

使用 Packager for iPhone[®] 构建 ADOBE[®] AIR[®] 应用程序

法律声明

有关法律声明，请参阅 http://help.adobe.com/zh_CN/legalnotices/index.html。

目录

第 1 章：为 iPhone 构建 AIR 应用程序快速入门	
重要概念	1
从 Adobe 获取开发人员工具	3
从 Apple 获取开发人员文件	3
使用 Flash Professional CS5 创建 Hello World iPhone 应用程序	7
第 2 章：编译和调试 iPhone 应用程序	
iPhone 图标和初始屏幕图像	11
iPhone 应用程序设置	12
编译 iPhone 应用程序安装程序 (IPA) 文件	17
安装 iPhone 应用程序	19
调试 iPhone 应用程序	20
向 App Store 提交 iPhone 应用程序	22
第 3 章：移动设备的 ActionScript 3.0 API 支持	
移动设备不支持 ActionScript 3.0 API	23
特定于移动 AIR 应用程序的 ActionScript API	25
非常适用于移动应用程序开发人员的 ActionScript 3.0 API	28
第 4 章：iPhone 应用程序设计注意事项	
硬件加速	30
提高显示对象性能的其他方式	31
信息密度	32
字体和文本输入	32
保存应用程序状态	33
屏幕方向更改	33
感应区目标	33
内存分配	34
绘图 API	34
事件冒泡	34
优化视频性能	34
Flex 和 Flash 组件	34
减少应用程序文件大小	35

第 1 章：为 iPhone 构建 AIR 应用程序快速入门

您可以使用 Adobe® Flash® 平台工具和 ActionScript® 3.0 代码构建 iPhone 和 iPod Touch 的 Adobe® AIR® 应用程序。这些应用程序的分发、安装和运行方式与其他 iPhone 应用程序相同。

注：此文档的其余部分将 iPhone 和 iPod Touch 合称为“iPhone”。

Packager for iPhone® 包含在 Adobe® Flash® Professional CS5 中。Packager for iPhone 将 ActionScript 3.0 字节代码编译为本地 iPhone 应用程序代码。iPhone 应用程序通过 iTunes Store 以 iPhone 应用程序安装程序文件（.ipa 文件）的形式进行分发。

您可以使用 Flash Professional CS5 或 Adobe® Flash® Builder™ 4 编辑应用程序的源 ActionScript 3.0 内容。

要开发 iPhone 应用程序，请使用 Flash Professional CS5。

还需要从 Apple 获取 iPhone 开发人员证书。

重要事项：在开发 iPhone 应用程序之前，请仔细阅读有关设计 iPhone 应用程序的信息。请参见第 30 页的“[iPhone 应用程序设计注意事项](#)”。另外，请了解构建 iPhone 应用程序所需的开发人员文件。请参见第 3 页的“[从 Apple 获取开发人员文件](#)”。

重要概念

在使用 ActionScript 3.0 开发 iPhone 应用程序之前，了解涉及的概念和 workflows 非常重要。

词汇表

了解下列术语对构建 iPhone 应用程序非常重要。

iPhone 开发人员中心站点 在 Apple Computer 网站 (<http://developer.apple.com/iphone/>) 中，您可以：

- 申请成为 iPhone 开发人员。
- 管理和创建 iPhone 开发证书、供给配置文件和应用程序 ID（相关定义稍后介绍）。
- 向应用程序存储提交应用程序。

iPhone 开发证书 用于标识以开发应用程序为目的的开发人员。

从 Apple 获取此文件。将此证书转换成 P12 证书文件以签署您使用 ActionScript 3.0 创建的 iPhone 应用程序。请参见 P12 证书文件。

如果您只是在开发计算机上调试和测试 Flash Professional CS5 应用程序，则不需要 iPhone 开发证书。然而，如果要在 iPhone 上安装并测试应用程序，则需要开发证书。

开发证书与分发证书不同，后者用于构建应用程序的最终版本。当构建应用程序的最终版本时，应从 Apple 获取分发证书。

证书签名请求 一个包含用于生成开发证书的个人信息的文件。也称为 CSR 文件。

供给配置文件 一个允许您测试或分发 iPhone 应用程序的文件。从 Apple 获取供给配置文件。供给配置文件被分配给特定开发证书、应用程序 ID 以及一个或多个设备 ID。以下是不同类型的供给配置文件：

- 开发供给配置文件 — 用于将应用程序的测试版本安装到开发人员的 iPhone 中。
- 测试供给配置文件 — 也称为临时供给配置文件。用于将应用程序的测试版本分发给多个用户（和 iPhone 单元）。借助此供给配置文件和此测试应用程序，用户可测试其应用程序，而无需将其提交给 App Store。注意：您还可以使用开发供给配置文件将测试应用程序分发给多个设备。

- 分发供给配置文件 — 用于构建 iPhone 应用程序，以将应用程序提交给 App Store。

应用程序 ID 标识由特定开发人员开发的 iPhone 应用程序（或多个应用程序）的唯一字符串。应在 iPhone 开发人员中心站点创建应用程序 ID。每个供给配置文件都具有一个关联的应用程序 ID 或应用程序 ID 模式。当开发应用程序时应使用此应用程序 ID（或模式）。应在 Flash Professional CS5 的“iPhone 设置”对话框中或在应用程序描述符文件中使用应用程序 ID。

iPhone 开发人员中心的应用程序 ID 包含一个绑定种子 ID（后面带有绑定标识符）。捆绑种子 ID 是 Apple 分配给应用程序 ID 的一个字符串，例如 5RM86Z4DJM。绑定标识符包含一个您选择的反向域名字符串。绑定标识符可能以星号 (*) 结尾，表示通配符应用程序 ID。例如：

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.*（通配符应用程序 ID）

iPhone 开发人员中心提供了两种应用程序 ID：

- 通配符应用程序 ID — 在 iPhone 开发人员中心，这些应用程序 ID 以星号 (*) 结尾，例如 96LPVWEASL.com.myDomain.* 或 96LPVWEASL.*。借助使用这种应用程序 ID 的供给配置文件，您可以生成测试应用程序，并且这些应用程序使用的应用程序 ID 与该模式匹配。对于应用程序的应用程序 ID，您可以将星号替换为任何有效字符串。例如，如果 iPhone 开发人员中心站点将 96LPVWEASL.com.example.* 指定为应用程序 ID，则您可以将 com.example.foo 或 com.example.bar 用作应用程序的应用程序 ID。
- 特定应用程序 ID — 它们定义在应用程序中使用的唯一应用程序 ID。在 iPhone 开发人员中心，这些应用程序 ID 不以星号结尾。例如：96LPVWEASL.com.myDomain.myApp。借助使用这种应用程序 ID 的供给配置文件，应用程序必须与该应用程序 ID 完全匹配。例如，如果 iPhone 开发人员中心站点将 96LPVWEASL.com.example.helloWorld 指定为应用程序 ID，您必须将 com.example.foo 用作应用程序的应用程序 ID。

在开发应用程序时，应在 Flash Professional CS5 的“iPhone 设置”对话框中或在应用程序描述符文件中指定应用程序 ID。有关应用程序 ID 的详细信息，请参阅第 13 页的“在 Flash Professional CS5 中设置 iPhone 应用程序属性”中的“部署”选项卡部分，或参阅第 14 页的“在应用程序描述符文件中设置 iPhone 应用程序属性”。

重要事项：指定应用程序 ID 时，请忽略应用程序 ID 的绑定种子 ID 部分。例如，如果 Apple 显示您的应用程序 ID 为 96LPVWEASL.com.example.bob.myApp，请忽略 96LPVWEASL，并将 com.example.bob.myApp 用作应用程序 ID。如果 Apple 显示您的应用程序 ID 为 5RM86Z4DJM.*，请忽略 5RM86Z4DJM，这是一个通配符应用程序 ID。

您可以在 iPhone 开发人员中心 (<http://developer.apple.com/iphone>) 找到与供给配置文件关联的应用程序 ID（或通配符应用程序 ID 模式）。转到“iPhone Developer Program Portal”，然后转到“Provisioning”部分。

P12 证书文件 P12 文件（扩展名为 .p12 的文件）是一种证书文件（个人信息交换文件）。Packager for iPhone 使用此类证书来构建 iPhone 应用程序。您应从 Apple 收到的开发人员证书转换为此形式的证书。

唯一设备 ID 用于标识特定 iPhone 的唯一代码。也称为 UDID 或设备 ID。

开发工作流程概述

请按照下列步骤开发 iPhone 的应用程序：

- 1 从 Adobe 下载并安装 Flash Professional CS5。
- 2 安装 iTunes。
- 3 从 Apple 获取开发人员文件。这些文件包括开发人员证书和供给配置文件。请参见第 3 页的“从 Apple 获取开发人员文件”。
- 4 将开发证书转换成 P12 证书文件。Flash CS5 要求使用 P12 证书。请参见第 3 页的“从 Apple 获取开发人员文件”。
- 5 使用 iTunes 将供给配置文件与 iPhone 相关联。
- 6 使用 Flash Professional CS5 编写应用程序。

了解设计和优化 iPhone 应用程序的代码的最佳实践非常重要。请参见第 30 页的“iPhone 应用程序设计注意事项”。

另外,某些 ActionScript 3.0 API 功能受限或不受 iPhone 支持。请参见第 23 页的“[移动设备的 ActionScript 3.0 API 支持](#)”。

您还可以使用 Flash Builder 4.0 编辑应用程序的 ActionScript 3.0 代码。

您可以使用 Flash Professional CS5 在开发计算机中测试您的应用程序。

7 为应用程序创建图标图片和初始屏幕图片。每个 iPhone 应用程序都包含一组用于向用户标识该程序的图标。iPhone 在加载程序时将显示初始屏幕图像。请参见第 11 页的“[iPhone 图标和初始屏幕图像](#)”。

8 编辑 iPhone 设置。这些设置包括:

- 应用程序的标识 (包括文件名、应用程序名称、版本号和应用程序 ID)
- 应用程序的源图标图片的位置
- 分配给应用程序的 P12 证书和供给配置文件
- 应用程序的初始高宽比

在 Flash Professional CS5 中,您可以在“iPhone 设置”对话框中编辑这些设置。有关详细信息,请参阅第 13 页的“[在 Flash Professional CS5 中设置 iPhone 应用程序属性](#)”。

您也可以直接在应用程序描述符文件中编辑这些设置。有关详细信息,请参阅第 14 页的“[在应用程序描述符文件中设置 iPhone 应用程序属性](#)”。

9 使用 Packager for iPhone 编译 IPA 文件。请参见第 17 页的“[编译 iPhone 应用程序安装程序 \(IPA\) 文件](#)”。

10 在您的 iPhone 上安装并测试应用程序。使用 iTunes 安装 IPA 文件。

对于临时分布,请重复此常用过程,但使用测试供给配置文件 (而不是开发供给配置文件)。对于应用程序的最终发布,请使用发布供给配置文件重复此过程。(有关不同类型的供给配置文件的信息,请参见第 1 页的“[词汇表](#)”。)

在构建应用程序的分发版本后,请参阅第 22 页的“[向 App Store 提交 iPhone 应用程序](#)”中的说明。

有关构建基础 iPhone 应用程序的快速教程,请参阅第 7 页的“[使用 Flash Professional CS5 创建 Hello World iPhone 应用程序](#)”。

从 Adobe 获取开发人员工具

要使用 ActionScript 3.0 开发 iPhone 应用程序,您需要 Flash Professional CS5。

重要信息:您应当对 Flash Professional CS5 中包含的预览版本的 Packager for iPhone 进行版本更新。在 Flash Professional CS5 中,选择“帮助”>“更新”。

您也可以使用 Flash Builder 4 编辑 ActionScript 代码。<http://www.adobe.com/cn/products/flashbuilder/> 上提供了 Flash Builder 4。

从 Apple 获取开发人员文件

与开发 iPhone 的任何应用程序一样,您必须首先从 Apple 获取 iPhone 开发人员文件。您需要获取 iPhone 开发人员证书和移动供给配置文件。您还需要获取其他供给配置文件。有关这些文件的定义,请参见第 1 页的“[词汇表](#)”。

注:获取这些文件是应用程序开发过程中的一个重要部分。请确保在开发应用程序之前完成此过程。获取开发人员文件不是一个简单的过程。请仔细阅读这些说明以及 Apple iPhone 开发人员中心站点上的说明。

获取并使用 iPhone 开发人员文件

您需要从 Apple 获取 iPhone 开发人员证书和供给配置文件。您还需要将此证书转换成 P12 证书。

安装 iTunes

您需要使用 iTunes 将应用程序安装到 iPhone 上。另外，还需要使用 iTunes 确定 iPhone 的设备 ID。申请 iPhone 开发人员证书时您需要知道设备 ID。

申请 iPhone 开发人员证书并创建供给配置文件

如果您尚未执行这些操作，请进行注册，成为 Apple iPhone 开发人员中心站点 (<http://developer.apple.com/iphone/>) 的 iPhone 注册开发人员。

注：您不需要使用 iPhone SDK 或 XCode 开发 iPhone 的 AIR 应用程序。您需要成为一名注册的 iPhone 开发人员。此外，您还需要获取一个开发人员证书和一个供给配置文件。

- 1 使用 iPhone 开发人员帐户 ID 登录到 iPhone 开发人员中心。
- 2 在 iPhone 开发人员中心，申请并购买 iPhone 开发人员证书。
您将收到一封来自 Apple 的电子邮件，其中包含 iPhone 开发人员计划激活代码。
- 3 返回到 iPhone 开发人员中心。按照说明激活开发人员计划（并根据提示输入激活代码）。
- 4 当您的激活代码被接受时，请转到 iPhone 开发人员中心的“iPhone Developer Program Portal”部分。
- 5 创建一个证书签名请求文件。您将使用此文件来获取 iPhone 开发证书。有关说明，请参阅第 5 页的“生成证书签名请求”。
- 6 在下一步骤中，系统会要求您提供 iPhone 的设备 ID（或唯一设备 ID）。您可以从 iTunes 获取 UDID：
 - a 将您的 iPhone 与 USB 电缆连接。然后，在 iTunes 中，选择 iPhone 的“摘要”选项卡。
 - b 从 iPhone 开发人员中心站点下载供给配置文件后，将它添加到 iTunes。
 - c 然后单击显示的序列号。现在显示了 UDID。在 Mac 上单击 Command-C 或在 Windows 上单击 Ctrl-C，将 UDID 复制到剪贴板。
- 7 创建并安装供给配置文件和 iPhone 开发证书。
按照 iPhone 开发人员中心的说明执行操作。相关说明位于“iPhone Developer Program Portal”部分。请使用 Development Provisioning Assistant 来获取开发证书并创建供给配置文件。
忽略涉及 XCode 的步骤。不需要使用 XCode 开发使用 Flash Professional CS5 的 iPhone 应用程序。
- 8 在 iTunes 中，选择“文件”>“添加到资料库”。然后选择供给配置文件（文件名扩展名为 mobileprovision）。然后将 iPhone 与 iTunes 同步。
这样，您可以在 iPhone 上测试与此供给配置文件相关联的应用程序。
要验证特定的供给配置文件是否已添加到 iTunes，您可以尝试将其添加到资料库。如果 iTunes 要求您替换现有的供给配置文件，您可以按“取消”按钮。（已安装该配置文件。）此外，您还可以检查 iPhone 上安装的供给配置文件：
 - a 打开 iPhone 上的“设置”应用程序。
 - b 打开“通用”类别。
 - c 点击“配置文件”。“配置文件”页列出了安装的供给配置文件。
- 9 如果您尚未执行此操作，请下载 iPhone 开发证书文件（一个 .cer 文件）。
Development Provisioning Assistant 可能已经为您提供了一个用于下载此文件的链接。您还可以在 Apple iPhone 开发人员中心站点 (<http://developer.apple.com/iphone/>) 的“Provisioning Portal”的“Certificates”部分找到此文件。
- 10 接下来，将 iPhone 开发人员证书转换为 P12 文件。有关说明，请参阅第 5 页的“将开发人员证书转换为 P12 文件”。

现在，您可以创建一个简单的 Hello World 应用程序了。请参阅第 7 页的“使用 Flash Professional CS5 创建 Hello World iPhone 应用程序”。

生成证书签名请求

要获取开发人员证书，应生成证书签名请求文件，然后将其提交给 Apple iPhone 开发人员中心站点。

在 Mac OS 上生成证书签名请求

在 Mac OS 上，您可以使用钥匙串访问应用程序生成代码签名请求。钥匙串应用程序位于“应用程序”目录的“实用工具”子目录中。在钥匙串访问菜单中，选择“证书助手”>“从证书颁发机构请求证书”。

- 1 打开钥匙串访问。
- 2 在钥匙串访问菜单中，选择“首选项”。
- 3 在“首选项”对话框中，单击“证书”。然后将“联机证书状态协议”和“证书吊销列表”设置为“关闭”。关闭此对话框。
- 4 在钥匙串访问菜单中，选择“证书助手”>“从证书颁发机构请求证书”。
- 5 输入与您的 iPhone 开发人员帐户 ID 匹配的电子邮件地址和名称。不要输入 CA 电子邮件地址。选择“将请求保存到磁盘”，然后单击“继续”按钮。
- 6 保存此文件 (CertificateSigningRequest.certSigningRequest)。
- 7 将 CSR 文件上载到 Apple 的 iPhone 开发人员站点。(请参阅“申请 iPhone 开发人员证书并创建供给配置文件”。)

在 Windows 上生成证书签名请求

对于 Windows 开发人员，在 Mac 计算机上获取 iPhone 开发人员证书可能最容易。但是，也可以在 Windows 计算机上获取证书。首先，使用 OpenSSL 创建一个证书签名请求 (CSR 文件)：

- 1 将 OpenSSL 安装在 Windows 计算机上。(请访问 <http://www.openssl.org/related/binaries.html>。)

您可能还需要安装 Open SSL 下载页上列出的 Visual C++ 2008 Redistributable 文件。(不需要在您计算机上安装 Visual C++。)

- 2 打开 Windows 命令会话，使用命令 CD 到 OpenSSL bin 目录 (例如 c:\OpenSSL\bin\)

- 3 通过在命令行中输入以下内容创建私钥：

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

保存此私钥文件。稍后您将使用它。

当使用 OpenSSL 时，请不要忽略错误消息。即使 OpenSSL 生成错误消息，它也可能输出文件。不过，这些文件可能不可用。如果发现错误，请检查语法并再次运行命令。

- 4 通过在命令行中输入以下内容创建 CSR 文件：

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

使用您自己的电子邮件地址、CN (证书名称) 和 C (国家 / 地区) 替换现有值。

- 5 将 CSR 文件上载到 Apple 的 iPhone 开发人员站点。(请参阅“申请 iPhone 开发人员证书并创建供给配置文件”。)

将开发人员证书转换为 P12 文件

要使用 Flash Professional CS5 开发 iPhone 应用程序，您必须使用 P12 证书文件。基于从 Apple 收到的 Apple iPhone 开发人员证书文件生成此证书。

在 **Mac OS** 上将 **iPhone** 开发人员证书转换为 **P12** 文件

从 **Apple** 下载 **Apple iPhone** 证书后，将其导出为 **P12** 证书格式。在 **Mac® OS** 上执行以下操作：

- 1 打开钥匙串访问应用程序（位于应用程序 / 实用工具文件夹中）。
- 2 如果尚未将该证书添加到钥匙串，请选择“文件”>“导入”。然后浏览到您从 **Apple** 获取的证书文件（.cer 文件）。
- 3 在钥匙串访问中选择密钥类别。
- 4 选择与 **iPhone** 开发证书相关联的私钥。
该私钥由 **iPhone** 开发人员识别：与之配对的 <名字><姓氏> 公共证书。
- 5 选择“文件”>“导出项目”。
- 6 以个人信息交换 (.p12) 文件格式保存您的密钥。
- 7 系统将提示您创建一个尝试在其他计算机上导入此密钥时需要使用的密码。

在 **Windows** 上将 **Apple** 开发人员证书转换为 **P12** 文件

要使用 **Flash CS5** 开发 **iPhone** 应用程序，则必须使用 **P12** 证书文件。基于从 **Apple** 收到的 **Apple iPhone** 开发人员证书文件生成此证书。

- 1 将从 **Apple** 收到的开发人员证书文件转换成 **PEM** 证书文件。从 **OpenSSL bin** 目录运行以下命令行语句：

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

- 2 如果您使用的是 **Mac** 计算机上钥匙串中的私钥，则将其转换成 **PEM** 密钥：

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

- 3 现在，您可以基于密钥和 **PEM** 版本的 **iPhone** 开发人员证书生成有效的 **P12** 文件：

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

如果您使用的是 **Mac OS** 钥匙串中的密钥，则使用上一步骤中生成的 **PEM** 版本。否则，请使用以前生成的 **OpenSSL** 密钥（位于 **Windows** 上）。

管理证书、设备 ID、应用程序 ID 和供给配置文件

您可以在 **Apple iPhone** 开发人员中心站点 (<http://developer.apple.com/iphone/>) 管理证书、设备 ID、应用程序 ID 和供给配置文件。转到此站点的“**iPhone Developer Program Portal**”部分。

- 单击“**Certificates**”链接以管理开发证书。您可以创建证书、下载证书或吊销证书。要创建证书，必须首先创建证书签名请求。请参阅第 5 页的“**生成证书签名请求**”。
- 单击“**Devices**”链接以管理可以安装开发应用程序的设备列表。
- 单击“**App IDs**”链接以管理应用程序 ID。创建供给配置文件后，该供给配置文件会绑定到应用程序 ID。
- 单击“**Provisioning**”链接以管理供给配置文件。您也可以使用 **Development Provisioning Assistant** 来创建开发供给配置文件。
- 单击“**Distribution**”链接以将应用程序提交到应用程序存储或创建应用程序的临时版本。此部分包含一个指向 **iTunes Connect** 站点的链接，用于将应用程序提交到 **App Store**。

使用 Flash Professional CS5 创建 Hello World iPhone 应用程序

重要事项：创建该应用程序之前，请下载所需的开发人员应用程序和文件。请参见第 3 页的“[从 Adobe 获取开发人员工具](#)”和第 3 页的“[从 Apple 获取开发人员文件](#)”。

创建 Flash Professional CS5 项目

您可以直接在 Flash Professional CS5 中生成 iPhone 应用程序：

- 1 打开 Flash CS5。
- 2 选择“文件”>“新建”。
- 3 选择 iPhone OS。
- 4 单击“确定”按钮。

向该应用程序中添加内容

接下来，将文本“Hello world!”添加到该应用程序中：

- 1 选择“文本工具”，然后单击舞台。
- 2 在文本字段的“属性”设置中，选择“传统文本”（而不是 TLF 文本）。
这是一个基本应用程序，使用“传统文本”就可以了。要使用 TLF 文本，需要应用某些其他设置。请参见第 32 页的“[字体和文本输入](#)”。
- 3 在新文本字段中，键入“Hello World!”
- 4 使用“选取”工具选择该文本字段。
- 5 然后打开“属性”检查器，设置下列内容：
 - “字符”>“系列”：_sans
 - “字符”>“大小”：50
 - “位置”>“X”：20
 - “位置”>“Y”：20
- 6 保存该文件。
- 7 选择“控制”>“测试影片”>“在 AIR Debug Launcher (移动设备) 中”。

Flash Professional CS5 将编译 SWF 内容，并在 AIR Debug Launcher (ADL) 中显示该应用程序的一个版本。这样，您可以快速预览应用程序。

创建应用程序的图标图片和初始屏幕图片

所有 iPhone 应用程序都具有在 iTunes 应用程序的用户界面和 iPhone 屏幕上显示的图标。

- 1 在您的项目目录中创建一个目录并将其命名为图标。
- 2 在该图标目录中创建三个 PNG 文件。将它们分别命名为 Icon29.png、Icon57.png 和 Icon512.png。
- 3 编辑这些 PNG 文件为应用程序创建适当的图片。这些文件必须为 29×29 像素、57×57 像素和 512×512 像素。对于此测试，您可以仅使用纯色正方形作为图片。

在 iPhone 上加载应用程序时，所有 iPhone 应用程序均显示初始图像。您可以在 PNG 文件中定义初始图像：

- 1 在主开发目录中，创建名为 `Default.png` 的 PNG 文件。（不要将此文件放在图标子目录中。确保将此文件命名为 `Default.png`，使用大写 D。）
- 2 编辑该文件，使其宽 320 像素，高 480 像素。此刻，内容可能显示为纯白色矩形。（您稍后可以进行更改。）

注：将应用程序提交给 Apple 应用程序库时，使用像素为 512 的 JPG 版本文件（而不是 PNG 版本）。测试应用程序的开发版本时使用 PNG 版本。

有关这些图形的详细信息，请参见第 11 页的“[iPhone 图标和初始屏幕图像](#)”。

编辑应用程序设置

重要事项：如果您尚未执行该操作，请下载开发 iPhone 所需的开发人员应用程序和文件。请参见第 3 页的“[从 Apple 获取开发人员文件](#)”。

在 Flash Professional CS5“iPhone 设置”对话框中，定义 iPhone 应用程序的许多基本属性。

- 1 选择“文件”>“iPhone OS 设置”。
- 2 在“常规”选项卡中，设置下列内容：

- 输出文件：HelloWorld.ipa

这是要生成的 iPhone 安装程序文件的文件名。

- 应用程序名称：Hello World

这是在 iPhone 中的应用程序图标下显示的应用程序的名称。

- 版本：1.0

应用程序的版本。

- 高宽比：纵向

- 全屏：选中。

- 自动方向：取消选中。

- 呈现：CPU

其他选项（如“GPU”和“自动”）使用硬件加速进行呈现。对于旨在利用硬件加速、使用大量图形的应用程序（例如游戏），此功能可以帮助改进其性能。有关详细信息，请参阅第 30 页的“[硬件加速](#)”。

- 包括的文件：将初始屏幕图片文件 (`Default.png`) 添加到“包括的文件”列表。

注：对于此 Hello World 示例，请不要根据这些说明中提供的内容来修改设置。一些设置（例如版本设置）存在具体的限制。第 12 页的“[iPhone 应用程序设置](#)”介绍了这些限制。

- 3 在“部署”选项卡中，设置下列内容：

- 证书：浏览并选择基于您从 Apple 获取的开发人员证书的 .p12 证书。

此证书用于签署文件。您必须将 Apple iPhone 证书转换为 .p12 格式。有关详细信息，请参见第 3 页的“[从 Adobe 获取开发人员工具](#)”。

- 密码：输入证书密码。

- 供给配置文件：浏览并选择您从 Apple 获取的开发人员供给配置文件。请参见第 3 页的“[从 Apple 获取开发人员文件](#)”。

- 应用程序 ID：如果此字段是可选的，您可以输入与您提供给 Apple 的应用程序 ID 相匹配的应用程序 ID（例如 `com.example.as3.HelloWorld`）。

应用程序 ID 将唯一标识您的应用程序。

如果此字段不可选，则将供给配置文件与特定的应用程序 ID 绑定。应用程序 ID 将显示在此字段中。

有关指定应用程序 ID 的详细信息，请参阅第 13 页的“在 [Flash Professional CS5 中设置 iPhone 应用程序属性](#)”中的“部署”选项卡部分。

- 4 在“图标”选项卡中，单击“图标”列表中的图标 29 x 29。然后，指定您先前创建的像素为 29 x 29 的 PNG 文件的位置（请参见第 7 页的“[创建应用程序的图标图片和初始屏幕图片](#)”）。然后为像素为 57 x 57 的图标和像素为 512 x 512 的图标指定 PNG 文件。
- 5 单击“确定”按钮。
- 6 保存该文件。

有关应用程序设置的详细信息，请参见第 12 页的“[iPhone 应用程序设置](#)”。

编译 IPA 文件

现在您可以编译 IPA 安装程序文件：

- 1 选择“文件”>“发布”。
- 2 在“iPhone 设置”对话框中，单击“确定”按钮。

iPhone 的包装程序在项目目录中生成 iPhone 应用程序安装程序文件 HelloWorld.ipa 文件。编译 IPA 文件可能需要几分钟的时间。

在 iPhone 上安装应用程序

安装 iPhone 应用程序以在 iPhone 上进行测试：

- 1 打开 iTunes 应用程序。
- 2 如果您尚未执行此操作，请将该应用程序的供给配置文件添加到 iTunes。在 iTunes 中，选择“文件”>“将文件添加到库”。然后选择供给配置文件（其文件类型为 mobileprovision）。

此刻，要在开发人员 iPhone 上测试该应用程序，请使用开发供给配置文件。

稍后，将应用程序分发给 iTunes Store 时，请使用分发配置文件。要临时分发应用程序（不通过 iTunes Store 分发给多个设备），请使用临时供给配置文件。

有关供给配置文件的详细信息，请参见第 3 页的“[从 Apple 获取开发人员文件](#)”。

- 3 如果已安装相同版本的应用程序，iTunes 的某些版本不会替换该应用程序。在这种情况下，从您的设备和 iTunes 中的应用程序列表中删除该应用程序。
- 4 双击您的应用程序的 IPA 文件。它应显示在应用程序列表中。
- 5 将您的 iPhone 连接到计算机上的 USB 端口。
- 6 在 iTunes 中，检查“应用程序”选项卡中是否存在该设备，并确保在要安装的应用程序的列表中选中了该应用程序。
- 7 选择左侧 iTunes 应用程序列表中的设备。然后单击“同步”按钮。完成同步后，Hello World 应用程序会显示在您的 iPhone 上。

如果未安装新版本，请将其从您的 iPhone 以及 iTunes 中的应用程序列表中删除，然后重新执行此过程。这可能是因为目前安装的版本使用的是相同的应用程序 ID 和版本。

如果在尝试安装应用程序时 iTunes 显示错误，请参阅第 19 页的“[安装 iPhone 应用程序](#)”中的“应用程序安装问题疑难解答”。

编辑初始屏幕图形

在编译应用程序之前，您创建了一个 `Default.png` 文件（请参见第 7 页的“[创建应用程序的图标图片和初始屏幕图片](#)”）。当加载应用程序时，此 PNG 文件充当启动图像。在 iPhone 上测试应用程序时，您可能已注意到启动时会出现空白屏幕。

您应更改此图像，使之与您的应用程序（“Hello World!”）的启动屏幕相匹配：

- 1 打开设备上的应用程序。当第一个“Hello World”文本出现时，按住“主屏幕”按钮（位于屏幕下方）。在按住“主屏幕”按钮的同时，按下“睡眠/唤醒”按钮（位于 iPhone 的顶部）。这会捕获一张屏幕快照并将其发送到摄像头卷。
- 2 通过从 iPhoto 或其他照片传输应用程序传输照片将此图像传输到您的开发计算机。（在 Mac OS 上，您也可以使用图像捕捉应用程序。）

还可以通过电子邮件将照片发送到您的开发计算机：

- 打开 Photos 应用程序。
 - 打开摄像头卷。
 - 打开您捕获的屏幕快照图像。
 - 点击此图像，然后点击左下角的“转发”（箭头）按钮。然后单击“用电子邮件发送照片”按钮并将图像发送给自己。
- 3 使用 PNG 版本的屏幕捕获图像替换 `Default.png` 文件（位于您的开发目录中）。
 - 4 重新编译该应用程序（请参见第 9 页的“[编译 IPA 文件](#)”）并将其重新安装在您的 iPhone 中。

该应用程序现在使用的是其加载的新启动屏幕。

注：您可以为 `Default.png` 文件创建任何图片，只要尺寸正确（320 × 480 像素）即可。但是，通常最好将 `Default.png` 图像与应用程序的初始状态相匹配。

第 2 章：编译和调试 iPhone 应用程序

您可以使用 **Packager for iPhone** 编译 iPhone 应用程序。Packager for iPhone 随附在 Flash Professional CS5 中。

您可以在开发计算机上调试应用程序，也可以在 iPhone 上安装调试版本并在 Flash Professional CS5 中接收 `trace()` 输出。

有关如何从头至尾构建 iPhone 应用程序的教程，请参见第 7 页的“[使用 Flash Professional CS5 创建 Hello World iPhone 应用程序](#)”。

iPhone 图标和初始屏幕图像

所有 iPhone 应用程序都具有在 iTunes 应用程序的用户界面和 iPhone 上显示的图标。

iPhone 应用程序图标

为 iPhone 应用程序定义下列图标：

- 29 x 29 像素图标 — iPhone 和 iPod 触摸屏上的 Spotlight 搜索结果使用此图标。
- 48 x 48 像素图标 — iPad 上的 Spotlight 搜索结果使用此图标。
- 57 x 57 像素图标 — iPhone 和 iPod Touch 主屏幕显示此图标。
- 72 x 72 像素图标（可选） — iPad 主屏幕显示此图标。
- 512 x 512 像素图标 — iTunes 显示此图标。当您最终应用程序提交给 Apple 应用程序库时，512 像素 PNG 文件仅用于测试应用程序的开发版本，须单独以 JPG 文件格式提交 512 图像。它不包含在 IPA 中。

在 Flash Professional CS5 中，在“iPhone 设置”对话框的“图标”选项卡中添加这些图标。请参阅第 13 页的“[在 Flash Professional CS5 中设置 iPhone 应用程序属性](#)”。

您还可以将图标的位置添加到应用程序描述符文件：

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

iPhone 为图标添加了眩光效果。不需要将其放在源图像中。要删除此默认眩光效果，请将以下内容添加到应用程序描述符文件中的 `InfoAdditions` 元素：

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

请参见第 14 页的“[在应用程序描述符文件中设置 iPhone 应用程序属性](#)”。

初始屏幕图片 (Default.png)

在 iPhone 上加载应用程序时，所有 iPhone 应用程序均显示初始图像。可以在名为 `Default.png` 的 PNG 文件中定义初始图像。在主开发目录中，创建名为 `Default.png` 的 PNG 文件。（不要将此文件放置在子目录中。确保将此文件命名为 `Default.png`，使用大写 D。）

Default.png 文件为 320 像素宽 × 480 像素高，不管应用程序的初始方向如何或是否为全屏。

如果应用程序的初始方向为横向，则使用的尺寸与纵向应用程序尺寸相同：320 像素宽 × 480 像素高。然而，要将 PNG 文件中的插图逆时针旋转 90°。在横向模式中，PNG 图片的左侧对应于 iPhone 屏幕的顶部。（有关设置初始应用程序方向的信息，请参阅第 12 页的“[iPhone 应用程序设置](#)”。）

对于非全屏的应用程序，将忽略默认图像作品顶部的 20 个像素。iPhone 将其状态栏显示在默认图像顶部的 20 像素宽的矩形之上。在横向应用程序中，此区域对应于 Default.png 文件的左侧 20 像素宽的矩形，而在横向模式中，显示在顶部。在纵向应用程序中，此区域是 Default.png 文件的顶部 20 像素宽的矩形。

对于大多数应用程序，Default.png 图像应该与应用程序的启动屏幕相匹配。获取应用程序启动屏幕的屏幕快照：

- 1 在 iPhone 上打开您的应用程序。用户界面的第一个屏幕出现后，按住“主屏幕”按钮（位于屏幕下方）。在按住“主屏幕”按钮的同时，按下“睡眠 / 唤醒”按钮（位于设备顶部）。这会捕获一张屏幕快照并将其发送到摄像头卷。
- 2 通过从 iPhoto 或其他照片传输应用程序传输照片将此图像传输到您的开发计算机。（在 Mac OS 上，您也可以使用图像捕捉应用程序。）

还可以通过电子邮件将照片发送到您的开发计算机：

- 打开 Photos 应用程序。
- 打开摄像头卷。
- 打开您捕获的屏幕快照图像。
- 点击此图像，然后点击左下角的“转发”（箭头）按钮。然后单击“用电子邮件发送照片”按钮并将图像发送给自己。

注：您可以根据需要为 Default.png 文件创建任何图片，只要尺寸正确。但是，通常最好将 Default.png 图像与应用程序的初始状态相匹配。

如果您的应用程序已本地化为多种语言，请不要在 Default.png 图像中包括文本。Default.png 是静态的，而且文本与其他语言不匹配。

在 Flash Professional CS5 中，请确保将 Default.png 文件添加到“iPhone 设置”对话框的“包括的文件”列表中。请参阅第 13 页的“[在 Flash Professional CS5 中设置 iPhone 应用程序属性](#)”。

当在命令行中使用 PFI 应用程序编译时，请确保引用包含的资源列表中的此文件。请参阅第 18 页的“[从命令行创建 iPhone 应用程序安装程序文件](#)”。

iPhone 应用程序设置

应用程序设置包括：

- 应用程序名称
- IPA 文件名
- 应用程序版本
- 应用程序的初始屏幕方向，以及当 iPhone 旋转时屏幕方向是否自动旋转
- 初始视图是否为全屏
- 有关应用程序的图标的信息
- 有关硬件加速的信息

您可以在 Flash Professional CS5 中编辑应用程序设置。

您还可以在应用程序描述符文件中编辑这些设置。应用程序描述符文件是一个包含应用程序的设置的 XML 文件。

在 Flash Professional CS5 中设置 iPhone 应用程序属性

使用 Flash Professional CS5“iPhone 设置”对话框，您可以定义 iPhone 应用程序的许多基本属性。

打开“iPhone 设置”对话框：

❖ 选择“文件”>“iPhone 设置”。

“常规”选项卡

“常规”选项卡包含下列与 iPhone 有关的设置：

- 输出文件 — 在 iPhone 中的应用程序图标下显示的应用程序的名称。输出文件名中不要包含加号 (+) 字符。
- 应用程序名称 — 在 iPhone 中的应用程序图标下显示的应用程序名称。应用程序名称中不要包含加号 (+) 字符。
- 版本 — 帮助用户确定安装哪个应用程序版本。此版本用作 iPhone 应用程序的 CFBundleVersion。必须使用类似于 nnnnn[.nn[.nn]] 的格式，其中 n 为 0-9 内的一个数字，中括号表示可选组件，如 1、1.0 或 1.0.1。iPhone 版本只能包含数字和小数点。iPhone 版本最多可以包含两个小数点。
- 高宽比 — 应用程序的初始高宽比（纵向或横向）。
- 全屏 — 应用程序是否使用全屏或它是否显示 iPhone 状态栏。
- 自动方向 — 选择此应用程序，以使应用程序的显示内容随 iPhone 的重新取向而重新取向。

在使用“自动方向”时，为获得最佳效果，应添加 **ActionScript** 代码以按照如下所示设置舞台的 align 属性：

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

- 呈现 — 显示对象在 iPhone 上的呈现方式：
 - CPU — 应用程序使用 CPU 呈现所有显示对象。没有使用硬件加速。
 - GPU — 应用程序使用 iPhone GPU 合成位图。
 - 自动 — 此功能尚未实现。

有关详细信息，请参阅第 30 页的“硬件加速”。

- 包括的文件 — 将所有文件和目录添加到 iPhone 应用程序中的软件包。默认情况下包含主 SWF 文件和应用程序描述符文件。将任何其他所需资源添加到“包括的文件”列表中。确保将初始屏幕图片文件 (Default.png) 添加到“包括的文件”列表中。

“部署”选项卡

“部署”选项卡包括应用程序的签名和编译设置：

- iPhone 数字签名 — 为证书指定 P12 证书文件和密码。您必须将 Apple iPhone 证书转换为 .p12 格式。有关详细信息，请参见第 3 页的“从 Apple 获取开发人员文件”。
- 供给文件 — 指向您从 Apple 获取的此应用程序的供给文件。有关详细信息，请参见第 3 页的“从 Apple 获取开发人员文件”。
- 应用程序 ID — 应用程序 ID 将唯一标识您的应用程序。如果供给配置文件与特定应用程序 ID 关联，则 Flash Professional CS5 将设置此字段，并且您无法编辑该字段。否则，供给配置文件会允许多个（通配符）应用程序 ID。提供的应用程序 ID 应与您向 Apple 提供的应用程序 ID 通配符模式匹配：
 - 如果您的 Apple 应用程序 ID 是 com.myDomain.*，则“iPhone 设置”对话框中的应用程序 ID 必须以 com.myDomain. 开头（例如 com.myDomain.myApp 或 com.myDomain.app22）。
 - 如果您的 Apple 应用程序 ID 是 *，则“iPhone 设置”对话框中的应用程序 ID 可以是任何有效字符串。

您可以在 iPhone 开发人员中心 (<http://developer.apple.com/iphone>) 查找与供给配置文件关联的 Apple 应用程序 ID（或通配符应用程序 ID 模式）。转到“iPhone Developer Program Portal”，然后转到“Provisioning”部分。

重要事项: 忽略 Apple 应用程序 ID 前面的字符。Apple 将此字符串称为绑定种子 ID。例如, 如果 Apple 显示您的应用程序 ID 为 96LPVWEASL.com.example.bob.myApp, 请忽略 96LPVWEASL, 并将 com.example.bob.myApp 用作应用程序 ID。如果 Apple 显示您的应用程序 ID 为 5RM86Z4DJM.*, 请忽略 5RM86Z4DJM, 这是一个通配符应用程序 ID。

- iPhone 部署类型:
 - 用于设备测试的快速发布 — 选择此选项可快速编译用于在开发人员 iPhone 上进行测试的应用程序版本。
 - 用于设备调试的快速发布 — 选择此选项可快速编译用于在开发人员 iPhone 上测试的应用程序的调试版本。借助此选项, Flash Professional CS5 调试器可以从 iPhone 应用程序接收 trace() 输出。(请参见第 20 页的“[调试 iPhone 应用程序](#)”。)
 - 部署 - 临时 — 选择此选项可创建用于临时部署的应用程序。请参见 Apple iPhone 开发人员中心
 - 部署 - Apple 应用程序库 — 选择此选项可创建用于部署到 Apple 应用程序库的 IPA 文件的最终版本。

“图标”选项卡

在“图标”选项卡上, 指定 29 x 29 像素图标图像、48 x 48 像素图标图像、57 x 57 像素图标图像、72 x 72 像素图标图像和 512 x 512 像素图标图像的位置。请参见第 11 页的“[iPhone 图标和初始屏幕图像](#)”。

注: Flash Professional CS5 附带的 Packager for iPhone Preview 的版本中不包含 48 x 48 像素和 72 x 72 像素的选项。在 Flash Professional CS5 中, 选择“帮助”>“更新”以添加这些选项。

在应用程序描述符文件中设置 iPhone 应用程序属性

应用程序描述符文件是一个包含整个应用程序的属性 (例如其名称、版本、版权及其他设置) 的 XML 文件。

Flash Professional CS5 基于“iPhone 设置”对话框中的设置生成应用程序描述符文件。但是, 您也可以在文本编辑器中编辑应用程序描述符文件。Flash Professional 通过将“- app.xml”添加到项目名称来命名应用程序描述符文件。例如, HelloWorld 项目的应用程序描述符文件名为 HelloWorld-app.xml。如果要定义 Flash Professional CS5“iPhone 设置”对话框中不支持的设置, 则编辑应用程序描述符文件。例如, 您可以定义 InfoAdditions 元素来定义应用程序的 info.Plist 设置。

重要事项: 请不要在 Flash Professional CS5 对话框处于打开状态时编辑应用程序描述符文件。在打开“iPhone 设置”对话框之前, 请保存对该应用程序描述符文件的更改。

下面是一个应用程序描述符文件的示例:

```
<?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>
```

以下是有关此应用程序描述符文件中的设置的详细信息：

- 在 `<application>` 元素中，构建 iPhone 应用程序需要使用 AIR 2.0 版命名空间：

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

- `<id>` 元素：

`<id>com.example.as3.HelloWorld</id>` 应用程序 ID 将唯一标识您的应用程序。建议采用的形式为以点分隔的反向 DNS 样式的字符串，如 "com.company.AppName"。编译器将此值用作 iPhone 应用程序的绑定 ID。

如果供给配置文件与特定应用程序 ID 关联，请在此元素中使用该应用程序 ID。请忽略 Apple 分配给 Apple 应用程序 ID 的开头的字符（称为绑定种子 ID）。例如，如果供给配置文件的应用程序 ID 是

96LPVWEASL.com.example.bob.myApp，请将 com.example.bob.myApp 用作应用程序描述符文件中的应用程序 ID。

如果供给配置文件允许多个（通配符）应用程序 ID，则其应用程序 ID 以星号结尾，例如 5RM86Z4DJM.*。提供的应用程序 ID 应与您向 Apple 提供的应用程序 ID 通配符模式匹配：

- 如果您的 Apple 应用程序 ID 是 com.myDomain.*，则应用程序描述符文件中的应用程序 ID 必须以 com.myDomain 开头。您可以指定一个应用程序 ID，例如 com.myDomain.myApp 或 com.myDomain.app22。
- 如果您的 Apple 应用程序 ID 是 *，则应用程序描述符文件中的应用程序 ID 可以是任何有效字符串。

您可以在 iPhone 开发人员中心 (<http://developer.apple.com/iphone>) 查找与供给配置文件关联的 Apple 应用程序 ID（或通配符应用程序 ID 模式）。转到“iPhone Developer Program Portal”，然后转到“Provisioning”部分。

重要事项：忽略 Apple 应用程序 ID 前面的字符。Apple 将此字符串称为绑定种子 ID。例如，如果 Apple 显示您的应用程序 ID 为 5RM86Z4DJM.*，请忽略 5RM86Z4DJM，这是一个通配符应用程序 ID。如果 Apple 显示您的应用程序 ID 为 96LPVWEASL.com.example.bob.myApp，请忽略 96LPVWEASL，并将 com.example.bob.myApp 用作应用程序 ID。

- `<filename>` 元素:

`<filename>HelloWorld</filename>` iPhone 安装程序文件所使用的名称。文件名中不要包含加号 (+) 字符。

- `<name>` 元素:

`<name>Hello World</name>` iTunes 应用程序和 iPhone 中显示的应用程序名称。名称中不要包含加号 (+) 字符。

- `<version>` 元素:

`<version>1.0</version>` 可帮助用户确定安装哪个版本的应用程序。此版本用作 iPhone 应用程序的 `CFBundleVersion`。必须使用类似于 `nnnnn[.nn[.nn]]` 的格式，其中 `n` 为 0-9 内的一个数字，中括号表示可选组件，如 1、1.0 或 1.0.1。iPhone 版本只能包含数字和小数点。iPhone 版本最多可以包含两个小数点。

- `<initialWindow>` 元素包含下列子元素，以指定应用程序的初始外观的属性:

`<content>HelloWorld.swf</content>` 标识要编译为 iPhone 应用程序的根 SWF 文件。

`<visible>true</visible>` 这是必需设置。

`<fullScreen>true</fullScreen>` 指定应用程序使用 iPhone 的整个屏幕。

`<aspectRatio>portrait</aspectRatio>` 指定应用程序的初始高宽比使用纵向模式（而不是横向）。请注意，不管此设置如何，用于定义应用程序的初始窗口的 `Default.png` 文件应宽 320 像素，高 480 像素。（请参见第 11 页的“[iPhone 图标和初始屏幕图像](#)”。）

`<autoOrients>true</autoOrients>`（可选）指定应用程序中的内容的方向是否随设备自身更改物理方向而自动重新取向。默认值为 `true`。您可以通过调用由 `Stage` 对象调度的 `orientationChanging` 事件的 `preventDefault()` 方法取消自动取向。有关详细信息，请参阅[设置和检测屏幕方向](#)。

在使用“自动方向”时，为获得最佳效果，应按照如下所示设置舞台的 `align` 属性:

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

`<renderMode>gpu</renderMode>`（可选）应用程序使用的呈现模式。存在三种可能的设置:

- `cpu` — 应用程序使用 CPU 呈现所有显示对象。没有使用硬件加速。
- `gpu` — 应用程序使用 iPhone GPU 合成位图。
- `auto` — 此功能尚未实现。

有关详细信息，请参阅第 30 页的“[硬件加速](#)”。

- `<profiles>` 元素:

`<profiles>mobileDevice</profiles>` 将应用程序限制为编译到移动设备配置文件。此配置文件目前仅支持 iPhone 应用程序。支持下列三种配置文件:

- `desktop` — 桌面 AIR 应用程序。
- `extendedDesktop` — 支持本机进程 API 的桌面 AIR 应用程序。
- `mobileDevice` — 适用于移动设备的 AIR 应用程序。目前，iPhone 是唯一受支持的移动设备。

将应用程序限制为特定配置文件可以阻止将其编译到其他配置文件中。如果不指定配置文件，则可以使用任何配置文件编译应用程序。您可以通过在 `<profiles>` 元素中列出每个配置文件并用空格隔开来指定多个配置文件。

请确保受支持的配置文件中包含 `mobileDevice`（或将 `<profiles>` 元素保留为空）。

- `<icon>` 元素包含下列子元素，以指定应用程序使用的图标:

`<image29x29>icons/icon29.png</image29x29>` 这是 Spotlight 搜索结果中使用的图像。

`<image48x48>icons/icon48.png</image48x48>` 这是 iPad 上 Spotlight 搜索结果中使用的图像。

`<image57x57>icons/icon57.png</image57x57>` 这是 iPhone 和 iPod Touch 主屏幕上使用的图像。

`<image72x72>icons/icon72.png</image72x72>` 这是 iPad 主屏幕上使用的图像。

`<image512x512>icons/icon512.png</image512x512>` 这是 iTunes 应用程序中使用的图像。

Packager for iPhone 工具使用在应用程序描述符文件中引用的 29、57 和 512 图标。该工具将它们分别复制到名为 `Icon-Small.png`、`Icon.png` 和 `iTunesArtwork` 的文件中。要避免进行复制，您可以直接打包这些文件。通过将它们放到包含应用程序描述符文件的目录中直接进行打包，并列出正确的名称和路径。

512 图像仅用于内部测试。将应用程序提交给 Apple 时，应单独提交 512 图像。它不包含在 IPA 中。请指定该图像，以便您可以确保在提交 512 图像之前它在 iTunes 中显示正常。

- `<iPhone>` 元素包含下列子元素，以指定 iPhone 特定的设置：

`<InfoAdditions></InfoAdditions>` 包含的子元素可指定用作应用程序的 `Info.plist` 设置的键 - 值对：

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

在此示例中，这些值设置应用程序的状态栏样式，并声明应用程序不要求永久的 Wi-Fi 访问。

`InfoAdditions` 设置括在 CDATA 标记中。

对于 iPad 支持，请包括 `UIDeviceFamily` 的键值设置。`UIDeviceFamily` 设置是一个字符串数组。每个字符串都定义受支持的设备。`<string>1</string>` 设置定义对 iPhone 和 iPod Touch 的支持。`<string>2</string>` 设置定义对 iPad 的支持。

`<string>3</string>` 设置定义对 tvOS 的支持。如果仅指定其中一个字符串，则仅支持该设备系列。例如，下面的设置限制对 iPad 的支持：

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

下面的设置支持两个设备系列（iPhone/iPod Touch 和 iPad）：

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

有关其他 `Info.plist` 设置的信息，请参见 Apple 开发人员文档。

编译 iPhone 应用程序安装程序 (IPA) 文件

使用 Packager for iPhone 将 ActionScript 3.0 应用程序编译为 IPA 安装程序文件。

使用 Flash Professional CS5 中随附的 Packager for iPhone 创建 iPhone 应用程序安装程序文件

要使用 Flash Professional CS5 附带的 Packager for iPhone：

- 1 选择“文件”>“发布”。
- 2 在“iPhone 设置”对话框中，确保为所有设置提供了相应的值。确保在“部署”选项卡中选择了正确的选项。请参阅第 13 页的“在 Flash Professional CS5 中设置 iPhone 应用程序属性”。

3 单击“发布”按钮。

Packager for iPhone 会生成 iPhone 应用程序安装程序 (IPA) 文件。编译 IPA 文件可能需要几分钟的时间。

您还可以从命令行运行 Packager for iPhone。请参阅第 18 页的“[从命令行创建 iPhone 应用程序安装程序文件](#)”。

从命令行创建 iPhone 应用程序安装程序文件

您可以从命令行运行 Packager for iPhone。Packager for iPhone 将 SWF 文件字节代码及其他源文件转换成本机 iPhone 应用程序。

1 打开命令外壳或终端并导航到您的 iPhone 应用程序的项目文件夹。

2 然后，使用 pfi 工具创建 IPA 文件，语法如下：

```
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
```

更改引用 pfi 以包括 pfi 应用程序的完整路径。pfi 应用程序安装在 Flash Professional CS5 安装目录的 pfi/bin 子目录中选择对应于您要创建的 iPhone 应用程序类型的 -target 选项：

- -target ipa-test — 选择此选项可快速编译要在开发人员 iPhone 上进行测试的应用程序版本。
- -target ipa-debug — 选择此选项可编译要在开发人员 iPhone 上测试的应用程序调试版本。通过此选项，您可以使用调试会话从 iPhone 应用程序接收 trace() 输出。

您可以包含以下 -connect 选项 (CONNECT_OPTIONS) 之一，以指定运行该调试器的开发计算机的 IP 地址：

- -connect — 应用程序将尝试连接到用于编译该应用程序的开发计算机上的调试会话。
- -connect IP_ADDRESS — 应用程序将尝试连接到具有指定 IP 地址的计算机上的调试会话。例如：

```
-target ipa-debug -connect 192.0.32.10
```
- -connect HOST_NAME — 应用程序将尝试连接到具有指定主机名称的计算机上的调试会话。例如：

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

注：在 Flash Professional CS5 附带的 Packager for iPhone Preview 中不包含 -connect 选项。通过在 Flash Professional CS5 中选择“帮助”>“更新”来更新 Packager for iPhone。

-connect 选项是可选项。如果不指定该选项，则生成的调试应用程序不会尝试连接到托管调试器。

如果调试连接尝试失败，应用程序会显示一个对话框，要求用户输入调试主机的 IP 地址。如果设备未连接到 wifi，连接尝试可能失败。如果设备已连接，但不在调试主机的防火墙之后，也可能失败。

有关详细信息，请参阅第 20 页的“[调试 iPhone 应用程序](#)”。

还可以包括 -renderingdiagnostics 选项以启用 GPU 呈现诊断功能。有关详细信息，请参阅第 20 页的“[调试 iPhone 应用程序](#)”中的“使用 GPU 呈现诊断进行调试”。

- -target ipa-ad-hoc — 选择此选项可创建用于临时部署的应用程序。请参见 [Apple iPhone 开发人员中心](#)
- -target ipa-app-store — 选择此选项可创建用于部署到 Apple 应用程序库的 IPA 文件的最终版本。

用应用程序的供给配置文件的路径替换 PROFILE_PATH。有关供给配置文件的详细信息，请参见第 3 页的“[从 Apple 获取开发人员文件](#)”。

替换 SIGNING_OPTIONS 以引用 iPhone 开发人员证书和密码。应使用以下语法：

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

使用 P12 证书文件的路径替换 P12_FILE_PATH。使用证书密码替换 PASSWORD。（请参见以下示例。）有关 P12 证书文件的详细信息，请参阅第 5 页的“[将开发人员证书转换为 P12 文件](#)”。

替换 APP_DESCRIPTOR 以引用应用程序描述符文件。

替换 SOURCE_FILES 以引用项目（后面跟有任何其他要包含的资源）的主 SWF 文件。包括在 Flash CS5 中的应用程序设置对话框中或在自定义应用程序描述符文件中定义的所有图标文件的路径。此外，还要添加初始屏幕图片文件 Default.png。

请看以下示例：

```
pfi -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"
```

此示例编译 HelloWorld.ipa 文件时使用了下列内容：

- 使用证书密码 dfb7VKL19 的特定 PKCS#12 证书
- HelloWorld-app.xml 应用程序描述符文件
- 源 HelloWorld.swf 文件
- 特定的 Default.png 和图标文件

pfi 应用程序根据应用程序描述符文件、SWF 文件及其他资源将应用程序编译为 IPA 文件。

在 Mac OS 上，您可以使用在钥匙串中存储的证书，方法是将下列选项添加到 pfi 命令：

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

使用要使用的证书的别名替换 ALIAS_NAME。当指向在 Mac 钥匙串中存储的证书时，应指定别名而不是指向证书文件位置。

安装 iPhone 应用程序

要在 iPhone 上安装开发应用程序，应将供给配置文件添加到 iPhone，然后在 iPhone 上安装应用程序。

将供给配置文件添加到 iPhone

要将供给配置文件添加到 iPhone：

- 1 在 iTunes 中，选择“文件”>“添加到资料库”。然后，选择供给配置文件（文件名扩展名为 mobileprovision）。

确保将 iPhone 添加到供给配置文件。您可以在 Apple iPhone 开发人员中心站点 (<http://developer.apple.com/iphone/>) 管理供给配置文件。转到此站点的“iPhone Developer Program Portal”部分。单击“Devices”链接以管理可以安装开发应用程序的设备列表。单击“Provisioning”链接以管理供给配置文件。

- 2 将 iPhone 连接到您的计算机并使它们同步。

有关获取供给配置文件的信息，请参阅第 3 页的“从 Apple 获取开发人员文件”。

安装应用程序

按照安装任何其他 IPA 文件的方式来安装开发应用程序：

- 1 如果您以前安装过此应用程序的某个版本，则从您的设备和从 iTunes 中的应用程序列表中删除该应用程序。

- 2 以下列任何方法在 iTunes 中添加应用程序：

- 在 iTunes 中的“文件”菜单上，选择“添加到资料库”命令。然后选择 IPA 文件并单击“打开”按钮。
- 双击 IPA 文件。
- 将 IPA 文件拖到 iTunes 资料库中。

- 3 将您的 iPhone 连接到计算机上的 USB 端口。

- 4 在 iTunes 中，检查“应用程序”选项卡中是否存在该设备，并确保在要安装的应用程序的列表中选中了该应用程序。

5 同步您的 iPhone。

应用程序安装问题疑难解答

如果尝试安装应用程序时 iTunes 显示错误，请检查下列各项：

- 确保将设备 ID 添加到供给配置文件。
- 为了确保已安装供给配置文件，可以将其拖动到 iTunes 中或使用“文件”>“添加到资料库”命令。

此外，查看应用程序的应用程序 ID 与 Apple 应用程序 ID 是否匹配：

- 如果您的 Apple 应用程序 ID 是 com.myDomain.*，则应用程序描述符文件中的应用程序 ID 或“iPhone 设置”对话框中的应用程序 ID 必须以 com.myDomain 开头（例如 com.myDomain.anythinghere）。
- 如果您的 Apple 应用程序 ID 是 com.myDomain.myApp，则应用程序描述符文件或 Flash Profession CS5 用户界面中的应用程序 ID 必须是 com.myDomain.myApp。
- 如果您的 Apple 应用程序 ID 是 *，则应用程序描述符文件或 Flash Profession CS5 用户界面中的应用程序 ID 可以是任何内容。

应在 Flash Professional CS5 的“iPhone 设置”对话框中或在应用程序描述符文件中设置应用程序的应用程序 ID。

调试 iPhone 应用程序

您可以在开发计算机中调试应用程序，同时使应用程序在 ADL 中运行。您也可以直接在 iPhone 上调试应用程序。

iPhone 中某个不支持的 AIR 功能在使用 ADL 测试应用程序（在开发计算机上）时仍然可用。在桌面上测试内容时，请注意这些区别。有关详细信息，请参阅第 23 页的“[移动设备不支持 ActionScript 3.0 API](#)”。

在开发计算机上调试应用程序

要在开发计算机上使用 Flash Professional CS5 调试应用程序：

- ❖ 选择“调试”>“调试影片”>“在 AIR Debug Launcher (移动设备) 中”。

您也可以通过从命令行调用 ADL 来调试应用程序。以下是对应的语法：

```
adl -profile mobileDevice appDescriptorFile
```

使用应用程序描述符文件的路径替换 appDescriptorFile。

请确保包括 -profile mobileDevice 选项。

在 iPhone 上调试应用程序

在 iPhone 上调试应用程序：

1 使用调试支持编译应用程序：

- 在 Flash Professional CS5 中，使用“用于设备调试的快速发布”设置进行编译。（请参阅第 17 页的“[使用 Flash Professional CS5 中随附的 Packager for iPhone 创建 iPhone 应用程序安装程序文件](#)”。）
- 使用 PFI 命令行应用程序以及 target ipa-debug 选项编译应用程序。（请参阅第 18 页的“[从命令行创建 iPhone 应用程序安装程序文件](#)”。）

2 在 iPhone 上安装应用程序。

3 在 iPhone 上，打开 Wi-Fi 并连接到开发计算机所在的网络。

4 在您的开发计算机上启动调试会话。在 Flash Professional CS5 中，选择“调试”>“开始远程调试会话”>“ActionScript 3.0”。

5 在 iPhone 上运行应用程序。

应用程序的调试版本将提示您输入开发人员计算机的 IP 地址。输入 IP 地址并点击“确定”按钮。获取开发计算机的 IP 地址。

- 在 Mac OS 中, 在“Apple”菜单中选择“系统首选项”。在“系统首选项”窗口中, 单击“网络”图标。“网络首选项”窗口列出了 IP 地址。
- 在 Windows 中, 启动命令行会话并运行 `ipconfig` 命令。

调试会话将显示从应用程序输出的任何 `trace()`。

调试安装在 iPhone 上的应用程序时, Flash Professional CS5 支持所有调试功能, 包括断点控制、跟踪代码和变量监视。

使用 GPU 呈现诊断进行调试

使用 GPU 呈现诊断功能可以查看应用程序使用硬件加速 (针对使用 GPU 呈现模式的应用程序) 的情况。要使用此功能, 请在命令行上使用 PFI 工具编译应用程序, 并包含 `-renderingdiagnostics` 选项:

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

`-renderingdiagnostics` 标志必须紧跟在 `-package` 标志后面。

GPU 呈现诊断功能为所有显示对象显示彩色矩形:

- 蓝色 — 显示对象不是位图, 也未缓存为位图, 并且正在呈现该对象。

如果对于没有发生改变的显示对象多次显示蓝色, 可能是因为该对象与移动显示对象交叠。例如, 该显示对象可能是移动显示对象的一个背景。请考虑将该显示对象缓存为位图。

如果针对您认为已经缓存的某个对象显示蓝色, 可能是由于该对象使用了 GPU 无法应用的效果。这些效果包括某些混合模式、颜色转换、`scrollRect` 属性和蒙版。

如果上载到 GPU 的显示对象超过内存限制, 应用程序也会显示蓝色。

应用程序会记录针对每个蓝色矩形的消息。应用程序会将这些消息随其他 `trace()` 和调试输出消息一起输出。

- 绿色 — 显示对象为位图或已缓存为位图, 正在向 GPU 首次上载该对象。

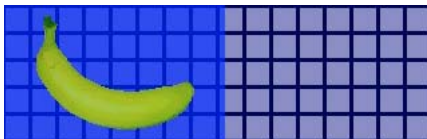
如果针对某个显示对象多次显示绿色, 则应用程序中的代码正在重新创建该显示对象。例如, 如果时间轴返回到创建该显示对象的帧, 就会出现这种情况。请考虑修改内容, 以防止重新创建同一对象。

- 红色 — 显示对象为位图或已缓存为位图, 并且正在向 GPU 重新上载该对象。

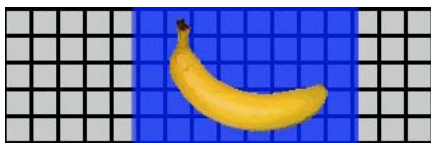
每次显示对象发生变化时, 如果要求应用程序重新呈现其位图表示, 就会显示红色。例如, 如果一个 2D 对象没有设置 `cacheAsBitmapMatrix` 属性, 则缩放或旋转该对象时将重新呈现该对象。当子显示对象移动或更改时, 也会发生重新呈现。

四个重绘周期后, 每个彩色矩形都会淡出, 前提是在这些周期中不再出现着色原因。但是, 如果屏幕上没有变化, 诊断颜色也不会变化。

例如, 请考虑矢量背景 (未缓存为位图) 前面的位图显示对象 (香蕉)。首次呈现香蕉时, 它显示为绿色。首次呈现背景时, 它显示为蓝色:

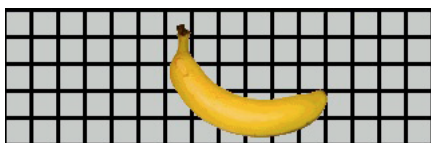


香蕉移动时，CPU 必须重新呈现背景，导致在背景上显示蓝色阴影：



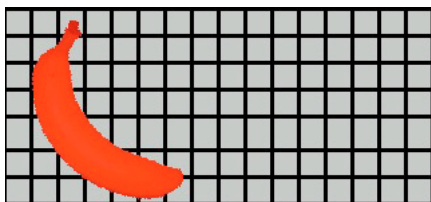
背景上的蓝色阴影反映的是需要发送到 GPU 的重绘区域。

但是，如果将背景缓存为位图，则当香蕉移动时，呈现诊断功能将不显示彩色色调：



由于 GPU 保留了背景位图，因此诊断功能不显示彩色色调。GPU 不涉及 CPU 即可将香蕉与背景复合。

假设香蕉是一个没有设置 `cacheAsBitmapMatrix` 属性的 2D 显示对象。只要该显示对象旋转（或缩放），呈现诊断功能都会显示红色。这表示应用程序需要向 GPU 上载该显示对象的新版本：



向 App Store 提交 iPhone 应用程序

要向 App Store 提交应用程序：

- 1 从 iPhone 开发人员中心网站 (<http://developer.apple.com/iphone/>) 获取分发证书和供给配置文件。

分发证书命名为“iPhone 开发人员：XXX.cer”，其中 XXX 表示您的姓名。

有关详细信息，请参见第 3 页的“从 Apple 获取开发人员文件”。

- 2 将分发证书转换为 P12 文件。

有关详细信息，请参阅第 5 页的“将开发人员证书转换为 P12 文件”。

- 3 使用 P12 文件和供给配置文件编译应用程序。

使用基于分发证书转换的 P12 文件。使用与分发供给配置文件关联的应用程序 ID。

有关详细信息，请参阅第 17 页的“编译 iPhone 应用程序安装程序 (IPA) 文件”。

- 4 在 iPhone 开发人员中心网站 (<http://developer.apple.com/iphone/>) 提交应用程序。

重要说明：Apple 要求您使用 Application Loader 程序，以便将应用程序上载到应用程序库。Apple 仅针对 Mac OS X 发布了 Application Loader。当您使用 Windows 计算机为 iPhone 开发 AIR 应用程序时，必须具有访问运行 OS X（版本 10.5.3 或更高版本）的计算机的权限，以便将应用程序提交到应用程序商店。您可以从 Apple iOS 开发人员中心获取 Application Loader 程序。

第 3 章：移动设备的 ActionScript 3.0 API 支持

在创建移动 AIR 应用程序时，您可以使用可用于开发 Adobe Flash Player 10.1 和 AIR 2 桌面应用程序的相同 ActionScript 3.0 API。但是，也有一些例外和新增内容。

移动设备不支持 ActionScript 3.0 API

不支持某些 ActionScript 3.0 API 用于在移动设备配置文件中运行的应用程序，如在 iPhone 上运行的应用程序。

当使用相同的 ActionScript 代码开发多个配置文件（例如桌面和移动设备配置文件）时，使用代码测试是否支持某个 API。例如，iPhone 应用程序中不支持 NativeWindow 类。（iPhone 应用程序无法使用或创建本机窗口。）要测试应用程序是否在支持本机窗口的配置文件上运行（例如桌面配置文件），请查看 NativeWindow.isSupported 属性。

下表列出了移动设备配置文件中不支持的 API。其中还列出了一些属性，通过这些属性可以确定应用程序何时在支持 API 的平台上运行。

API	测试是否支持
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 support	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported

API	测试是否支持
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

您不能为移动设备配置文件编写基于 HTML 和 JavaScript 的 AIR 应用程序。

仅部分支持某些 ActionScript 3.0 类：

文件

iPhone 应用程序只能访问应用程序目录和应用程序存储目录。您也可以调用 `File.createTempFile()` 和 `File.createTempDirectory()`。调用某个操作以访问另一个目录（例如 `FileStream` 读取或写入方法）会导致 `IOError` 异常。

iPhone 应用程序不支持本机文件浏览器对话框，例如 `File.browseForOpen()` 方法提供的对话框。

Loader

在 iPhone 应用程序中，无法使用 `Loader.load()` 方法。但在使用 `Loader.load()` 方法加载的 SWF 内容中又无法运行任何 ActionScript 代码。不过您可以使用 SWF 文件中的资源（例如库中的影片剪辑、图像、字体和声音）。您还可以使用 `Loader.load()` 方法加载图像文件。

视频

iPhone 上的 AIR 应用程序中仅支持 Sorensen 视频和 ON2 VP6 视频。

您可以使用 `navigateToURL()` 方法打开该应用程序之外的 H.264 视频。作为 `request` 参数，该方法传递 `URLRequest` 对象，其中 URL 指向该视频。视频将在 iPhone 的视频播放器中启动。

Text fields

iPhone 中文本字段的字体和其他设置存在一些限制。请参见第 32 页的“[字体和文本输入](#)”。

不支持的 API 和使用 ADL 进行调试

iPhone 中某个不支持的 AIR 功能在使用 ADL 测试应用程序（在开发计算机上）时仍然可用。在使用 ADL 测试内容时，请注意这些区别。

此功能包含下列视频和音频编解码器：`Speex`（音频）、`H.264/AVC`（视频）和 `AAC`（音频）。这些编解码器对在 iPhone 上运行的 AIR 应用程序不可用。然而，它们可以继续桌面上正常运行。

辅助功能和屏幕阅读器支持在 Windows 上的 ADL 中工作。然而，iPhone 中不支持这些 API。

如果从桌面上的 ADL 使用 RTMPE 协议，该协议可以正常运行。然而，尝试使用 RTMPE 协议连接的 `NetConnection` 在 iPhone 上无法正常运行。

如果使用 ADL 执行内容，则 `Loader` 类可以正常运行，没有其他限制。然而，当在 iPhone 中执行时，尝试加载包含 ActionScript 字节代码的 SWF 内容会导致错误消息。

`Shader` 实例在 ADL 中执行。然而，在 iPhone 中无法解释 `Pixel Bender` 字节代码，而且着色器没有图形效果。

有关详细信息，请参阅第 20 页的“[调试 iPhone 应用程序](#)”。

特定于移动 AIR 应用程序的 ActionScript API

下列 API 仅在移动设备上的 AIR 应用程序中可用。当前在 Flash Player 或 AIR 的桌面版本中不可用。

屏幕方向 API

屏幕方向 API 允许使用舞台方向和 iPhone:

- `Stage.autoOrients` — 应用程序是否设置为使舞台随设备旋转自动重新取向。当在 Flash Professional CS5 “iPhone 设置” 对话框中选择 “自动方向” 选项时, 此属性将设置为 `true`。(也可以在应用程序描述符文件中将 `autoOrients` 元素设置为 `true`。)请参见第 12 页的 “iPhone 应用程序设置”。可通过为 `Stage` 对象添加 `orientationChanging` 事件侦听器来取消自动重新定向。调用此事件对象的 `preventDefault()` 方法可取消自动重新定向。

在使用 “自动方向” 时, 为获得最佳效果, 应按照如下所示设置舞台的 `align` 属性:

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

- `Stage.deviceOrientation` — 设备的物理方向。 `StageOrientation` 类定义此属性的值。
- `Stage.orientation` — 舞台的目前方向。 `StageOrientation` 类定义此属性的值。
- `Stage.supportsOrientationChange` — 在 iPhone 中设置为 `true`, 在 AIR 应用程序中设置为 `false`。
- `Stage.setOrientation()` — 设置舞台的方向。此方法包含一个参数, 该参数是一个定义新舞台方向的字符串。 `StageOrientation` 类中的常量用于定义此参数的可能值。
- `StageOrientation` — 定义舞台方向值。例如, `StageOrientation.ROTATED_RIGHT` 表示舞台相对于设备的默认方向向右旋转。
- `StageOrientationEvent` — 定义当屏幕方向更改时舞台调度的事件。当用户旋转 iPhone 时发生此事件。有两种类型的事件。当设备旋转时, `Stage` 调度 `orientationChanging` 事件。要防止舞台重新取向, 则调用 `orientationChanging` 事件对象的 `preventDefault()` 方法。舞台重新取向完成后, `Stage` 将调度 `orientationChange` 事件。

当前, 屏幕方向 API 仅在移动设备上的 AIR 应用程序中可用。如果移动 AIR 应用程序和桌面 AIR 应用程序共享源代码, 请使用 `Stage.supportsOrientationChange` 属性来检查是否支持 API。

以下示例显示如何响应旋转设备的用户:

```
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;  
    }  
}
```

在此示例中，出现不同的舞台方向时，将显示注释而不是功能代码。

您可以通过调用 **Stage** 对象的 `setOrientation()` 方法来更改舞台的方向。设置方向是一种异步操作。您可以通过侦听 `orientationChange` 事件来检查取向完成的时间。以下代码显示如何将舞台设置为向右：

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

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

舞台在旋转时会进行大小调整，并且 **Stage** 对象将调度 `resize` 事件。您可以调整显示对象大小，并将其重新定位在舞台上，以响应 `resize` 事件。

NativeApplication.systemIdleMode 和 SystemIdleMode

借助 `NativeApplication.systemIdleMode` 属性，您可以阻止 iPhone 进入闲置模式。默认情况下，如果一段时间内没有触摸屏交互，iPhone 将进入闲置模式。闲置模式可以导致屏幕变暗，也可以导致 iPhone 进入锁定模式。此属性可以设置为下列值之一：

- `SystemIdleMode.NORMAL` — iPhone 执行标准闲置模式行为。
- `SystemIdleMode.KEEP_AWAKE` — 应用程序尝试阻止 iPhone 进入闲置模式。

仅在移动设备上支持此功能。在桌面操作系统上运行的 AIR 应用程序不支持此功能。对于在桌面上运行的应用程序，设置 `NativeApplication.systemIdleMode` 属性不起任何作用。

以下代码显示如何禁用 iPhone 闲置模式：

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

CameraRoll

借助 `CameraRoll` 类，您可以向 iPhone 相机胶卷添加图像。`addBitmapData()` 方法将向 iPhone 相机胶卷添加图像。此方法包含一个参数 `bitmapData`。此参数是 `BitmapData` 对象，其中包含要添加到相机胶卷的图像。

仅移动设备支持 `CameraRoll` 功能。在桌面操作系统上运行的 AIR 应用程序不支持此功能。要在运行时查看应用程序是否支持 `CameraRoll` 功能，请查看 `CameraRoll.supportsAddBitmapData` 静态属性。

在调用 `addBitmapData()` 方法之后，`CameraRoll` 对象将调度下列事件之一：

- `complete` — 成功完成操作。
- `error` — 存在错误。例如，iPhone 上可能没有足够的可用空间来存储图像。

以下代码将舞台的图像（屏幕捕获）添加到相机胶卷：

```
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` 属性是一个 `Matrix` 对象，用于定义当 `cacheAsBitmap` 设置为 `true` 时显示对象的呈现方式。当呈现显示对象的位图版本时，应用程序将此矩阵用作转换矩阵。

设置 `cacheAsBitmapMatrix` 后，应用程序将保留缓存的位图图像，该图像使用此矩阵（而不是显示矩阵）呈现。（显示矩阵是显示对象的 `transform.concatenatedMatrix` 的值。）如果此矩阵与显示矩阵不匹配，则根据需要缩放和旋转位图。

仅当 `cacheAsBitmapMatrix` 的值更改时，设置了 `cacheAsBitmapMatrix` 的显示对象才重新呈现。位图会适当地缩放或旋转以符合显示矩阵。

`cacheAsBitmapMatrix` 属性有益于基于 CPU 和基于 GPU 的呈现，但通常对 GPU 呈现更有益。

注 要使用硬件加速，请在 Flash Professional CS5 的“iPhone 设置”对话框中的“常规”选项卡上将“呈现”设置为 GPU。（或在应用程序描述符文件中将 `renderMode` 属性设置为 `gpu`。）

例如，以下代码使用显示对象的未转换的位图表示形式：

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

以下代码使用与当前呈现匹配的位图表示形式：

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

通常，恒等矩阵 (`new Matrix()`) 或 `transform.concatenatedMatrix` 就足够了。然而，您可以使用其他矩阵（例如按比例缩小的矩阵）将其他位图上载到 GPU。例如，以下示例应用了一个在 X 轴和 Y 轴上缩放了 0.5 的 `cacheAsBitmapMatrix` 矩阵。GPU 使用的位图对象较小，但 GPU 会调整其大小以与显示对象的 `transform.matrix` 属性匹配：

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

通常，选择的矩阵会将显示对象转换为其将在应用程序中的显示大小。例如，如果您的应用程序显示缩小一半的 **Sprite** 的位图版本，则应使用缩小一半的矩阵。如果您的应用程序显示的 **Sprite** 大于其当前尺寸，则使用的矩阵应按该系数放大。

对于设置了 `cacheAsBitmapMatrix` 属性的显示对象的大小，具有实际限制。该限制为 1020×1020 像素。对于设置了 `cacheAsBitmapMatrix` 属性的所有显示对象的像素总数，具有实际限制。该限制大约为四百万像素。

使用 `cacheAsBitmapMatrix` 和硬件加速时，需要谨记许多注意事项。有必要了解哪些显示对象应设置该属性，而哪些显示对象不应设置该属性。有关使用此属性的重要信息，请参阅第 30 页的“[硬件加速](#)”。

您可以使用 GPU 呈现诊断功能来诊断应用程序调试版本中的 GPU 使用情况。有关详细信息，请参阅第 20 页的“[调试 iPhone 应用程序](#)”。

网络注意事项

将以下 URL 方案和 `navigateToURL()` 函数配合使用会导致文档在外部应用程序中打开：

URL 方案	调用 <code>navigateToURL()</code> 的结果	示例
<code>mailto:</code>	在邮件应用程序中打开新消息。	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>sms:</code>	在文本消息应用程序中打开消息。	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>tel:</code>	拨打手机中的电话号码（需要用户同意）。	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

iPhone 应用程序可以依赖安装的内签名根证书在安全事务（例如 `https` 请求）期间进行服务器身份验证。服务器不仅应该发送分支证书，还应该发送与根证书关联的所有中间证书。

非常适用于移动应用程序开发人员的 ActionScript 3.0 API

下列 ActionScript 3.0 API 定义了可在移动设备上使用的功能。

Accelerometer API

下列类允许应用程序从设备的加速计获取事件：

- `Accelerometer`
- `AccelerometerEvent`

有关详细信息，请参阅[加速器输入](#)。

Geolocation API

下列类允许应用程序从设备的位置传感器获取事件：

- `Geolocation`
- `GeolocationEvent`

有关详细信息，请参阅[Geolocation](#)。

触摸 **API**、多点触控 **API** 和手势 **API**

下列类允许应用程序接收 touch 和 gesture 事件：

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

有关详细信息，请参阅[触摸、多点触控和手势输入](#)。

第 4 章 : iPhone 应用程序设计注意事项

iPhone 的处理速度和屏幕大小决定了特殊的设计和编码注意事项。但是，许多设计注意事项对于所有应用程序的开发或移动应用程序的开发来说都是通用的。

有关优化应用程序的详细信息，请参阅[优化 Flash Platform 的性能](#)。通常，此文档包含许多优化移动内容、Flash Player 内容、AIR 内容和基于 ActionScript 的内容的性能的建议。其中大多数建议也适用于 iPhone 的 AIR 应用程序。

重要事项：要开发能够在 iPhone 上正常运行的应用程序，必须了解其中诸多设计注意事项并掌握优化技术。

硬件加速

可以使用 OpenGL ES 1.1 硬件加速来增强一些应用程序中的图形性能。其中对显示对象进行了动画处理的游戏和其他应用程序可受益于硬件加速。使用硬件加速的应用程序可以将某些图形进程的负载从 CPU 卸载到 iPhone GPU，这可以大大提高性能。

当设计一个应用程序使用 GPU 时，一定要遵循这样的原则，即确保内容是有效支持 GPU 加速的。

要使用硬件加速，请在 Flash Professional CS5 的“iPhone 设置”对话框中的“常规”选项卡上将“呈现”设置为 GPU。也可以在应用程序描述符文件中将 renderMode 属性设置为 gpu：

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

请参阅第 13 页的“[在 Flash Professional CS5 中设置 iPhone 应用程序属性](#)”和第 14 页的“[在应用程序描述符文件中设置 iPhone 应用程序属性](#)”。

有四类显示对象，如果其内容频繁变化，可在硬件加速中快速呈现。

- 位图对象
- 2D 显示对象，cacheAsBitmap 属性设置为 true，也可以设置 cacheAsBitmapMatrix 属性（参见下面内容）
- 3D 显示对象（即设置了 z 属性）
- 使用单一纯色矩形填充以及边缘与屏幕上的像素对齐的显示对象。

当其他 Sprite 在基于矢量的对象上方或下方执行动画效果时，这些对象将重新呈现。因此，充当某个动画的背景或前景的任何对象也应属于这些类别之一。

对于 cacheAsBitmap 设置为 true 的显示对象，设置 cacheAsBitmapMatrix 将导致 GPU 使用通过矩阵转换生成的位图。GPU 将使用位图表示形式，即使旋转或缩放对象也是如此。与 CPU 重绘矢量渲染的对象相比，GPU 合成此位图并为其添加动画效果要快得多。

单独将 cacheAsBitmap 设置为 true 会导致缓存显示对象（以及所有子对象）。新区域暴露或整个组合图形转换时，显示对象不重绘。

设置显示对象的 cacheAsBitmapMatrix 属性后，此应用程序可为其构建表示形式，即使在显示对象不可见时也是如此。在开始下一帧时，此应用程序将为显示对象构建缓存表示形式。然后，当您显示对象添加到舞台时，此应用程序将快速呈现该对象。此外，应用程序还能够以动画方式快速旋转或缩放对象。对于旋转或缩放的对象，如果还未设置 cacheAsBitmapMatrix 属性，则不要设置 cacheAsBitmap 属性。

对于缓存为位图的对象，应用程序也可以快速执行 alpha 转换。但是，对于硬件加速 alpha 转换，仅支持 0 与 1.0 之间的 alpha 值。这与 colorTransform.alphaMultiplier 设置为 0 到 256 相对应。

对于频繁更新的对象，例如文本字段，请不要将 cacheAsBitmap 属性设置为 true。

频繁更改图形内容的显示对象通常不太适合进行 GPU 呈现。在 GPU 功能不太强大的早期设备上尤其如此。相比将图形上载到 GPU 所需的开销，CPU 呈现可能是更好的选择。

重构包含相对于父显示对象移动的子显示对象的显示对象。进行更改，使子显示对象变为与父显示对象同级。这可确保每个对象都有自己的位图表示形式。同时，每个显示对象将能够相对于其他显示对象移动，而无需将新图形上载到 GPU。

在子显示对象没有动画效果的最高级别的显示列表中，将 `cacheAsBitmap` 属性设置为 `true`。换句话说，为不包含移动部分的显示对象容器进行此设置。请不要对子显示对象进行此设置。请不要对包含具有动画效果的其他显示对象的 `Sprite` 进行此设置。

当设置显示对象的 `z` 属性时，应用程序始终使用缓存的位图表示形式。此外，设置显示对象的 `z` 属性后，该应用程序将使用缓存的位图表示形式，即使旋转或缩放该对象也是如此。此应用程序不会为设置了 `z` 属性的显示对象使用 `cacheAsBitmapMatrix` 属性。相同的规则同样适用于设置任何三维显示对象属性（包括 `rotationX`、`rotationY`、`rotationZ` 和 `transform.matrix3D` 属性）的情况。

如果显示对象容器中包含要对其使用硬件加速的内容，请不要设置 `scrollRect` 或 `mask` 属性。设置这些属性会对显示对象容器及其子对象禁用硬件加速功能。作为设置 `mask` 属性的替代选择，将遮罩显示对象置于被遮罩的显示对象之上。

对于可用于硬件加速的显示对象，有大小限制。在旧设备上，宽高均不能超过 1024 像素。在较新的设备上，宽高均不能超过 2048 像素。您可以使用 GPU 呈现诊断工具来测试设备上的性能。

GPU 也使用 iPhone RAM 存储位图图像。它使用的内存至少与位图图像所需的内存相当。

GPU 使用的内存分配是每个位图图像尺寸的 2 的整数次幂。例如，GPU 占用的内存大小可以是 512 x 1024 或 8 x 32。因此 9 x 15 像素的图像的内存使用量与 16 x 16 像素的图像相同。对于缓存的显示对象，您可能想要使用每个方向上接近（但不超过）2 的整数次幂的尺寸。例如，使用 32 x 16 像素的显示对象比使用 33 x 17 像素的显示对象更有效。

不要依赖调整舞台大小来减少为其他平台（例如桌面）保留的资源，而应该使用 `cacheAsBitmapMatrix` 属性或在发布 iPhone 之前调整资源的大小。3D 对象在缓存表面图像时会忽略 `cacheAsBitmapMatrix` 属性。因此，如果显示对象将在 3D 表面上呈现，最好在发布之前调整这些显示对象的大小。

硬件加速的优点和 RAM 占用量之间存在一个平衡点。当内存不足时，iPhone 操作系统将通知其他处于运行状态的本机 iPhone 应用程序释放内存。当这些应用程序处理此通知并释放内存时，它们可能与您的应用程序竞争 CPU 循环。这会立刻降低您的应用程序的性能。请确保在较旧的设备上测试您的应用程序，因为这些设备上为您运行的进程提供的内存可能会大大减少。

在 iPhone 上调试应用程序时，可以启用 GPU 呈现诊断功能。此功能帮助您了解应用程序使用 GPU 呈现的状况。有关详细信息，请参阅第 20 页的“[调试 iPhone 应用程序](#)”中的“[使用 GPU 呈现诊断进行调试](#)”。

有关使用 `cacheAsBitmapMatrix` 属性的信息，请参阅第 25 页的“[特定于移动 AIR 应用程序的 ActionScript API](#)”中的“`DisplayObject.cacheAsBitmapMatrix`”部分。

提高显示对象性能的其他方式

硬件加速可以提高一些类型的显示对象的图形性能。下面是一些有关如何最大程度增强图形性能的提升：

- 尝试限制舞台上可见项目的数量。每个项目都需要花费一些时间来呈现其周围的其他项目并与它们合成。
当您不再需要显示显示对象时，请将其 `visible` 属性设置为 `false`，或将其从舞台 (`removeChild()`) 删除。不要仅将其 `alpha` 属性设置为 0。
- 通常需要避免使用混合模式，特别是图层混合模式。请尽可能使用正常混合模式。
- 显示对象滤镜计算成本很高。请尽量少用。例如，可以在简介屏幕上使用一些滤镜。但是，请避免对许多对象或对其设置动画效果的对象使用滤镜，或在必须使用高帧速率时使用滤镜。
- 避免使形状变形。
- 避免使用裁剪。

- 如果可能，在调用 `Graphic.beginBitmapFill()` 方法时将 `repeat` 参数设置为 `false`。
- 不要过分夸张。将背景色作为背景。不要将较大形状进行叠放。每个必须绘制的像素都需要成本。对于非硬件加速的显示对象来说尤其如此。
- 避免使用带有狭长突起、自相交的边缘或边缘周围存在许多细节的形状。呈现这些形状需要的时间比边缘平滑的显示对象长。对于非硬件加速的显示对象来说尤其如此。
- 使位图的大小接近但小于 2^n 乘 2^m 位。尺寸不必为 2 的整数次幂，但是应该接近 2 的整数次幂，而不需要更大。例如，31 乘 15 像素的图像比 33 乘 17 像素的图像的呈现速度更快。(31 和 15 刚刚小于 2 的整数次幂: 32 和 16。) 此类图像也能更高效地使用内存。
- 将显示对象的尺寸限制为 1024 x 1024 像素（或在较新的设备上为 2048 x 2048 像素）。

信息密度

移动设备屏幕的实际大小小于其在桌面上的大小，尽管前者的像素密度更高。更清楚的文本看起来更美观，但是字型必须是易于阅读的最小实际大小。

移动设备通常在活动时以及照明条件很差的情况下使用。考虑您实际可以在屏幕上清晰显示多少信息。它可能比您在具有相同像素尺寸的台式机屏幕上显示的内容要少。

使用印刷层次结构突出显示重要信息。使用字体大小、粗细、位置和间距来表示用户界面的元素的相对重要性。您可以在层次结构的每层上使用一个或多个提示。在整个应用程序内统一应用这些提示。提示可以是空间形式的（缩进、行距、位置），也可以是图形形式的（大小、样式、字体颜色）。应用重复的提示可以有效确保清晰表示层次结构。但是，对每层分组使用的提示不要超过三个。

尝试简化所需的标签和说明文本。例如，请在文本字段中使用示例输入来建议内容和避免使用单独的标签。

字体和文本输入

要实现最佳外观，请使用设备字体。例如，下列几种字体是 iPhone 上的设备字体：

- Serif: Times New Roman、Georgia 和 `_serif`
- Sans-serif: Helvetica、Arial、Verdana、Trebuchet、Tahoma 和 `_sans`
- 等宽字体: Courier New、Courier 和 `_typewriter`

请使用 14 像素或更大的字体。

对可编辑的文本字段使用设备字体。文本字段中的设备字体的呈现速度比嵌入的字体更快。

不要在输入文本字段中使用带有下划线的文本。并且不要设置文本字段的对齐方式。iPhone 上的输入文本字段仅支持左对齐（默认值）。

如果对 Flash Professional CS5 中的文本字段使用 TLF 文本设置，请关闭 ActionScript 3.0 设置中默认链接中的运行时共享库。否则，应用程序不会在 iPhone 上工作，因为它将尝试使用运行时共享库 SWF 文件：

- 1 选择“文件”>“发布设置”。
- 2 在“发布设置”对话框中，单击“Flash”选项卡。
- 3 单击“脚本”(ActionScript 3.0) 下拉列表右侧的“设置”按钮。
- 4 单击“库路径”选项卡。
- 5 在“默认链接”下拉列表中，选择“合并到代码”。

考虑实现替代方法以使用输入文本字段。例如，要让用户输入数值，您不需要文本字段。您可以提供两个按钮来增加或减少此值。

注意虚拟键盘使用的空间。激活虚拟键盘时（例如，当用户在文本字段中点击时），应用程序将调整舞台的位置。自动重新定位确保可以看到所选的输入文本字段：

- 舞台顶部的文本字段将移动到可见舞台区域的顶部。（可见舞台区域对于容纳虚拟键盘有些小。）
- 舞台底部的文本字段位于新舞台区域的底部。
- 舞台其余部分中的文本字段将移动到舞台的垂直中心。

当用户单击文本字段对其进行编辑（将显示虚拟键盘）时，`TextField` 对象将调度 `focusIn` 事件。您可以为此事件添加一个事件侦听器来重新定位文本字段。

当用户编辑单行文本字段中的文本时，该字段中会包含一个清除按钮（位于文本右侧）。然而，如果文本字段太窄，则不会显示此清除按钮。

用户在单行文本字段中编辑完文本后，可通过轻击键盘上的“完成”键来使虚拟键盘消失。

用户在多行文本字段中编辑完文本后，可通过在文本字段外点击来使虚拟键盘消失。这可从文本字段上移除焦点。请确保显示虚拟键盘时，您的设计包括文本字段以外的区域。如果文本字段过大，可能看不到其他区域。

使用某些 **Flash Professional CS5** 组件可以阻止您从文本字段上移除焦点。这些组件是针对在桌面计算机上使用而设计的，桌面计算机上需要此焦点行为。其中一个组件是 `TextArea` 组件。当该组件处于焦点中（并且正在对其进行编辑）时，则无法通过单击其他显示对象来移除焦点。在舞台上放置一些其他 **Flash Professional CS5** 组件也可以阻止更改正在对其进行编辑的文本字段上的焦点。

不要依赖键盘事件。例如，针对 **Web** 设计的某些 **SWF** 内容使用键盘来允许用户控制应用程序。但是，在 **iPhone** 上，仅当用户编辑文本字段时才显示虚拟键盘。**iPhone** 应用程序在存在虚拟键盘时仅调度键盘事件。

保存应用程序状态

您的应用程序可能会随时退出（例如，当电话响起时）。考虑在应用程序状态更改时进行保存。例如，您可以将设置保存到应用程序存储目录中的文件或数据库中。您还可以将数据保存到本地共享对象。然后，当重新启动应用程序时，您可以还原其状态。如果电话呼叫中断应用程序，当呼叫结束时应用程序将重新启动。

在应用程序退出时不要依赖调度 `exiting` 事件的 `NativeApplication` 对象；它可能不调度该事件。

屏幕方向更改

可以纵向或横向查看 **iPhone** 内容。考虑您的应用程序如何处理屏幕方向更改。有关详细信息，请参阅[检测设备方向](#)。

感应区目标

在设计按钮和用户点击的其他用户界面元素时，需要考虑感应区目标的大小。这些元素要足够大，以便在触摸屏上使用一个手指即可激活。此外，还要确保目标之间具有足够的空间。命中目标应该大约介于 44 像素到 57 像素。

内存分配

分配新的内存块代价很高。会降低应用程序运行速度，或导致在动画播放或交互期间性能延迟，因此此操作会触发垃圾回收。

尽可能设法回收对象，而不是将其弃之不用，然后再创建一个新对象。

请记住，矢量对象占用的内存比数组少。请参阅 [Vector 类](#) 和 [Array 类](#)。

有关内存使用的详细信息，请参阅 [节省内存](#)。

绘图 API

尽量避免使用 **ActionScript** 绘图 API (**Graphics** 类) 创建图形。在舞台上使用绘图 API 动态创建对象，然后将其呈现到光栅器中。如果可能，在舞台上处于编辑状态时静态创建这些对象。

每次执行 **ActionScript** 时，如果重复调用使用绘图 API 创建的对象，将破坏并重新创建这些对象。但是，静态对象通过不同的时间轴存放在内存中。

事件冒泡

对于深层嵌套的显示对象容器，事件冒泡成本很高。可以通过完全在目标对象中处理事件，然后调用事件对象的 `stopPropagation()` 方法来降低此成本。调用此方法会阻止事件冒泡。调用此方法还意味着父对象不会接收该事件。

相关结果可以通过将显示对象嵌套平面化来实现，以避免事件链较长。

请尽可能注册 **MouseEvent** 事件而不是 **TouchEvent** 事件。**MouseEvent** 事件使用的处理器开销比 **TouchEvent** 事件少。

尽可能将 `mouseEnabled` 和 `mouseChildren` 属性设置为 `false`。

优化视频性能

要优化移动视频播放，请确保在视频播放期间应用程序中几乎不运行其他内容。这样，就可以将尽可能多的 CPU 用于视频解码和呈现过程。

当播放视频时，运行很少的 **ActionScript** 代码或根本不运行。尽可能避免运行在频繁的间隔计时器或时间轴上运行的代码。

尽量减少重绘非视频显示对象。特别是避免重绘与视频区域相交的显示对象。即使这些对象隐藏在视频下面，也要避免这样做。否则，仍会重绘显示对象并用尽处理资源。例如，为位置指示符使用简单形状，并在一秒钟内仅更新几次位置指示符而不是在每个帧上都更新。请不要将视频控件与视频区域重叠，而是直接将它们放在下方。对于视频缓冲动画，在不使用时不要将其隐藏在视频后，而是将其设置为不可见。

Flex 和 Flash 组件

许多 **Flex** 组件和 **Flash** 组件设计为在桌面应用程序中使用。这些组件（尤其是显示组件）在移动设备上性能比较缓慢。除了性能缓慢之外，桌面组件的有些交互模型可能对于移动设备不适合。

Adobe 正在开发 **Flex** 框架的移动优化版本。有关详细信息，请参阅 <http://labs.adobe.com/technologies/flex/mobile/>。

减少应用程序文件大小

以下是如何减小 IPA 文件的文件大小的一些提示：

- 查看背景位图以确认它们的大小是否合适（不能大于所需大小）。
- 查看以确认是否嵌入了任何多余字体。
- 查看 PNG 资源是否有 Alpha 通道，并在不需要时将其删除。使用实用程序（如 PNG Crunch）减小 PNG 资源的大小。
- 尽可能将 PNG 资源转换为 JPG 资源。
- 考虑使用较低比特率压缩声音文件
- 删除任何不使用的资源。