

Criando aplicativos Adobe® AIR® com o Packager for iPhone®

Avisos legais

Para ver os avisos legais, consulte http://help.adobe.com/pt_BR/legalnotices/index.html.

Conteúdo

Capítulo 1: Introdução à criação de aplicativos AIR para iPhone

Conceitos importantes	1
Obtenção das ferramentas de desenvolvedor da Adobe	4
Obtenção dos arquivos de desenvolver da Apple	4
Criação do aplicativo Hello World para iPhone com o Flash Professional CS5	8

Capítulo 2: Compilação e depuração de aplicativos para iPhone

Ícone do iPhone e imagens de tela iniciais	13
Configurações de aplicativos para iPhone	15
Compilando um IPA (arquivo do instalador do aplicativo)	21
Instalar um aplicativo iPhone	23
Depurando um aplicativo para iPhone	24
Submetendo seu aplicativo iPhone à App Store	27

Capítulo 3: Suporte ActionScript 3.0 API para dispositivos móveis

ActionScript 3.0 APIs sem suporte em dispositivos móveis	28
ActionScript APIs específica para aplicativos AIR móveis	30
Arquivos API do ActionScript 3.0 de interesse especial para desenvolvedores de aplicativos	35

Capítulo 4: Considerações a respeito do design do aplicativo para iPhone

Aceleração por hardware	36
Outras maneiras para melhorar o desempenho de objetos de exibição	38
Densidade das informações	39
Fontes e entrada de texto	39
Como salvar o estado do aplicativo	41
Alterações na orientação da tela	41
Destinos de ocorrências	41
Alocação de memória	41
API de desenho	41
Fase de bubbling de eventos	42
Otimizar desempenho de vídeo	42
Componentes do Flex e Flash	42
Redução do tamanho do arquivo do aplicativo	42

Capítulo 1: Introdução à criação de aplicativos AIR para iPhone

Você pode utilizar as ferramentas da plataforma Adobe® Flash® e o código ActionScript® 3.0 para criar aplicativos AIR para iPhone e iPod Touch. Esses aplicativos são distribuídos, instalados e executados da mesma maneira que outros aplicativos para iPhone.

Nota: O restante deste documento refere-se a iPhone e iPod Touch simplesmente como “iPhone”.

O Packager for iPhone® está incluído no Adobe® Flash® Professional CS5. O Packager for iPhone compila os bytecodes do ActionScript 3.0 para o código do aplicativo nativo para iPhone. Os aplicativos para iPhone são distribuídos como arquivos do instalador do aplicativo para iPhone (arquivos .ipa) por meio da iTunes Store.

É possível usar o Flash Professional CS5 ou Adobe® Flash® Builder™ 4 para editar o conteúdo de origem do ActionScript 3.0 de seu aplicativo.

Para desenvolver aplicativos para o iPhone, utilize o Flash Professional CS5.

Obtenha também os certificados de desenvolvedor de iPhone da Apple.

Importante: Antes de desenvolver aplicativos para iPhone, reveja as informações sobre design de aplicativos para iPhone. Consulte “[Considerações a respeito do design do aplicativo para iPhone](#)” na página 36. Além disso, informe-se a respeito dos arquivos de desenvolvedor necessários para criar um aplicativo para iPhone. Consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.

Conceitos importantes

É importante compreender os conceitos e o fluxo de trabalho envolvidos antes de desenvolver um aplicativo para iPhone usando o ActionScript 3.0.

Glossário

Os termos abaixo são importantes para compreender a criação de um aplicativo para iPhone.

Site do iPhone Dev Center Site da Apple Computer (<http://developer.apple.com/iphone/>) onde você pode encontrar as informações a seguir:

- Registre-se para tornar-se um desenvolvedor para iPhone.
- Gerencie e crie certificados de desenvolvimento para o iPhone, perfis de provisionamento e IDs de aplicativos (definidos abaixo).
- Envie aplicativos para a App Store.

Certificado de desenvolvimento de iPhone Usado para identificar um desenvolvedor para propósitos de desenvolvimento de aplicativos.

Obtenha esse arquivo na Apple. Converta esse certificado para um arquivo de certificado P12 para assinar o aplicativo para iPhone que cria usando o ActionScript 3.0. Consulte *Arquivo de certificado P12*.

Não é necessário um certificado de desenvolvimento de iPhone para depurar e testar aplicativos do Flash Professional CS5 no computador de desenvolvimento. No entanto, é necessário obter um certificado de desenvolvimento para instalar e testar o aplicativo em um iPhone.

O certificado de desenvolvimento é diferente de um certificado de distribuição, utilizado para criar uma versão final de seu aplicativo. Você obtém um certificado de distribuição da Apple quando cria a versão final de seu aplicativo.

Solicitação de assinatura de certificado Arquivo que contém informações pessoais usadas para criar um certificado de desenvolvimento. Também conhecido como arquivo CSR.

Perfil de aprovisionamento Arquivo que permite que você teste ou distribua um aplicativo para iPhone. Obtenha os arquivos do perfil de aprovisionamento na Apple. O perfil de aprovisionamento é designado para um certificado de desenvolvimento específico, um ID do aplicativo e um ou mais IDs de dispositivos. Há diferentes tipos de perfis de aprovisionamento:

- **Perfil de aprovisionamento de desenvolvimento:** Usado para instalar a versão de teste de um aplicativo no iPhone do desenvolvedor.
- **Perfil de aprovisionamento de teste:** Conhecido também como perfil de aprovisionamento ad hoc. Usado para distribuir a versão de teste do aplicativo para múltiplos usuários (e unidades iPhone). Com o perfil de aprovisionamento e o aplicativo de teste, os usuários podem testar seu aplicativo, sem submetê-lo à App Store. Observação: É possível também usar o perfil de aprovisionamento de desenvolvimento para distribuir aplicativos de teste para múltiplos dispositivos.
- **Perfil de aprovisionamento de distribuição:** Usado para criar um aplicativo para iPhone para submeter seu aplicativo à App Store.

ID do aplicativo Sequência de caracteres única que identifica um aplicativo para iPhone (ou múltiplos aplicativos) de um desenvolvedor específico. Crie IDs e aplicativo no site do iPhone Dev Center. Cada perfil de aprovisionamento possui um ID de aplicativo ou padrão de ID associado. Utilize este ID de aplicativo ID (ou padrão) no desenvolvimento de um aplicativo. Utilize o ID de aplicativo na caixa de diálogos Configurações iPhone para Flash Professional CS5 (ou no arquivo descritor do aplicativo).

IDs de aplicativo do iPhone Dev Center contém um ID da distribuição de conjunto seguido por um identificador de conjunto. A ID da distribuição de conjunto é uma sequência de caracteres como, por exemplo, 5RM86Z4DJM, que a Apple atribui à ID de aplicativo. O identificador de conjunto contém a sequência de caracteres de nome de domínio reverso selecionada. Um identificador de conjunto pode terminar em um asterisco (*), indicando uma ID de aplicativo curinga. A seguir, alguns exemplos:

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.* (ID de aplicativo curinga)

Existem dois tipos de ID de aplicativo no iPhone Dev Center:

- IDs de aplicativo curinga—No iPhone Dev Center, estes IDs de aplicativo terminam em asterisco (*), como 96LPVWEASL.com.myDomain.* ou 96LPVWEASL.*. Com um perfil de aprovisionamento que utiliza este tipo de ID de aplicativo, você pode gerar aplicativos de teste que utilizam um ID de aplicativo correspondente ao padrão. Para o ID de aplicativo (do aplicativo), você pode substituir o asterisco com qualquer sequência de caracteres válidos. Por exemplo, o site do iPhone Dev Center especifica 96LPVWEASL.com.example.* como ID de aplicativo e você pode utilizar com.example.foo ou com.example.bar como ID de aplicativo (do aplicativo).

- IDs de aplicativo específicos—Definem um ID de aplicativo único para ser utilizado no aplicativo. No iPhone Dev Center, estes IDs de aplicativo não terminam em asterisco. Um exemplo é 96LPVWEASL.com.myDomain.myApp. Com um perfil de provisionamento que utiliza este tipo de ID de aplicativo, os aplicativos devem corresponder exatamente ao ID de aplicativo. Por exemplo, de o site do iPhone Dev Center especificar 96LPVWEASL.com.example.helloWorld como ID de aplicativo, você deve utilizar com.example.foo como ID de aplicativo.

Durante o desenvolvimento de seu aplicativo, especifique o ID de aplicativo na caixa de diálogo de configurações do iPhone no Flash Professional CS5 ou no arquivo descritor do aplicativo. Para obter mais detalhes sobre IDs de aplicativo, consulte a seção “Guia de implantação” de “[Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5](#)” na página 15 ou consulte “[Definição de propriedades de aplicativo do iPhone no arquivo do descritor do aplicativo](#)” na página 17.

Importante: Ao especificar o ID do aplicativo, ignore o ID da distribuição do conjunto do ID de aplicativo. Por exemplo, se a Apple indicar seu ID de aplicativo como 96LPVWEASL.com.example.bob.myApp, ignore a parte 96LPVWEASL—use com.example.bob.myApp como ID do aplicativo. Se a Apple indicar seu ID de aplicativo como 5RM86Z4DJM.*, ignore a parte 5RM86Z4DJM—isto é um ID de aplicativo curinga.

É possível localizar o ID de aplicativo (ou padrão de ID de aplicativo curinga) associado com um perfil de provisionamento no iPhone Dev Center (<http://developer.apple.com/iphone>). Acesse a o iPhone Developer Program Portal e, em seguida, acesse a seção Provisioning.

Arquivo de certificado P12 O arquivo P12 (arquivo com extensão .p12) é um tipo de arquivo de certificado (arquivo Personal Information Exchange). O Packager for iPhone utiliza esse tipo de certificado para criar um aplicativo para iPhone. Converta o certificado de desenvolvedor recebido da Apple para este formulário de certificado.

ID do dispositivo único Código único que identifica um iPhone específico. Conhecido também como UDID ou ID do dispositivo.

Visão geral do fluxo de trabalho do desenvolvimento

Ao desenvolver um aplicativo para iPhone, siga as etapas abaixo:

- 1 Instale o Flash Professional CS5 da Adobe.
- 2 Instale o iTunes.
- 3 Obtenha os arquivos de desenvolver da Apple. Esses arquivos incluem certificado de desenvolvedor e perfis de provisionamento. Consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.
- 4 Converta o certificado de desenvolvimento em um arquivo de certificado P12. O Flash CS5 requer que o certificado seja P12. Consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.
- 5 Use o iTunes para associar seu perfil de provisionamento ao seu iPhone.
- 6 Escreva o aplicativo no Flash Professional CS5.

É importante compreender as melhores práticas para criar o design e otimizar o código do aplicativo para iPhone. Consulte “[Considerações a respeito do design do aplicativo para iPhone](#)” na página 36.

Além disso, alguns arquivos API do ActionScript 3.0 são limitados ou não são suportados no iPhone. Consulte “[Suporte ActionScript 3.0 API para dispositivos móveis](#)” na página 28.

É possível também usar o Flash Builder 4.0 para editar o código do ActionScript 3.0 do aplicativo.

É possível usar o Flash Professional CS5 para testar seu aplicativo no computador de desenvolvimento.

- 7 Crie um ícone arte e uma tela inicial arte para o aplicativo. Cada aplicativo para iPhone inclui um conjunto de ícones que o identificam para os usuários. O iPhone exibe a imagem da tela inicial quando o programa carrega. Consulte [“Ícone do iPhone e imagens de tela iniciais”](#) na página 13.
- 8 Edite as configurações do iPhone. Essas configurações incluem o seguinte:
 - Identidade do aplicativo (incluindo nome do arquivo, nome do aplicativo, número da versão e ID do aplicativo)
 - Local do ícone arte de origem do aplicativo
 - Certificado P12 e perfil de provisionamento designados para o aplicativo
 - Proporção do aspecto inicial do aplicativoNo Flash Professional CS5, é possível editar essas configurações na caixa de diálogo Configurações do iPhone. Para obter detalhes, consulte [“Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5”](#) na página 15.

Também é possível editar essas configurações diretamente no arquivo do descritor do aplicativo. Para mais informações, consulte [“Definição de propriedades de aplicativo do iPhone no arquivo do descritor do aplicativo”](#) na página 17.
- 9 Compile o arquivo IPA usando o Packager for iPhone. Consulte [“Compilando um IPA \(arquivo do instalador do aplicativo\)”](#) na página 21.

10 Instale e teste o aplicativo em seu iPhone. Use o iTunes para instalar o arquivo IPA.

Para distribuição ad hoc, repita esse processo geral; no entanto, use o perfil de provisionamento de teste em vez do perfil de provisionamento de desenvolvimento. Para distribuição final do aplicativo, repita esse processo usando o perfil de provisionamento de distribuição. (Consulte o [“Glossário”](#) na página 1 para obter informações sobre os diferentes tipos de perfis de provisionamento).

Após construir a versão de distribuição de seu aplicativo, consulte as instruções [“Submetendo seu aplicativo iPhone à App Store”](#) na página 27.

Para ver o tutorial rápido sobre a criação de um aplicativo básico para iPhone usando, consulte [“Criação do aplicativo Hello World para iPhone com o Flash Professional CS5”](#) na página 8.

Obtenção das ferramentas de desenvolvedor da Adobe

Para desenvolver aplicativos iPhone utilizando ActionScript 3.0, é necessário utilizar o Flash Professional CS5.

Importante: Atualize a versão do Packager for iPhone a partir da versão de visualização incluída no Flash Professional CS5. No Flash Professional CS5, selecione Ajuda > Atualizações.

Também é possível utilizar o Flash Builder 4 para editar o código do ActionScript. O Flash Builder 4 está disponível em <http://www.adobe.com/br/products/flashbuilder/>.

Obtenção dos arquivos de desenvolver da Apple

Ao desenvolver qualquer aplicativo para iPhone, primeiro é necessário obter os arquivos de desenvolvedor de iPhone da Apple. É necessário obter o certificado de desenvolvedor de iPhone e o perfil de provisionamento de portáteis. É necessário também obter outros perfis de provisionamento. Consulte o [“Glossário”](#) na página 1 para obter as definições desses arquivos.

Nota: Obter esses arquivos é uma parte importante do processo de desenvolvimento do aplicativo. Certifique-se de completar esse processo antes de desenvolver seu aplicativo. Obter os arquivos do desenvolvedor não é um processo simples. Leia atentamente as seguintes instruções e as instruções no site Apple iPhone Dev Center.

Obtendo e trabalhando com arquivos de desenvolvedor de iPhone

É necessário obter o certificado de desenvolvedor de iPhone e os perfis de provisionamento da Apple. É necessário também converter o certificado em certificado P12.

Instalando o iTunes

É necessário o iTunes para instalar seu aplicativo em seu iPhone. Além disso, use o iTunes para verificar o ID do dispositivo de seu iPhone. Será necessário saber o ID do dispositivo ao requerer o certificado de desenvolvedor de iPhone.

Requerendo o certificado de desenvolvedor de iPhone e crie um perfil de provisionamento

Caso ainda não tenha feito isto, registre-se no centro de desenvolvedores da Apple iPhone para ser um desenvolvedor de iPhone (<http://developer.apple.com/iphone/>).

Nota: Você não precisa do iPhone SDK ou XCode para desenvolver aplicativos AIR para iPhone. Você não precisa ser um desenvolvedor registrado de iPhone. E você precisa obter um certificado de desenvolvedor e um perfil de provisionamento.

- 1 Acesse o iPhone Dev Center usando sua conta de desenvolvedor de iPhone.
- 2 No iPhone Dev Center, solicite (e compre) o certificado de desenvolvedor de iPhone.
A Apple irá enviar-lhe uma mensagem por e-mail com seu código de ativação do Programa de Desenvolvedores de iPhone.
- 3 Retorne ao iPhone Dev Center. Siga as instruções sobre como ativar o programa de desenvolvedor (e digite seu código de ativação quando solicitado).
- 4 Quando o código de ativação for aceito, acesse a seção do portal Programa de Desenvolvedores de iPhone do iPhone Dev Center.
- 5 Crie um arquivo e solicitação de assinatura de certificado. Este arquivo será utilizado para obter um iPhone Development Certificate. Para obter instruções, consulte “[Gerar uma solicitação de assinatura de certificado](#)” na página 6.
- 6 Na etapa seguinte, você será solicitado a fornecer o ID do dispositivo (o ID do dispositivo único) de seu iPhone. Você pode obter o UDID no iTunes:
 - a Conecte seu iPhone ao cabo USB. Em seguida, no iTunes, selecione a guia de resumo do iPhone.
 - b Depois de baixar o perfil de provisionamento do site do centro de desenvolvedores de iPhone, adicione-o ao iTunes.
 - c Em seguida, clique no número de série exibido. O UDID é exibido. Clique em Command-C no Mac ou em Control-C no Windows para copiar o UDID para a área de transferência.
- 7 Crie e instale o perfil de provisionamento e o certificado de desenvolvimento de iPhone.

Siga as instruções no site iPhone Dev Center. Procure as instruções na seção do iPhone Developer Program Portal. Você pode querer utilizar o Assistente de provisionamento de desenvolvimento para obter seu certificado de desenvolvedor e criar seu perfil de provisionamento.

Ignore as etapas que envolvem o XCode. Não é necessário usar o XCode para desenvolver aplicativos para iPhone usando o Flash Professional CS5.

- 8 No iTunes, selecione Arquivo > Adicionar à Biblioteca. Em seguida, selecione o arquivo de perfil de aprovisionamento (que tem mobileprovision como extensão do arquivo). Em seguida, sincronize seu iPhone com o iTunes.

Isso permite que você teste o aplicativo associado a esse perfil de aprovisionamento em seu iPhone.

Para verificar se o perfil de aprovisionamento específico é adicionado ao iTunes, tente adicioná-lo à Biblioteca. Se o iTunes perguntar se você deseja substituir o perfil de aprovisionamento existente, pressione o botão Cancelar. (O perfil já está instalado). Além disso, é possível verificar os perfis de aprovisionamento instalados em seu iPhone:

- a Abra o aplicativo Ajustes em seu iPhone.
 - b Abra a categoria Geral.
 - c Pressione Perfis. A página Perfis lista os perfis de aprovisionamento instalados.
- 9 Se ainda não fez isso, baixe o arquivo do certificado de desenvolvimento para iPhone (arquivo .cer).
O Assistente de Aprovisionamento de Desenvolvimento forneceu-lhe um link para baixar esse arquivo. Também é possível encontrar o arquivo na seção Certificates no portal Provisioning no site Apple iPhone Dev Center (<http://developer.apple.com/iphone/>).
 - 10 Em seguida, converta seu certificado de desenvolvedor iPhone para um arquivo P12. Para obter instruções, consulte “[Converter um certificado de desenvolvedor em um arquivo P12](#)” na página 7.

Você agora pode criar um aplicativo Hello World simples. “[Criação do aplicativo Hello World para iPhone com o Flash Professional CS5](#)” na página 8.

Gerar uma solicitação de assinatura de certificado

Para obter um certificado de desenvolvedor gere um arquivo de solicitação de assinatura de certificado, que será enviado para o site Apple iPhone Dev Center.

Gerar a solicitação de assinatura de certificado no Mac OS

No Mac OS, é possível utilizar o aplicativo Acesso Porta-chaves para gerar uma solicitação de assinatura de código. O aplicativo Acesso ao Porta-chaves está no subdiretório Utilitários do diretório Aplicações. No menu Acesso ao Porta Chaves, selecione “Assistente de Certificado” > “Solicitar um Certificado de uma Autoridade de Certificados”.

- 1 Abra o Acesso ao Porta-chaves.
- 2 No menu do Acesso ao Porta-chaves, selecione Preferências.
- 3 Na caixa de diálogo Preferências, clique em Certificates. Em seguida, configure as opções Protocolo de Estado de Certificado On-line e Lista de Anulação de Certificados (CRL) como Desativado. Feche a caixa de diálogo.
- 4 No menu Acesso ao Porta Chaves, selecione “Assistente de Certificado” > “Solicitar um Certificado de uma Autoridade de Certificados”.
- 5 Digite o endereço de e-mail e o nome do seu ID de conta de desenvolvedor iPhone. Não insira um endereço de e-mail de uma autoridade de certificação. Selecione Salvo no disco; em seguida, clique no botão Continuar.
- 6 Salve o arquivo (CertificateSigningRequest.certSigningRequest).
- 7 Faça o upload do arquivo CSR para a Apple [no site de desenvolvedor iPhone](#). (Consulte “Registrar-se para obter um certificado de desenvolvedor iPhone e criar um arquivo de aprovisionamento”).

Gerar a solicitação de assinatura de certificado no Windows

Para desenvolvedores do Windows, talvez seja mais fácil obter o certificado de desenvolvedor de iPhone em um computador Mac. No entanto, é possível obter o certificado em um computador Windows. Primeiro, crie o arquivo CSR (Certificate Signing Request) usando o OpenSSL:

- 1 Instale o OpenSSL em seu computador Windows. (Vá para <http://www.openssl.org/related/binaries.html>).

Você deve instalar os arquivos do Visual C++ 2008 Redistributable, indicados na página de download do Open SSL. (A instalação do Visual C++ no computador *não* é necessária).

- 2 Abra a sessão de comandos do Windows e CD para o diretório OpenSSL bin (por exemplo, c:\OpenSSL\bin\).

- 3 Crie a chave particular digitando as informações abaixo na linha de comando:

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

Salve o arquivo de chave particular. O arquivo será utilizado posteriormente.

Não ignore as mensagens de erro ao utilizar o OpenSSL. Mesmo que o OpenSSL gere uma mensagem de erro, ele ainda pode gerar os arquivos. No entanto, estes arquivos podem não ser utilizáveis. Se ocorrerem erros, verifique a sintaxe e execute o comando novamente.

- 4 Crie o arquivo CSR digitando as informações abaixo na linha de comando:

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

Substitua os valores de endereço de email, CN (nome de certificado) e C (país) pelos seus valores.

- 5 Faça o upload do arquivo CSR para a Apple [no site de desenvolvedor iPhone](#). (Consulte “Registrar-se para obter um certificado de desenvolvedor iPhone e criar um arquivo de provisionamento”).

Converter um certificado de desenvolvedor em um arquivo P12

Para desenvolver aplicativos para iPhone usando o Flash Professional CS5, é necessário usar o arquivo de certificado P12. Crie esse certificado com base no arquivo de certificado de desenvolvedor de iPhone da Apple que você recebe da Apple.

Converter o certificado de desenvolvedor de iPhone em um arquivo P12 no Mac OS

Após baixar o certificado de Apple iPhone, exporte-o para o formato de certificado P12. Para executar isso no Mac® OS:

- 1 Abra o aplicativo Acesso ao Porta-chaves (na pasta Aplicações/Utilitários).
- 2 Se ainda não adicionou o certificado às Chaves, selecione Ficheiro > Importar elementos... Navegue até o diretório do arquivo de certificado (arquivo .cer) obtido da Apple.
- 3 Selecione a categoria Chaves no Acesso ao Porta-chaves.
- 4 Selecione a chave particular associada ao seu certificado de desenvolvimento de iPhone.
O desenvolvedor de iPhone identifica a chave particular: Certificado público <nome> <sobrenome> com o qual coincide.
- 5 Selecione Ficheiro> Exportar elementos...
- 6 Salve a chave no formato de arquivo .p12 (Personal Information Exchange).
- 7 Você será solicitado a criar a senha que é usada ao tentar importar essa chave para outro computador.

Converter um certificado de desenvolvedor da Apple em um arquivo P12 no Windows

Para desenvolver aplicativos para iPhone usando o Flash CS5, é necessário usar o arquivo de certificado P12. Crie esse certificado com base no arquivo de certificado de desenvolvedor de iPhone da Apple que você recebe da Apple.

- 1 Converta o arquivo de certificado de desenvolvedor que recebe da Apple em um arquivo de certificado PEM. Digite o seguinte comando na linha de comando do diretório bin do OpenSSL:

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

- 2 Se estiver usando a chave particular de um keychain em um computador Mac, converta-a em uma chave PEM:

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

- 3 É possível criar um arquivo P12 válido, com base na chave e na versão PEM do certificado de desenvolvedor de iPhone:

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

Se estiver usando a chave do Mac OS Keychain, use a versão PEM que criou na etapa anterior. Do contrário, use a chave do OpenSSL que criou anteriormente (no Windows).

Gerenciar certificados, IDs de dispositivos, IDs de aplicativos e perfis de provisionamento

Você pode gerenciar certificados, IDs de dispositivos, IDs de aplicativos e perfis de provisionamento no site Apple iPhone Dev Center (<http://developer.apple.com/iphone/>). Acesse a seção do iPhone Developer Program Portal no site.

- Clique no link Certificates para gerenciar seus certificados de desenvolvimento. Você pode criar, fazer o download ou anular um certificado. Para criar um certificado, primeiro é necessário criar uma solicitação de assinatura de certificado. Consulte “[Gerar uma solicitação de assinatura de certificado](#)” na página 6.
- Clique no link Devices para gerenciar a lista de dispositivos nos quais seu aplicativo de desenvolvimento pode ser instalado.
- Clique no link App IDs para gerenciar seus IDs de aplicativo. O perfil de provisionamento é vinculado ao ID de aplicativo durante a criação do perfil.
- Clique no link Provisioning para gerenciar seus perfis de provisionamento. Você também pode utilizar o Development Provisioning Assistant para criar perfis de desenvolvimento de provisionamento.
- Clique no link Distribution quando desejar enviar seu aplicativo para a App Store ou criar uma versão Ad Hoc de seu aplicativo. Esta seção possui um link para o site iTunes Connect, no qual você pode enviar um aplicativo para a App Store.

Criação do aplicativo Hello World para iPhone com o Flash Professional CS5

Importante: Antes de criar o aplicativo, baixe os arquivos e os aplicativos de desenvolvedor necessários. Consulte “[Obtenção das ferramentas de desenvolvedor da Adobe](#)” na página 4 “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.

Criação de um projeto com o Flash Professional CS5

É possível criar um aplicativo para iPhone diretamente no Flash Professional CS5:

- 1 Abra o Flash CS5.
- 2 Selecione Arquivo > Novo.
- 3 Selecione iPhone OS.
- 4 Clique no botão OK.

Adicionar conteúdo ao aplicativo

Em seguida, adicione o texto “Hello world!” ao aplicativo:

- 1 Selecione a ferramenta Texto e clique em Palco.
- 2 Nas configurações de Propriedades para o campo de texto, selecione Texto clássico (não Texto TLF).
Esse é um aplicativo básico e Texto clássico é adequado. Para usar Texto TLF, é necessário aplicar outras configurações. Consulte “[Fontes e entrada de texto](#)” na página 39.
- 3 No campo Novo texto, digite “Hello World!”
- 4 Selecione o campo de texto com a ferramenta Seleção.
- 5 Em seguida, abra o inspector de propriedades, especifique as seguintes configurações:

- Caractere > Família: _sans
- Caractere >> Tamanho: 50
- Posição > X: 20
- Posição > Y: 20

- 6 Salve o arquivo.
- 7 Selecione Controle > Testar filme > No AIR Debug Launcher (Móvel).

O Flash Professional CS5 compila o conteúdo SWF e exibe a versão do aplicativo no ADL (AIR Debug Launcher). Isso permite que você obtenha uma rápida visualização do aplicativo.

Criar ilustração do ícone arte e ilustração da tela inicial do aplicativo

Todos os aplicativos para iPhone têm ícones que aparecem na interface de usuário do aplicativo iTunes e na tela do iPhone.

- 1 Crie um diretório no diretório de seu projeto e nomeie-o “ícones”.
- 2 Crie três arquivos PNG no diretório “ícones”. Nomeie os arquivos Icon29.png, Icon57.png e Icon512.png.
- 3 Edite os arquivos PNG para criar a arte adequada ao seu aplicativo. Os arquivos devem ter 29 por 29 pixels, 57 por 57 pixels e 512 por 512 pixels. Neste teste, use simplesmente quadrados de cor sólida como arte.

Todos os aplicativos para iPhone exibem uma imagem inicial quando o aplicativo carrega no iPhone. Defina a imagem inicial em um arquivo PNG:

- 1 No diretório principal de desenvolvimento, crie um arquivo PNG com o nome Default.png. (Não coloque esse arquivo no subdiretório “ícones”. Certifique-se de nomear o arquivo Default.png com D maiúsculo).
- 2 Edite o arquivo para 320 pixels de largura e 480 pixels de altura. Por enquanto, o conteúdo pode ser retângulo branco liso. (Você mudará isso mais tarde).

Nota: Ao submeter um aplicativo à Apple App Store, use a versão JPG (não a versão PNG) do arquivo com 512 pixels. Use a versão PNG ao testar as versões de desenvolvimento de um aplicativo.

Para obter informações detalhadas desses gráficos, consulte “[Ícone do iPhone e imagens de tela iniciais](#)” na página 13.

Editar as configurações do aplicativo

Importante: Se ainda não fez isso, baixe os arquivos e os aplicativos de desenvolvedor necessários ao desenvolvimento de iPhone. Consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.

Na caixa de diálogo Configurações do iPhone do Flash Professional CS5, defina as propriedades básicas do aplicativo para iPhone.

1 Escolha Arquivo > Configurações do iPhone OS.

2 Na guia Geral, especifique as seguintes configurações:

- Arquivo de saída: HelloWorld.ipa
Esse é o nome do arquivo do instalador do iPhone que será criado.
- Nome do aplicativo: Hello World
Esse é o nome do aplicativo exibido no ícone do aplicativo no iPhone.

- Versão: 1.0

Versão do aplicativo.

- Proporção do aspecto: retrato
- Tela cheia: selecionada.
- Orientação automática: deselegionada.
- Renderização: CPU

As outras opções, GPU e Automático, usam aceleração por hardware para renderização. Este recurso pode ajudar a melhorar o desempenho de aplicativos com uso intensivo de gráficos (por exemplo, jogos) que foram desenvolvidos para utilizar os recursos de aceleração de hardware. Para obter mais informações, consulte “[Aceleração por hardware](#)” na página 36.

- Arquivos incluídos: Adicione o arquivo de arte de tela inicial (Default.png) na lista de Arquivos incluídos.

Nota: Neste exemplo do Hello World, não modifique as configurações fornecidas nessas instruções. Algumas configurações, como a configuração Versão, têm restrições específicas. Essas restrições são descritas em “[Configurações de aplicativos para iPhone](#)” na página 15.

3 Na guia Implantação, especifique as seguintes configurações:

- Certificado: Procure e selecione o certificado .p12 com base no certificado de desenvolvedor obtido na Apple. Esse certificado é usado para assinar o arquivo. É necessário converter o certificado Apple iPhone no formato .p12. Para obter mais informações, consulte “[Obtenção das ferramentas de desenvolvedor da Adobe](#)” na página 4.
- Senha: Digite a senha do certificado.
- Arquivo de provisionamento: Procure e selecione o arquivo de provisionamento de desenvolvedor obtido na Apple. Consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.
- ID do aplicativo: Se esse campo é selecionável, insira o ID do aplicativo que coincida com o ID do aplicativo que você forneceu à Apple (como com.example.as3.HelloWorld).

O ID do aplicativo identifica exclusivamente seu aplicativo.

Se esse campo não é selecionável, então o arquivo de provisionamento está ligado a um ID do aplicativo específico. O ID do aplicativo é exibido no campo.

Para obter detalhes sobre como especificar o ID do aplicativo, consulte a seção “Guia Implantação” de [“Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5”](#) na página 15.

- 4 Na guia Ícones, clique no ícone 29 x 29, na lista Ícones. Em seguida, especifique o local do arquivo PNG com 29 x 29 pixels que criou anteriormente (consulte [“Criar ilustração do ícone arte e ilustração da tela inicial do aplicativo”](#) na página 9). Em seguida, especifique os arquivos PNG para o ícone com 57 x 57 pixels e o ícone com 512 x 512 pixels.
- 5 Clique no botão OK.
- 6 Salve o arquivo.

Para obter detalhes das configurações de aplicativos, consulte [“Configurações de aplicativos para iPhone”](#) na página 15.

Compilação do arquivo IPA

Agora é possível compilar o arquivo do instalador do IPA:

- 1 Selecione Arquivo > Publicar.
- 2 Na caixa de diálogo Configurações do iPhone, clique no botão OK.

O Packager for iPhone cria o arquivo do instalador do aplicativo para iPhone, arquivo HelloWorld.ipa, no diretório do projeto. A compilação do arquivo IPA pode demorar alguns minutos.

Instalar o aplicativo no iPhone

Para instalar o aplicativo no iPhone para testá-lo no iPhone:

- 1 Abra o aplicativo iTunes.
- 2 Se você ainda não fez isso, adicione o perfil de provisionamento desse aplicativo ao iTunes. No iTunes, selecione Arquivo > Adicionar arquivo à Biblioteca. Em seguida, selecione o arquivo de perfil de provisionamento (que tem mobileprovision como tipo de arquivo).

Por enquanto, para testar o aplicativo em seu iPhone do desenvolvedor, use o perfil de provisionamento de desenvolvimento.

Mais tarde, ao distribuir o aplicativo para a iTunes Store, use o perfil de distribuição. Para distribuir o aplicativo ad hoc (para múltiplos dispositivos sem passar pela iTunes Store), use o arquivo de provisionamento ad hoc.

Para obter mais informações sobre perfis de provisionamento, consulte [“Obtenção dos arquivos de desenvolver da Apple”](#) na página 4.

- 3 Algumas versões do iTunes não substituem o aplicativo, se a mesma versão do aplicativo estiver instalada. Nesse caso, exclua o aplicativo do seu dispositivo e da lista de aplicativos no iTunes.
- 4 Clique duas vezes no arquivo IPA do aplicativo. O arquivo deve aparecer na lista de aplicativos.
- 5 Conecte seu iPhone à porta USB em seu computador.
- 6 No iTunes, selecione a guia Aplicativos do dispositivo e certifique-se de que o aplicativo está selecionado na lista de aplicativos para instalação.

7 Selecione o dispositivo na lista à esquerda do aplicativo iTunes. Em seguida, clique no botão Sync. Quando a sincronização terminar, o aplicativo Hello World aparecerá em seu iPhone.

Se a nova versão não estiver instalada, exclua-a do iPhone e da lista de aplicativos no iTunes e, em seguida, execute novamente esse procedimento. Talvez esse seja o caso, se a versão instalada atualmente usa a mesma ID do aplicativo e versão.

Se o iTunes exibir um erro durante a instalação do aplicativo, consulte “Solução de problemas de instalação de aplicativos” in “[Instalar um aplicativo iPhone](#)” na página 23.

Editar o gráfico da tela inicial

Antes de compilar o aplicativo, você criou o arquivo Default.png (consulte “[Criar ilustração do ícone arte e ilustração da tela inicial do aplicativo](#)” na página 9). Esse arquivo PNG atua como a imagem de inicialização quando o aplicativo é carregado. Quando você testou o aplicativo em seu iPhone, talvez tenha observado a tela branca na inicialização.

Altere essa imagem para coincidir com a tela de inicialização de seu aplicativo (“Hello World!”):

- 1 Abra o aplicativo em seu dispositivo. Quando o primeiro texto “Hello World” aparecer, pressione e mantenha pressionado o botão Início (abaixo da tela). Mantendo pressionado o botão Início, pressione o botão Despertar/Repousar (na parte superior do iPhone). Isso executa uma captura de tela e envia-a para o rolo da câmera.
- 2 Transfira a imagem para seu computador de desenvolvimento, transferindo as fotos do iPhoto ou de outro aplicativo de transferência de fotos. (No Mac OS, é possível também usar o aplicativo Captura de Imagem).
É possível também enviar a foto por email para seu computador de desenvolvimento:
 - Abra o aplicativo Fotos.
 - Abra Rolo da câmera.
 - Abra a imagem da captura de tela que capturou.
 - Pressione a imagem e, em seguida, pressione o botão (seta) “Avançar” no canto inferior esquerdo. Em seguida, clique no botão Enviar foto por email e envie a imagem para você mesmo.
- 3 Substitua o arquivo Default.png (no seu diretório de desenvolvimento) pela versão PNG da imagem da captura de tela.
- 4 Compile novamente o aplicativo, (consulte “[Compilação do arquivo IPA](#)” na página 11) e reinstale-o em seu iPhone.

O aplicativo usa a nova tela de inicialização ao carregar.

Nota: É possível criar qualquer arte que deseja para o arquivo Default.png, desde que tenha as dimensões corretas (320 x 480 pixels). No entanto, frequentemente é melhor que a imagem do arquivo Default.png coincida com o estado inicial do aplicativo.

Capítulo 2: Compilação e depuração de aplicativos para iPhone

É possível compilar um aplicativo para iPhone utilizando o Packager for iPhone. O Packager for iPhone está incluído no Flash Professional CS5.

É possível depurar o aplicativo no computador de desenvolvimento. É possível também instalar a versão de depuração no iPhone e receber a saída `trace()` no Flash Professional CS5.

Para ver o tutorial sobre como criar um aplicativo para iPhone do início ao fim, consulte, “[Criação do aplicativo Hello World para iPhone com o Flash Professional CS5](#)” na página 8.

Ícone do iPhone e imagens de tela iniciais

Todos os aplicativos para iPhone têm ícones que aparecem na interface de usuário do aplicativo iTunes e no iPhone.

Ícones de aplicativos para iPhone

Defina os seguintes ícones de aplicativos para um aplicativo iPhone:

- Um ícone de 29 por 29 pixels — os resultados de pesquisa Spotlight no iPhone e no iPod touch utilizam esse ícone.
- Um ícone de 48 por 48 pixels — os resultados da pesquisa Spotlight no iPad utilizam esse ícone.
- Um ícone de 57 por 57 pixels — as telas iniciais do iPhone e do iPod touch exibem esse ícone.
- Um ícone de 72 por 72 pixels (opcional)— a tela inicial do iPad exibe esse ícone.
- Ícone 512 x 512 pixels — o iTunes exibe este ícone. O arquivo de 512 pixels PNG é utilizado somente para testar versões de desenvolvimento do aplicativo. Ao enviar o aplicativo final para a Apple App Store, envie a imagem de 512 pixels separadamente, como arquivo JPG. Isso não está incluído no IPA.

No Flash Professional CS5, adicione esses ícones à guia Ícones na caixa de diálogo Configurações do iPhone. “[Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5](#)” na página 15.

Você também pode adicionar os locais dos ícones no arquivo descritor do aplicativo:

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

O iPhone adiciona o efeito de brilho ao ícone. Não é necessário incluí-lo na imagem de origem. Para remover o efeito de brilho padrão, adicione o seguinte ao elemento `InfoAdditions` no arquivo do descritor do aplicativo:

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


Consulte [“Definição de propriedades de aplicativo do iPhone no arquivo do descritor do aplicativo”](#) na página 17.

Tela inicial arte (Default.png)

Todos os aplicativos para iPhone exibem uma imagem inicial quando o aplicativo carrega no iPhone. Defina a imagem inicial em um arquivo PNG nomeado Default.png. No diretório principal de desenvolvimento, crie um arquivo PNG com o nome Default.png. (Não coloque esse arquivo em um subdiretório. Certifique-se de nomear o arquivo Default.png com D maiúsculo).

O arquivo Default.png possui 320 pixels de largura por 480 pixels de altura, independentemente da orientação inicial do aplicativo ou se ele é tela cheia ou não.

Se a orientação inicial de seu aplicativo for do tipo paisagem, utilize as mesmas dimensões do aplicativo do tipo retrato: 320 pixels de largura por 480 pixels de altura. No entanto, gire a arte 90° no sentido anti-horário no arquivo PNG. O lado esquerdo da arte PNG corresponde à parte superior a tela do iPhone no modo paisagem. (Para obter mais informações sobre como configurar a orientação inicial do aplicativo, consulte [“Configurações de aplicativos para iPhone”](#) na página 15).

Para aplicativos que não sejam do tipo tela cheia, os 20 pixels superiores da arte da imagem padrão são ignorados. O iPhone exibe sua barra de status no retângulo de 20 pixels de largura na parte superior da imagem padrão. Em uma orientação de paisagem, esta região corresponde ao retângulo esquerdo de 20 pixels de largura do arquivo Default.png (exibido na parte superior do modo paisagem). Em um aplicativo de orientação de retrato, esta região corresponde ao retângulo superior de 20 pixels do arquivo Default.png.

Na maioria dos aplicativos, a imagem Default.png deveria coincidir com a tela de inicialização do aplicativo: Para realizar uma captura da tela de inicialização do aplicativo:

- 1 Abra o aplicativo no iPhone. Quando a primeira tela da interface do usuário aparecer, pressione e mantenha pressionado o botão Início (abaixo da tela). Mantendo pressionado o botão Início, pressione o botão Despertar/Repousar (na parte superior do dispositivo). Isso executa uma captura de tela e envia-a para o rolo da câmera.
- 2 Transfira a imagem para seu computador de desenvolvimento, transferindo as fotos do iPhoto ou de outro aplicativo de transferência de fotos. (No Mac OS, é possível também usar o aplicativo Captura de Imagem).

É possível também enviar a foto por email para seu computador de desenvolvimento:

- Abra o aplicativo Fotos.
- Abra Rolo da câmera.
- Abra a imagem da captura de tela que capturou.
- Pressione a imagem e, em seguida, pressione o botão (seta) “Avançar” no canto inferior esquerdo. Em seguida, clique no botão Enviar foto por email e envie a imagem para você mesmo.

Nota: É possível criar qualquer arte que desejar para o arquivo Default.png, desde que ela tenha as dimensões corretas. No entanto, frequentemente é melhor que a imagem do arquivo Default.png coincida com o estado inicial do aplicativo.

Não inclua texto na imagem Default.png, se o aplicativo estiver localizado em múltiplos idiomas. Default.png é estático e o texto não coincidiria com outros idiomas.

No Flash Professional CS5, certifique-se de adicionar o arquivo Default.png na lista de Arquivos incluídos na caixa de diálogo de configurações do iPhone. [“Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5”](#) na página 15.

Ao compilar utilizando o aplicativo PFI na linha de comando, certifique-se de criar uma referência a esse arquivos na lista de ativos incluídos. Consulte [“Criação do arquivo do instalador do aplicativo para iPhone na linha de comando”](#) na página 21.

Configurações de aplicativos para iPhone

As configurações do aplicativo incluem:

- Nome do aplicativo
- Nome do arquivo IPA
- Versão do aplicativo
- Orientação da tela inicial do aplicativo e se a orientação da tela gira automaticamente quando o iPhone é girado
- Se a exibição inicial é em tela cheia ou não
- Informações sobre ícones dos aplicativos
- Informações sobre aceleração por hardware

É possível editar as configurações de aplicativos no Flash Professional CS5.

Também é possível editar essas configurações diretamente no arquivo do descritor do aplicativo. O arquivo do descritor do aplicativo é um arquivo XML que contém as configurações para o aplicativo.

Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5

A caixa de diálogo Configurações do iPhone do Flash Professional CS5 permite que você defina as propriedades básicas do aplicativo para iPhone.

Para abrir a caixa de diálogo Configurações do iPhone:

- ❖ Escolha Arquivo > Configurações do iPhone.

Guia Geral

A guia Geral inclui as seguintes configurações relacionadas ao iPhone:

- Arquivo de saída: Nome do aplicativo exibido no ícone do aplicativo no iPhone. Não inclua o caractere do sinal de adição (+) no nome de arquivo de saída.
- Nome do aplicativo: Nome do aplicativo exibido no ícone Aplicativos no iPhone. Não inclua o caractere do sinal de adição (+) no nome do aplicativo.
- Versão: Ajuda os usuários a verificar qual versão do aplicativo eles estão instalando. Esta versão é utilizada como CFBundleVersion do aplicativo para iPhone. O formato deve ser semelhante à nnnnn[.nn[.nn]], onde n é um dígito de 0 a 9 e os colchetes indicam componentes opcionais como 1, 1.0 ou 1.0.1. As versões do iPhone devem conter somente dígitos e casas decimais. As versões do iPhone podem conter até duas casas decimais.
- Proporção do aspecto: Proporção do aspecto inicial do aplicativo (retrato ou paisagem).
- Tela cheia: Se o aplicativo usa a tela cheia ou se é exibido na barra de status do iPhone.
- Orientação automática: Selecione esse aplicativo para reorientar o conteúdo da tela do aplicativo quando o iPhone é reorientado.

Ao utilizar a orientação automática, para obter melhores resultados, adicione o código do ActionScript para definir a propriedade `align` de palco para a seguinte:

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

- Renderização: A forma como os objetos de exibição são renderizados no iPhone:
 - CPU: O aplicativo usa a CPU para renderizar todos os objetos de exibição. Não se usa aceleração por hardware.
 - GPU: O aplicativo usa o iPhone GPU para compor bitmaps.
 - Automático: Esse recurso não foi implantado.

Para obter mais informações, consulte “[Aceleração por hardware](#)” na página 36.

- Arquivos incluídos: Adiciona todos os arquivos e diretórios ao pacote do aplicativo para iPhone. O arquivo SWF principal e o arquivo descritor do aplicativo são incluídos por padrão. Adicione quaisquer ativos necessários na lista Arquivos incluídos. Certifique-se de adicionar o arquivo de arte de tela inicial (Default.png) na lista de Arquivos incluídos.

Guia Implantação

A guia Implantação inclui configurações de assinatura e de compilação do aplicativo:

- Assinatura digital iPhone: Especifique o arquivo de certificado P12 e a senha do certificado. É necessário converter o certificado Apple iPhone no formato .p12. Para obter mais informações, consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.
- Arquivo de aprovisionamento: Aponta para o arquivo de aprovisionamento desse aplicativo, obtido na Apple. Para obter mais informações, consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.
- ID do aplicativo—O ID do aplicativo único que identifica o aplicativo. Se o arquivo de aprovisionamento estiver ligado a um ID de aplicativo específico, o Flash Professional CS5 define este campo e não é possível editá-lo. Do contrário, o perfil de aprovisionamento permite múltiplos IDs do aplicativo (caractere curinga). Fornece o ID do aplicativo que corresponde ao padrão do caractere curinga do ID do aplicativo que você forneceu à Apple:
 - Se o seu ID de aplicativo da Apple é com.myDomain.*, o ID de aplicativo na caixa de diálogo Configurações do iPhone deve começar com com.myDomain. (por exemplo: com.myDomain.myApp ou com.myDomain.app22).
 - Se o seu ID de aplicativo da Apple for *, o ID de aplicativo na caixa de diálogo Configurações do iPhone pode ser qualquer cadeia de caracteres válidos.

É possível localizar o ID de aplicativo Apple (ou padrão de ID de aplicativo curinga) associado com um perfil de aprovisionamento no iPhone Dev Center (<http://developer.apple.com/iphone>). Acesse a o iPhone Developer Program Portal e, em seguida, acesse a seção Provisioning.

Aviso importante: Ignore os caracteres no início do ID de aplicativo da Apple. A Apple chama essa cadeia de caracteres de ID da distribuição de conjunto. Por exemplo, se a Apple indicar seu ID de aplicativo como 96LPVWEASL.com.example.bob.myApp, ignore a parte 96LPVWEASL—use com.example.bob.myApp como ID do aplicativo. Se a Apple indicar seu ID de aplicativo como 5RM86Z4DJM.*, ignore a parte 5RM86Z4DJM—isto é um ID de aplicativo curinga.

- Tipo de implantação de iPhone:
 - Publicação rápida para testar dispositivo: Escolha essa opção para compilar rapidamente a versão do aplicativo para testar em seu iPhone do desenvolvedor.
 - Publicação rápida para depurar dispositivo: Escolha essa opção para compilar rapidamente a versão do aplicativo para testar em seu iPhone do desenvolvedor. Com essa opção, o depurador do Flash Professional CS5 pode receber a saída `trace()` do aplicativo para iPhone. (Consulte “[Depurando um aplicativo para iPhone](#)” na página 24).
 - Implantação - Ad Hoc: Escolha essa opção para criar o aplicativo para implantação ad hoc. Consulte o centro de desenvolvedores da Apple iPhone

- Implantação - Apple App Store: Escolha essa opção para criar a versão final do arquivo IPA para implantação na Apple App Store.

Guia Ícones

Na guia Ícones, especifique o local da imagem do ícone de 29 x 29 pixels, o da imagem do ícone de 48 x 48 pixels, o da imagem de 57 x 57 pixels, o da imagem de 72 x 72 pixels e o da imagem do ícone de 512 x 512 pixels. Consulte “[Ícone do iPhone e imagens de tela iniciais](#)” na página 13.

Nota: As opções de 48 x 48 pixels e 72 x 72 pixels não estão incluídas na versão do Packager for iPhone Preview incluída no Flash Professional CS5. No Flash Professional CS5, selecione Ajuda > Atualizações para adicionar essas opções.

Definição de propriedades de aplicativo do iPhone no arquivo do descritor do aplicativo

O arquivo do descritor do aplicativo é um arquivo XML que contém propriedades do aplicativo inteiro, tais como nome, versão, direitos autorais e outras configurações.

O Flash Professional CS5 cria o arquivo do descritor do aplicativo com base nas configurações na caixa de diálogo Configurações do iPhone. No entanto, é possível também editar o arquivo do descritor do aplicativo em um editor de texto. O Flash Professional nomeia o arquivo do descritor do aplicativo adicionando “-app.xml” ao nome do projeto. Por exemplo, o arquivo do descritor do aplicativo do projeto HelloWorld é nomeado HelloWorld-app.xml. Edite o arquivo do descritor do aplicativo se desejar definir as configurações não suportadas na caixa de diálogo Configurações do iPhone do Flash Professional CS5. Por exemplo, é possível definir o elemento `InfoAdditions` para definir as configurações de info.Plist para o aplicativo.

Aviso importante: Não edite o arquivo descritor do aplicativo enquanto a caixa de diálogo do Flash Professional CS5 estiver aberta. Salve as alterações do arquivo descritor do aplicativo antes de abrir a caixa de diálogo Configurações do iPhone.

Veja um exemplo do arquivo do descritor do aplicativo:

```
<?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>
```

Veja os detalhes das configurações nesse arquivo do descritor do aplicativo:

- No elemento `<application>`, é necessário “AIR 2.0 namespace” para criar aplicativos para iPhone:

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

- O elemento `<id>`:

`<id>com.example.as3.HelloWorld</id>` O ID do aplicativo identifica exclusivamente seu aplicativo. A forma recomendada é uma sequência de caractere separados por pontos, do tipo DNS reverso, como, por exemplo, "com.company.AppName". O compilador usa esse valor como ID de conjunto para o aplicativo para iPhone.

Se o arquivo de provisionamento estiver ligado a um ID de aplicativo específico, utilize o ID de aplicativo deste elemento. Ignore os caracteres que a Apple atribui no início de do ID de aplicativo Apple (conhecido como ID da distribuição de conjunto). Por exemplo, se o ID de aplicativo do perfil de provisionamento for 96LPVWEASL.com.example.bob.myApp, utilize com.example.bob.myApp como ID de aplicativo no arquivo descritor do aplicativo.

Se o perfil de provisionamento permitir múltiplos IDs de aplicativo (caractere curinga), o ID do aplicativo terminará em asterisco (exemplo: 5RM86Z4DJM.*). Forneça um ID de aplicativo que corresponda ao padrão do caractere curinga do ID do aplicativo que você forneceu à Apple:

- Se o seu ID de aplicativo da Apple é com.myDomain.*, o ID de aplicativo no arquivo descritor de aplicativo deve iniciar com com.myDomain. É possível especificar um ID de aplicativo como, por exemplo, com.myDomain.myApp ou com.myDomain.app22.

- Se o seu ID de aplicativo da Apple for *, o ID de aplicativo do arquivo descritor pode ser qualquer cadeia de caracteres válidos.

É possível localizar o ID de aplicativo Apple (ou padrão de ID de aplicativo curinga) associado com um perfil de provisionamento no iPhone Dev Center (<http://developer.apple.com/iphone>). Acesse a o iPhone Developer Program Portal e, em seguida, acesse a seção Provisioning.

Aviso importante: Ignore os caracteres no início do ID de aplicativo da Apple. A Apple chama essa cadeia de caracteres de ID da distribuição de conjunto. Por exemplo, se a Apple indicar seu ID de aplicativo como 5RM86Z4DJM.*, ignore a parte 5RM86Z4DJM—isto é um ID de aplicativo curinga. Se a Apple indicar seu ID de aplicativo como 96LPVWEASL.com.example.bob.myApp, ignore a parte 96LPVWEASL—use com.example.bob.myApp como ID do aplicativo.

- O elemento `<filename>`:
`<filename>HelloWorld</filename>` Nome utilizado no arquivo do instalador do iPhone. Não inclua o caractere do sinal de adição (+) no nome de arquivo.
- O elemento `<name>`:
`<name>Hello World</name>` Nome do aplicativo exibido no aplicativo iTunes e no iPhone. Não inclua o caractere do sinal de adição (+) no nome.
- O elemento `<version>`:
`<version>1.0</version>` Ajuda os usuários a determinar qual versão do aplicativo eles estão instalando. Esta versão é utilizada como `CFBundleVersion` do aplicativo para iPhone. O formato deve ser semelhante à `nnnnn[.nn[.nn]]`, onde `n` é um dígito de 0 a 9 e os colchetes indicam componentes opcionais como 1, 1.0 ou 1.0.1. As versões do iPhone devem conter somente dígitos e casas decimais. As versões do iPhone podem conter até duas casas decimais.
- O elemento `<initialWindow>` contém os seguintes elementos filho para especificar as propriedades da aparência inicial do aplicativo:
`<content>HelloWorld.swf</content>` Identifica o arquivo SWF raiz a fim de compilar para o aplicativo para iPhone.
`<visible>true</visible>` Essa configuração é necessária.
`<fullScreen>true</fullScreen>` Especifica se o aplicativo usa a tela inteira do iPhone.
`<aspectRatio>portrait</aspectRatio>` Especifica se a proporção do aspecto inicial do aplicativo está no modo retrato (em vez de paisagem). O arquivo `Default.png` usado para definir a janela inicial do aplicativo deve ter 320 pixels de largura e 480 pixels de altura, independentemente dessa configuração. (Consulte “Ícone do iPhone e imagens de tela iniciais” na página 13.)
`<autoOrients>true</autoOrients>` (Opcional) Especifica se a orientação do conteúdo no aplicativo reorienta automaticamente quando o próprio dispositivo altera a orientação física. O valor padrão é `true`. É possível cancelar a orientação automática chamando o método `preventDefault()` de um evento `orientationChanging`, despachado pelo objeto `Palco`. Para obter mais informações, consulte [Definindo e detectando a orientação da tela](#).
Ao utilizar orientação automática, para obter melhores resultados, defina a propriedade `align` de `palco` para a seguinte:

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


`<renderMode>gpu</renderMode>` (Opcional) Modo de renderização que o aplicativo utiliza. Há três configurações possíveis:
 - `cpu`: O aplicativo usa a CPU para renderizar todos os objetos de exibição. Não se usa aceleração por hardware.

- `gpu`: O aplicativo usa o iPhone GPU para compor bitmaps.
- Automático: Esse recurso não foi implantado.

Para obter mais informações, consulte [“Aceleração por hardware”](#) na página 36.

- O elemento `<profiles>`:

`<profiles>mobileDevice</profiles>` Limita o aplicativo que será compilado para o perfil de dispositivo portátil. Atualmente, esse perfil suporta apenas aplicativos para iPhone. Há três perfis suportados:

- `desktop`—Aplicativo AIR.
- `extendedDesktop`: Aplicativo AIR com suporte a processo nativo API.
- `mobileDevice`: Aplicativo AIR para dispositivo portátil. Atualmente, o iPhone é o único dispositivo portátil suportado.

Limitar o aplicativo a um perfil específico impede que ele seja compilado para outros perfis. Se especificar nenhum perfil, então é possível compilar um aplicativo para qualquer perfil. É possível especificar múltiplos perfis, listando-os, separados por espaços, no elemento `<profiles>`.

Certifique-se de incluir `mobileDevice` como perfil suportado (ou deixe o elemento `<profiles>` vazio).

- O elemento `<icon>` contém os seguintes elementos filho para especificar os ícones que o aplicativo utiliza:

`<image29x29>icons/icon29.png</image29x29>` Essa é a imagem usada em resultados da pesquisa Spotlight.

`<image48x48>icons/icon48.png</image48x48>` Esta é a imagem usada em resultados da pesquisa Spotlight no iPad.

`<image57x57>icons/icon57.png</image57x57>` Esta é a imagem usada na tela inicial do iPhone e do iPod Touch.

`<image72x72>icons/icon72.png</image72x72>` Esta é a imagem usada na tela inicial do iPad.

`<image512x512>icons/icon512.png</image512x512>` Essa é a imagem usada no aplicativo iTunes.

A ferramenta Packager for iPhone usa ícones 29, 57 e 512 indicados no arquivo do descritor do aplicativo. A ferramenta copia os ícones para arquivos `Icon-Small.png`, `Icon.png` e `iTunesArtwork`, respectivamente. Para evitar fazer a cópia, é possível empacotar os arquivos diretamente. Empacote os arquivos diretamente, colocando-os no diretório que contém o arquivo do descritor do aplicativo e liste os nomes e os caminhos corretos.

A imagem com 512 pixels destina-se a testes internos apenas. Ao submeter um aplicativo à Apple, submeta a imagem com 512 pixels separadamente. Isso não está incluído no IPA. Especifique-o para certificar-se de que a imagem com 512 pixels tem boa aparência no iTunes antes de submetê-la.

- O elemento `<iPhone>` contém os seguintes elementos filho para especificar as configurações específicas do iPhone:

`<InfoAdditions></InfoAdditions>` contém os elementos filho, especificando os pares com valor-chave para usar como configurações de `Info.plist` para o aplicativo:

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

Nesse exemplo, os valores definem o estilo da barra de status do aplicativo e determinam que o aplicativo não requer acesso WiFi contínuo.

As configurações de `InfoAdditions` são anexadas a uma tag `CDATA`.

Para o suporte ao iPad, inclua as configurações de valor chave de `UIDeviceFamily`. A configuração `UIDeviceFamily` é uma matriz de sequências de caracteres. Cada sequência de caracteres define os dispositivos suportados. O parâmetro `<string>1</string>` define o suporte para iPhone e iPod Touch. A configuração `<string>2</string>` define o suporte ao iPad. A configuração `<string>3</string>` define o suporte ao tvOS. Se você especificar somente uma dessas sequências de caracteres, somente aquela família de dispositivos será suportada. Por exemplo, a configuração a seguir limita o suporte ao iPad:

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

Os conjuntos a seguir suportam as duas famílias de dispositivos (iPhone/iPod Touch e iPad):

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

Para obter mais informações sobre outras configurações de Info.plist, consulte a documentação do desenvolvedor da Apple.

Compilando um IPA (arquivo do instalador do aplicativo)

Utilize o Packager for iPhone para compilar o aplicativo do ActionScript-3.0 para um arquivo do instalador do IPA.

Criar um arquivo de instalador de aplicativo para iPhone utilizando o Packager for iPhone incluído no Flash Professional CS5

Para incluir o Packager for iPhone incluído no Flash Professional CS5:

- 1 Selecione Arquivo > Publicar.
- 2 Na caixa de diálogo Configurações do iPhone, certifique-se de que forneceu os valores para todas as configurações. Certifique-se de que selecionou a opção correta na guia Implantação. Consulte [“Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5”](#) na página 15.
- 3 Clique no botão Publicar.

O Packager for iPhone cria o IPA (arquivo do instalador do aplicativo) do iPhone. A compilação do arquivo IPA pode demorar alguns minutos.

Também é possível executar o Packager for iPhone a partir da linha de comando. Consulte [“Criação do arquivo do instalador do aplicativo para iPhone na linha de comando”](#) na página 21.

Criação do arquivo do instalador do aplicativo para iPhone na linha de comando

É possível executar o Packager for iPhone na linha de comando. O Packager for iPhone converte o bytecode do arquivo SWF e outros arquivos de origem em um aplicativo nativo para iPhone.

- 1 Abra o comando shell ou um terminal e navegue para a pasta de projetos do aplicativo para iPhone.

2 Em seguida, utilize a ferramenta `pfi` para criar um arquivo IPA utilizando a sintaxe a seguir:

```
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
```

Altere a referência ao aplicativo `pfi` para incluir o caminho completo para o aplicativo `pfi`. O aplicativo `pfi` está instalado no subdiretório `pfi/bin` do diretório de instalação do Flash Professional CS5.

Selecione a opção `-target` correspondente ao tipo de aplicativo para iPhone que deseja criar:

- `-target ipa-test`: escolha essa opção para compilar rapidamente a versão do aplicativo para testar em seu iPhone do desenvolvedor.
- `-target ipa-debug`: Escolha essa opção para compilar a versão de depuração para testar em seu iPhone de desenvolvimento. Com essa opção, você pode utilizar uma sessão de depuração para receber a saída de `trace()` do aplicativo iPhone.

É possível incluir uma das seguintes opções `-connect` (`CONNECT_OPTIONS`) para especificar o endereço IP do computador de desenvolvimento que está executando o depurador:

- `-connect` — o aplicativo tentará se conectar a uma sessão de depuração no computador de desenvolvimento utilizado para compilar o aplicativo.
- `-connect IP_ADDRESS` — o aplicativo tentará se conectar a uma sessão de depuração no computador com o endereço IP especificado. Por exemplo:

```
-target ipa-debug -connect 192.0.32.10
```
- `-connect HOST_NAME` — o aplicativo tentará se conectar a uma sessão de depuração no computador com o nome de host especificado. Por exemplo:

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

Nota: A opção `-connect` não está incluída no *Packager for iPhone Preview* incluído no Flash Professional CS5. Atualize o *Packager for iPhone* selecionando *Ajuda > Atualizações no Flash Professional CS5*.

A opção `-connect` é facultativa. Caso não especificado, o aplicativo de depuração resultante não tentará se conectar a um depurador hospedado.

Se houver falha de uma tentativa de conexão de depuração, o aplicativo apresentará uma caixa de diálogo que solicitará que o usuário insira o endereço IP da máquina de host da depuração. Uma tentativa de conexão pode falhar se o dispositivo não estiver conectado ao wifi. Isto também pode ocorrer se o dispositivo estiver conectado mas não estiver protegido por um firewall da máquina de host de depuração.

Para obter mais informações, consulte “[Depurando um aplicativo para iPhone](#)” na página 24.

Também é possível incluir a opção `-renderingdiagnostics` para habilitar o recurso de diagnóstico de renderização por GPU. Para obter mais informações, consulte “[Depurando com diagnóstico de renderização de GPU](#)” em “[Depurando um aplicativo para iPhone](#)” na página 24.

- `-target ipa-ad-hoc`: Escolha essa opção para criar um aplicativo para implantação ad hoc. Consulte o centro de desenvolvedores da Apple iPhone.
- `-target ipa-app-store`: Escolha essa opção para criar a versão final do arquivo IPA para implantação na Apple App Store.

Substitua `PROFILE_PATH` pelo caminho para o arquivo de perfil de aprovisionamento de seu aplicativo. Para obter mais informações sobre perfis de aprovisionamento, consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.

Substitua `SIGNING_OPTIONS` para fazer referência ao certificado e senha de desenvolvedor de iPhone. Use a seguinte sintaxe:

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

Substitua *P12_FILE_PATH* pelo caminho para o arquivo de certificado P12. Substitua *PASSWORD* pela senha do certificado. (Veja o exemplo abaixo). Para obter mais informações sobre o arquivo de certificado P12, consulte “[Converter um certificado de desenvolvedor em um arquivo P12](#)” na página 7.

Substitua o *APP_DESCRIPTOR* para fazer referência ao arquivo do descritor do aplicativo.

Substitua *SOURCE_FILES* para fazer referência ao arquivo principal SWF de seu projeto seguido de outros ativos para inclusão. Certifique-se de incluir os caminhos para todos os arquivos de ícones definidos na caixa de diálogo de configurações do aplicativo no Flash CS5 ou em um arquivo personalizado do descritor do aplicativo. Além disso, inclua o arquivo Default.png da arte de tela inicial.

Considere o seguinte exemplo:

```
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"
```

O arquivo HelloWorld.ipa é compilado usando o seguinte:

- O certificado PKCS#12 específico usando a senha de certificado dfb7VKL19
- O arquivo do descritor do aplicativo HelloWorld-app.xml
- O arquivo de origem HelloWorld.swf
- Default.png específico e arquivos de ícones

O aplicativo pfi compila o aplicativo, com base no arquivo do descritor do aplicativo, arquivo SWF e outros ativos para um arquivo IPA.

No Mac OS, você pode utilizar um certificado armazenado na sequência de chaves, adicionando as opções a seguir ao comando pfi:

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

Substitua *ALIAS_NAME* pelo apelido do certificado que deseja utilizar. Ao apontar para um certificado armazenado na sequência de chaves do Mac, especifique o apelido em vez de apontar para o local do arquivo do certificado.

Instalar um aplicativo iPhone

Para instalar um aplicativo de desenvolvimento no iPhone, adicione o arquivo de aprovisionamento ao iPhone e, em seguida, instale o aplicativo no iPhone.

Adicionar um perfil de aprovisionamento ao iPhone

Para adicionar um perfil de aprovisionamento ao iPhone:

- 1 No iTunes, selecione Arquivo > Adicionar à Biblioteca. Selecione o arquivo de perfil de aprovisionamento (que tem mobileprovision como tipo de arquivo).

Certifique-se de que seu iPhone está adicionado ao perfil de aprovisionamento. Você pode gerenciar perfis de aprovisionamento no site Apple iPhone Dev Center (<http://developer.apple.com/iphone/>). Acesse a seção do iPhone Developer Program Portal no site. Clique no link Devices para gerenciar a lista de dispositivos nos quais seu aplicativo de desenvolvimento pode ser instalado. Clique no link Provisioning para gerenciar seus perfis de aprovisionamento.

2 Conecte seu iPhone ao computador e execute a sincronização.

Para obter mais informações sobre como obter um perfil de provisionamento, consulte [“Obtenção dos arquivos de desenvolver da Apple”](#) na página 4.

Instalar o aplicativo

Instale o aplicativo de desenvolvimento da mesma maneira que você instalaria qualquer arquivo IPA:

- 1 Caso possua uma versão anterior instalada do aplicativo, remova o aplicativo de seu dispositivo e da lista de aplicativos do iTunes.
- 2 Adicione o aplicativo no iTunes, utilizando uma das formas a seguir:
 - No menu Arquivo (no iTunes), selecione o comando Adicionar à biblioteca. Em seguida, selecione o arquivo IPA e clique no botão Abrir.
 - Clique duas vezes no arquivo do IPA.
 - Arraste o arquivo IPA para a biblioteca do iTunes.
- 3 Conecte seu iPhone à porta USB em seu computador.
- 4 No iTunes, selecione a guia Aplicativos do dispositivo e certifique-se de que o aplicativo está selecionado na lista de aplicativos para instalação.
- 5 Sincronize o iPhone.

Resolução de problemas de instalação de aplicativos

Se o iTunes exibir um erro durante a instalação de um aplicativo, verifique os itens a seguir:

- Certifique-se de que o ID do dispositivo está adicionado ao perfil de provisionamento.
- Para certificar-se de que o perfil de provisionamento está instalado, arraste-o para o iTunes ou utilize o comando Arquivo > Adicionar à biblioteca.

Verifique também se o ID do aplicativo corresponde ao ID de aplicativo da Apple:

- Se o seu ID de aplicativo da Apple é com.myDomain.*, o ID de aplicativo no arquivo descritor do aplicativo ou no ID de aplicativo na caixa de diálogo Configurações do iPhone deve começar com com.myDomain (como com.myDomain.anythinghere).
- Se o ID de aplicativo da Apple for com.myDomain.myApp, o ID do aplicativo no arquivo do descritor do aplicativo ou na interface de usuário do Flash Profession CS5 deve ser com.myDomain.myApp.
- Se o ID de aplicativo da Apple for *, o ID do aplicativo no arquivo do descritor do aplicativo ou na interface de usuário do Flash Profession CS5 pode ser qualquer um.

Defina o ID de aplicativo na caixa de diálogo Configurações do iTunes no Flash Professional CS5 ou no arquivo do descritor do aplicativo.

Depurando um aplicativo para iPhone

É possível depurar o aplicativo no computador do desenvolvimento com o aplicativo executando no ADL. Também é possível depurar o aplicativo no iPhone.

Algumas funcionalidades do AIR não suportadas no iPhone ainda estão disponíveis ao testar um aplicativo utilizando o ADL (no computador no desenvolvimento). Esteja ciente dessas diferenças ao testar o conteúdo na área de trabalho. Para obter mais informações, consulte [“ActionScript 3.0 APIs sem suporte em dispositivos móveis”](#) na página 28.

Depurando o aplicativo no computador do desenvolvimento

Para depurar o aplicativo no computador do desenvolvimento utilizando o Flash Professional CS5:

- ❖ Selecione Depurar > Depurar filme > Em AIR Debug Launcher (Móvel).

Também é possível depurar o aplicativo chamando o ADL na linha de comando. Esta é a sintaxe:

```
adl -profile mobileDevice appDescriptorFile
```

Substitua *appDescriptorFile* pelo caminho para o arquivo do descritor do aplicativo.

Certifique-se de incluir a opção `-profile mobileDevice`.

Depurando o aplicativo no iPhone

Para depurar o aplicativo no iPhone:

- 1 Compile o aplicativo com suporte à depuração:
 - No Flash Professional CS5, compile utilizando a opção “Publicação rápida para depuração no dispositivo”. (Consulte [“Criar um arquivo de instalador de aplicativo para iPhone utilizando o Packager for iPhone incluído no Flash Professional CS5”](#) na página 21.)
 - Utilizando o aplicativo de linha de comando PFI, compile o aplicativo com a opção `target ipa-debug`. (Consulte [“Criação do arquivo do instalador do aplicativo para iPhone na linha de comando”](#) na página 21.)
- 2 Instale o aplicativo no iPhone.
- 3 No iPhone, habilite o Wi-Fi e conecte na mesma rede do computador de desenvolvimento.
- 4 Inicie uma sessão de depuração no computador de desenvolvimento. No Flash Professional CS5, escolha Depurar > Iniciar sessão remota para depurar > ActionScript 3.0.
- 5 Execute o aplicativo no iPhone.

A versão de depuração do aplicativo solicitará o endereço IP do computador do desenvolvedor. Digite o endereço IP e pressione o botão OK. Para obter o endereço IP do computador de desenvolvimento.

- No Mac OS, no menu Apple, escolha Preferência de sistema. Na janela Preferências de sistema, clique no ícone Rede. A janela Preferências de rede lista o endereço IP.
- No Windows, inicie a sessão na linha de comando e execute o comando `ipconfig`.

A sessão de depuração do exibe qualquer saída `trace()` do aplicativo.

Ao fazer a depuração de um aplicativo instalado no iPhone, o Flash Professional CS5 possui suporte a todos os recursos, incluindo controle de ponto de interrupção, avanço pelas linhas e monitoramento de variável.

Depurando com o diagnóstico de renderização por GPU

O recurso de diagnóstico de renderização por GPU permite que você veja como o aplicativo utiliza a aceleração de hardware (para aplicativos que utilizam o modo de renderização por GPU). Para usar esse recurso, compile o aplicativo utilizando a ferramenta PFI na linha de comando e inclua a opção `-renderingdiagnostics`:

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

O indicador `-renderingdiagnostics` deve seguir diretamente o indicador `-package`.

O recurso de diagnóstico de renderização por GPU exibe retângulos coloridos para todos os objetos de exibição:

- Azul — o objeto de exibição não é um bitmap ou armazenado em cache como bitmap, e está sendo renderizado. Se for exibido azul repetidamente para um objeto de exibição que não estiver sendo alterado, isto pode ter ocorrido porque ele encontra objetos de exibição que se movimentam. Por exemplo, o objeto de exibição pode ser o plano de fundo de objetos de exibição que se movimentam. Considere armazenar o objeto de exibição em cache como bitmap.

Se for exibido azul para um objeto de exibição que você acha que deve ser armazenado em cache, pode ser porque o objeto está utilizando um efeito que a GPU não pode aplicar. Esses efeitos incluem alguns modos de mesclagem, transformações de cor, a propriedade `scrollRect` e máscaras.

O aplicativo também exibe azul se os objetos de exibição carregados para a GPU excederem os limites de memória.

O aplicativo registra as mensagens de cada retângulo azul. O aplicativo fornece essas mensagens juntamente com outras mensagens de saída de depuração e `trace()`.

- Verde — O objeto de exibição é um bitmap ou armazenado em cache como bitmap e está sendo carregado para a GPU pela primeira vez.

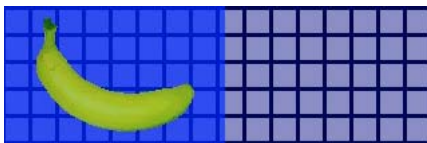
Se verde for exibido repetidamente para um objeto de exibição, o código do aplicativo está recriando o objeto de exibição. Por exemplo, isto pode ocorrer se a linha de tempo retornar para um quadro que cria o objeto de exibição. Considere modificar o conteúdo para evitar a recriação de objetos idênticos.

- Vermelho — o objeto de exibição é um bitmap ou armazenado em cache como bitmap, e está sendo recarregado para a GPU.

Vermelho é exibido toda vez que um objeto de exibição muda de forma a exigir que o aplicativo renderize novamente a representação de bitmap. Por exemplo, se um objeto 2D não possui a propriedade `cacheAsBitmapMatrix` definida, ele é renderizado novamente quando for redimensionado ou girado. Também é executada uma nova renderização quando os objetos de exibição filhos forem movidos ou alterados.

Cada retângulo colorido esmaece depois de quatro ciclos de redesenho de tela, desde que a coloração não ocorra novamente durante esses ciclos. Entretanto, se não houver mudanças na tela, a coloração de diagnóstico não mudará.

Por exemplo, considere um objeto de exibição de bitmap (uma banana) em frente a um segundo plano vetorial não armazenado em cache como bitmap. Quando a banana for renderizada, sua cor será verde. Quando o plano de fundo for renderizado pela primeira vez, sua cor será azul:

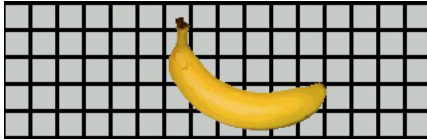


Conforme a banana se mover, a CPU precisará renderizar novamente o plano de fundo, fazendo com que a sombra azul seja exibida sobre o fundo:



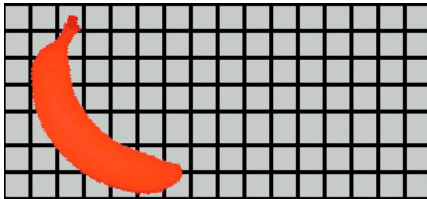
A sombra azul sobre o fundo reflete as regiões redesenhadas que precisam ser enviadas para a GPU.

Entretanto, se o fundo for armazenado em cache como bitmap, quando a banana se mover, o recurso de diagnóstico de renderização não exibirá cores:



O recurso de diagnóstico não exibirá cores porque a GPU reterá o bitmap do fundo. A GPU pode compor a banana com o fundo sem envolver a CPU.

Suponha que a banana seja um objeto de exibição 2D que não tenha a propriedade `cacheAsBitmapMatrix` definida. Sempre que o objeto de exibição for girado (ou redimensionado), o recurso de diagnóstico de renderização mostrará vermelho. Isto indica que o aplicativo tem de carregar uma nova versão do objeto de exibição para a GPU:



Submetendo seu aplicativo iPhone à App Store

Para submeter seu aplicativo à App Store:

- 1 Obtenha um certificado de distribuição e perfil de provisionamento no site iPhone Dev Center (<http://developer.apple.com/iphone/>).

O certificado de distribuição é nomeado “iPhone Developer: XXX.cer,” onde XXX é seu nome.

Para obter mais informações, consulte “[Obtenção dos arquivos de desenvolver da Apple](#)” na página 4.

- 2 Converta o certificado de distribuição em um arquivo P12.

Para obter mais informações, consulte “[Converter um certificado de desenvolvedor em um arquivo P12](#)” na página 7.

- 3 Compile seu aplicativo utilizando o arquivo P12 e o perfil de provisionamento.

Utilize o arquivo P12 criado com base no certificado de distribuição. Utilize o ID do aplicativo associado ao perfil de provisionamento de distribuição.

Para obter mais informações, consulte “[Compilando um IPA \(arquivo do instalador do aplicativo\)](#)” na página 21.

- 4 Submeta o aplicativo no site iPhone Dev Center (<http://developer.apple.com/iphone/>).

Importante: Apple exige que você use o programa Application Loader para carregar aplicativos para a App Store. A Apple somente publica o Application Loader para Mac OS X. Assim, enquanto você pode desenvolver um aplicativo do AIR para o iPhone usando um computador com Windows, você deve ter acesso a um computador com Mac OS X (versão 10.5.3 ou posterior) para enviar o aplicativo para a App Store. Você pode obter o programa Application Loader no site iOS Dev Center da Apple.

Capítulo 3: Suporte ActionScript 3.0 API para dispositivos móveis

Ao criar aplicativos AIR móveis, é possível usar os mesmos arquivos API do ActionScript 3.0 disponíveis para desenvolver aplicativos Adobe Flash Player 10.1 e AIR 2. No entanto, há adições e exceções.

ActionScript 3.0 APIs sem suporte em dispositivos móveis

Algumas APIs ActionScript 3.0 não estão disponíveis para aplicativos que executam em perfil de dispositivo móvel (como aplicativos que executam no iPhone).

Ao usar o mesmo código do ActionScript para desenvolver múltiplos perfis (como área de trabalho e móvel), use o código para testar se o API é suportado. Por exemplo, a classe `NativeWindow` não é suportada nos aplicativos para iPhone. (Os aplicativos para iPhone não podem usar ou criar janelas nativas). Para testar se o aplicativo está executando em um perfil que suporta janelas nativas (como o perfil da área de trabalho), selecione a propriedade `NativeWindow.isSupported`.

A tabela abaixo lista APIs que não possuem suporte no perfil portátil. Lista também as propriedades que você seleciona para verificar quando um aplicativo está em execução em uma plataforma que oferece suporte ao arquivo API.

API	Teste para suporte
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

API	Teste para suporte
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
Suporte a arquivo PDF	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

Não é possível criar aplicativos AIR com base em HTML e JavaScript no perfil do dispositivo móvel.

Algumas classes do ActionScript 3.0 são apenas suportadas parcialmente:

Arquivo

Os aplicativos para iPhone somente têm acesso ao diretório do aplicativo e ao diretório de armazenamento do aplicativo. Também é possível chamar `File.createTempFile()` e `File.createTempDirectory()`. Chamar uma operação para acessar outro diretório (como o método de leitura ou gravação `FileStream`) resulta na exceção `IOError`.

Os aplicativos para iPhone não suportam caixas de diálogo do navegador do arquivo nativo, como aquelas que o método `File.browseForOpen()` oferece.

Carregador

No aplicativo para iPhone, não é possível utilizar o método `Loader.load()`. No entanto, não é possível executar o código do ActionScript no conteúdo carregado com o método `Loader.load()`. No entanto, é possível utilizar os ativos no arquivo SWF (tais como clipes de filmes, imagens, fontes e sons na biblioteca). É possível utilizar o método `Loader.load()` para carregar arquivos de imagem.

Vídeo

Somente os vídeos Sorensen e ON2 VP6 são suportados em um aplicativo AIR para iPhone.

É possível usar o método `navigateToURL()` para abrir um vídeo H.264 fora do aplicativo. Como o parâmetro `request`, passe o objeto `URLRequest` com URL apontando para o vídeo. O vídeo é aberto no leitor de vídeo do iPhone.

Campos de texto

Existem limitações para fontes e outras configurações de campos de texto no iPhone. Consulte [“Fontes e entrada de texto”](#) na página 39.

APIs não suportados e depuração utilizando o ADL

Algumas funcionalidades do AIR não suportadas no iPhone ainda estão disponíveis ao testar um aplicativo utilizando o ADL (no computador no desenvolvimento). Esteja ciente dessas diferenças ao testar o conteúdo utilizando o ADL.

Essa funcionalidade inclui os seguintes codecs de vídeo e áudio: Speex (áudio), H.264/AVC (vídeo) e AAC (áudio). Esses códigos não estão disponíveis para aplicativos AIR executando no iPhone. No entanto, os códigos continuam a funcionar normalmente na área de trabalho.

O suporte à accessibility e à screen reader funciona no ADL no Windows. No entanto, esses APIs não são suportados no iPhone.

O protocolo RTMPE funciona normalmente quando utilizado no ADL na área de trabalho. No entanto, uma NetConnection que tenta conectar utilizando o protocolo RTMPE não é bem-sucedida no iPhone.

A classe loader funciona sem restrições adicionais quando o conteúdo é executado com o ADL. No entanto, ao executar no iPhone, as tentativas para carregar o conteúdo SWF que contém o bytecode do ActionScript resultam em uma mensagem de erro.

As ocorrências de sombreadores executam no ADL. No entanto, o bytecode do pixel bender do iPhone não é interpretado e as ocorrências de sombreadores não têm nenhum efeito gráfico.

Para obter mais informações, consulte [“Depurando um aplicativo para iPhone”](#) na página 24.

ActionScript APIs específica para aplicativos AIR móveis

As APIs a seguir somente estão disponíveis nos aplicativos AIR de dispositivos móveis. Elas não são funcionais na versão de estação de trabalho do Flash Player ou AIR.

API da orientação da tela

O arquivo API da orientação da tela permite que você trabalhe com a orientação do palco e do iPhone:

- `Stage.autoOrients`: o aplicativo é definido para reorientar automaticamente o palco quando o dispositivo é girado. Essa propriedade é definida em `true` quando a opção Orientação automática é selecionada na caixa de diálogo Configurações do iPhone do Flash Professional CS5. (Também é possível definir o elemento `autoOrients` em `true` no arquivo do descritor do aplicativo). Consulte [“Configurações de aplicativos para iPhone”](#) na página 15. É possível cancelar a reorientação automática adicionando um ouvinte de evento `orientationChanging` ao objeto Palco. Chamar o método `preventDefault()` desse objeto de evento cancela a reorientação automática.

Ao utilizar orientação automática, para obter melhores resultados, defina a propriedade `align` de palco para a seguinte:

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

- `Stage.deviceOrientation`: Orientação física do dispositivo. A classe `StageOrientation` define valores para essa propriedade.
- `Stage.orientation`: Orientação atual do palco. A classe `StageOrientation` define valores para essa propriedade.
- `Stage.supportsOrientationChange`: Definida em `true` no iPhone e `false` no aplicativo AIR.
- `Stage.setOrientation()`: Define a orientação do palco. Este método possui um parâmetro, que é uma cadeia de caracteres que define a nova orientação do palco. As constantes na classe `StageOrientation` definem possíveis valores para o parâmetro.
- `StageOrientation`: Define os valores da orientação do palco. Por exemplo, `StageOrientation.ROTATED_RIGHT` indica o palco que é girado à direita em relação à orientação padrão do dispositivo.

- `StageOrientationEvent`: Define os eventos que o palco despacha quando a orientação da tela altera. Esse evento ocorre quando o usuário gira o iPhone. Existem dois tipos de eventos. O palco despacha o evento `orientationChanging` quando o dispositivo é girado. Para evitar que o palco reorienta, chame o método `preventDefault()` do objeto de evento `orientationChanging`. O palco despacha o evento `orientationChange` assim que a reorientação do palco completar.

Atualmente, a API de orientação de tela é útil em aplicativos AIR em dispositivos móveis. Se um aplicativo AIR e um aplicativo AIR de área de trabalho compartilharem o mesmo código-fonte, use a propriedade `Stage.supportsOrientationChange` para verificar se existe suporte à API.

O exemplo a seguir mostra como responder ao usuário girando o dispositivo:

```
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;
    }
}
```

Nesse exemplo, no caso de diferentes orientações de palco, há comentários em vez do código funcional.

É possível alterar a orientação do palco chamando o método `setOrientation()` do objeto do palco. A configuração da orientação é uma operação assíncrona. É possível verificar quando a orientação é completa ouvindo o evento `orientationChange`. O código a seguir mostra como definir o palco com orientação à direita:

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

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

Conforme o Palco pode ser girado e redimensionado e o objeto de Palco envia um evento `resize`. Você pode redimensionar e reposicionar objetos de exibição no Palco como resposta a um evento `Redimensionar`.

NativeApplication.systemIdleMode e SystemIdleMode

A propriedade `NativeApplication.systemIdleMode` permite evitar que o iPhone entre em modo ocioso. Por padrão, o iPhone entra em modo ocioso se não existir uma interação de toque de tela em um período específico. O modo ocioso pode fazer com que a tela escureça. Ele também pode fazer com que o iPhone entre em modo bloqueado. Esta propriedade pode ser definida para um dos dois valores a seguir:

- `SystemIdleMode.NORMAL`—O iPhone segue o comportamento de modo ocioso normal.
- `SystemIdleMode.KEEP_AWAKE`—O aplicativo tenta impedir que o iPhone entre em modo ocioso.

Esta funcionalidade é suportada somente em dispositivos portáteis. Ela não é suportada em aplicativos AIR executados em sistemas operacionais de áreas de trabalho. Em um aplicativo executado na área de trabalho, definir a propriedade `NativeApplication.systemIdleMode` não causa efeitos.

O código a seguir mostra como desabilitar o modo ocioso do iPhone:

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

CameraRoll

A classe `CameraRoll` permite adicionar uma imagem ao rolo da câmera do iPhone. O método `addBitmapData` adiciona uma imagem ao rolo de câmera do iPhone. Este método possui um parâmetro, `bitmapData`. Este parâmetro é o `BitmapData` que contém a imagem a ser adicionada ao rolo da câmera.

A funcionalidade `CameraRoll` é suportada somente em dispositivos portáteis. Ela não é suportada em aplicativos AIR executados em sistemas operacionais de áreas de trabalho. Para verificar em tempo de execução se o aplicativo suporta a funcionalidade `CameraRoll`, selecione a propriedade estática `CameraRoll.supportsAddBitmapData`.

Após chamar o método `addBitmapData()`, o objeto `CameraRoll` envia um de dois eventos:

- `complete`—A operação foi concluída com sucesso.
- `error`—Ocorreu um erro. Por exemplo, talvez não haja espaço livre suficiente no iPhone para armazenar a imagem.

O código a seguir adiciona uma imagem ao palco (uma captura de tela) ao rolo da câmera:

```
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

A propriedade `cacheAsBitmapMatrix` é um objeto Matriz que define como um objeto de exibição é renderizado quando `cacheAsBitmap` está definido para `true`. O aplicativo utiliza esta matriz como matriz de transformação ao renderizar a versão bitmap do objeto de exibição.

Com `cacheAsBitmapMatrix` definido, o aplicativo retém a imagem bitmap em cache renderizada utilizando a matriz, em vez da matriz de exibição. (A matriz de exibição é o valor de `transform.concatenatedMatrix` do objeto de exibição). Se esta matriz não corresponder à matriz de exibição, o bitmap é dimensionado e girado conforme necessário.

Um objeto de exibição com `cacheAsBitmapMatrix` definido somente é renderizado quando o valor de `cacheAsBitmapMatrix` é alterado. O bitmap é dimensionado ou girado conforme necessário, para corresponder à matriz de exibição.

A renderização por meio de CPU e HOJE se beneficiam da utilização da propriedade `cacheAsBitmapMatrix`, embora a renderização por meio de HOJE geralmente se beneficie mais.

Nota: Para usar a aceleração por hardware, configure a GPU (Unidade de processamento gráfico) na guia Geral da caixa de diálogo Configurações do iPhone no Flash Professional CS5. (Ou defina a propriedade `renderMode` para `gpu` no arquivo do descritor do aplicativo).

Por exemplo, o código a seguir utiliza uma representação de bitmap não transformada do objeto de exibição:

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

O código a seguir utiliza uma representação de bitmap que corresponde à renderização atual:

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

Geralmente, a matriz de identificação (`new Matrix()`) ou `transform.concatenatedMatrix` é suficiente. No entanto, você pode utilizar outra matriz como, por exemplo, uma matriz reduzida, para fazer o upload de um bitmap diferentes para a UGP. O exemplo a seguir aplica uma matriz `cacheAsBitmapMatrix` dimensionada para 0.5 nos eixos `x` e `y`. O objeto de bitmap utilizado pela UGP é menor. No entanto, a UGP ajusta seu tamanho para corresponder à propriedade `transform.matrix` do objeto de exibição:

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

Geralmente, selecione uma matriz que transforma o objeto de exibição para o tamanho que será exibido no aplicativo. Por exemplo, se o aplicativo exibe uma versão de bitmap reduzida pela metade da sprite, utilize uma matriz que reduz pela metade. Se o aplicativo exibir uma sprite maior que as dimensões atuais, utilize uma matriz que dimensiona aumentando por este fator.

Existe um limite prático para o tamanho dos objetos de exibição para os quais a propriedade `cacheAsBitmapMatrix` está definida. O limite é 1020 por 1020 pixels. Existe um limite prático para o número total de pixels para todos os objetos de exibição para os quais a propriedade `cacheAsBitmapMatrix` está definida. Este limite é aproximadamente de quatro milhões de pixels.

Existem diversos pontos a serem observados ao utilizar `cacheAsBitmapMatrix` e aceleração por hardware. É importante saber quais objetos devem possuir ou não a propriedade definida. Para obter informações importante sobre como utilizar esta propriedade, consulte [“Aceleração por hardware”](#) na página 36.

É possível utilizar o recurso de diagnóstico de renderização por GPU para diagnosticar o uso da GPU em versões de depuração do seu aplicativo. Para obter mais informações, consulte [“Depurando um aplicativo para iPhone”](#) na página 24.

Observações sobre redes

Utilizando os seguintes esquemas de URL com a função `navigateToURL()` faz com que o documento abra em uma aplicativo externo:

Esquema de URL	Resultado da chamada para <code>navigateToURL()</code>	Exemplo
mailto:	Abre uma nova mensagem no aplicativo do mail.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	Abre uma mensagem no aplicativo de mensagem de texto.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel.:	Disca um número de telefone (com a aprovação do usuário).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Um aplicativo para iPhone pode contar com certificados raiz autoassinados para autenticação de servidor durante uma transação segura, tal como uma solicitação https. Um servidor deveria enviar não apenas a folha do certificado mas também certificados intermediários encadeando para o certificado raiz.

Arquivos API do ActionScript 3.0 de interesse especial para desenvolvedores de aplicativos

As ActionScript 3.0 APIs a seguir definem funcionalidades úteis em dispositivos móveis.

API do medidor de aceleração

As classes abaixo permitem que o aplicativo obtenha eventos do medidor de aceleração do dispositivo:

- Acelerômetro
- AccelerometerEvent

Para obter mais informações, consulte [Entrada de accelerometer](#).

API de geolocalização

As classes abaixo permitem que o aplicativo obtenha eventos do sensor de localização do dispositivo:

- Geolocalização
- GeolocationEvent

Para obter mais informações, consulte [Geolocation](#).

Toque, múltiplos toques e API de gesto

As classes abaixo permitem que o aplicativo receba toques e eventos de gestos:

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

Para obter mais informações, consulte [Entrada de toque, múltiplos toques e gesto](#).

Capítulo 4: Considerações a respeito do design do aplicativo para iPhone

A velocidade do processamento e o tamanho da tela do iPhone proporcionam design especial e considerações a respeito da codificação. No entanto, muitas considerações a respeito do design são comuns ao desenvolvimento de todos os aplicativos ou aplicativos para portáteis.

Para obter mais informações sobre como otimizar aplicativos, consulte [Otimizando conteúdo para a plataforma Flash](#). Esse documento inclui muitas sugestões para otimizar o desempenho do conteúdo de portáteis, conteúdo do Flash Player, conteúdo do AIR e conteúdo com base no ActionScript em geral. A maioria das sugestões também se aplicam a aplicativos AIR para o iPhone.

Aviso importante: A maioria destas considerações de design e técnicas de otimização são essenciais no desenvolvimento de aplicativos que possuem um bom desempenho no iPhone.

Aceleração por hardware

É possível usar a aceleração por hardware OpenGL ES 1.1 para aprimorar o desempenho de gráficos em alguns aplicativos. Jogos e outros aplicativos em que os objetos de exibição são animados podem tirar proveito da aceleração por hardware. Os aplicativos que usam aceleração por hardware podem transferir alguns processos de gráficos da CPU para o UPG do iPhone, o que pode melhorar bastante o desempenho.

Ao desenvolver um aplicativo para uso da GPU, é importante seguir as regras para garantir que o conteúdo seja acelerado corretamente pela GPU.

Para usar a aceleração por hardware, configure a GPU (Unidade de processamento gráfico) na guia Geral da caixa de diálogo Configurações do iPhone no Flash Professional CS5. Também é possível definir a propriedade `renderMode` para `gpu` no arquivo do descritor do aplicativo:

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

Consulte “[Configuração de propriedades de aplicativos do iPhone no Flash Professional CS5](#)” na página 15 e “[Definição de propriedades de aplicativo do iPhone no arquivo do descritor do aplicativo](#)” na página 17.

Existem quatro classes de objetos de exibição que podem ser renderizadas rapidamente em aceleração de hardware se o conteúdo não for alterado frequentemente:

- Objetos bitmap
- Objetos de exibição 2D para os quais a propriedade `cacheAsBitmap` está definida como `true` e também podem ter a propriedade `cacheAsBitmapMatrix` definida (consulte a seguir)
- Objetos de exibição 3D (que possuem a propriedade `z` definida)
- Exibe objetos que possuem um preenchimento retangular de cor sólida e que possuem bordas que se alinham com os pixels da tela.

Os objetos com base em vetor são renderizados sempre que outra `sprite` é animada sobre ou sob eles. Dessa forma, qualquer objeto que atua como fundo ou primeiro plano de uma animação também deve pertencer a uma dessas categorias.

Considerações a respeito do design do aplicativo para iPhone

Para objetos de exibição que possuem `cacheAsBitmap` definido como `true`, definir `cacheAsBitmapMatrix` faz com que a UPG utilize o bitmap resultante da transformação da matriz. A UPG utilize a representação de bitmap sempre que o objeto é girado ou dimensionado. A UPG pode compor e animar este bitmap de forma mais rápida o que a CPU pode redesenhar um objeto renderizado vetor.

Definir `cacheAsBitmap` como `true` faz com que o objeto de exibição (e seus filhos) sejam armazenados em cache. O objeto de exibição não é redesenhado quando as novas regiões são expostas, ou todo o gráfico combinado é traduzido.

Quando a propriedade `cacheAsBitmapMatrix` do objeto de exibição é definida, o aplicativo pode criar uma representação do objeto de exibição mesmo quando não estiver visível. O aplicativo cria uma representação em cache do objeto de exibição ao início do próximo quadro. Ao adicionar o objeto de exibição ao palco, o aplicativo o renderiza de forma mais rápida. De forma similar, o aplicativo pode rapidamente girar ou dimensionar o objeto de forma animada. Não defina a propriedade `cacheAsBitmap` para objetos que são girados ou dimensionados sem também definir a propriedade `cacheAsBitmapMatrix`.

O aplicativo também pode executar transformações alfa rapidamente de um objeto armazenado em cache como bitmap. Entretanto, somente os valores `alpha` entre 0 e 1.0 podem ser utilizados em transformações alfa aceleradas por hardware. Isso corresponde a uma configuração `colorTransform.alphaMultiplier` entre 0 e 256.

Não defina a propriedade `cacheAsBitmap` como `true` para objetos atualizados com frequência, como os campos de texto.

Objetos de exibição que possuem conteúdo gráfico alterado com frequência normalmente não são bons candidatos para renderização pela GPU. Isto é verdade especialmente em dispositivos antigos com GPUs menos poderosas. A sobrecarga de carregamento dos gráficos para a GPU pode tornar a renderização por CPU uma melhor escolha.

Reestruturar objetos de exibição que possuam objetos de exibição filhos que se movem de forma relativa ao pai. Mude-os para que os objetos de exibição filhos se tornem irmãos do pai. Isto garante que eles tenham uma representação própria em bitmap. Além disso, cada objeto de exibição poderá se mover de forma relativa aos outros sem a necessidade de novos gráficos serem carregados para a GPU.

Defina a propriedade `cacheAsBitmap` em `true` no mais alto nível da lista de exibição onde os objetos de exibição filho não são animados. Em outras palavras, defina a propriedade em contêineres de objetos de exibição que não contenham partes móveis. Não defina a propriedade em objetos de exibição filho. *Não* defina a propriedade em sprites que possuem outros objetos de exibição que sejam animados.

Ao definir a propriedade `z` para um objeto de exibição, o aplicativo sempre utiliza a representação do bitmap em cache. Após definir a propriedade `z` para um objeto de exibição, o aplicativo utiliza a representação do bitmap em cache, mesmo que você gire ou dimensione o objeto. O aplicativo não utiliza a propriedade `cacheAsBitmapMatrix` para exibir objetos que possuem a propriedade `z` definida. Aplicam-se as mesmas regras quando você define qualquer propriedade de objeto de exibição tridimensional, incluindo as propriedades `rotationX`, `rotationY`, `rotationZ`, and `transform.matrix3D`.

Não defina a propriedade `scrollRect` ou `mask` para um contêiner de objeto de exibição no qual deseja utilizar aceleração por hardware. A definição dessas propriedades desativa a aceleração por hardware para um contêiner de objeto de exibição e seus objetos filho. Opcionalmente, para configurar a propriedade `mask` crie um objeto de exibição de máscara em camada sobre o objeto que está sendo posicionado na máscara.

Existem limites de tamanho dos objetos de exibição disponíveis para aceleração de hardware. Em dispositivos mais antigos, o limite é de 1024 pixels ou menos em largura e altura. Em dispositivos mais recentes, o limite é de 2048 pixels ou menos. É possível utilizar a ferramenta de diagnóstico de renderização por GPU para testar o desempenho de um dispositivo.

A GPU utiliza também iPhone RAM para armazenar imagens em bitmap. Utiliza o mínimo de memória necessária para imagens em bitmap.

A GPU utiliza distribuições de memória que são potências de 2 em cada dimensão da imagem em bitmap. Por exemplo, a GPU pode reservar memória para tamanhos 512 x 1024 ou 8 x 32. Dessa forma, uma imagem com 9 x 15 pixels ocupa a mesma quantidade de memória que uma imagem com 16 x 16 pixels. Para objetos de exibição armazenados em cache, pode ser interessante utilizar dimensões mais próximas a potências de 2 (mas não mais) em cada direção. Por exemplo, é mais eficiente utilizar um objeto de exibição com 32 x 16 pixels que um objeto de exibição de 33 x 17 pixels.

Não conte com um palco redimensionado para reduzir os ativos que foram dimensionados para outras plataformas (tais como a área de trabalho). Em vez disso, utilize a propriedade `cacheAsBitmapMatrix` ou redimensione os ativos antes de publicar para o iPhone. Os objetos em 3D ignoram a propriedade `cacheAsBitmapMatrix` ao armazenar em cache uma imagem de superfície. Por esse motivo, é melhor redimensionar os objetos antes da publicação, se forem renderizados em uma superfície em 3D.

Existe uma troca de benefícios entre a aceleração por hardware e o uso da RAM. À medida que a memória é preenchida, o iPhone OS avisa outros aplicativos nativos em execução iPhone para liberar memória. À medida que esses aplicativos processam essa notificação e trabalham para liberar memória podem concorrer com seu aplicativos para ciclos da CPU. Isso pode diminuir momentaneamente o desempenho do aplicativo. Certifique-se de testar o aplicativo em dispositivos mais antigos visto que eles possuem substancialmente menos memória disponível para o processo em execução.

Ao depurar o aplicativo no iPhone, é possível habilitar o recurso de diagnóstico de renderização por GPU. Este recurso ajuda você a ver como o aplicativo utiliza a renderização por GPU. Para obter mais informações, consulte “Depuração com diagnóstico de renderização de GPU” em “[Depurando um aplicativo para iPhone](#)” na página 24.

Para obter mais informações sobre como utilizar a propriedade `cacheAsBitmapMatrix`, consulte a seção “`DisplayObject.cacheAsBitmapMatrix`” em “[ActionScript APIs específica para aplicativos AIR móveis](#)” na página 30.

Outras maneiras para melhorar o desempenho de objetos de exibição

A aceleração por hardware pode acelerar o desempenho de gráficos em algumas classes de objetos de exibição. Veja algumas dicas de como maximizar o desempenho de gráficos:

- Tente limitar o número de itens visíveis no palco. Cada item demora algum tempo a renderizar e compor outros itens ao seu redor.

Quando não precisar mais exibir o objeto de exibição, defina sua propriedade `visible` em `false` ou remova-o do palco `removeChild()`. Não defina simplesmente sua propriedade `alpha` em 0.

- Evite modos de mesclagem em geral e o modo de mesclagem de camadas em particular. Use o modo de mesclagem normal sempre que possível.
- Os filtros do objeto de exibição são computacionalmente dispendiosos. Use-os raramente. Por exemplo, usar poucos filtros em uma tela de introdução pode ser aceitável. No entanto, evite usar filtros em muitos objetos ou em objetos sendo animados ou quando for necessário usar uma taxa de quadros alta.
- Evite formas mutáveis.
- Evite usar recorte.
- Se possível, defina o parâmetro `repeat` em `false` ao chamar o método `Graphic.beginBitmapFill()`.

- Não exagere. Use a cor do fundo como fundo. Não estenda em camadas formas grandes umas em cima das outras. Há um custo para cada pixel que é necessário delinear. Isso é especialmente verdadeiro para objetos de exibição que não são acelerados por hardware.
- Evite formas com aumentos longos e finos, extremidades com autointerseção, ou com muitos detalhes finos ao longo das extremidades. Essas formas demoram mais a renderizar que os objetos de exibição com bordas suaves. Isso é especialmente verdadeiro para objetos de exibição que não são acelerados por hardware.
- Crie bitmaps com tamanhos próximos, porém, menores que 2^n por 2^m bits. As dimensões não precisam ser potência de 2, mas devem ser próximas à potência de 2, sem ser maiores. Por exemplo, uma imagem com 31 por 15 pixels renderiza mais rápido que uma imagem 33 por 17 pixels. (31 e 15 são justamente potências menores de 2: 32 e 16). Essas imagens também utilizam a memória de forma mais eficiente.
- Limite o tamanho dos objetos de exibição a 1024 x 1024 pixels (ou 2048 x 2048 em dispositivos mais novos).

Densidade das informações

O tamanho físico da tela de dispositivos móveis é menor que na área de trabalho, embora sua densidade em pixels seja mais alta. É agradável olhar para texto mais acentuado, mas os glifos têm de ter tamanho físico mínimo legível.

Dispositivos móveis são frequentemente usados em movimento e em fracas condições de iluminação. Considere a quantidade de informações que pode exibir de forma realista na tela legivelmente. Talvez seja menos que poderia exibir em uma tela com as mesmas dimensões em pixels em uma área de trabalho.

Use a hierarquia tipográfica para destacar informações importantes. Use tamanho da fonte, espessura, posicionamento e espaçamento para expressar a importância relativa do elemento da interface do usuário. Você pode utilizar uma ou mais sinalizações em cada nível da hierarquia. Aplique as sinalizações de forma consistente em todo o aplicativo. Uma sinalização pode ser espacial (indentação, espaçamento entre linhas, posicionamento) ou gráfica (tamanho, estilo, cor da typeface). A aplicação de sinalizações redundantes pode ser uma maneira eficaz de certificar-se de que a hierarquia é expressa claramente. No entanto, não tente usar mais de três sinalizações para cada nível de agrupamento.

Tente simplificar as etiquetas e o texto explicatório necessário. Por exemplo, use entrada de exemplo no campo de texto para sugerir o conteúdo e evitar uma etiqueta separada.

Fontes e entrada de texto

Para obter melhor aparência, use fontes do dispositivo. Por exemplo, as fontes abaixo são fontes do dispositivo no iPhone:

- Serif: Times New Roman, Georgia, e `_serif`
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma e `_sans`
- Fixed-width: Courier New, Courier e `_typewriter`

Use fontes que têm 14 pixels ou maiores.

Use fontes do dispositivo em campos de texto editáveis. As fontes do dispositivo em campos de texto renderizam também mais rapidamente que fontes incorporadas.

Não use texto sublinhado em campos de entrada de texto. Além disso, não defina o alinhamento do campo de texto. Campos de entrada de texto no iPhone suportam somente alinhamento à esquerda (o padrão).

Se usar a definição Texto TLF em um campo de texto no Flash Professional CS5, desative a biblioteca compartilhada na ligação padrão nas configurações do ActionScript 3.0. Do contrário, o aplicativo não funcionará no iPhone porque tentaria usar o arquivo SWF da biblioteca compartilhada na ligação:

- 1 Selecione Arquivo > Configurações de publicação.
- 2 Na caixa de diálogo Configurações de publicação, clique na guia Flash.
- 3 Clique no botão Script à direita da lista suspensa Script (ActionScript 3.0).
- 4 Clique na guia Caminho da biblioteca.
- 5 Na lista suspensa Ligação padrão, selecione Mesclado no código.

Considere a implantação de alternativas para usar campos de texto de entrada. Por exemplo, não é necessário um campo de texto para o usuário digitar um valor numérico. É possível fornecer dois botões para aumentar ou diminuir o valor.

Esteja ciente do espaço que o teclado virtual utiliza. Quanto o teclado virtual é ativado (por exemplo, quando o usuário pressiona o campo de texto), o aplicativo ajusta a posição do palco. O reposicionamento automático assegura que o campo de texto da entrada selecionada fique visível:

- O campo de texto na parte superior do palco move para a parte superior da área visível do palco. (A área visível do palco é menor para acomodar o teclado virtual).
- O campo de texto na parte inferior do palco fica na parte inferior da nova área do palco.
- O campo de texto em outra parte do palco é movido para o centro vertical do palco.

Quando o usuário clica em um campo de texto ou edita-o (e o teclado virtual é exibido), o objeto TextField despacha um evento `focusIn`. É possível adicionar um ouvinte de evento a esse evento para reposicionar o campo de texto.

Um campo de texto com uma única linha inclui um botão para limpar (à direita do texto) quando o usuário edita o texto. No entanto, esse botão para limpar não é exibido caso o campo de texto seja estreito demais.

Após editar o texto em um campo de texto com uma única linha, o usuário libera o teclado virtual, pressionando a tecla Concluído no teclado.

Após editar o texto em um campo de texto com múltiplas linhas, o usuário libera o teclado virtual, pressionando fora do campo de texto. Isso remove o foco do campo de texto. Certifique-se de que o design inclua uma área fora do campo de texto quando o teclado virtual é exibido. Se o campo de texto for grande demais, nenhuma outra área poderá ficar visível.

A utilização de alguns componentes do Flash Professional CS5 pode impedir que você remova o foco de um campo de texto. Esses componentes são projetados para utilização em áreas de trabalho onde esse comportamento de foco é desejável. Um desses componentes é o componente TextArea. Quando esse componente está em foco (e sendo editado), não é possível remover o foco clicando em outro objeto de exibição. A colocação de alguns outros componentes do Flash Professional CS5 no palco também pode impedir que o foco altere no campo de texto sendo editado.

Não conte com eventos para teclado. Por exemplo, algum conteúdo SWF projetado para usuários da internet usa o teclado para permitir que o usuário controle o aplicativo. No entanto, no iPhone, o teclado virtual está presente somente quando o usuário edita o campo de texto. O aplicativo para iPhone somente despacha eventos para o teclado quando o teclado virtual está presente.

Como salvar o estado do aplicativo

O aplicativo pode fechar a qualquer momento (por exemplo, quando o telefone toca). Considere salvar o estado do aplicativo quando ele muda. Por exemplo, é possível salvar as configurações para um arquivo ou banco de dados no diretório de armazenamento do aplicativo. Opcionalmente, você pode salvar dados em um objeto compartilhado local. Você pode restaurar o estado do aplicativo quando ele for aberto novamente. Se uma chamada telefônica interromper um aplicativo, ele será reiniciado quando a chamada for encerrada.

Não conte com o objeto `NativeApplication` despachando um evento `exitting` quando o aplicativo fecha. Isso pode não ocorrer.

Alterações na orientação da tela

É possível visualizar o conteúdo do iPhone na orientação retrato ou paisagem. Considere a forma como o aplicativo lidará com as alterações na orientação da tela. Para obter mais informações, consulte [Exibição de telas no AIR](#).

Destinos de ocorrências

Considere o tamanho dos destinos de ocorrência ao projetar botões e outros elementos da interface do usuário que o usuário pressiona. Aumente esses elementos o suficiente para serem ativados confortavelmente com o dedo na tela de toque. Além disso, certifique-se de que existe espaço suficiente entre os destinos. Os destinos de ocorrências devem ter aproximadamente de 44 pixels a 57 pixels.

Alocação de memória

É dispendioso alocar blocos de memória recentes. O desempenho do aplicativo pode diminuir ou ocorrer atraso no desempenho durante a animação ou interação quando a coleta de lixo é acionada.

Tente reciclar objetos sempre que possível em vez de livrar-se de um objeto e criar um novo objeto.

Tenha em mente que objetos vetoriais podem consumir menos memória que as matrizes. Consulte [Classe Vector versus classe Array](#).

Para obter mais informações sobre utilização de memória, consulte [Manutenção de memória](#).

API de desenho

Evite usar o API de desenho do ActionScript (classe `Graphics`) para criar gráficos. A utilização do API de desenho cria objetos dinamicamente no palco, em seguida, renderiza-os para o conversor de bitmap. Se possível, crie os objetos estaticamente no palco no momento da autoria, como alternativa.

Os objetos criados usando API de desenho, quando chamados repetidamente, são destruídos e recriados sempre que o ActionScript é executado. No entanto, os objetos estáticos residem na memória através de linhas de tempo diferentes.

Fase de bubbling de eventos

Em contêiner de objeto de exibição profundamente aninhado, a fase de bubbling de eventos pode ser dispendiosa. Reduza essa despesa manuseando o evento completamente no objeto de destino, chamando em seguida, o método `stopPropagation()` do objeto do evento. Chamar esse método evita que ocorra bubbling no evento. Chamar esse método significa também que os objetos pai não recebem o evento.

É possível perceber os ganhos relacionados aplainando o ninho do objeto de exibição para evitar longas cadeias de eventos.

Registre eventos `MouseEvent` em vez de eventos `TouchEvent` quando for possível. Os eventos `MouseEvent` sobrecarregam menos o processador que os eventos `TouchEvent`.

Defina as propriedades `mouseEnabled` e `mouseChildren` para `false`, quando possível.

Otimizar desempenho de vídeo

Para otimizar a reprodução de vídeo móvel, certifique-se de que não existe muita atividade ocorrendo no aplicativo enquanto o vídeo é reproduzido. Isto faz com que o processo de renderização e decodificação do vídeo utilize o máximo possível da CPU.

Durante a reprodução do vídeo, tente não executar nenhum código `ActionScript`. Tente evitar executar código em intervalos frequentes ou na linha de tempo.

Minimize o desenho recorrente de objetos de exibição que não são do vídeo. Evite especificamente redesenhar objetos de exibição que cruzam a área do vídeo. Isto se aplica mesmo que eles estejam ocultos atrás do vídeo. Eles ainda serão redesenhados e utilizarão recursos de processamento. Por exemplo, utiliza formas simples para o indicador de posição e atualize-o somente algumas vezes, em vez de atualizá-lo a cada quadro. Não permita que os controles de vídeo de sobreponham à área de vídeo, posicione-os diretamente abaixo. Caso possua uma animação de buffer de vídeo, não a oculte atrás do vídeo quando não estiver em uso, defina-a como invisível.

Componentes do Flex e Flash

Muitos componentes Flex e Flash são desenvolvidos para uso em aplicativos de área de trabalho. Esses componentes, em especial os componentes de exibição, podem ter um desempenho lento em dispositivos portáteis. Além do desempenho lento, os componentes de área de trabalho podem ter modelos de interação inadequados para dispositivos portáteis.

A Adobe está desenvolvendo uma versão otimizada da estrutura Flex para dispositivos móveis. Para obter mais informações, consulte <http://labs.adobe.com/technologies/flex/mobile/>.

Redução do tamanho do arquivo do aplicativo

A seguir, algumas dicas para reduzir o tamanho de arquivos IPA:

- Verifique os bitmaps de plano de fundo para certificar-se de que possuem o tamanho correto (o tamanho não é maior do que o necessário).
- Verifique se existem fontes adicionais integradas.

Considerações a respeito do design do aplicativo para iPhone

- Verifique se existem ativos PNG para os canais alfa e remova-os se não forem necessários. Use um utilitário como o “PNG crunch” para reduzir o tamanho dos ativos PNG.
- Converta os ativos PNG para ativos JPG sempre que possível.
- Avalie a possibilidade de compactar os arquivos de áudio (utilizando um bitrate mais baixo)
- Remova quaisquer ativos que não estiverem em uso.