程式的 `CFBundleVersion`。它必須是類似 `nnnnn[.nn[.nn]]` 的格式，其中 `n` 是 0-9 的數字，而方括號代表選擇性組件（例如 1、1.0 或 1.0.1）。iPhone 版本必須僅包含數字以及小數點。iPhone 版本最多可包含兩個小數點。
- 外觀比例：應用程式的起始外觀比例（直向或橫向）。
- 全螢幕：應用程式是否使用全螢幕，或是否顯示 iPhone 狀態列。
- 自動方向：若選取此應用程式，當轉動 iPhone 的方向時，應用程式的顯示內容也會隨之轉向。

使用自動方向時，如果要得到最佳結果，請新增 `ActionScript` 程式碼，將舞台的 `align` 屬性設定成以下項目：

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- 顯示：如何在 iPhone 上顯示物件：
 - CPU：應用程式使用 CPU 來顯示所有的顯示物件。不使用硬體加速。
 - GPU：應用程式使用 iPhone GPU 來複合點陣圖。
 - 自動：此功能尚未實際應用。

如需詳細資訊，請參閱第 30 頁「[硬體加速](#)」。

- 包含的檔案：將所有檔案和目錄新增至 iPhone 應用程式的套件中。主要 SWF 檔與應用程式描述器檔案預設即包含在其中。將所有需要的資源加入「包含的檔案」清單。請務必將起始螢幕圖案檔案 (`Default.png`) 加入「包含的檔案」清單中。

部署標籤

「部署」標籤包括應用程式的簽名及編譯設定：

- iPhone 數位簽名：指定憑證的 P12 憑證檔案和密碼。您必須將 Apple iPhone 憑證轉換成 `.p12` 格式。如需詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。
- 佈建檔案：指向此應用程式的佈建檔案（您可以從 Apple 取得此檔案）。如需詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。
- 應用程式 ID：可識別唯一應用程式的應用程式 ID。如果佈建描述檔與特定應用程式 ID 關聯，這表示此欄位由 Flash Professional CS5 設定，您將無法編輯。否則，佈建描述檔允許多個（萬用字元）應用程式 ID。請提供一個應用程式 ID，此 ID 必須符合您提供給 Apple 的應用程式 ID 萬用字元樣式：
 - 如果您的 Apple 應用程式 ID 為 `com.myDomain.*`，則「iPhone 設定」對話方塊中的應用程式 ID 必須以 `com.myDomain` 開頭（例如 `com.myDomain.myApp` 或 `com.myDomain.app22`）。
 - 如果您的 Apple 應用程式 ID 為 `*`，則「iPhone 設定」對話方塊中的應用程式 ID 可以是任何有效的字元字串。

您可以在 iPhone Dev Center (<http://developer.apple.com/iphone>) 中找到與佈建描述檔相關的 Apple 應用程式 ID（或萬用字元應用程式 ID 樣式）。移至 iPhone Developer Program Portal，然後移至「Provisioning」區段。

重要事項：忽略 Apple 應用程式 ID 前方的字元。Apple 稱此字串為組合種子 ID。例如，如果 Apple 列出您的應用程式 ID 為 96LPVWEASL.com.example.bob.myApp，則請忽略 96LPVWEASL，使用 com.example.bob.myApp 做為您的應用程式 ID。如果 Apple 列出您的應用程式 ID 為 5RM86Z4DJM.*，則請忽略 5RM86Z4DJM。這是萬用字元應用程式 ID。

- iPhone 部署類型：
 - 快速發佈以進行裝置測試：選擇此選項可快速編譯應用程式的版本，方便在您的開發人員 iPhone 中進行測試。
 - 快速發佈以進行裝置除錯：選擇此選項可快速編譯應用程式的除錯版本，方便在您的開發人員 iPhone 中進行測試。透用此選項，Flash Professional CS5 除錯程式就可以收到 iPhone 應用程式的 trace() 輸出。(請參閱第 20 頁「iPhone 應用程式除錯」)。
 - 部署 - 臨機操作：選擇此選項可建立進行臨時部署的應用程式。請參閱 Apple iPhone 開發人員中心
 - 部署 - Apple App Store：選擇此選項可建立要部署到 Apple App Store 的最終版 IPA 檔。

圖示標籤

在「圖示」標籤上，指定 29 x 29 像素圖示影像、48 x 48 像素圖示影像、57 x 57 像素圖示影像、72 x 72 像素圖示影像及 512 x 512 像素圖示影像的位置。請參閱第 11 頁「iPhone 圖示與起始螢幕影像」。

備註：Flash Professional CS5 隨附的 Packager for iPhone Preview 版本，不包含 48 x 48 像素與 72 x 72 像素的選項。在 Flash Professional CS5 中，選取「說明 > 更新」以新增這些選項。

在應用程式描述器檔案中設定 iPhone 應用程式屬性

應用程式描述器檔案是內含整個應用程式屬性的 XML 檔案，例如其名稱、版本以及其他設定。

Flash Professional CS5 會根據「iPhone 設定」對話方塊中的設定，產生一個應用程式描述器檔案。但您也可以直接在文字編輯程式中編輯應用程式描述器檔案。Flash Professional 會在專案名稱後方加上 "-app.xml"，做為應用程式描述器檔案的名稱。例如，HelloWorld 專案的應用程式描述器檔案名稱會是 HelloWorld-app.xml。如果要定義 Flash Professional CS5 「iPhone 設定」對話方塊中不支援的設定，請編輯應用程式描述器檔案。例如，您可以定義 InfoAdditions 元素，以便為應用程式定義 info.Plist 設定。

重要事項：當 Flash Professional CS5 對話方塊開啟時，請勿編輯應用程式描述器檔案。請在開啟「iPhone 設定」對話方塊之前，先儲存應用程式描述器檔案的變更。

以下為範例應用程式描述器檔案：

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/2.0">
  <id>com.example.HelloWorld</id>
  <filename>HelloWorld</filename>
  <name>Hello World</name>
  <version>v1</version>
  <initialWindow>
    <renderMode>gpu</renderMode>
    <content>HelloWorld.swf</content>
    <fullScreen>true</fullScreen>
    <aspectRatio>portrait</aspectRatio>
    <autoOrients>true</autoOrients>
  </initialWindow>
  <supportedProfiles>mobileDevice desktop</supportedProfiles>
  <icon>
    <image29x29>icons/icon29.png</image29x29>
    <image57x57>icons/icon57.png</image57x57>
    <image512x512>icons/icon512.png</image512x512>
  </icon>
  <iPhone>
    <InfoAdditions>
      <![CDATA[
        <key>UIStatusBarStyle</key>
        <string>UIStatusBarStyleBlackOpaque</string>
        <key>UIRequiresPersistentWiFi</key>
        <string>NO</string>
      ]]>
    </InfoAdditions>
  </iPhone>
</application>
```

以下是此應用程式描述器檔案設定的詳細資訊：

- 在 <application> 元素中，建立 iPhone 應用程式所需的 AIR 2.0 命名空間：

```
<application xmlns="http://ns.adobe.com/air/application/2.0">
```

- <id> 元素：

<id>com.example.as3.HelloWorld</id> 應用程式 ID 可識別唯一的應用程式。建議的格式為逗點分隔的反向 DNS 樣式字串，例如 "com.company.AppName"。編譯器會使用此值做為 iPhone 應用程式的組合 ID。

如果佈建檔與特定應用程式 ID 關聯，請在此元素使用該應用程式 ID。忽略 Apple 在 Apple 應用程式 ID 開頭指派的字元（稱為組合種子 ID）。例如，如果佈建描述檔的應用程式 ID 為 96LPVWEASL.com.example.bob.myApp，請使用 com.example.bob.myApp 做為應用程式描述器檔案的應用程式 ID。

如果佈建描述檔允許多個（萬用字元）應用程式 ID，則應用程式 ID 會以星號結尾（such as 5RM86Z4DJM.*）。請提供一個應用程式 ID，此 ID 必須符合您提供給 Apple 的應用程式 ID 萬用字元樣式：

- 如果您的 Apple 應用程式 ID 為 com.myDomain.*，則應用程式描述器檔案中的應用程式 ID 必須以 com.myDomain 開頭。您可以指定應用程式 ID，例如 com.myDomain.myApp 或 com.myDomain.app22。
- 如果您的 Apple 應用程式 ID 為 *，則應用程式描述器檔案中的應用程式 ID 可以是任何有效的字元字串。

您可以在 iPhone Dev Center (<http://developer.apple.com/iphone>) 中找到與佈建描述檔相關的 Apple 應用程式 ID（或萬用字元應用程式 ID 樣式）。移至 iPhone Developer Program Portal，然後移至「Provisioning」區段。

重要事項：忽略 Apple 應用程式 ID 前方的字元。Apple 稱此字串為組合種子 ID。例如，如果 Apple 列出您的應用程式 ID 為 5RM86Z4DJM.*，則請忽略 5RM86Z4DJM。這是萬用字元應用程式 ID。如果 Apple 列出您的應用程式 ID 為 96LPVWEASL.com.example.bob.myApp，則請忽略 96LPVWEASL，使用 com.example.bob.myApp 做為您的應用程式 ID。

- <filename> 元素：

`<filename>HelloWorld</filename>` 用於 iPhone 安裝程式檔案的名稱。請勿在檔案名稱中使用加號 (+) 字元。

- `<name>` 元素：

`<name>Hello World</name>` 顯示於 iTunes 應用程式及 iPhone 中的應用程式名稱。請勿在名稱中使用加號 (+) 字元。

- `<version>` 元素：

`<version>1.0</version>`：協助使用者判斷所安裝的應用程式版本。該版本將做為 iPhone 應用程式的 `CFBundleVersion`。它必須是類似 `nnnnn[.nn[.nn]]` 的格式，其中 `n` 是 0-9 的數字，而方括號代表選擇性組件（例如 1、1.0 或 1.0.1）。iPhone 版本必須僅包含數字以及小數點。iPhone 版本最多可包含兩個小數點。

- `<initialWindow>` 元素包含以下子元素，可用來指定應用程式起始外觀的屬性：

`<content>HelloWorld.swf</content>` 識別要編譯成 iPhone 應用程式的根 SWF 檔。

`<visible>true</visible>` 這是必要的設定。

`<fullScreen>true</fullScreen>` 指定應用程式使用 iPhone 的整個螢幕。

`<aspectRatio>portrait</aspectRatio>` 指定應用程式的起始外觀比例為直向模式（不是橫向）。請注意，無論此設定為何，定義應用程式起始視窗之 `Default.png` 檔案的寬度必須是 320 像素，而高度必須是 480 像素。（請參閱第 11 頁「[iPhone 圖示與起始螢幕影像](#)」）。

`<autoOrients>true</autoOrients>`（選擇性）指定應用程式的內容方向是否會隨著裝置的實際方向自動變換方向。預設值是 `true`。您可以呼叫 `orientationChanging` 事件的 `preventDefault()` 方法，取消自動方向。由 Stage 物件傳送。如需詳細資訊，請參閱[設定以及偵測螢幕方向](#)。

使用自動方向時，如果要得到最佳結果，請將舞台的 `align` 屬性設定成以下項目：

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

`<renderMode>gpu</renderMode>`（選擇性）應用程式使用的顯示模式。目前有三種可能的設定：

- `cpu`：應用程式使用 CPU 來顯示所有顯示物件。不使用硬體加速。
- `gpu`：應用程式使用 iPhone GPU 來複合點陣圖。
- `auto`：此功能尚未實際應用。

如需詳細資訊，請參閱第 30 頁「[硬體加速](#)」。

- `<profiles>` 元素：

`<profiles>mobileDevice</profiles>` 限制應用程式只能編譯成行動裝置描述檔。此描述檔目前只支援 iPhone 應用程式。目前有三種支援的描述檔：

- `desktop`：桌上型 AIR 應用程式。
- `extendedDesktop`：支援原生程序 API 的桌上型 AIR 應用程式。
- `mobileDevice`：行動裝置專用的 AIR 應用程式。目前，iPhone 是唯一支援的行動裝置。

限制應用程式只能編譯成特定的描述檔，避免其被編譯成其他描述檔。如果您未指定描述檔，則可以將應用程式編譯成任何描述檔。您可以在 `<profiles>` 元素中逐一列出描述檔並以空格區隔，如此便能指定多個描述檔。

請務必包含 `mobileDevice` 做為支援的描述檔（或讓 `<profiles>` 元素空白）。

- `<icon>` 元素包含以下的子元素，這些元素可指定應用程式使用的圖示：

`<image29x29>icons/icon29.png</image29x29>` 這是用於 Spotlight 搜尋結果的影像。

`<image48x48>icons/icon48.png</image48x48>` 這是用於 iPad 上 Spotlight 搜尋結果的影像。

`<image57x57>icons/icon57.png</image57x57>` 這是用於 iPhone 與 iPod Touch 主畫面的影像。

`<image72x72>icons/icon72.png</image72x72>` 這是用於 iPad 主畫面的影像。

`<image512x512>icons/icon512.png</image512x512>` 這是 iTunes 應用程式使用的影像。

Packager for iPhone 工具會使用應用程式描述器檔案中所指示的 29、57、512 圖示。工具會將它們分別複製到 Icon-Small.png、Icon.png 及 iTunesArtwork 檔案中。若要避免此複製動作，您可以直接封裝這些檔案。將檔案放入內含應用程式描述器檔案的目錄以封裝這些檔案，然後列出正確的名稱和路徑。

512 影像僅供內部測試之用。將應用程式送出至 Apple 時，您會另外送出 512 影像。它不會包含在 IPA 之中。指定 512 影像可讓您在送出前，確定該影像可在 iTunes 中正常顯示。

- `<iPhone>` 元素包含以下的子元素，可指定 iPhone 特有的設定：

`<InfoAdditions></InfoAdditions>` 包含子元素，這些子元素會指定索引鍵值配對，做為應用程式的 Info.plist 設定：

```
<![CDATA [
  <key>UIStatusBarStyle</key>
  <string>UIStatusBarStyleBlackOpaque</string>
  <key>UIRequiresPersistentWiFi</key>
  <string>NO</string>
]]>
```

在此範例中，這些值會設定應用程式的狀態列樣式，並說明應用程式不需要持續存取 Wi-Fi。

InfoAdditions 設定包括在 CDATA 標籤中。

若要支援 iPad，請在 UIDeviceFamily 中包含索引鍵值設定。UIDeviceFamily 設定為字串陣列。每個字串定義支援的裝置。`<string>1</string>` 設定定義支援 iPhone 與 iPod Touch。`<string>2</string>` 設定定義支援 iPad。`<string>3</string>` 設定定義支援 tvOS。如果僅指定其中一個字串，則只會支援該裝置系列。例如，下列設定僅支援 iPad：

```
<key>UIDeviceFamily</key>
  <array>
    <string>2</string>
  </array>>
```

以下設定支援兩種裝置系列 (iPhone/iPod Touch 與 iPad)：

```
<key>UIDeviceFamily</key>
<array>
  <string>1</string>
  <string>2</string>
</array>
```

如需其他 Info.plist 設定的相關資訊，請參閱 Apple 開發人員文件。

編譯 iPhone 應用程式安裝程式 (IPA) 檔案

使用 Packager for iPhone 將 ActionScript-3.0 應用程式編譯成 IPA 安裝程式檔案。

使用 Flash Professional CS5 中隨附的 Packager for iPhone 來建立 iPhone 應用程式安裝程式檔案

若要使用 Flash Professional CS5 內含的 Packager for iPhone：

- 1 選取「檔案」>「發佈」。
- 2 確認已在「iPhone 設定」對話方塊中提供各項設定值。確認已在「部署」標籤中選取正確的選項。請參閱第 13 頁「[在 Flash Professional CS5 中設定 iPhone 應用程式屬性](#)」。
- 3 按一下「發佈」按鈕。

Packager for iPhone 會產生 iPhone 應用程式安裝程式 (IPA) 檔案。編譯 IPA 檔需要花費數分鐘的時間。

您也可以從命令列執行 Packager for iPhone 請參閱第 18 頁「[從命令列建立 iPhone 應用程式安裝程式檔案](#)」。

從命令列建立 iPhone 應用程式安裝程式檔案

您可以從命令列執行 Packager for iPhone。Packager for iPhone 會將 SWF 檔案位元組碼及其他來源檔案轉換成原生 iPhone 應用程式。

- 1 開啟命令殼層或終端機，然後移至 iPhone 應用程式的專案資料夾。
- 2 接下來，使用 pfi 工具以下列語法建立 IPA 檔案：

```
pfi -package -target [ipa-test ipa-debug ipa-app-store ipa-ad-hoc] -provisioning-profile PROFILE_PATH  
SIGNING_OPTIONS TARGET_IPA_FILE APP_DESCRIPTOR SOURCE_FILES
```

變更 pfi 參考，使其包含 pfi 應用程式的完整路徑。pfi 應用程式安裝在 Flash Professional CS5 安裝目錄的 pfi/bin 子目錄中。

選取與您想建立之 iPhone 應用程式類型對應的 -target 選項：

- -target ipa-test：選擇此選項可快速編譯應用程式的版本，方便在您的開發人員 iPhone 中進行測試。
- -target ipa-debug：選擇此選項以編譯應用程式的除錯版本，方便在您的開發人員 iPhone 中進行測試。透過此選項，您可以使用除錯工作階段從 iPhone 應用程式接收 trace() 輸出。

您可以包含下列 -connect 選項 (CONNECT_OPTIONS) 之一，指定執行除錯程式的開發電腦 IP 位址：

- -connect：應用程式會嘗試連線至編譯應用程式開發電腦上的除錯工作階段。
- -connect IP_ADDRESS：應用程式會嘗試連線至指定 IP 位址電腦上的除錯工作階段。例如：

```
-target ipa-debug -connect 192.0.32.10
```
- -connect HOST_NAME：應用程式會嘗試連線至指定主機名稱電腦上的除錯工作階段。例如：

```
-target ipa-debug -connect bobroberts-mac.example.com
```

備註：Flash Professional CS5 隨附的 Packager for iPhone Preview 不包含 -connect 選項。請在 Flash Professional CS5 中選取「說明 > 更新」，以更新 Packager for iPhone。

-connect 為選擇性選項。若未指定，產生的除錯應用程式將不會嘗試連線裝載的除錯程式。

如果除錯連線嘗試失敗，應用程式將會顯示對話方塊，要求使用者輸入除錯主機的 IP 位址。如果裝置未連線 wifi，連線嘗試將會失敗。如果裝置已連線，但是不在除錯主機的防火牆之後，也會失敗。

如需詳細資訊，請參閱第 20 頁「[iPhone 應用程式除錯](#)」。

您也可以包含 -renderingdiagnostics 選項，啟用 GPU 顯示診斷功能。如需詳細資訊，請參閱第 20 頁「[iPhone 應用程式除錯](#)」中的「使用 GPU 顯示診斷來除錯」。

- -target ipa-ad-hoc：選擇此選項可建立進行臨時部署的應用程式。請參閱 [Apple iPhone 開發人員中心](#)
- -target ipa-app-store：選擇此選項可建立要部署到 Apple App Store 的最終版 IPA 檔。

以應用程式的佈建描述檔路徑來取代 PROFILE_PATH。如需瞭解佈建描述檔的詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

取代 SIGNING_OPTIONS 以參照您的 iPhone 開發人員憑證和密碼 請使用下列語法：

```
-storetype pkcs12 -keystore P12_FILE_PATH -storepass PASSWORD
```

以您的 P12 憑證檔案來取代 P12_FILE_PATH。以憑證密碼來取代 PASSWORD。(請參閱以下範例。) 如需 P12 憑證的詳細資訊，請參閱第 5 頁「[將開發人員憑證轉換成 P12 檔案](#)」。

取代 APP_DESCRIPTOR 以參照應用程式描述器檔案。

取代 SOURCE_FILES 以參照專案的主要 SWF 檔案，並在後方加上所有要包括的資源。包括您在 Flash CS5 的應用程式設定對話方塊中，或在自訂的應用程式描述器檔案中，所有定義的圖示檔案路徑。同時，請加入起始螢幕圖案檔案 (Default.png)。

請參考下列範例：

```
pfcli -package -target ipa-test -storetype pkcs12 -keystore
"/Users/Jeff/iPhoneCerts/iPhoneDeveloper_Jeff.p12" -storepass dfb7VKL19 "HelloWorld.ipa" "HelloWorld-
app.xml" "HelloWorld.swf" "Default.png" "icons/icon29.png" "icons/icon57.png" "icons/icon512.png"
```

使用以下項目來編譯 HelloWorld.ipa 檔案：

- 使用憑證密碼 dfb7VKL19 的特定 PKCS#12 憑證
- HelloWorld-app.xml 應用程式描述器檔案
- 來源 HelloWorld.swf 檔案
- 特定的 Default.png 和圖示檔案

pfcli 應用程式會根據應用程式描述器檔案、SWF 檔案及其他資源，將應用程式編譯成 IPA 檔。

您可以在 Mac OS 使用儲存於鑰匙圈中的憑證，方法是將下列選項新增到 pfcli 命令：

```
-alias ALIAS_NAME -storetype KeychainStore -providerName Apple
```

以您要使用的憑證別名取代 ALIAS_NAME。當您指向儲存在 Mac 鑰匙圈中的憑證時，您要做的就是指定別名而不是指向憑證檔位置。

安裝 iPhone 應用程式

若要在 iPhone 安裝您開發的應用程式，必須先將佈建描述檔加入 iPhone，然後再將應用程式安裝到 iPhone。

將佈建描述檔加入 iPhone

將佈建描述檔加入 iPhone：

- 1 在 iTunes 中，選取「檔案 > 加到資料庫」。選取佈建描述檔 (副檔名為 mobileprovision)。

請確認已在佈建描述檔中加入您的 iPhone。您可以在 Apple iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>) 管理佈建描述檔。移至網站的「iPhone Developer Program Portal」區段。按一下「Devices」連結，管理可安裝開發應用程式的裝置清單。按一下「Provisioning」連結，管理您的佈建描述檔。

- 2 將 iPhone 連接到您的電腦並進行同步。

如需取得佈建描述檔的詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

安裝應用程式

安裝您所開發應用程式的方式與安裝其他 IPA 檔相同：

- 1 如果您曾經安裝應用程式的其他版本，請從裝置及 iTunes 的應用程式清單刪除該應用程式。
- 2 請使用下列任一方法將應用程式加入 iTunes：
 - 在「檔案」選單 (iTunes) 選擇「加到資料庫」命令。然後選取 IPA 檔，再按一下「開啟」按鈕。
 - 按兩下 IPA 檔。
 - 將 IPA 檔拖曳到 iTunes 資料庫。
- 3 將 iPhone 連接至電腦的 USB 連接埠。
- 4 在 iTunes 中，選取裝置的「應用程式」標籤，並確認在要安裝的應用程式清單中已選取該應用程式。

5 同步您的 iPhone。

疑難排解應用程式安裝問題

如果當您嘗試安裝應用程式時 iTunes 顯示錯誤，請檢查下列項目：

- 請確認已在佈建描述檔中加入裝置 ID。
- 請確認已安裝佈建描述檔，您可以將檔案拖曳到 iTunes，或使用「檔案 > 加到資料庫」命令。

此外，請檢查您的應用程式 ID 是否與 Apple 應用程式 ID 相符：

- 如果您的 Apple 應用程式 ID 為 com.myDomain.*，則應用程式描述器檔案的應用程式 ID，或「iPhone 設定」對話方塊中的應用程式 ID 必須以 com.myDomain 開頭（例如 com.myDomain.anythinghere）。
- 如果您的 Apple 應用程式 ID 為 com.myDomain.myApp，則應用程式描述器檔案或 Flash Profession CS5 使用者介面中的應用程式 ID 必須是 com.myDomain.myApp。
- 如果您的 Apple 應用程式 ID 為 *，則應用程式描述器檔案或 Flash Profession CS5 使用者介面中的應用程式 ID 可以是任意值。

您可以在 Flash Professional CS5 的「iPhone 設定」對話方塊或應用程式描述器檔案中，指定應用程式的應用程式 ID。

iPhone 應用程式除錯

在 ADL 中執行應用程式，便可以在開發電腦上進行應用程式的除錯。您也可以在 iPhone 中進行應用程式的除錯。

使用 ADL 測試應用程式（在開發電腦上）時，您仍然可以使用某些 iPhone 所不支援的 AIR 功能。在桌上型電腦測試內容時，請瞭解其中的差異。如需詳細資訊，請參閱第 23 頁「[行動裝置不支援 ActionScript 3.0 API](#)」。

在開發電腦上進行應用程式的除錯

若要使用 Flash Professional CS5 在開發電腦上進行應用程式的除錯：

- ❖ 選擇「除錯 > 影片除錯 > 在 AIR Debug Launcher（行動裝置）中」。

您也可以從命令列呼叫 ADL 來進行應用程式的除錯。語法如下：

```
adl -profile mobileDevice appDescriptorFile
```

以應用程式描述器檔案的路徑來取代 appDescriptorFile。

請務必包括 -profile mobileDevice 選項。

在 iPhone 上進行應用程式的除錯

在 iPhone 對應用程式進行除錯：

1 使用除錯支援來編譯應用程式：

- 在 Flash Professional CS5 中，使用「快速發佈以進行裝置除錯」設定來編譯應用程式（請參閱第 17 頁「[使用 Flash Professional CS5 中隨附的 Packager for iPhone 來建立 iPhone 應用程式安裝程式檔案](#)」）。
- 透過 PFI 命令列應用程式，使用 target ipa-debug 選項來編譯應用程式。（請參閱第 18 頁「[從命令列建立 iPhone 應用程式安裝程式檔案](#)」）。

2 在 iPhone 上安裝應用程式。

3 在 iPhone 上開啟 Wi-Fi，並連線至與開發電腦相同的網路。

4 在開發電腦上啟動除錯工作階段。在 Flash Professional CS5 中，選取「除錯 > 開始遠端除錯工作階段 > ActionScript 3.0」。

5 在 iPhone 上執行應用程式。

應用程式的除錯版本將提示您輸入開發人員電腦的 IP 位址。輸入 IP 位址，並點選「確認」按鈕。取得開發電腦的 IP 位址。

- 在 Mac OS 的 Apple 選單中選擇「系統偏好設定」。在「系統偏好設定」視窗中，按一下「網路」圖示。「網路」偏好設定視窗中便會列出 IP 位址。
- 在 Windows 中，啟動命令列工作階段，然後執行 ipconfig 命令。

除錯工作階段會顯示應用程式的任何 trace() 輸出。

對 iPhone 安裝的應用程式除錯時，Flash Professional CS5 支援所有除錯功能（包括中斷點控制、逐步執行程式碼以及變數監視）。

使用 GPU 顯示診斷來除錯

GPU 顯示診斷功能可讓您查看應用程式使用硬體加速的情形（針對使用 GPU 顯示模式的應用程式）。若要使用此功能，請使用命令列的 PFI 工具來編譯應用程式，並包含 -renderingdiagnostics 選項：

```
pfi -package -renderingdiagnostics -target ipa-debug -connect ...
```

-renderingdiagnostics 旗標必須緊接著 -package 旗標。

GPU 顯示診斷功能會以彩色矩形顯示所有顯示物件：

- 藍色：顯示物件不是點陣圖或未快取為點陣圖，且正在顯示。

如果未變更的顯示物件重複出現藍色，可能是因為它與移動的顯示物件相交。例如，此顯示物件可能是移動的顯示物件的背景。考慮將顯示物件快取為點陣圖。

如果某個您認為應該快取的物件重複出現藍色，可能是因為該物件使用 GPU 無法應用的效果。這些效果包括混合模式、顏色變形、scrollRect 屬性以及遮罩。

如果上傳至 GPU 的顯示物件超過記憶體限制，應用程式也會顯示藍色。

應用程式會記錄每個藍色矩形的訊息。應用程式會輸出這些訊息，以及其他 trace() 與除錯輸出訊息。

- 綠色：顯示物件為點陣圖或快取為點陣圖，且正首次上傳至 GPU。

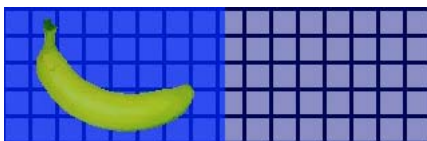
如果顯示物件重複出現綠色，表示應用程式中的程式碼正在重新建立顯示物件。例如，如果時間軸返回建立顯示物件的影格，便會發生此情況。考慮修改內容，以避免重新建立相同的物件。

- 紅色：顯示物件為點陣圖或快取為點陣圖，且正重新上傳至 GPU。

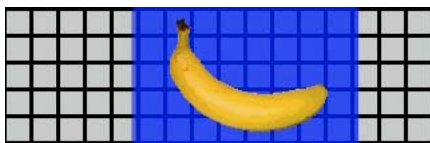
當顯示物件變更而使得應用程式需要重新顯示點陣圖形式時，便會出現紅色。例如，如果 2D 物件未設定 cacheAsBitmapMatrix 屬性，當它縮放或旋轉時便會重新顯示。當子顯示物件移動或變更時，也會重新顯示。

四個畫面重新繪製循環後，只要在這些循環期間不再發生出現彩色矩形的原因，每個彩色矩形便會淡出。不過，如果畫面沒有變更，診斷色彩也不會變更。

例如，假設在沒有快取為點陣圖的向量背景前有一個點陣圖顯示物件（一根香蕉）。當首次顯示香蕉時，色彩為綠色。當首次顯示背景時，色彩為藍色：

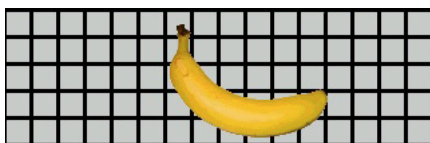


香蕉移動時，CPU 必須重新顯示背景，使得藍色陰影出現在背景上：



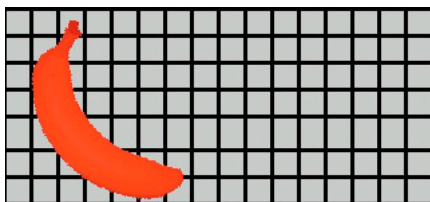
背景上的藍色陰影代表必須傳送至 GPU 的重新繪製區域。

不過，如果背景快取為點陣圖，當香蕉移動時，顯示診斷功能將不會顯示任何上色色彩：



由於 GPU 保留背景點陣圖，因此診斷功能不會顯示上色色彩。毋須涉及 CPU，GPU 便能複合香蕉與點陣圖。

假設香蕉為 2D 顯示物件，且未設定 `cacheAsBitmapMatrix` 屬性。當此顯示物件旋轉（或縮放）時，顯示診斷功能將顯示紅色。這代表應用程式必須將顯示物件的新版本上傳至 GPU：



將您的 iPhone 應用程式送出至 App Store

將您的應用程式送出至 App Store：

- 1 從 iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>) 取得散發憑證和佈建描述檔。

散發憑證會命名為「iPhone Developer: XXX.cer」，其中 XXX 是您的名稱。

如需詳細資訊，請參閱第 3 頁「[從 Apple 取得開發人員檔案](#)」。

- 2 將散發憑證轉換成 P12 檔案。

如需詳細資訊，請參閱第 5 頁「[將開發人員憑證轉換成 P12 檔案](#)」。

- 3 使用 P12 檔案以及佈建描述檔編譯您的應用程式。

使用您根據散發憑證轉換的 P12 檔案。使用與散發佈建描述檔相關的應用程式 ID。

如需詳細資訊，請參閱第 17 頁「[編譯 iPhone 應用程式安裝程式 \(IPA\) 檔案](#)」。

- 4 將應用程式送出至 iPhone Dev Center 網站 (<http://developer.apple.com/iphone/>)。

重要事項：Apple 要求您使用 Apple Application Loader 程式，以上傳應用程式至 App Store。Apple 僅發佈 Mac OS X 的 Application Loader。因此，使用 Windows 電腦開發 iPhone 的 AIR 應用程式時，您必須使用執行 OS X (版本 10.5.3，或更新的版本) 的電腦，將應用程式提交至 App Store。您可以從 Apple iOS Dev Center 取得 Application Loader 程式。

第 3 章 行動裝置的 ActionScript 3.0 API 支援

建立行動 AIR 應用程式時，您可以使用與開發 Adobe Flash Player 10.1 及 AIR 2 桌上型電腦應用程式相同的 ActionScript 3.0 API。不過，有一些例外情況和附帶條件。

行動裝置不支援 ActionScript 3.0 API

以行動裝置描述檔執行的應用程式不支援部分 ActionScript 3.0 API (例如，在 iPhone 上執行的應用程式)。

使用相同的 ActionScript 程式碼開發多個描述檔時 (例如桌上型電腦和行動裝置)，請使用程式碼來測試是否支援 API。例如，iPhone 應用程式不支援 NativeWindow 類別 (iPhone 應用程式無法使用或建立原生視窗)。若要測試應用程式是否在支援原生視窗的描述檔中執行 (例如桌上型電腦描述檔)，請檢查 NativeWindow.isSupported 屬性。

下表列出行動裝置描述檔不支援的 API。您也可以使用列出的屬性來檢查並判斷，在平台上執行的應用程式是否提供 API 的支援。

API	支援測試
Accessibility	Capabilities.hasAccessibility
Camera	Camera.isSupported
DatagramSocket	DatagramSocket.isSupported
DNSResolver	DNSResolver.isSupported
DockIcon	NativeApplication.supportsDockIcon
DRMManager	DRMManager.isSupported
EncryptedLocalStore	EncryptedLocalStore.isSupported
HTMLLoader	HTMLLoader.isSupported
LocalConnection	LocalConnection.isSupported
Microphone	Microphone.isSupported
NativeApplication.exit()	—
NativeApplication.menu	NativeApplication.supportsMenu
NativeApplication.isSetAsDefaultApplication()	NativeApplication.supportsDefaultApplication
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
PDF 支援	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported

API	支援測試
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

您無法為行動裝置描述檔撰寫以 HTML 和 JavaScript 為基礎的 AIR 應用程式。

僅局部支援某些 ActionScript 3.0 類別：

檔案

iPhone 應用程式只能存取應用程式目錄及應用程式儲存目錄。您也可以呼叫 `File.createTempFile()` 和 `File.createTempDirectory()`。呼叫作業來存取另一個目錄 (例如 `FileStream` 讀取或寫入方法) 會導致 `IOError` 例外。

iPhone 應用程式不支援原生檔案瀏覽器對話方塊，例如 `File.browseForOpen()` 方法提供的對話方塊。

Loader

在 iPhone 應用程式中，您無法使用 `Loader.load()` 方法。您無法在使用 `Loader.load()` 方法載入的 SWF 內容中，執行任何 ActionScript 程式碼。不過，您可以使用該 SWF 檔案中的資源 (例如元件庫中的影片片段、影像、字體以及聲音)。您也可以使用 `Loader.load()` 方法來載入影像檔案。

視訊

iPhone 上的 AIR 應用程式只支援 Sorensen 視訊和 ON2 VP6 視訊。

您可以使用 `navigateToURL()` 方法，在應用程式外面開啟 H.264 視訊。若為 `request` 參數，請傳送 `URLRequest` 物件及指向視訊的 URL。視訊會在 iPhone 的視訊播放程式中啟動。

文字欄位

iPhone 文字欄位的字體及其他設定各有不同的限制。請參閱第 32 頁「[字體及文字輸入](#)」。

不支援的 API 與使用 ADL 除錯

使用 ADL 測試應用程式 (在開發電腦上) 時，您仍然可以使用某些 iPhone 所不支援的 AIR 功能。使用 ADL 進行內容的測試時，請瞭解其中的差異。

這些功能包括以下的視訊和音效轉碼器：`Speex` (音效)、`H.264/AVC` (視訊) 以及 `AAC` (音效)。雖然在 iPhone 中執行的 AIR 應用程式無法使用這些轉碼器。但這些轉碼器可在桌上型電腦中正常運作。

輔助功能以及螢幕朗讀程式支援 Windows 中的 ADL 作業。不過，iPhone 並不支援這些 API。

從桌上型電腦的 ADL 使用 RTMPE 協定時，該協定可正常運作。不過，嘗試使用 RTMPE 協定連線的 `NetConnection` 則會在 iPhone 失敗。

使用 ADL 執行內容時，`Loader` 類別可以正常運作，沒有其他限制。不過，在 iPhone 中執行時，如果嘗試載入的 SWF 內容包含 ActionScript 位元組碼，就會產生錯誤訊息。

`Shader` 實體可在 ADL 中執行。然而，iPhone 不但無法解譯 `Pixel Bender` 位元組碼而且 `shader` 也無任何圖形效果。

如需詳細資訊，請參閱第 20 頁「[iPhone 應用程式除錯](#)」。

行動 AIR 應用程式專屬的 ActionScript API

下列 API 只能用於行動裝置上的 AIR 應用程式。它們目前無法在 Flash Player 或 AIR 的桌上型電腦版本上運作。

螢幕方向 API

螢幕方向 API 可讓您操作舞台和 iPhone 的方向：

- `Stage.autoOrients`：應用程式是否設定成允許舞台在裝置旋轉時，自動調整方向。在 Flash Professional CS5 「iPhone 設定」對話方塊中選取「自動方向」選項時，此屬性會設定成 `true` (您也可以在應用程式描述器檔案中，將 `autoOrients` 元素設定成 `true`)。請參閱第 12 頁「[iPhone 應用程式設定](#)」。替 `Stage` 物件新增 `orientationChanging` 事件偵聽程式，就可以取消自動方向調整。呼叫此事件物件的 `preventDefault()` 方法會取消自動方向調整。

使用自動方向時，如果要得到最佳結果，請將舞台的 `align` 屬性設定成以下項目：

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- `Stage.deviceOrientation`：裝置的實際方向。`StageOrientation` 類別會定義此屬性的值。
- `Stage.orientation`：舞台目前的方向。`StageOrientation` 類別會定義此屬性的值。
- `Stage.supportsOrientationChange`：在 iPhone 設定為 `true`，而在 AIR 應用程式設定為 `false`。
- `Stage.setOrientation()`：設定舞台的方向。此方法有一個參數，該參數是定義新舞台方向的字串。`StageOrientation` 類別的常數會定義該參數的可能值。
- `StageOrientation`：定義舞台方向值。例如，`StageOrientation.ROTATED_RIGHT` 表示舞台會以裝置的預設方向為基準向右旋轉。
- `StageOrientationEvent`：定義螢幕方向變更後，`Stage` 要傳送的事件。當使用者旋轉 iPhone 時即會發生此事件。目前有兩種類型的事件。當裝置旋轉時，`Stage` 會傳送 `orientationChanging` 事件。若要禁止舞台重新轉向，請呼叫 `orientationChanging` 事件物件的 `preventDefault()` 方法。當舞台重新轉向後，`Stage` 會傳送 `orientationChange` 事件。

目前，螢幕方向 API 只能用於行動裝置的 AIR 應用程式。如果行動裝置 AIR 應用程式與桌上型電腦 AIR 應用程式共享原始碼，請使用 `Stage.supportsOrientationChange` 屬性檢查是否支援此 API。

以下範例顯示使用者旋轉裝置時將產生的回應：

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);

function onOrientationChange(event:StageOrientationEvent):void
{
    switch (event.afterOrientation) {
        case StageOrientation.DEFAULT:
            // re-orient display objects based on
            // the default (right-side up) orientation.
            break;
        case StageOrientation.ROTATED_RIGHT:
            // Re-orient display objects based on
            // right-hand orientation.
            break;
        case StageOrientation.ROTATED_LEFT:
            // Re-orient display objects based on
            // left-hand orientation.
            break;
        case StageOrientation.UPSIDE_DOWN:
            // Re-orient display objects based on
            // upside-down orientation.
            break;
    }
}
```

在此範例中，我們將針對不同的舞台方向提供註解（並非功能程式碼）。

您可以呼叫 **Stage** 物件的 `setOrientation()` 方法來變更舞台方向。設定方向是一項非同步作業。您可以偵聽 `orientationChange` 事件，檢查方向轉換何時完成。以下的程式碼顯示如何將舞台設定成向右轉：

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);
stage.setOrientation(StageOrientation.ROTATED_RIGHT);

function onOrientationChange(event:StageOrientationEvent):void
{
    // Code to handle the new Stage orientation
}
```

當 **Stage** 旋轉時即會調整大小，而 **Stage** 物件會傳送 `resize` 事件。您可以在 **Stage** 調整並重新放置顯示物件，以回應 `resize` 事件。

NativeApplication.systemIdleMode 與 **SystemIdleMode**

`NativeApplication.systemIdleMode` 屬性可防止 iPhone 進入閒置模式。根據預設，若一段時間沒有使用觸控螢幕，iPhone 便會進入閒置模式。閒置模式不但會讓螢幕變暗，也會讓 iPhone 進入鎖定模式。此屬性可設定為下列其中一個值：

- `SystemIdleMode.NORMAL`：iPhone 會執行正常閒置模式行為。
- `SystemIdleMode.KEEP_AWAKE`：應用程式會嘗試避免 iPhone 進入閒置模式。

僅支援在行動裝置上使用此功能。不支援在桌上型電腦作業系統上執行的 AIR 應用程式。在桌上型電腦上執行的應用程式中，設定 `NativeApplication.systemIdleMode` 屬性不會有任何作用。

以下程式碼顯示如何停用 iPhone 閒置模式：

```
NativeApplication.nativeApplication.systemIdleMode = SystemIdleMode.KEEP_AWAKE;
```

CameraRoll

`CameraRoll` 類別可讓您在 iPhone 「相機膠捲」中新增影像。`addBitmapData()` 方法可新增影像到 iPhone 「相機膠捲」。此方法含參數 `bitmapData`。此參數為 `BitmapData` 物件，其中包含要新增到「相機膠捲」的影像。

僅支援在行動裝置上使用「相機膠捲」功能。不支援在桌上型電腦作業系統上執行的 AIR 應用程式。若要在執行階段時，檢查應用程式是否支援 `CameraRoll` 功能，請檢查靜態 `CameraRoll.supportsAddBitmapData` 屬性。

呼叫 `addBitmapData()` 方法後，`CameraRoll` 物件會傳送下列其中一個事件：

- `complete`：作業已順利完成。
- `error`：發生錯誤。例如，iPhone 可能沒有足夠的可用空間來儲存影像。

下列程式碼會將舞台的影像（螢幕擷取）新增到「相機膠捲」：

```

if (CameraRoll.supportsAddBitmapData)
{
    var cameraRoll:CameraRoll = new CameraRoll();
    cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);
    cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);
    var bitmapData:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight);
    bitmapData.draw(stage);
    cameraRoll.addBitmapData(bitmapData);
}
else
{
    trace("not supported.");
}

function onCrError(event:ErrorEvent):void
{
    // Notify user.
}

function onCrComplete(event:Event):void
{
    // Notify user.
}

```

DisplayObject.cacheAsBitmapMatrix

`cacheAsBitmapMatrix` 屬性為 `Matrix` 物件，定義當 `cacheAsBitmap` 設定為 `true` 時，該如何呈現顯示物件。當您要呈現顯示物件的點陣圖版本時，應用程式會使用此矩陣做為變形矩陣。

設定 `cacheAsBitmapMatrix` 後，應用程式會使用該矩陣（而不是顯示矩陣）來取得顯示的快取點陣圖影像。（顯示矩陣為顯示物件的 `transform.concatenatedMatrix` 值）。如果此矩陣與顯示矩陣不符，則點陣圖會視需要加以縮放或旋轉。

設定 `cacheAsBitmapMatrix` 的顯示物件，只會在 `cacheAsBitmapMatrix` 值變更時顯示。點陣圖會適當縮放或旋轉以符合顯示矩陣。

CPU 和 GPU 顯示都同時受益於使用 `cacheAsBitmapMatrix` 屬性，但一般而言 GPU 顯示受益較多。

備註：若要使用硬體加速，請在 Flash Professional CS5 「iPhone 設定」對話方塊中，將「一般」標籤的「顯示：」設定為「GPU」。（或在應用程式描述器檔案中將 `renderMode` 屬性設定為 `gpu`）。

例如，下列程式碼使用顯示物件的未變形點陣圖形式：

```

matrix:Matrix = new Matrix(); // creates an identity matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;

```

下列程式碼使用符合目前顯示方式的點陣圖形式：

```

mySprite.cacheAsBitmapMatrix = mySprite.transform.concatenatedMatrix;
mySprite.cacheAsBitmap = true;

```

一般而言，使用單位矩陣 (`new Matrix()`) 或 `transform.concatenatedMatrix` 就已足夠。但您可以使用其他矩陣（例如縮小矩陣）將不同的點陣圖上傳至 GPU。例如，下列範例套用將 `x` 和 `y` 軸縮小為 0.5 倍的 `cacheAsBitmapMatrix` 矩陣。GPU 使用的點陣圖物件雖然較小，但 GPU 會調整大小以符合顯示物件的 `transform.matrix` 屬性：

```

matrix:Matrix = new Matrix(); // creates an identity matrix
matrix.scale(0.5, 0.5); // scales the matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;

```

一般而言，您所選擇的矩陣必須會將顯示物件變形為應用程式中顯示的大小。例如，如果應用程式顯示縮小一半的 **Sprite** 點陣圖版本，請使用會縮小一半的矩陣。如果應用程式將顯示大於目前尺寸的 **Sprite**，請使用會放大比率的矩陣。

設定 `cacheAsBitmapMatrix` 屬性的顯示物件大小是有限制的。限制為 1020 x 1020 像素。設定 `cacheAsBitmapMatrix` 屬性的所有顯示物件像素總數是有限制的。限制大約為四百萬像素。

使用 `cacheAsBitmapMatrix` 與硬體加速時需考量多項因素。您必須知道哪些顯示物件需要設定屬性，而哪些不需要。如需使用此屬性的重要資訊，請參閱第 30 頁「[硬體加速](#)」。

您可以使用 GPU 顯示診斷功能，診斷應用程式除錯版本的 GPU 使用量。如需詳細資訊，請參閱第 20 頁「[iPhone 應用程式除錯](#)」。

網路注意事項

使用以下的 URL 配置加上 `navigateToURL()` 函數，可在外部的應用程式中開啟文件：

URL 配置	呼叫 <code>navigateToURL()</code> 之後的結果	範例
<code>mailto:</code>	在郵件應用程式開啟新郵件。	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>sms:</code>	在文字訊息應用程式中開啟一則訊息。	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>tel:</code>	在電話上撥打電話號碼（經使用者核准）。	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

iPhone 應用程式在安全交易時（例如 `https` 要求），會依賴安裝的自我簽署根憑證來進行伺服器驗證。伺服器不應只傳送分葉憑證，而是必須將所有中繼憑證鏈結傳送至根憑證。

行動應用程式開發人員特別感興趣的 ActionScript 3.0 API

下列 ActionScript 3.0 API 會定義適用於行動裝置的功能。

加速計 API

以下類別允許應用程式從裝置的加速計取得事件：

- `Accelerometer`
- `AccelerometerEvent`

如需詳細資訊，請參閱[加速計輸入](#)。

地理位置 API

以下類別允許應用程式從裝置的位置感應器取得事件：

- `Geolocation`
- `GeolocationEvent`

如需詳細資訊，請參閱[地理位置](#)。

觸控、多點觸控以及筆勢 API

以下類別允許應用程式接收觸碰和筆勢事件：

- `GestureEvent`
- `GesturePhase`
- `MultiTouch`
- `MultitouchInputMode`
- `TouchEvent`
- `TransformGestureEvent`

如需詳細資訊，請參閱[觸控、多點觸控和手勢輸入](#)。

第 4 章 iPhone 應用程式設計注意事項

進行特殊設計及編碼時，您必須考量 iPhone 的處理速度及螢幕大小。對您而言，開發所有應用程式或開發行動應用程式兩者間有著許多共通的设计考量。

如需最佳化應用程式的詳細資訊，請參閱[最佳化 Flash Platform 效能](#)。這份文件包含許多最佳化行動內容、Flash Player 內容、AIR 內容及 ActionScript 內容效能的建議。大部分的建議也適用於 iPhone 的 AIR 應用程式。

重要事項：開發可在 iPhone 上順利執行的應用程式時，這些设计考量和最佳化技巧都極為重要。

硬體加速

您可以使用 OpenGL ES 1.1 硬體加速來提昇部分應用程式的圖形效能。顯示動畫物件的遊戲和其他應用程式也都受益於硬體加速。使用硬體加速的應用程式可以從 CPU 卸載部分圖形程序給 iPhone GPU，如此可以大幅改善效能。

设计使用 GPU 的應用程式時，務必遵循相關規則，確保內容可由 GPU 有效加速。

若要使用硬體加速，請在 Flash Professional CS5 「iPhone 設定」對話方塊中，將「一般」標籤的「顯示:」設定為「GPU」。您也可以在此應用程式描述器檔案中，將 renderMode 屬性設定成 gpu：

```
<initialWindow>
  <renderMode>gpu</renderMode>
  ...
```

請參閱第 13 頁「[在 Flash Professional CS5 中設定 iPhone 應用程式屬性](#)」和第 14 頁「[在應用程式描述器檔案中設定 iPhone 應用程式屬性](#)」。

有四種類別的顯示物件，如果內容不常變更，可在硬體加速下迅速顯示。

- 點陣圖物件
- cacheAsBitmap 屬性設為 true 以及選擇性設定其 cacheAsBitmapMatrix 屬性的 2D 顯示物件 (參閱下文)
- 3D 顯示物件 (亦即，設定其 z 屬性)
- 填滿單一實色的矩形，且其邊緣在畫面上以像素為單位對齊的顯示物件。

只要另一個 Sprite 在向量物件的上方或下方製作動畫，向量物件就會重新顯示。因此，任何做為動畫背景或前景的物件也屬於這些類別。

對 cacheAsBitmap 設定成 true 的顯示物件而言，設定 cacheAsBitmapMatrix 會導致 GPU 使用矩陣變形所產生的點陣圖。即使旋轉或縮放物件，GPU 仍會使用點陣圖形式。GPU 複合並對此點陣圖製作動畫的速度，比 CPU 重新繪製向量顯示物件的速度快。

只將 cacheAsBitmap 設定為 true，將會快取顯示物件 (及任何子系)。公開新區域或轉譯整個組合的圖形時，不會重新繪製顯示物件。

當設定顯示物件的 cacheAsBitmapMatrix 屬性時，應用程式便可建立顯示物件的形式，即使物件是隱藏的。應用程式會在下個影格開始時，建立顯示物件的快取形式。接著，當您將顯示物件新增至舞台時，應用程式便可快速顯示該物件。應用程式同時也可以快速地執行旋轉或縮放物件等動畫效果。對旋轉或縮放的物件設定 cacheAsBitmap 屬性時，請務必同時設定 cacheAsBitmapMatrix 屬性。

應用程式也能針對快取為點陣圖的物件，快速地執行 Alpha 色彩變形。不過，硬體加速的 Alpha 色彩變形僅支援 0 至 1.0 之間的 alpha 值。該範圍可對應至 0 至 256 之間的 colorTransform.alphaMultiplier 設定。

請勿將經常更新物件 (例如文字欄位) 的 cacheAsBitmap 屬性設定成 true。

經常變更圖形內容的顯示物件，通常不適用於 GPU 顯示。針對 GPU 能力較弱的早期裝置更是如此。上傳圖形至 GPU 的額外負荷，使得 CPU 顯示成為較佳的選擇。

針對相對於父系移動的子顯示物件，重新建立包含這些物件的顯示物件。變更這些物件，讓子顯示物件成為父系的旁系。如此可確保這些物件每個都有各自的點陣圖形式。此外，每個顯示物件可以相對於其他物件移動，不需要上傳新圖形至 GPU。

在顯示清單的最高層級（此處的子顯示物件不是動畫），將 `cacheAsBitmap` 屬性設定成 `true`。換言之，針對不含移動組件的顯示物件容器設定此值。不要在子顯示物件上設定此值。當 `Sprite` 包含要製成動畫的其他顯示物件時，請不要在 `Sprite` 上設定此值。

設定顯示物件的 `z` 屬性時，應用程式一律會使用快取的點陣圖形式。在您設定顯示物件的 `z` 屬性後，即使您旋轉或縮放物件，應用程式還是會使用快取的點陣圖形式。對於已設定 `z` 屬性的顯示物件，應用程式不會使用 `cacheAsBitmapMatrix` 屬性。相同的規則也適用於設定任何 3D 顯示物件的屬性，這些屬性包括 `rotationX`、`rotationY`、`rotationZ` 以及 `transform.matrix3D` 屬性。

如果想在容器包含的內容中使用硬體加速，請勿設定顯示物件容器的 `scrollRect` 或 `mask` 屬性。因為設定這些屬性會停用顯示物件容器及其子物件的硬體加速。除了使用 `mask` 屬性外，您也可以在被遮蓋的顯示物件上放置遮色片顯示物件圖層。

使用硬體加速時，顯示物件的大小有所限制。在較舊的裝置上，長度與寬度不得超過 1024 個像素。在較新的裝置上，不得超過 2048 個像素。您可以使用 GPU 顯示診斷工具，測試裝置的效能。

GPU 也會使用 iPhone RAM 儲存點陣圖影像，而所使用的記憶體數量剛好是點陣圖影像所需的記憶體數量。

GPU 會針對點陣圖影像的每一個維度配置 2 乘方的記憶體。例如，GPU 可以保留 512 x 1024 或 8 x 32 大小的記憶體。因此，一個 9 x 15 像素的影像與 16 x 16 像素的影像所佔用的記憶體是相同的。針對快取的顯示物件，建議每個方向的維度接近 2 的乘方（不可超過）。例如，使用 32 x 16 像素顯示物件遠比 33 x 17 像素顯示物件更有效率。

請勿藉由調整舞台的大小來縮小已根據其他平台（例如桌上型電腦）調整大小的資源。請使用 `cacheAsBitmapMatrix` 屬性，或於發佈至 iPhone 之前先調整資源的大小。快取表面影像時，3D 物件會忽略 `cacheAsBitmapMatrix` 屬性。基於這個理由，如果要在 3D 表面上呈現顯示物件，您最好在發佈之前先調整顯示物件的大小。

硬體加速的效果及 RAM 使用量是一種取捨關係。隨著記憶體空間漸漸不足，iPhone OS 會通知其他執行中的原生 iPhone 應用程式釋放記憶體。當這些應用程式處理這項通知並開始釋放記憶體時，這些應用程式便可能與您的應用程式競爭 CPU 循環。這樣會暫時降低應用程式的效能。請務必在較舊的裝置上測試應用程式，因為基本上舊裝置可提供給執行中程序使用的記憶體較少。

在 iPhone 上執行應用程式除錯時，可啟用 GPU 顯示診斷功能。此功能可協助您查看應用程式使用 GPU 顯示的情形。如需詳細資訊，請參閱第 20 頁「iPhone 應用程式除錯」中的「使用 GPU 顯示診斷來除錯」。

如需使用 `cacheAsBitmapMatrix` 屬性的詳細資訊，請參閱第 25 頁「行動 AIR 應用程式專屬的 ActionScript API」的「`DisplayObject.cacheAsBitmapMatrix`」一節。

其他改善顯示物件效能的方法

硬體加速會提昇部分類別之顯示物件的圖形效能。以下是最大化圖形效能的秘訣：

- 嘗試限制舞台的可見項目數。每個項目都需要花費一些時間來顯示，並與周圍的其他項目複合。
當您不再需要呈現某個顯示物件時，可以將其 `visible` 屬性設定成 `false`，或從舞台移除該物件 (`removeChild()`)。不要只將其 `alpha` 屬性設定成 0。
- 一般而言，避免混合模式，特別是圖層混合模式。儘可能使用正常混合模式。
- 顯示物件濾鏡是高成本的計算。請勿頻繁使用它們。例如，您可以在介紹螢幕使用濾鏡。但請避免在下列情況使用濾鏡：眾多物件、正在製作動畫的物件，或必須使用高影格速率時。
- 避免變形形狀。
- 避免使用剪裁。

- 如有可能，請在呼叫 `Graphic.beginBitmapFill()` 方法時，將 `repeat` 參數設定成 `false`。
- 不要過度繪圖。使用背景顏色當做背景。避免大的圖形相互重疊。繪製每個像素都需要付出成本。針對沒有硬體加速的顯示物件更是如此。
- 避免細長尖狀、邊緣自交或邊緣包含大量細節的形狀。比起邊緣平滑的顯示物件，這些形狀需要更長的時間才能顯示。針對沒有硬體加速的顯示物件更是如此。
- 讓點陣圖的大小接近，但小於 $2^n \times 2^m$ 位元。尺寸不必是 2 乘方，但必須接近 2 的乘方（不一定要大於 2 的乘方）。例如， 31×15 像素影像的顯示速度快於 33×17 像素影像的顯示速度（31 和 15 剛好小於 2 的乘方：32 和 16）。此類影像也會更有效率地使用記憶體。
- 限制顯示物件的大小為 1024×1024 像素（在較新裝置上則為 2048×2048 ）。

資訊密度

行動裝置螢幕的實際大小小於桌上型電腦，但行動裝置螢幕的像素密度較高。銳利化的文字看起來很美觀，但字符必須等於最小的實際大小才能達到效果。

您通常會在移動及光線不足的情況下使用行動裝置。請考慮螢幕實際能顯示的資訊數量。與相同像素尺寸的桌上型電腦螢幕相比，行動裝置所能顯示的資訊量可能較少。

使用印刷階層來反白標示重要的資訊。使用字體大小、寬度、位置以及間距，來表示使用者介面元素的相對重要性。您可以在階層的每個層級使用一或多個視覺提示。在您的應用程式中一致地套用這些視覺提示。視覺提示可以是空間（內縮、行距、位置）或圖形（字樣的大小、樣式或顏色）。套用繁複視覺提示是一種確認已明確表示階層的有效方法。不過，最好不要在群組的每個階層中使用三個以上的視覺提示。

試著簡化必要的標籤及說明文字。例如，在文字欄位中使用範例輸入，藉此建議內容並避免標籤分散。

字體及文字輸入

若要取得最佳的外觀，請使用裝置字體。例如，以下字體是 iPhone 的裝置字體：

- Serif：Times New Roman、Georgia 及 `_serif`
- Sans-serif：Helvetica、Arial、Verdana、Trebuchet、Tahoma 及 `_sans`
- 等寬字體：Courier New、Courier 及 `_typewriter`

使用 14（或更多）像素的字體。

在可編輯的文字欄位使用裝置字體。文字欄位中的裝置字體會比內嵌字體更快顯示出來。

請勿在輸入文字欄位中使用加底線的文字。另外，請勿設定文字欄位的對齊方式。iPhone 的輸入文字欄位僅支援靠左對齊（預設）。

如果您在 Flash Professional CS5 的文字欄位使用「TLF 文字」設定，請到「ActionScript 3.0 設定」的預設連結中關閉執行階段共享元件庫。否則，應用程式將無法在 iPhone 中作用，因為它會嘗試使用執行階段共享元件庫 SWF 檔：

- 1 選取「檔案 > 發佈設定」。
- 2 在「發佈設定」對話方塊中，按一下「Flash」標籤。
- 3 按一下「Script (ActionScript 3.0)」下拉式清單右方的「Script」按鈕。
- 4 按一下「元件庫路徑」標籤。
- 5 在「預設連結」下拉式清單中，選取「合併至程式碼」。

考慮使用其他方法，而不使用輸入文字欄位。例如，若要讓使用者輸入數字，則不需文字欄位。您可以提供兩個按鈕來增加或減少數值。

請瞭解虛擬鍵盤使用的空間。啟用虛擬鍵盤時（例如，使用者在文字欄位內點一下），應用程式會調整舞台的位置。自動重新定位可確保顯示選取的輸入文字欄位：

- 舞台最上方的文字欄位會移至可見舞台區域的頂端（為了配合虛擬鍵盤，可見舞台區域會變小）。
- 舞台最下方的文字欄位還是會出現在新舞台區域的底端。
- 舞台其他部分的文字欄位會移至舞台的垂直中心處。

當使用者按一下文字欄位進行編輯時（並顯示虛擬鍵盤），`TextField` 物件會傳送一個 `focusIn` 事件。您可以對此事件新增事件偵聽程式，藉以重新定位文字欄位。

當使用者編輯文字時，單行文字欄位會包含一個清除按鈕（在文字的右方）。不過，如果文字欄位太窄，將無法顯示此清除按鈕。

在單行文字欄位編輯文字之後，使用者可以輕點鍵盤上的「完成」按鍵來隱藏虛擬鍵盤。

在多行文字欄位中編輯文字之後，使用者可以輕點文字欄位外的位置來隱藏虛擬鍵盤。這樣會將焦點移開文字欄位。請確定設計時，已在文字欄位外加入一個可顯示虛擬鍵盤的區域。如果文字欄位太大，使用者將看不到其他區域。

使用某些 **Flash Professional CS5** 組件可以防止焦點離開文字欄位。這些組件是為了桌上型電腦所設計的，因為桌上型電腦需要此種焦點行為。其中一個組件是 `TextArea` 組件。當該組件為焦點（且可被編輯）時，按一下其他顯示物件並不會移開焦點。此外，將某些其他 **Flash Professional CS5** 組件放置於舞台，也可防止焦點離開可編輯的文字欄位。

請不要依賴鍵盤事件。例如，部分為網站設計的 `SWF` 內容會使用鍵盤來允許使用者控制應用程式。不過，**iPhone** 只有當使用者編輯文字欄位時，才會顯示虛擬鍵盤。**iPhone** 應用程式只有在虛擬鍵盤存在時，才會傳送鍵盤事件。

儲存應用程式狀態

您的應用程式隨時都可能結束（例如，當手機響起時）。請考慮一旦應用程式變更狀態，就加以儲存。例如，您可以將設定儲存至應用程式儲存目錄的檔案或資料庫中。或者，您可以將資料儲存至本機共享物件。當應用程式重新啟動時，您便可以還原應用程式的狀態。如果來電中斷了應用程式，應用程式將在通話結束後重新啟動。

應用程式結束時，請不要依賴傳送 `exiting` 事件的 `NativeApplication` 物件；因為它可能不進行傳送。

螢幕方向變更

iPhone 內容可以直向或橫向檢視。請考量您的應用程式將如何處理螢幕方向變更。如需詳細資訊，請參閱[偵測裝置方向](#)。

感應區目標

設計使用者可以點選的按鈕及其他使用者介面元素時，請考量感應區目標的大小。這些元素的大小，必須足以讓手指在觸控螢幕上輕易的啟動它們。此外，您也必須確定各個目標之間有足夠的空間。感應區目標的大小必須大約在 44 像素到 57 像素之間。

記憶體配置

配置全新記憶體區塊的成本很高。觸發記憶體回收後，它會於動畫或互動期間減慢應用程式速度或造成效能延遲。

請盡可能隨時回收多個物件，而不是回收一個物件後，再重新建立一個新的物件。

請記住，向量物件消耗的記憶體比陣列少。請參閱 [Vector 類別與 Array 類別](#)。

如需記憶體使用量的詳細資訊，請參閱 [節省記憶體](#)。

繪圖 API

試著避免使用 **ActionScript 繪圖 API (Graphics 類別)** 來建立圖形。在舞台上使用繪圖 API 動態建立物件，然後將這些物件顯示於點陣處理器。如有可能，請改為在舞台的編寫期間以靜態方式建立這些物件。

使用繪圖 API 建立的物件，在重複呼叫時會於每次執行 **ActionScript** 時遭破壞並重新建立。不過，記憶體中的靜態物件位於不同的時間軸。

事件反昇

在深度巢狀的顯示物件容器中，事件反昇的成本相當高。在目標物件中徹底處理事件，然後呼叫事件物件的 `stopPropagation()` 方法，便可降低此成本。呼叫此方法可避免事件反昇。呼叫此方法也表示父物件不會接收事件。

將顯示物件的巢狀結構扁平化，可藉由避免冗長的事件鏈結來降低成本。

如有可能，請登錄 **MouseEvent** 事件而不是 **TouchEvent** 事件。因為 **MouseEvent** 事件的處理器使用量比 **TouchEvent** 事件來得少。

如有可能，請將 `mouseEnabled` 和 `mouseChildren` 屬性設定成 `false`。

最佳化視訊效能

若要最佳化行動視訊播放，請確認應用程式在播放視訊的同時，沒有執行太多其他的作業。如此一來，視訊便可在解碼與顯示過程中盡可能使用大量的 CPU。

播放視訊時，請盡量執行少量 (或完全不執行) **ActionScript** 程式碼。試著避免執行經常於間隔計時器或時間軸上執行的程式碼。

將重新繪製非視訊物件的次數減至最少。尤其是避免重新繪製與視訊區域交集的顯示物件。即使這些顯示物件隱藏在視訊下方，也要避免重新繪製。如果重新繪製這些隱藏的物件，仍會耗用處理資源。例如，使用簡單形狀的位置指示器，每秒只更新位置指示器一兩次，而不要每個影格都更新一次。請勿將視訊控制項與視訊區域重疊，您應將視訊控制項放在視訊區域的下方。如果視訊使用緩衝動畫功能，請勿將未使用的視訊控制項藏在視訊的後方，應該要將視訊控制項設定為隱藏。

Flex 與 Flash 組件

許多 **Flex** 組件與 **Flash** 組件是設計用於桌上型電腦應用程式。這些組件 (特別是顯示組件) 在行動裝置上可能會執行緩慢。除了執行緩慢之外，桌上型電腦組件的互動模型可能不適用於行動裝置。

Adobe 正在開發 Flex 架構的行動裝置最佳化版本。如需詳細資訊，請參閱 <http://labs.adobe.com/technologies/flex/mobile/>。

縮小應用程式檔案

以下提供一些縮小 IPA 檔案大小的秘訣：

- 檢查背景點陣圖是否為正確大小（不要大於所需的尺寸）。
- 檢查是否內嵌多餘的字體。
- 尋找 PNG 資源中的 Alpha 色版，若無需要請加以移除。使用公用程式（如 PNG Crunch）來縮小 PNG 資源的大小。
- 儘可能將 PNG 資源轉換成 JPG 資源。
- 考慮壓縮聲音檔案（使用較低的位元速率）
- 移除所有不使用的資源。