

Creazione di applicazioni Adobe AIR con il Packager per iPhone



Note legali

Per le note legali, vedete http://help.adobe.com/it_IT/legalnotices/index.html.

Sommario

Capitolo 1: Introduzione alla creazione di applicazioni AIR per l'iPhone

Concetti importanti	1
Come ottenere strumenti per sviluppatori da Adobe	4
Come ottenere file per sviluppatori da Apple	4
Creazione di un'applicazione iPhone Hello World con Flash Professional CS5	8

Capitolo 2: Compilazione e debug di applicazioni iPhone

Icona iPhone e immagini schermata iniziale	13
Impostazioni applicazione iPhone	15
Compilazione di un file del programma di installazione dell'applicazione iPhone (IPA)	21
Installazione di un'applicazione iPhone	23
Debug di un'applicazione iPhone	25
Invio dell'applicazione iPhone ad App Store	27

Capitolo 3: Supporto API di ActionScript 3.0 per dispositivi mobili

API di ActionScript 3.0 non supportate nei dispositivi mobili	29
API di ActionScript specifiche per applicazioni AIR per dispositivi mobili	31
API di ActionScript 3.0 di particolare interesse per gli sviluppatori di applicazioni per dispositivi mobili	35

Capitolo 4: Considerazioni sulla progettazione di applicazioni iPhone

Accelerazione hardware	37
Altri modi per migliorare le prestazioni degli oggetti di visualizzazione	39
Densità di informazione	40
Caratteri e immissione di testo	40
Salvataggio dello stato dell'applicazione	42
Modifica dell'orientamento dello schermo	42
Destinazioni di tocco	42
Allocazione di memoria	42
API di disegno	42
Bubbling degli eventi	43
Ottimizzazione delle prestazioni video	43
Componenti Flex e Flash	43
Riduzione delle dimensioni del file dell'applicazione	43

Capitolo 1: Introduzione alla creazione di applicazioni AIR per l'iPhone

Potete utilizzare gli strumenti Adobe® Flash® Platform e il codice ActionScript® 3.0 per creare applicazioni Adobe® AIR® per l'iPhone e l'iPod Touch. Queste applicazioni sono distribuite, installate ed eseguite come altre applicazioni iPhone.

Nota: nella parte restante di questo documento si fa riferimento a iPhone e iPod Touch semplicemente come "l'iPhone".

Il Packager per iPhone® è incluso con Adobe® Flash® Professional CS5. Il Packager per iPhone compila tutto il bytecode ActionScript 3.0 in codice applicazione iPhone nativo. Le applicazioni iPhone vengono distribuite come file del programma di installazione dell'applicazione iPhone (file .ipa), mediante iTunes Store.

Potete utilizzare Flash Professional CS5 o Adobe® Flash® Builder™ 4 per modificare il contenuto ActionScript 3.0 di origine per l'applicazione.

Per sviluppare applicazioni iPhone, utilizzate Flash Professional CS5.

Dovete anche ottenere certificati per sviluppatori iPhone da Apple.

Importante: prima di sviluppare applicazioni iPhone, esaminate le informazioni sulla progettazione di applicazioni per iPhone. Vedete "[Considerazioni sulla progettazione di applicazioni iPhone](#)" a pagina 37. Informatevi, inoltre, su quali sono i file per sviluppatori richiesti per creare un'applicazione iPhone. Vedete "[Come ottenere file per sviluppatori da Apple](#)" a pagina 4.

Concetti importanti

È importante comprendere i concetti e il flusso di lavoro coinvolto prima di sviluppare un'applicazione iPhone utilizzando ActionScript 3.0.

Glossario

Quando si crea un'applicazione iPhone, è importante comprendere i termini riportati di seguito.

Sito iPhone Dev Center Il sito Web Apple Computer (<http://developer.apple.com/iphone/>) dove è possibile eseguire le seguenti operazioni:

- Fare domanda per diventare uno sviluppatore iPhone.
- Gestire e creare certificati di sviluppo iPhone, profili di provisioning e ID applicazione (definiti di seguito).
- Inviare applicazioni per App Store.

Certificato di sviluppo di iPhone Utilizzato per identificare uno sviluppatore ai fini dello sviluppo di applicazioni.

Questo file viene fornito da Apple. Potete convertire questo certificato in un certificato P12 per firmare l'applicazione iPhone creata utilizzando ActionScript 3.0. Vedete *File di certificato P12*.

Per eseguire semplicemente il debug e provare le applicazioni Flash Professional CS5 sul computer di sviluppo, non avete bisogno di un certificato di sviluppo di iPhone. Tuttavia, un certificato di sviluppo è richiesto per installare e provare l'applicazione su un iPhone.

Il certificato di sviluppo è diverso da un certificato di distribuzione, utilizzato per creare una versione finale dell'applicazione. Quando create una versione finale dell'applicazione, Apple fornisce un certificato di distribuzione.

Richiesta di firma del certificato Un file contenente informazioni personali utilizzate per generare un certificato di sviluppo. Noto anche come file CSR.

Profilo di provisioning Un file che consente di provare o distribuire un'applicazione iPhone. I file del profilo di provisioning vengono forniti da Apple. Un profilo di provisioning è assegnato a un certificato di sviluppo specifico, un ID applicazione e uno o più ID dispositivo. Sono disponibili diversi tipi di profili di provisioning:

- **Profilo di provisioning di sviluppo:** utilizzato per installare una versione di prova di un'applicazione nell'iPhone dello sviluppatore.
- **Profilo di provisioning di prova:** noto anche come profilo di provisioning ad hoc. Utilizzato per distribuire una versione di prova dell'applicazione a più utenti (e dispositivi iPhone). Utilizzando questo profilo di provisioning e l'applicazione di prova, gli utenti possono provare l'applicazione senza inviarla ad App Store. Nota: potete anche utilizzare un file di provisioning di sviluppo per distribuire le applicazioni di prova a più dispositivi.
- **Profilo di provisioning di distribuzione:** utilizzato per creare un'applicazione iPhone e inviarla ad App Store.

App ID (ID applicazione) Una stringa univoca che identifica un'applicazione iPhone (o più applicazioni) di uno sviluppatore specifico. Gli ID applicazione vengono creati nel sito iPhone Dev Center. Ad ogni profilo di provisioning è associato un ID applicazione o un carattere jolly ID applicazione. Potete utilizzare l'ID applicazione (o il carattere jolly) durante lo sviluppo di un'applicazione. Potete utilizzare l'ID applicazione nella finestra di dialogo Impostazioni iPhone di Flash Professional CS5 (o nel file descrittore dell'applicazione).

Gli ID applicazione a livello di iPhone Dev Center contengono un ID inizializzazione pacchetto seguito da un identificatore pacchetto. L'ID inizializzazione pacchetto è una stringa di caratteri, ad esempio 5RM86Z4DJM, assegnata da Apple all'ID applicazione. L'identificatore pacchetto contiene una stringa nome di dominio inversa che potete selezionare. L'identificatore pacchetto può terminare con un carattere asterisco (*), che indica un carattere jolly ID applicazione. Esempi sono:

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.* (un carattere jolly ID applicazione)

In iPhone Dev Center esistono due tipi di ID applicazione:

- **Carattere jolly ID applicazione:** in iPhone Dev Center, questi ID applicazione terminano con un asterisco (*), ad esempio 96LPVWEASL.com.myDomain.* o 96LPVWEASL.*. Con un file di provisioning che utilizza questo tipo di ID applicazione, potete generare applicazioni di prova che utilizzano un ID applicazione che corrisponde al carattere jolly. Per l'ID applicazione dell'applicazione, potete sostituire l'asterisco con qualsiasi stringa di caratteri validi. Ad esempio, se il sito iPhone Dev Center specifica 96LPVWEASL.com.example.* come ID applicazione, potete utilizzare com.example.foo o com.example.bar come ID applicazione dell'applicazione.
- **ID applicazione specifici:** definiscono un ID applicazione univoco da utilizzare nell'applicazione. In iPhone Dev Center, questi ID applicazione non terminano con un asterisco. Un esempio è 96LPVWEASL.com.myDomain.myApp. Con un file di provisioning che utilizza questo tipo di ID applicazione, per le applicazioni è necessaria la corrispondenza esatta con l'ID applicazione. Ad esempio, se il sito iPhone Dev Center specifica 96LPVWEASL.com.example.helloWorld come ID applicazione, è necessario utilizzare com.example.foo come ID applicazione dell'applicazione.

Durante lo sviluppo dell'applicazione, specificate l'ID applicazione nella finestra di dialogo Impostazioni iPhone in Flash Professional CS5 o nel file descrittore dell'applicazione. Per ulteriori informazioni sugli ID applicazione, vedete la sezione “Scheda Distribuzione” di [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15 oppure vedete [“Impostazione delle proprietà dell'applicazione iPhone nel file descrittore dell'applicazione”](#) a pagina 17.

Importante: quando si specifica l'ID applicazione, ignorate la parte ID inizializzazione pacchetto dell'ID applicazione. Ad esempio, se Apple elenca l'ID applicazione come 96LPVWEASL.com.example.bob.myApp, ignorate 96LPVWEASL e utilizzate com.example.bob.myApp come ID applicazione. Se Apple elenca l'ID applicazione come 5RM86Z4DJM.*, ignorate 5RM86Z4DJM (questo è il carattere jolly ID applicazione).

Potete trovare l'ID applicazione (o il carattere jolly ID applicazione) associato con un profilo di provisioning in iPhone Dev Center (<http://developer.apple.com/iphone>). Andate alla sezione iPhone Developer Program Portal e quindi passate alla sezione Provisioning.

File di certificato P12 Un file P12 (un file con estensione .p12) è un tipo di file di certificato (un file Personal Information Exchange). Il packager per iPhone utilizza questo tipo di certificato per creare un'applicazione iPhone. Il certificato per sviluppatori ricevuto da Apple viene convertito in questa forma di certificato.

ID dispositivo univoco Un codice univoco che identifica un iPhone specifico. Noto anche come UDID o ID dispositivo.

Panoramica del flusso di lavoro di sviluppo

Durante lo sviluppo di un'applicazione per iPhone, procedete nel modo seguente:

- 1 Installate Flash Professional CS5 da Adobe.
- 2 Installate iTunes.
- 3 Ottenete i file per sviluppatori da Apple. Questi file includono il certificato per sviluppatori e i profili di provisioning. Vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.
- 4 Convertite il certificato di sviluppo in un file di certificato P12. Flash CS5 richiede che il certificato sia un certificato P12. Vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.
- 5 Utilizzate iTunes per associare il profilo di provisioning all'iPhone.
- 6 Scrivete l'applicazione in Flash Professional CS5.

È importante comprendere le procedure consigliate per la progettazione e l'ottimizzazione del codice per un'applicazione iPhone. Vedete “[Considerazioni sulla progettazione di applicazioni iPhone](#)” a pagina 37.

Inoltre, il supporto per alcune applicazioni API di ActionScript 3.0 è limitato o assente sull'iPhone. Vedete “[Supporto API di ActionScript 3.0 per dispositivi mobili](#)” a pagina 29.

Potete anche utilizzare Flash Builder 4.0 per modificare il codice ActionScript 3.0 per l'applicazione.

Potete utilizzare Flash Professional CS5 per provare l'applicazione sul computer di sviluppo.

- 7 Create un disegno icona e un disegno schermata iniziale per l'applicazione. Ogni applicazione iPhone include un insieme di icone che la identificano agli utenti. L'iPhone visualizza l'immagine della schermata iniziale durante il caricamento del programma. Vedete “[Icona iPhone e immagini schermata iniziale](#)” a pagina 13.
- 8 Modificate le impostazioni iPhone, incluse le seguenti:
 - L'identità dell'applicazione (compreso il nome file, il nome applicazione, il numero di versione e l'ID applicazione)
 - La posizione del disegno icona di origine per l'applicazione
 - Il certificato P12 e il profilo di provisioning assegnato all'applicazione
 - Le proporzioni iniziali dell'applicazione

In Flash Professional CS5, potete modificare queste impostazioni nella finestra di dialogo Impostazioni iPhone. Per maggiori dettagli, vedete “[Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5](#)” a pagina 15.

Potete anche modificare queste impostazioni direttamente nel file descrittore dell'applicazione. Per ulteriori informazioni, vedete “[Impostazione delle proprietà dell'applicazione iPhone nel file descrittore dell'applicazione](#)” a pagina 17.

9 Compilate il file IPA utilizzando il Packager per iPhone. Vedete “[Compilazione di un file del programma di installazione dell'applicazione iPhone \(IPA\)](#)” a pagina 21.

10 Installate e provate l'applicazione sull'iPhone. Utilizzate iTunes per installare il file IPA.

Per la distribuzione ad hoc, ripetete questo processo generale, ma utilizzate un profilo di provisioning di prova anziché un profilo di provisioning di sviluppo. Per la distribuzione finale dell'applicazione, ripetete questo processo utilizzando il profilo di provisioning di distribuzione. (Per informazioni sui diversi tipi di profili di provisioning, vedete “[Glossario](#)” a pagina 1.)

Quando avete creato una versione di distribuzione dell'applicazione, vedete le istruzioni in “[Invio dell'applicazione iPhone ad App Store](#)” a pagina 27.

Per un'esercitazione rapida sulla creazione di un'applicazione iPhone di base, vedete “[Creazione di un'applicazione iPhone Hello World con Flash Professional CS5](#)” a pagina 8.

Come ottenere strumenti per sviluppatori da Adobe

Per sviluppare applicazioni iPhone utilizzando ActionScript 3.0, è necessario Flash Professional CS5.

Importante: dovete aggiornare la versione del Packager per iPhone dalla versione di anteprima inclusa in Flash Professional CS5. In Flash Professional CS5, selezionate Aiuto > Aggiornamenti.

Potete anche utilizzare Flash Builder 4 per modificare il codice ActionScript. Flash Builder 4 è disponibile all'indirizzo <http://www.adobe.com/it/products/flashbuilder/>.

Come ottenere file per sviluppatori da Apple

Analogamente allo sviluppo di qualsiasi applicazione per iPhone, è necessario innanzitutto ottenere i file per sviluppatori iPhone da Apple. Dovete ottenere un certificato per sviluppatori iPhone e un profilo di provisioning per dispositivi mobili. Inoltre, dovete ottenere altri profili di provisioning. Per le definizioni di questi file, vedete il “[Glossario](#)” a pagina 1.

***Nota:** ottenere questi file è una parte importante del processo di sviluppo dell'applicazione. Accertatevi di completare questo processo prima di sviluppare l'applicazione. Ottenere i file per sviluppatori non è un processo semplice. Leggete attentamente le istruzioni sul sito Apple iPhone Dev Center.*

Come ottenere e utilizzare i file per sviluppatori iPhone

Dovete ottenere un certificato per sviluppatori iPhone e profili di provisioning da Apple. Dovete inoltre convertire il certificato in un certificato P12.

Installare iTunes

Per installare l'applicazione sull'iPhone occorre iTunes. Inoltre, utilizzate iTunes per determinare l'ID dispositivo dell'iPhone. Quando fate richiesta di un certificato per sviluppatori iPhone, dovete conoscere l'ID dispositivo.

Richiesta di un certificato per sviluppatori iPhone e creazione di un profilo di provisioning

Se non l'avete già fatto, iscrivetevi come sviluppatore iPhone registrato presso il sito Apple iPhone Dev Center (<http://developer.apple.com/iphone/>).

Nota: sebbene non sia necessario iPhone SDK o XCode per sviluppare applicazioni AIR per l'iPhone, dovete però essere uno sviluppatore iPhone registrato. Inoltre, dovete ottenere un certificato per sviluppatori e un profilo di provisioning.

- 1 Eseguite il login a iPhone Dev Center utilizzando l'ID account per sviluppatori iPhone.
- 2 Presso il sito iPhone Dev Center, richiedete (e acquistate) un certificato per sviluppatori iPhone.
Riceverete un messaggio e-mail da Apple contenente il codice di attivazione dell'iPhone Developer Program.
- 3 Tornate a iPhone Dev Center. Seguite le istruzioni per l'attivazione del programma per sviluppatori e immettete il codice di attivazione quando richiesto.
- 4 Quando il codice di attivazione è accettato, andate alla sezione iPhone Developer Program Portal del sito iPhone Dev Center.
- 5 Create un file CSR (Certificate Signing Request). Utilizzerete questo file per ottenere un certificato di sviluppo di iPhone. Per le istruzioni, vedete “Generazione di una richiesta di firma del certificato” a pagina 6.
- 6 Nel passaggio successivo, vi verrà chiesto di fornire l'ID dispositivo (o ID dispositivo univoco) per l'iPhone. Potete ottenere l'UDID da iTunes:
 - a Collegate l'iPhone con un cavo USB. Quindi, in iTunes, selezionate la scheda di riepilogo per l'iPhone.
 - b Dopo aver scaricato il profilo di provisioning dal sito del centro per sviluppatori iPhone, aggiungetelo a iTunes.
 - c Quindi, fate clic sul numero di serie visualizzato per visualizzare l'UDID. Premete Comando (Mac OS) o Ctrl (Windows) per copiare l'UDID negli Appunti.
- 7 Create e installate un profilo di provisioning e un certificato di sviluppo di iPhone.
Seguite le istruzioni sul sito iPhone Dev Center. Cercate le istruzioni nella sezione iPhone Developer Program Portal. Per ottenere il certificato di sviluppo e creare il profilo di provisioning potete scegliere di utilizzare il Development Provisioning Assistant.
Ignorate i passaggi relativi a XCode, in quanto questo codice non è richiesto per sviluppare applicazioni iPhone utilizzando Flash Professional CS5.
- 8 In iTunes, selezionate File > Aggiungi alla Libreria, quindi scegliete il file del profilo di provisioning (il cui tipo è mobileprovision). Eseguite la sincronizzazione di iPhone con iTunes.
Questo consente di provare l'applicazione associata a questo profilo di provisioning sull'iPhone.
Per verificare che un profilo di provisioning specifico è stato aggiunto a iTunes, potete provare ad aggiungerlo alla libreria. Se iTunes chiede di sostituire un profilo di provisioning esistente, potete premere il pulsante Annulla. (Il profilo è già installato.) Inoltre, potete controllare i profili di provisioning installati nell'iPhone:
 - a Aprite l'applicazione Impostazioni sull'iPhone.
 - b Aprite la categoria Generali.
 - c Toccate Profili. Nella pagina Profili sono elencati i profili di provisioning installati.
- 9 Se non l'avete fatto, scaricate il file del certificato di sviluppo di iPhone (un file .cer).
È possibile che venga fornito Development Provisioning Assistant con un collegamento per scaricare questo file. Potete anche trovare il file nella sezione Certificates (Certificati) del portale di provisioning sul sito Apple iPhone Dev Center (<http://developer.apple.com/iphone/>).

10 Infine, convertite il certificato per sviluppatori iPhone in un file P12. Per le istruzioni, vedete “[Conversione di un certificato per sviluppatori in un file P12](#)” a pagina 7.

Potete ora creare una semplice applicazione Hello World. Vedete “[Creazione di un'applicazione iPhone Hello World con Flash Professional CS5](#)” a pagina 8.

Generazione di una richiesta di firma del certificato

Per ottenere un certificato per sviluppatori, generate una richiesta di firma del certificato, che inviate al sito Apple iPhone Dev Center.

Generare una richiesta di firma del certificato in Mac OS

In Mac OS, potete utilizzare l'applicazione Accesso Portachiavi per generare una richiesta di firma del certificato. L'applicazione Accesso Portachiavi si trova nella sottodirectory Utility della directory Applicazioni. Dal menu Accesso Portachiavi, selezionate Assistente Certificato > Richiedi un certificato da un'autorità di certificazione.

- 1 Aprite Accesso Portachiavi.
- 2 Nel menu Accesso Portachiavi, selezionate Preferenze.
- 3 Nella finestra di dialogo Preferenze, fate clic su Certificati. Quindi impostate OCSP (Online Certificate Status Protocol) e CRL (Certificate Revocation List) su Disattivato. Chiudete la finestra di dialogo.
- 4 Dal menu Accesso Portachiavi, selezionate Assistente Certificato > Richiedi un certificato da un'autorità di certificazione.
- 5 Immettete l'indirizzo e-mail e il nome che corrispondono all'ID account per sviluppatori di iPhone. Non immettete l'indirizzo e-mail dell'autorità di certificazione. Selezionate Richiesta salvata su disco, quindi fate clic sul pulsante Continua.
- 6 Salvate il file (CertificateSigningRequest.certSigningRequest).
- 7 Caricate il file CSR in Apple sul [sito per sviluppatori di iPhone](#). (Vedete “Richiesta di un certificato per sviluppatori iPhone e creazione di un file di provisioning”.)

Generare una richiesta di firma del certificato in Windows

Per gli sviluppatori Windows, potrebbe essere più semplice ottenere il certificato per sviluppatori iPhone su un computer Mac. Tuttavia, è possibile ottenere un certificato su un computer Windows. Create, innanzitutto, un file CSR (Certificate Signing Request) utilizzando OpenSSL:

- 1 Installate OpenSSL sul computer Windows. (Visitate <http://www.openssl.org/related/binaries.html>.)
Potrebbe anche essere necessario installare i file redistribuibili di Visual C++ 2008, elencati nella pagina di download di Open SSL. (*Non* è necessario che Visual C++ sia installato sul computer.)
- 2 Aprite una sessione di comandi Windows e passate alla directory bin OpenSSL (ad esempio, c:\OpenSSL\bin\).
- 3 Create la chiave privata immettendo l'istruzione seguente nella riga di comando:

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

Salvate questo file della chiave privata per utilizzarlo in seguito.

Quando utilizzate OpenSSL, non ignorate i messaggi di errore. Se OpenSSL genera un messaggio di errore, gli eventuali file creati non sono utilizzabili. In caso di errori, controllate la sintassi ed eseguite nuovamente il comando.

- 4 Create il file CSR immettendo l'istruzione seguente nella riga di comando:

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

Sostituite l'indirizzo e-mail e i valori CN (nome certificato) e C (paese) con quelli personali.

- 5 Caricate il file CSR in Apple sul [sito per sviluppatori di iPhone](#). (Vedete “Richiesta di un certificato per sviluppatori iPhone e creazione di un file di provisioning”.)

Conversione di un certificato per sviluppatori in un file P12

Per sviluppare applicazioni iPhone utilizzando Flash Professional CS5, dovete utilizzare un file di certificato P12. Questo file viene generato in base al certificato per sviluppatori iPhone Apple ricevuto da Apple.

Convertire il certificato per sviluppatori iPhone in un file P12 in Mac OS

Dopo aver scaricato il certificato iPhone Apple da Apple, esportatelo nel formato di certificato P12. Per eseguire questa operazione in Mac® OS:

- 1 Aprite l'applicazione Accesso Portachiavi (nella cartella Applicazioni/Utility).
- 2 Se non avete già aggiunto il certificato a Portachiavi, selezionate File > Importa. Quindi passate al file di certificato (.cer) ottenuto da Apple.
- 3 Selezionate la categoria Chiavi nell'Accesso Portachiavi.
- 4 Selezionate la chiave privata associata al certificato di sviluppo iPhone.
La chiave privata è identificata dallo sviluppatore iPhone: <Nome> <Cognome> certificato pubblico con cui è associato.
- 5 Selezionate File > Esporta elementi.
- 6 Salvate la chiave nel formato file Personal Information Exchange (.p12).
- 7 Vi verrà chiesto di creare una password che verrà utilizzata quando tentate di importare questa chiave in un altro computer.

Convertire un certificato per sviluppatori di Apple in un file P12 in Windows

Per sviluppare applicazioni iPhone utilizzando Flash CS5, dovete utilizzare un file di certificato P12. Questo file viene generato in base al certificato per sviluppatori iPhone Apple ricevuto da Apple.

- 1 Convertite il file di certificato per sviluppatori ricevuto da Apple in un file di certificato PEM. Eseguite l'istruzione della riga di comando seguente dalla directory bin di OpenSSL:

```
openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
```
- 2 Se state utilizzando la chiave privata del portachiavi su un computer Mac, convertitela in una chiave PEM:

```
openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
```
- 3 Potete ora generare un file P12 valido, basato sulla chiave e la versione PEM del certificato per sviluppatori iPhone:

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


Se state utilizzando una chiave del portachiavi Mac OS, utilizzate la versione PEM generata nel passaggio precedente. In caso contrario, utilizzate la chiave OpenSSL generata in precedenza (in Windows).

Gestione di certificati, ID dispositivi, ID applicazioni e profili di provisioning

Potete gestire certificati, ID dispositivi, ID applicazioni e profili di provisioning sul sito Apple iPhone Dev Center (<http://developer.apple.com/iphone/>). Andate alla sezione iPhone Developer Program Portal del sito.

- Fate clic sul collegamento Certificati per gestire i certificati di sviluppo. Potete creare, scaricare o revocare un certificato. Per creare un certificato, dovete innanzitutto creare una richiesta di firma del certificato. Vedete “[Generazione di una richiesta di firma del certificato](#)” a pagina 6.
- Fate clic sul collegamento Dispositivi per gestire l'elenco dei dispositivi in cui è possibile installare l'applicazione di sviluppo.
- Fate clic sul collegamento App ID (ID applicazione) per gestire gli ID applicazione. Quando create un profilo di provisioning, questo è legato a un ID applicazione.
- Fate clic sul collegamento Provisioning per gestire i profili di provisioning. Potete anche utilizzare il Development Provisioning Assistant per creare profili di provisioning di sviluppo.
- Fate clic sul collegamento Distribuzione quando desiderate inviare l'applicazione all'App Store o creare una versione ad hoc dell'applicazione. Questa sezione contiene un collegamento al sito iTunes Connect, utilizzato per inviare un'applicazione all'App Store.

Creazione di un'applicazione iPhone Hello World con Flash Professional CS5

Importante: prima di creare l'applicazione, scaricate le applicazioni e i file per sviluppatori richiesti. Vedete “[Come ottenere strumenti per sviluppatori da Adobe](#)” a pagina 4 e “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.

Creare un progetto Flash Professional CS5

Potete generare un'applicazione iPhone direttamente in Flash Professional CS5:

- 1 Aprite Flash CS5.
- 2 Selezionate File > Nuovo.
- 3 Selezionate iPhone OS.
- 4 Fate clic sul pulsante OK.

Aggiungere contenuto all'applicazione

Aggiungete il testo “Hello world!” all'applicazione:

- 1 Selezionate lo Strumento Testo e fate clic sullo stage.
- 2 Nelle impostazioni Proprietà per il campo di testo, selezionate Testo classico (non Testo TLF).
Poiché si tratta di un'applicazione di base, l'impostazione Testo classico è adeguata. Per utilizzare Testo TLF, dovete applicare qualche altra impostazione. Vedete “[Caratteri e immissione di testo](#)” a pagina 40.
- 3 Nel nuovo campo di testo, digitate “Hello World!”
- 4 Selezionate il campo di testo con lo strumento Selezione.
- 5 Quindi aprite la finestra di ispezione Proprietà e configurate le seguenti impostazioni:
 - Carattere > Famiglia: _sans

- Carattere > Dimensione: 50
- Posizione > X: 20
- Posizione > Y: 20

6 Salvate il file.

7 Selezionate Controllo > Prova filmato > In AIR Debug Launcher (Mobile).

Flash Professional CS5 compila il contenuto SWF e visualizza una versione dell'applicazione in AIR Debug Launcher (ADL). Questo offre un'anteprima rapida dell'applicazione.

Creare un disegno icona e un disegno schermata iniziale per l'applicazione

Tutte le applicazioni iPhone dispongono di icone che vengono visualizzate nell'interfaccia utente dell'applicazione iTunes e sullo schermo dell'iPhone.

- 1 Create una directory all'interno della directory di progetto e denominatela Icone.
- 2 Create tre file PNG nella directory Icone e denominateli Icon29.png, Icon57.png e Icon512.png.
- 3 Modificate i file PNG per creare il disegno appropriato per l'applicazione. Le risoluzioni dei file devono essere 29x29 pixel, 57x57 pixel e 512x512 pixel. Per questa prova, potete semplicemente utilizzare quadrati a tinta unita.

Tutte le applicazioni iPhone visualizzano un'immagine iniziale mentre l'applicazione viene caricata sull'iPhone. Definite l'immagine iniziale in un file PNG:

- 1 Nella directory di sviluppo principale, create un file PNG denominato Default.png. (*Non* collocate questo file nella sottodirectory icons. Accertatevi di denominare il file Default.png, con una D maiuscola.)
- 2 Modificate il file in modo che sia largo 320 pixel e alto 480 pixel. Per il momento, il contenuto può essere un semplice rettangolo bianco. (Potrete cambiare questo aspetto in seguito.)

Nota: quando inviate un'applicazione ad App Store di Apple, utilizzate una versione JPG (non una versione PNG) del file da 512 pixel. Potete utilizzare la versione PNG durante la prova delle versioni di sviluppo di un'applicazione.

Per informazioni dettagliate su questi grafici, vedete “[Icona iPhone e immagini schermata iniziale](#)” a pagina 13.

Modificare le impostazioni dell'applicazione

Importante: se non lo avete già fatto, scaricate le applicazioni e i file per sviluppatori richiesti per lo sviluppo di iPhone. Vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.

Nella finestra di dialogo delle impostazioni iPhone di Flash Professional CS5, potete definire numerose proprietà dell'applicazione iPhone.

- 1 Scegliete File > Impostazioni iPhone.
- 2 Nella scheda Generali, configurate le seguenti impostazioni:
 - File di output: HelloWorld.ipa
Questo è il nome del file del programma di installazione di iPhone che deve essere generato.
 - Nome applicazione: Hello World
Questo è il nome dell'applicazione visualizzata sotto l'icona applicazione nell'iPhone.
 - Versione: 1.0
La versione dell'applicazione.
 - Proporzioni: verticale

- Schermo intero: selezionato
- Orientamento automatico: deselezionato
- Rendering: CPU

Le altre opzioni, GPU e Auto, utilizzano l'accelerazione hardware per il rendering. Questa funzione consente di migliorare le prestazioni delle applicazioni che fanno uso intensivo della grafica (ad esempio i giochi) e progettate per sfruttare l'accelerazione hardware. Per ulteriori informazioni, vedete [“Accelerazione hardware”](#) a pagina 37.

- File inclusi: aggiungete il file grafico della schermata iniziale (Default.png) all'elenco File inclusi.

Nota: per questo esempio Hello World, non modificate le impostazioni rispetto a quelle fornite in queste istruzioni. Alcune impostazioni, ad esempio l'impostazione Versione, hanno limitazioni specifiche descritte in [“Impostazioni applicazione iPhone”](#) a pagina 15.

3 Nella scheda Distribuzione, configurate le seguenti impostazioni:

- Certificato: sfogliate e selezionate il certificato .p12 in base al certificato per sviluppatori ottenuto da Apple. Questo certificato è usato per firmare il file. Dovete convertire il certificato iPhone di Apple nel formato .p12. Per ulteriori informazioni, vedete [“Come ottenere strumenti per sviluppatori da Adobe”](#) a pagina 4.
- Password: immettete la password per il certificato.
- File di provisioning: sfogliate e selezionate il file di provisioning per sviluppatori ottenuto da Apple. Vedete [“Come ottenere file per sviluppatori da Apple”](#) a pagina 4.
- App ID (ID applicazione): se questo campo è selezionabile, potete immettere un ID applicazione che corrisponde all'ID applicazione inviato ad Apple (ad esempio, com.example.as3.HelloWorld).

L'ID applicazione identifica in maniera univoca l'applicazione.

Se il campo non è selezionabile, il profilo di provisioning è vincolato a un ID applicazione specifico. L'ID applicazione è visualizzato nel campo.

Per informazioni dettagliate sulla specifica di un ID applicazione, vedete la sezione “Scheda Distribuzione” di [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15.

4 Nella scheda Icone, fate clic su Icon 29 x 29 nell'elenco Icone. Quindi specificate il percorso del file PNG da 29x29 pixel creato in precedenza (vedete [“Creare un disegno icona e un disegno schermata iniziale per l'applicazione”](#) a pagina 9). Infine, specificate i file PNG per l'icona da 57x57 pixel e l'icona da 512x512 pixel.

5 Fate clic sul pulsante OK.

6 Salvate il file.

Per ulteriori informazioni sulle impostazioni dell'applicazione, vedete [“Impostazioni applicazione iPhone”](#) a pagina 15.

Compilazione del file IPA

Potete ora compilare il file del programma di installazione IPA:

- 1 Selezionate File > Pubblica.
- 2 Nella finestra di dialogo Impostazioni iPhone, fate clic sul pulsante OK.

Il Packager per iPhone genera il file del programma di installazione dell'applicazione iPhone, HelloWorld.ipa, nella directory del progetto. La compilazione del file IPA può richiedere alcuni minuti.

Installare l'applicazione sull'iPhone

Per installare l'applicazione iPhone per la prova su un iPhone:

- 1 Aprite l'applicazione iTunes.
- 2 Se non l'avete già fatto, aggiungete il profilo di provisioning per questa applicazione a iTunes. In iTunes, selezionate File > Aggiungi file alla Libreria, quindi scegliete il file del profilo di provisioning (il cui tipo è mobileprovision).
Per provare l'applicazione sull'iPhone per sviluppatori, utilizzate per il momento il profilo di provisioning di sviluppo.
In seguito, durante la distribuzione di un'applicazione a iTunes Store, utilizzate il profilo di distribuzione. Per distribuire l'applicazione ad hoc a più dispositivi senza passare per iTunes Store, utilizzate il profilo di provisioning ad hoc.
Per ulteriori informazioni sui profili di provisioning, vedete [“Come ottenere file per sviluppatori da Apple”](#) a pagina 4.
- 3 Alcune versioni di iTunes non sostituiscono l'applicazione se la stessa versione dell'applicazione è già installata. In questo caso, cancellate l'applicazione dal dispositivo e dall'elenco delle applicazioni in iTunes.
- 4 Fate doppio clic sul file IPA per l'applicazione. Il file verrà visualizzato nell'elenco delle applicazioni.
- 5 Collegate l'iPhone alla porta USB sul computer.
- 6 In iTunes, controllate la scheda Applicazione per il dispositivo e verificate che l'applicazione sia selezionata nell'elenco delle applicazioni da installare.
- 7 Selezionate il dispositivo nell'elenco a sinistra dell'applicazione iTunes. Quindi fate clic sul pulsante Sincronizza. Al termine della sincronizzazione, l'applicazione Hello World viene visualizzata sull'iPhone.

Se la nuova versione non è installata, eliminatela dall'iPhone e dall'elenco delle applicazioni in iTunes, quindi ripetete questa procedura. Questa situazione può verificarsi se la versione installata utilizza lo stesso ID applicazione e la stessa versione.

Se iTunes visualizza un errore quando tentate di installare l'applicazione, vedete [“Risoluzione dei problemi di installazione dell'applicazione”](#) in [“Installazione di un'applicazione iPhone”](#) a pagina 23.

Modificare il grafico della schermata iniziale

Prima di compilare l'applicazione, create un file Default.png (vedete [“Creare un disegno icona e un disegno schermata iniziale per l'applicazione”](#) a pagina 9). Questo file PNG serve come immagine di avvio durante il caricamento dell'applicazione. Durante la prova dell'applicazione sull'iPhone, potreste aver notato questo schermo vuoto all'avvio.

Modificate questa immagine per farla corrispondere alla schermata di avvio dell'applicazione (“Hello World!”):

- 1 Aprite l'applicazione sul dispositivo. Quando viene visualizzato il primo testo “Hello World”, tenete premuto il pulsante Home (sotto lo schermo). Tenendo premuto il pulsante Home, premete il pulsante Power/Sleep (nella parte superiore dell'iPhone). Questa operazione consente di scattare un'istantanea e di inviarla all'album Rullino fotografico.
- 2 Caricate l'immagine nel computer di sviluppo trasferendo le fotografie da iPhoto o da un'altra applicazione di trasferimento foto. (In Mac OS, potete anche utilizzare l'applicazione Acquisizione immagine.)

Potete anche inviare la foto via e-mail al computer di sviluppo:

- Aprite l'applicazione Foto.
- Aprite l'album Rullino fotografico.
- Aprite l'istantanea acquisita.

- Toccate l'immagine, quindi toccate il pulsante (freccia) “avanti” nell'angolo inferiore sinistro. Fate clic su Invia foto per e-mail e inviate l'immagine a voi stessi.
- 3 Sostituite il file Default.png (nella directory di sviluppo) con una versione PNG dell'immagine acquisita dallo schermo.
 - 4 Ricompilate l'applicazione (vedete “[Compilazione del file IPA](#)” a pagina 10) e reinstallatela sull'iPhone.

L'applicazione utilizza ora la nuova schermata di avvio durante il caricamento.

***Nota:** potete creare qualsiasi disegno preferite per il file Default.png, purché sia delle dimensioni corrette (320 x 480 pixel). Tuttavia, spesso è preferibile che l'immagine Default.png corrisponda allo stato iniziale dell'applicazione.*

Capitolo 2: Compilazione e debug di applicazioni iPhone

Potete compilare un'applicazione iPhone utilizzando il Packager per iPhone. Il Packager per iPhone è incluso con Flash® Professional CS5.

Potete eseguire il debug dell'applicazione sul computer di sviluppo. Potete inoltre installare una versione di debug sull'iPhone e ricevere l'output `trace()` in Flash Professional CS5.

Per un'esercitazione sulla creazione di un'applicazione iPhone dall'inizio alla fine, vedete [“Creazione di un'applicazione iPhone Hello World con Flash Professional CS5”](#) a pagina 8.

Icona iPhone e immagini schermata iniziale

Tutte le applicazioni iPhone dispongono di icone che vengono visualizzate nell'interfaccia utente dell'applicazione iTunes e sull'iPhone.

Icone applicazione iPhone

Sono definite le seguenti icone per un'applicazione iPhone:

- Icona 29x29 pixel: utilizzata dai risultati di ricerca Spotlight su l'iPhone e l'iPod Touch.
- Icona 48x48 pixel: utilizzata dai risultati di ricerca Spotlight sull'iPad.
- Icona 57x57 pixel: visualizzata sugli schermi sensibili di iPhone e iPod Touch.
- Icona 72x72 pixel (opzionale): visualizzata sulla schermata principale dell'iPad.
- Icona 512x512: visualizzata da iTunes. Il file da 512 pixel è utilizzato solo per la prova delle versioni di sviluppo dell'applicazione. Quando inviate l'applicazione finale a Apple App Store, l'immagine da 512 pixel viene inviata separatamente come un file JPG che non fa parte dell'IPA.

In Flash Professional CS5, aggiungete queste icone nella scheda Icons (Icone) della finestra di dialogo Impostazioni iPhone. Vedete [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15.

Potete anche aggiungere i percorsi delle icone nel file descrittore dell'applicazione:

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

L'iPhone aggiunge un effetto bagliore all'icona che non è necessario includere nell'immagine sorgente. Per rimuovere questo effetto bagliore predefinito, aggiungete le istruzioni seguenti all'elemento `InfoAdditions` nel file descrittore dell'applicazione:


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

Vedete [“Impostazione delle proprietà dell'applicazione iPhone nel file descrittore dell'applicazione”](#) a pagina 17.

Disegno schermata iniziale (Default.png)

Tutte le applicazioni iPhone visualizzano un'immagine iniziale mentre l'applicazione viene caricata sull'iPhone. Potete definire l'immagine iniziale in un file PNG denominato Default.png. Nella directory di sviluppo principale, create un file PNG denominato Default.png. (Non collocate questo file in una sottodirectory. Accertatevi di denominare il file Default.png, con una D maiuscola.)

Il file Default.png è largo 320 pixel e alto 480 pixel, indipendentemente dall'orientamento iniziale dell'applicazione o dalla modalità di visualizzazione (ad esempio, schermo intero).

Se l'orientamento iniziale dell'applicazione è orizzontale, utilizzate le stesse dimensioni di un'applicazione verticale: 320 pixel di larghezza e 480 pixel di altezza. Tuttavia, ruotate il disegno di 90° in senso antiorario nel file PNG. Il lato sinistro della grafica PNG corrisponde alla parte superiore dello schermo dell'iPhone in modalità orizzontale. (Per informazioni sull'impostazione dell'orientamento dell'applicazione iniziale, vedete [“Impostazioni applicazione iPhone”](#) a pagina 15.)

Per un'applicazione che non è visualizzata a schermo intero, i 20 pixel in alto dell'immagine predefinita vengono ignorati. L'iPhone visualizza la sua barra di stato sul rettangolo largo 20 pixel nella parte superiore dell'immagine predefinita. In un'applicazione con orientamento orizzontale, questa regione corrisponde al rettangolo largo 20 pixel del file Default.png (che viene visualizzato nella parte superiore in modalità orizzontale). In un'applicazione con orientamento verticale, questa regione è il rettangolo largo 20 pixel nella parte superiore del file Default.png.

Per la maggior parte delle applicazioni, l'immagine Default.png deve corrispondere alla schermata di avvio dell'applicazione. Per acquisire un'istantanea della schermata di avvio dell'applicazione:

- 1 Aprite l'applicazione sull'iPhone. Quando appare la prima schermata dell'interfaccia utente, tenete premuto il pulsante Home (sotto la schermata). Tenendo premuto il pulsante Home, premete il pulsante Power/Sleep (nella parte superiore del dispositivo). Questa operazione consente di scattare un'istantanea e di inviarla all'album Rullino fotografico.
- 2 Caricate l'immagine nel computer di sviluppo trasferendo le fotografie da iPhoto o da un'altra applicazione di trasferimento foto. (In Mac OS, potete anche utilizzare l'applicazione Acquisizione immagine.)

Potete anche inviare la foto via e-mail al computer di sviluppo:

- Aprite l'applicazione Foto.
- Aprite l'album Rullino fotografico.
- Aprite l'istantanea acquisita.
- Toccate l'immagine, quindi toccate il pulsante (freccia) “avanti” nell'angolo inferiore sinistro. Fate clic su Invia foto per e-mail e inviate l'immagine a voi stessi.

Nota: potete creare qualsiasi disegno preferite per il file Default.png, purché sia delle dimensioni corrette. Tuttavia, spesso è preferibile che l'immagine Default.png corrisponda allo stato iniziale dell'applicazione.

Non includete testo nell'immagine Default.png se l'applicazione viene localizzata in più lingue. L'immagine Default.png è statica e il testo non verrebbe localizzato in altre lingue.

In Flash Professional CS5, accertatevi di aggiungere il file Default.png all'elenco File inclusi nella finestra di dialogo Impostazioni iPhone. Vedete [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15.

Quando eseguite la compilazione utilizzando l'applicazione PFI nella riga di comando, accertatevi di fare riferimento a questo file nell'elenco delle risorse incluse. Vedete [“Creazione di un file del programma di installazione dell'applicazione iPhone dalla riga di comando”](#) a pagina 22.

Impostazioni applicazione iPhone

Le impostazioni dell'applicazione includono:

- Il nome applicazione
- Il nome file IPA
- La versione dell'applicazione
- L'orientamento della schermata iniziale dell'applicazione e se l'orientamento della schermata ruota automaticamente quando l'iPhone viene ruotato
- Se la vista iniziale è a schermo intero o no
- Informazioni sulle icone dell'applicazione
- Informazioni sull'accelerazione hardware

Potete modificare le impostazioni dell'applicazione in Flash Professional CS5.

Potete inoltre modificarle nel file descrittore dell'applicazione. Il file descrittore dell'applicazione è un file XML che contiene impostazioni per l'applicazione.

Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5

La finestra di dialogo delle impostazioni iPhone di Flash Professional CS5 consente di definire numerose proprietà di base dell'applicazione iPhone.

Per aprire la finestra di dialogo Impostazioni iPhone:

- ❖ Scegliete File > Impostazioni iPhone.

Scheda Generali

La scheda Generali include le seguenti impostazioni correlate a iPhone:

- File di output: il nome dell'applicazione visualizzata sotto l'icona applicazione nell'iPhone. Non includere un carattere segno più (+) nel nome del file di output.
- Nome applicazione: il nome dell'applicazione visualizzata sotto l'icona applicazione nell'iPhone. Non includere un carattere segno più (+) nel nome applicazione.
- Versione: semplifica l'identificazione della versione dell'applicazione che si sta installando. Questa versione viene utilizzata come CFBundleVersion dell'applicazione iPhone. Il formato deve essere simile a nnnnn[.nn[.nn]] dove n è una cifra compresa tra 0 e 9 e le parentesi quadre indicano componenti opzionali, ad esempio 1, 1.0 o 1.0.1. Le versioni iPhone devono contenere solo cifre e un massimo di due punti decimali.
- Proporzioni: le proporzioni iniziali dell'applicazione (verticale o orizzontale).

- Schermo intero: indica se l'applicazione utilizza la visualizzazione a schermo intero o se visualizza la barra di stato di iPhone.
- Orientamento automatico: consente all'applicazione di riorientare il contenuto del display quando l'iPhone viene ruotato.

Quando utilizzate l'orientamento automatico, per risultati ottimali aggiungete codice ActionScript per impostare la proprietà `align` dello Stage sui valori seguenti:

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

- Rendering: indica il modo in cui gli oggetti di visualizzazione vengono riprodotti sull'iPhone:
 - CPU: l'applicazione utilizza la CPU per eseguire il rendering di tutti gli oggetti di visualizzazione. Non viene utilizzata nessuna accelerazione hardware.
 - GPU: l'applicazione utilizza la GPU dell'iPhone per creare composizioni bitmap.
 - Auto: questa funzione non è implementata.

Per ulteriori informazioni, vedete “[Accelerazione hardware](#)” a pagina 37.

- File inclusi: tutti i file e le directory da includere nel pacchetto dell'applicazione iPhone. Il file SWF principale e il file descrittore dell'applicazione sono inclusi per impostazione predefinita. Aggiungete le eventuali altre risorse richieste nell'elenco File inclusi. Accertatevi di aggiungere il file grafico della schermata iniziale (Default.png) all'elenco File inclusi.

Scheda Distribuzione

La scheda Distribuzione include impostazioni di firma e compilazione per l'applicazione:

- Firma digitale iPhone: specifica un file di certificato P12 e la password per il certificato. Dovete convertire il certificato iPhone di Apple nel formato .p12. Per ulteriori informazioni, vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.
- File di provisioning: percorso al file di provisioning per questa applicazione ottenuto da Apple. Per ulteriori informazioni, vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.
- App ID (ID applicazione): l'ID applicazione identifica in maniera univoca l'applicazione. Se il file di provisioning è collegato a un ID applicazione specifico, Flash Professional CS5 imposta questo campo e non può essere modificato. In caso contrario, il profilo di provisioning consente più ID applicazione (caratteri jolly). Si consiglia di fornire un ID applicazione che corrisponde al carattere jolly ID applicazione fornito a Apple:
 - Se l'ID applicazione Apple è `com.myDomain.*`, il campo App ID nella finestra di dialogo Impostazioni iPhone deve iniziare con `com.myDomain`. (ad esempio, `com.myDomain.myApp` o `com.myDomain.app22`).
 - Se l'ID applicazione Apple è `*`, il campo App ID nella finestra di dialogo Impostazioni iPhone può essere qualsiasi stringa di caratteri validi.

Potete trovare l'ID applicazione (o il carattere jolly ID applicazione) associato con un profilo di provisioning in iPhone Dev Center (<http://developer.apple.com/iphone>). Andate alla sezione iPhone Developer Program Portal e quindi passate alla sezione Provisioning.

Importante: ignorate i caratteri davanti all'ID applicazione Apple. Apple chiama questa stringa ID inizializzazione pacchetto. Ad esempio, se Apple elenca l'ID applicazione come `96LPVWEASL.com.example.bob.myApp`, ignorate `96LPVWEASL` e utilizzate `com.example.bob.myApp` come ID applicazione. Se Apple elenca l'ID applicazione come `5RM86Z4DJM.*`, ignorate `5RM86Z4DJM` (questo è il carattere jolly ID applicazione).

- Tipo di distribuzione iPhone:
 - Quick publishing for device testing (Pubblicazione rapida per prova dispositivo): scegliete questa opzione per compilare rapidamente una versione dell'applicazione per la prova sull'iPhone per sviluppatori.
 - Quick publishing for device debugging (Pubblicazione rapida per il debug dispositivo): scegliete questa opzione per compilare rapidamente una versione di debug dell'applicazione per la prova sull'iPhone per sviluppatori. Grazie a questa opzione, il debugger di Flash Professional CS5 può ricevere output `trace()` dall'applicazione iPhone. (Vedete “[Debug di un'applicazione iPhone](#)” a pagina 25.)
 - Deployment - Ad Hoc (Distribuzione - Ad hoc): scegliete questa opzione per creare un'applicazione per la distribuzione ad hoc. Visitate il centro per sviluppatori Apple iPhone.
 - Deployment - Apple App Store (Distribuzione - App Store di Apple): scegliete questa opzione per creare una versione finale del file IPA per la distribuzione all'App Store di Apple.

Scheda Icone

Nella scheda Icone, specificate la posizione dell'immagine icona da 29x29 pixel, 48x48 pixel, 57x57 pixel, 72x72 pixel e 512x512 pixel. Vedete “[Icona iPhone e immagini schermata iniziale](#)” a pagina 13.

***Nota:** le opzioni per 48x48 pixel e 72x72 pixel non sono incluse nella versione di Packager per iPhone Preview inclusa con Flash Professional CS5. In Flash Professional CS5, selezionate Aiuto > Aggiornamenti per aggiungere queste opzioni.*

Impostazione delle proprietà dell'applicazione iPhone nel file descrittore dell'applicazione

Il file descrittore dell'applicazione è un file XML contenente le proprietà per l'intera applicazione, ad esempio nome, versione copyright e altre impostazioni.

Flash Professional CS5 genera un file descrittore dell'applicazione in base alle impostazioni nella finestra di dialogo Impostazioni iPhone. Tuttavia, potete anche modificare il file descrittore dell'applicazione in un editor di testo. Flash Professional assegna un nome al file descrittore dell'applicazione aggiungendo “-app.xml” al nome del progetto. Ad esempio, il file descrittore dell'applicazione per un progetto HelloWorld è denominato HelloWorld-app.xml. Modificate il file descrittore dell'applicazione se desiderate definire le impostazioni non supportate nella finestra di dialogo delle impostazioni iPhone di Flash Professional CS5. Ad esempio, potete definire l'elemento `InfoAdditions` per specificare le impostazioni info.Plist per l'applicazione.

Importante: non modificate il file descrittore dell'applicazione mentre la finestra di dialogo Flash Professional CS5 è aperta. Salvate le modifiche apportate al file descrittore dell'applicazione prima di aprire la finestra di dialogo Impostazioni iPhone.

Di seguito è riportato un esempio di file descrittore dell'applicazione:

```
<?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>
```

Di seguito sono riportati i dettagli sulle impostazioni nel file descrittore dell'applicazione:

- Nell'elemento `<application>`, lo spazio dei nomi AIR 2.0 è richiesto per la creazione di applicazioni iPhone:

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

- L'elemento `<id>`:

`<id>com.example.as3.HelloWorld</id>` L'ID applicazione identifica in modo univoco l'applicazione. La forma consigliata è una stringa in stile DNS inverso delimitata da punti, ad esempio "com.company.AppName". Il compilatore utilizza questo valore come ID pacchetto per l'applicazione iPhone.

Se il file di provisioning è collegato a un ID applicazione specifico, utilizzate questo ID nell'elemento. Ignorate i caratteri assegnati da Apple all'inizio dell'ID applicazione Apple (noto anche come ID inizializzazione pacchetto). Ad esempio, se l'ID applicazione per il profilo di provisioning è 96LPVWEASL.com.example.bob.myApp, utilizzate com.example.bob.myApp come ID applicazione nel file descrittore dell'applicazione.

Se il profilo di provisioning consente più ID applicazione (caratteri jolly), il suo ID applicazione termina con un asterisco (ad esempio, 5RM86Z4DJM.*). Si consiglia di fornire un ID applicazione che corrisponde al carattere jolly ID applicazione fornito a Apple:

- Se l'ID applicazione Apple è com.myDomain.*, l'ID applicazione nel file descrittore dell'applicazione deve iniziare con com.myDomain. È possibile specificare un ID applicazione quale com.myDomain.myApp o com.myDomain.app22.
- Se l'ID applicazione Apple è *, l'ID applicazione nel file descrittore dell'applicazione può essere qualsiasi stringa di caratteri validi.

Potete trovare l'ID applicazione (o il carattere jolly ID applicazione) associato con un profilo di provisioning in iPhone Dev Center (<http://developer.apple.com/iphone>). Andate alla sezione iPhone Developer Program Portal e quindi passate alla sezione Provisioning.

Importante: ignorate i caratteri davanti all'ID applicazione Apple. Apple chiama questa stringa ID inizializzazione pacchetto. Ad esempio, se Apple elenca l'ID applicazione come 5RM86Z4DJM.*, ignorate 5RM86Z4DJM (questo è il carattere jolly ID applicazione). Se Apple elenca l'ID applicazione come 96LPVWEASL.com.example.bob.myApp, ignorate 96LPVWEASL, utilizzate com.example.bob.myApp come ID applicazione.

- L'elemento `<filename>`:

`<filename>HelloWorld</filename>` Il nome utilizzato per il file del programma di installazione di iPhone. Non includete un carattere segno più (+) nel nome file.

- L'elemento `<name>`:

`<name>Hello World</name>` Il nome dell'applicazione visualizzata nell'applicazione iTunes e nell'iPhone. Non includete un carattere segno più (+) nel nome.

- L'elemento `<version>`:

`<version>1.0</version>` Semplifica l'identificazione della versione dell'applicazione installata dagli utenti. Questa versione viene utilizzata come `CFBundleVersion` dell'applicazione iPhone. Il formato deve essere simile a `nnnnn[.nn[.nn]]` dove `n` è una cifra compresa tra 0 e 9 e le parentesi quadre indicano componenti opzionali, ad esempio 1, 1.0 o 1.0.1. Le versioni iPhone devono contenere solo cifre e un massimo di due punti decimali.

- L'elemento `<initialWindow>` contiene gli elementi subordinati seguenti per specificare le proprietà per l'aspetto iniziale dell'applicazione:

`<content>HelloWorld.swf</content>` Identifica il file SWF radice da compilare nell'applicazione iPhone.

`<visible>true</visible>` Questa è un'impostazione richiesta.

`<fullScreen>true</fullScreen>` Specifica che l'applicazione utilizza tutto lo schermo dell'iPhone.

`<aspectRatio>portrait</aspectRatio>` Specifica che le proporzioni iniziali dell'applicazioni sono in modalità verticale (anziché orizzontale). Osservate che il file `Default.png` utilizzato per definire la finestra iniziale dell'applicazione deve essere largo 320 pixel e alto 480 pixel, indipendentemente dalle impostazioni. (Vedete “[Icona iPhone e immagini schermata iniziale](#)” a pagina 13.)

`<autoOrients>true</autoOrients>` (Facoltativo) Specifica se il contenuto dell'applicazione si riorienta automaticamente quando l'orientamento fisico del dispositivo cambia. Il valore predefinito è `true`. Potete annullare l'orientamento automatico chiamando il metodo `preventDefault()` di un evento `orientationChanging` inviato dall'oggetto Stage. Per ulteriori informazioni, vedete [Impostazione e rilevamento dell'orientamento schermo](#).

Quando utilizzate l'orientamento automatico, per risultati ottimali impostate la proprietà `align` dello Stage sui valori seguenti:

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

`<renderMode>gpu</renderMode>` (Facoltativo) La modalità di rendering utilizzata dall'applicazione. Sono disponibili tre impostazioni:

- `cpu`: l'applicazione utilizza la CPU per eseguire il rendering di tutti gli oggetti di visualizzazione. Non viene utilizzata nessuna accelerazione hardware.
- `gpu`: l'applicazione utilizza la GPU dell'iPhone per creare bitmap composite.

- `auto`: questa funzione non è stata implementata.

Per ulteriori informazioni, vedete “[Accelerazione hardware](#)” a pagina 37.

- L'elemento `<profiles>`:

`<profiles>mobileDevice</profiles>` Limita la compilazione dell'applicazione nel profilo del dispositivo mobile. Questo profilo supporta attualmente solo applicazioni iPhone. Sono supportati tre profili:

- `desktop`: un'applicazione AIR desktop.
- `extendedDesktop`: un'applicazione AIR desktop con supporto per l'API NativeProcess.
- `mobileDevice`: un'applicazione AIR per dispositivi mobili. Attualmente, l'iPhone è il solo dispositivo mobile supportato.

La limitazione dell'applicazione a un profilo specifico impedisce che venga compilata in altri profili. Se non specificate nessun profilo, potete compilare un'applicazione per qualsiasi profilo. Potete specificare più profili elencandoli separatamente nell'elemento `<profiles>`.

Accertatevi di includere `mobileDevice` come profilo supportato (o lasciate l'elemento `<profiles>` vuoto).

- L'elemento `<icon>` contiene gli elementi subordinati seguenti per specificare le icone utilizzate dall'applicazione:

`<image29x29>icons/icon29.png</image29x29>` Questa è l'immagine utilizzata nei risultati di ricerca Spotlight.

`<image48x48>icons/icon48.png</image48x48>` Questa è l'immagine utilizzata nei risultati di ricerca Spotlight sull'iPad.

`<image57x57>icons/icon57.png</image57x57>` Questa è l'immagine utilizzata sulla schermata principale dell'iPhone e dell'iPod Touch.

`<image72x72>icons/icon72.png</image72x72>` Questa è l'immagine utilizzata sulla schermata principale dell'iPad.

`<image512x512>icons/icon512.png</image512x512>` Questa è l'immagine utilizzata nell'applicazione iTunes.

Il Packager per lo strumento iPhone utilizza le icone 29, 57 e 512 a cui si fa riferimento nel file descrittore dell'applicazione. Lo strumento copia le icone nei file denominati, rispettivamente, `Icon-Small.png`, `Icon.png` e `iTunesArtwork`. Per evitare che venga eseguita questa operazione di copia, potete creare direttamente un pacchetto di questi file inserendo i file nella directory contenente il file descrittore dell'applicazione ed elencando i nomi e i percorsi corretti.

L'immagine a 512 pixel è solo per scopi di prova interna. Durante l'invio di un'applicazione a Apple, l'immagine a 512 pixel viene inviata separatamente e non fa parte dell'IPA. In questo modo potete verificare l'aspetto dell'immagine a 512 pixel in iTunes prima di inviarla.

- L'elemento `<iPhone>` contiene gli elementi subordinati seguenti per specificare le impostazioni specifiche di iPhone:

`<InfoAdditions></InfoAdditions>` contiene gli elementi subordinati che specificano le coppie chiave-valore da utilizzare come impostazioni Info.plist per l'applicazione:

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

In questo esempio, i valori impostano lo stile della barra di stato dell'applicazione e affermano che l'applicazione non richiede accesso Wi-Fi continuo.

Le impostazioni InfoAdditions sono racchiuse in un tag CDATA.

Per il supporto iPad, includete impostazioni chiave-valore per `UIDeviceFamily`. L'impostazione `UIDeviceFamily` è un array di stringhe. Ogni stringa definisce i dispositivi supportati. L'impostazione `<string>1</string>` definisce il supporto per l'iPhone e l'iPod Touch. L'impostazione `<string>2</string>` definisce il supporto per l'iPad. L'impostazione `<string>3</string>` definisce il supporto per tvOS. Se specificate una sola di queste stringhe, solo quella famiglia di dispositivi è supportata. Ad esempio, l'impostazione seguente limita il supporto all'iPad:

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

Di seguito viene impostato il supporto per entrambe le famiglie di dispositivi (iPhone/iPod Touch e iPad):

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

Per informazioni su altre impostazioni Info.plist, vedete la documentazione per sviluppatori di Apple.

Compilazione di un file del programma di installazione dell'applicazione iPhone (IPA)

Utilizzate il Packager per iPhone per compilare un'applicazione ActionScript 3.0 in un file del programma di installazione IPA.

Creazione di un file del programma di installazione dell'applicazione iPhone utilizzando il Packager per iPhone incluso con Flash Professional CS5

Per utilizzare il Packager per iPhone incluso con Flash Professional CS5:

- 1 Selezionate File > Pubblica.
- 2 Nella finestra di dialogo Impostazioni iPhone, verificate di aver specificato i valori per tutte le impostazioni. Accertatevi di aver selezionato le opzioni corrette nella scheda Distribuzione. Vedete [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15.
- 3 Fate clic sul pulsante Pubblica.

Il Packager per iPhone genera il file (IPA) del programma di installazione dell'applicazione iPhone. La compilazione del file IPA può richiedere alcuni minuti.

Potete anche eseguire il Packager per iPhone dalla riga di comando. Vedete [“Creazione di un file del programma di installazione dell'applicazione iPhone dalla riga di comando”](#) a pagina 22.

Creazione di un file del programma di installazione dell'applicazione iPhone dalla riga di comando

Potete eseguire il Packager per iPhone dalla riga di comando. Il Packager per iPhone converte il bytecode del file SWF e altri file sorgente in un'applicazione iPhone nativa.

- 1 Aprite una shell dei comandi o un terminale e passate alla cartella del progetto dell'applicazione iPhone.
- 2 Quindi, utilizzate lo strumento `pfi` per creare il file IPA, utilizzando la sintassi seguente:

```
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
```

Modificate il `pfi` di riferimento per includere il percorso completo all'applicazione `pfi`. L'applicazione `pfi` è installata nella sottodirectory `pfi/bin` della directory di installazione di Flash Professional CS5.

Selezionate l'opzione `-target` che corrisponde al tipo di applicazione iPhone che desiderate creare:

- `-target ipa-test`: scegliete questa opzione per compilare rapidamente una versione dell'applicazione per la prova sull'iPhone per sviluppatori.
- `-target ipa-debug`: scegliete questa opzione per compilare una versione di debug dell'applicazione per la prova sull'iPhone per sviluppatori. Con questa opzione, potete utilizzare una sessione di debug per ricevere output `trace()` dall'applicazione iPhone.

Potete includere una delle opzioni `-connect` seguenti (`CONNECT_OPTIONS`) per specificare l'indirizzo IP del computer di sviluppo che esegue il debugger:

- `-connect`: l'applicazione tenta di connettersi a una sessione di debug sul computer di sviluppo utilizzato per compilare l'applicazione.
- `-connect IP_ADDRESS`: l'applicazione tenta di connettersi a una sessione di debug sul computer con l'indirizzo IP specificato. Ad esempio:

```
-target ipa-debug -connect 192.0.32.10
```
- `-connect HOST_NAME`: l'applicazione tenta di connettersi a una sessione di debug sul computer con il nome host specificato. Ad esempio:

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

Nota: l'opzione `-connect` non è inclusa nel Packager per iPhone Preview contenuto in Flash Professional CS5. Aggiornate il Packager per iPhone selezionando **Aiuto > Aggiornamenti in Flash Professional CS5**.

L'opzione `-connect` è opzionale. Se non specificato, l'applicazione di debug risultante non tenterà di connettersi a un debugger residente.

Se una connessione di debug non riesce, l'applicazione visualizza una finestra di dialogo in cui si chiede all'utente di inserire l'indirizzo IP della macchina di debug. Un tentativo di connessione può non riuscire se il dispositivo non è connesso a una rete wifi. Inoltre, questa evenienza si può verificare se il dispositivo è connesso ma non dietro il firewall della macchina host di debug.

Per ulteriori informazioni, vedete “[Debug di un'applicazione iPhone](#)” a pagina 25.

Potete anche includere l'opzione `-renderingdiagnostics` per abilitare la funzione di diagnostica del rendering GPU. Per ulteriori informazioni, vedete “[Debug con la diagnostica del rendering GPU](#)” in “[Debug di un'applicazione iPhone](#)” a pagina 25.

- `-target ipa-ad-hoc`: scegliete questa opzione per creare un'applicazione per la distribuzione ad hoc. Visitate il centro per sviluppatori Apple iPhone.

- `-target ipa-app-store`: scegliete questa opzione per creare una versione finale del file IPA per la distribuzione all'App Store di Apple.

Sostituite `PROFILE_PATH` con il percorso al file del profilo di provisioning per l'applicazione. Per ulteriori informazioni sui profili di provisioning, vedete [“Come ottenere file per sviluppatori da Apple”](#) a pagina 4.

Sostituite `SIGNING_OPTIONS` per fare riferimento al certificato e alla password per sviluppatori iPhone. Utilizzate la sintassi seguente:

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

Sostituite `P12_FILE_PATH` con il percorso al file di certificato P12. Sostituite `PASSWORD` con la password del certificato. (Vedete l'esempio riportato di seguito.) Per ulteriori informazioni sul file del certificato P12, vedete [“Conversione di un certificato per sviluppatori in un file P12”](#) a pagina 7.

Sostituite `APP_DESCRIPTOR` per fare riferimento al file descrittore dell'applicazione.

Sostituite `SOURCE_FILES` per fare riferimento al file SWF principale del progetto seguito dalle eventuali altre risorse da includere. Includete i percorsi a tutti i file delle icone definiti nella finestra di dialogo delle impostazioni dell'applicazione in Flash CS5 o in un file descrittore dell'applicazione personalizzato. Inoltre, aggiungete il file grafico della schermata iniziale, `Default.png`.

Considerare l'esempio seguente:

```
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"
```

Il file `HelloWorld.ipa` viene compilato utilizzando:

- Un certificato PKCS#12 specifico utilizzando la password del certificato `dfb7VKL19`
- Il file descrittore dell'applicazione `HelloWorld-app.xml`
- Un file `HelloWorld.swf` sorgente
- File `Default.png` e di icona specifici

L'applicazione `pfi` compila l'applicazione, in base al file descrittore dell'applicazione, il file SWF e le altre risorse, in un file IPA.

In Mac OS, potete utilizzare un certificato memorizzato nel portachiavi aggiungendo le opzioni seguenti al comando `pfi`:

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

Sostituite `ALIAS_NAME` con l'alias del certificato che volete utilizzare. Quando fate riferimento a un certificato memorizzato nel portachiavi Mac, specificate l'alias anziché fare riferimento al percorso del file del certificato.

Installazione di un'applicazione iPhone

Per installare l'applicazione di sviluppo su un'iPhone, aggiungete il profilo di provisioning all'iPhone, quindi installate l'applicazione.

Aggiungere il profilo di provisioning all'iPhone

Per aggiungere il profilo di provisioning all'iPhone:

- 1 In iTunes, selezionate File > Aggiungi alla Libreria, quindi scegliete il file del profilo di provisioning (il cui tipo è mobileprovision).

Verificate che l'iPhone sia aggiunto al profilo di provisioning. Potete gestire profili di provisioning sul sito Apple iPhone Dev Center (<http://developer.apple.com/iphone/>). Andate alla sezione iPhone Developer Program Portal del sito. Fate clic sul collegamento Dispositivi per gestire l'elenco dei dispositivi in cui è possibile installare l'applicazione di sviluppo. Fate clic sul collegamento Provisioning per gestire i profili di provisioning.

- 2 Collegate l'iPhone al computer ed eseguite la sincronizzazione.

Per informazioni su come ottenere un profilo di provisioning, vedete “Come ottenere file per sviluppatori da Apple” a pagina 4.

Installare l'applicazione

La modalità di installazione dell'applicazione di sviluppo è identica a quella di qualsiasi altro file IPA:

- 1 Se avete installato in precedenza una versione dell'applicazione, eliminate l'applicazione dal dispositivo e dall'elenco delle applicazioni in iTunes.
- 2 Aggiungete l'applicazione in iTunes, utilizzando uno dei metodi seguenti:
 - Dal menu File (in iTunes), scegliete il comando Aggiungi alla Libreria. Quindi selezionate il file IPA e fate clic sul pulsante Apri.
 - Fate doppio clic sul file IPA.
 - Trascinate il file IPA nella libreria iTunes.
- 3 Collegate l'iPhone alla porta USB sul computer.
- 4 In iTunes, controllate la scheda Applicazione per il dispositivo e verificate che l'applicazione sia selezionata nell'elenco delle applicazioni da installare.
- 5 Sincronizzate l'iPhone.

Risoluzione dei problemi di installazione dell'applicazione

Se iTunes visualizza un errore quando si cerca di installare l'applicazione, controllate quanto segue:

- Accertatevi che l'ID dispositivo sia aggiunto al profilo di provisioning.
- Per accertarvi che il profilo di provisioning sia installato, potete trascinarlo in iTunes o utilizzare il comando File > Aggiungi alla Libreria.

Inoltre, verificate che l'ID applicazione dell'applicazione corrisponda all'ID applicazione di Apple:

- Se l'ID applicazione Apple è com.myDomain.*, l'ID applicazione nel file descrittore dell'applicazione o l'ID applicazione nella finestra di dialogo Impostazioni iPhone deve iniziare con com.myDomain (ad esempio, com.myDomain.anythinghere).
- Se l'ID applicazione di Apple è com.myDomain.myApp, l'ID applicazione nel file descrittore dell'applicazione o interfaccia utente Flash Professional CS5 deve essere com.myDomain.myApp.
- Se l'ID applicazione di Apple è *, l'ID applicazione nel file descrittore dell'applicazione o interfaccia utente Flash Profession CS5 può essere qualsiasi.

Potete impostare l'ID applicazione nella finestra di dialogo Impostazioni di iPhone in Flash Professional CS5 o nel file descrittore dell'applicazione.

Debug di un'applicazione iPhone

Potete eseguire il debug dell'applicazione sul computer di sviluppo, con l'applicazione in esecuzione in ADL. Potete anche eseguire il debug dell'applicazione sull'iPhone.

Alcune funzionalità AIR che non sono supportate sull'iPhone sono ancora disponibili quando si esegue il test di un'applicazione utilizzando ADL (sul computer di sviluppo). Tenete presente queste differenze quando eseguite il test del contenuto sul desktop. Per ulteriori informazioni, vedete [“API di ActionScript 3.0 non supportate nei dispositivi mobili”](#) a pagina 29.

Debug dell'applicazione sul computer di sviluppo

Per eseguire il debug dell'applicazione sul computer di sviluppo utilizzando Flash Professional CS5:

- ❖ Scegliete Debug > Debug filmato > In AIR Debug Launcher (Mobile).

Potete anche eseguire il debug dell'applicazione chiamando ADL dalla riga di comando. La sintassi è la seguente:

```
adl -profile mobileDevice appDescriptorFile
```

Sostituite *appDescriptorFile* con il percorso al file descrittore dell'applicazione.

Accertatevi di includere l'opzione `-profile mobileDevice`.

Debug dell'applicazione sull'iPhone

Per eseguire il debug dell'applicazione sull'iPhone:

- 1 Compilate l'applicazione con il supporto debug:
 - In Flash Professional CS5, compilate utilizzando l'impostazione “Quick publishing for device debugging” (Pubblicazione rapida per debug dispositivo). (Vedete [“Creazione di un file del programma di installazione dell'applicazione iPhone utilizzando il Packager per iPhone incluso con Flash Professional CS5”](#) a pagina 21.)
 - Utilizzando l'applicazione della riga di comando PFI, compilate l'applicazione con l'opzione `target ipa-debug`. (Vedete [“Creazione di un file del programma di installazione dell'applicazione iPhone dalla riga di comando”](#) a pagina 22.)
- 2 Installate l'applicazione sull'iPhone.
- 3 Sull'iPhone, attivate la modalità Wi-Fi e collegatevi alla stessa rete del computer di sviluppo.
- 4 Avviate una sessione di debug sul computer di sviluppo. In Flash Professional CS5, scegliete Debug > Inizia sessione di debug remota > ActionScript 3.0.
- 5 Eseguite l'applicazione sull'iPhone.

La versione di debug dell'applicazione richiede l'indirizzo IP del computer per sviluppatori. Immettete l'indirizzo IP e toccate il pulsante OK. Per ottenere l'indirizzo IP del computer di sviluppo:

- In Mac OS, nel menu Apple, scegliete Preferenze di Sistema. Nella finestra Preferenze di Sistema, fate clic sull'icona Rete. La finestra Preferenze di Network contiene l'indirizzo IP.
- In Windows, avviate una riga di comando ed eseguite il comando `ipconfig`.

La sessione di debug visualizza qualsiasi output `trace()` dall'applicazione.

Quando eseguite il debug di un'applicazione installata sull'iPhone, Flash Professional CS5 supporta tutte le funzioni di debug, comprese il controllo dei punti di interruzione, l'esecuzione istruzione per istruzione e il monitoraggio delle variabili.

Debug con la diagnostica del rendering GPU

La funzione di diagnostica del rendering GPU vi consente di vedere come l'applicazione utilizza l'accelerazione hardware (per applicazioni che usano la modalità rendering GPU). Per utilizzare questa funzione, compilate l'applicazione utilizzando lo strumento PFI sulla riga di comando e includete l'opzione `-renderingdiagnostics`:

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

Il flag `-renderingdiagnostics` deve seguire direttamente il flag `-package`.

La funzione di diagnostica del rendering GPU visualizza rettangoli colorati per tutti gli oggetti di visualizzazione:

- **Blu:** l'oggetto di visualizzazione non è una bitmap o memorizzato nella cache come una bitmap, ed è in corso il rendering.

Se il blu appare ripetutamente per un oggetto di visualizzazione che non cambia, la causa può essere che si interseca con oggetti di visualizzazione in movimento. Ad esempio, l'oggetto di visualizzazione potrebbe essere uno sfondo per oggetti di visualizzazione in movimento. Memorizzate l'oggetto di visualizzazione nella cache come una bitmap.

Se il blu appare per un oggetto che pensate debba essere memorizzato nella cache, può essere che l'oggetto utilizzi un effetto che la GPU non può applicare. Questi effetti includono alcuni metodi di fusione, le trasformazioni cromatiche, la proprietà `scrollRect` e le maschere.

L'applicazione visualizza inoltre il blu se gli oggetti di visualizzazione caricati nella GPU superano i limiti di memoria.

L'applicazione registra i messaggi per ogni rettangolo blu. L'applicazione restituisce questi messaggi insieme con altri `trace()` e messaggi di output di debug.

- **Verde:** l'oggetto di visualizzazione è una bitmap, o memorizzato nella cache come una bitmap, e viene caricato nella GPU per la prima volta.

Se il verde appare ripetutamente per un oggetto di visualizzazione, il codice nell'applicazione sta ricreando l'oggetto di visualizzazione. Ad esempio, questo si verifica se la timeline ritorna a un fotogramma creato dall'oggetto di visualizzazione. Modificate il contenuto per impedire che vengano ricreati oggetti identici.

- **Rosso:** l'oggetto di visualizzazione è una bitmap, o memorizzato nella cache come una bitmap, e viene ricaricato nella GPU.

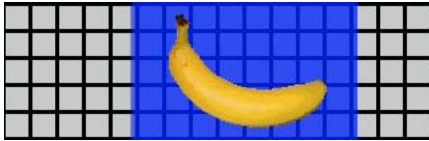
Il rosso appare ogni volta che un tale oggetto di visualizzazione cambia in un modo che richiede una nuova esecuzione del rendering della rappresentazione bitmap. Ad esempio, se viene nuovamente eseguito il rendering di un oggetto 2D la cui proprietà `cacheAsBitmapMatrix` non è impostata quando l'oggetto viene ridimensionato o ruotato. La ripetizione del rendering si verifica anche quando gli oggetti di visualizzazione secondari vengono spostati o modificati.

Ogni rettangolo colorato si dissolve dopo che lo schermo viene ridisegnato quattro volte, purché durante questi cicli non si verifichi nuovamente il motivo della colorazione. Tuttavia, se non ci sono modifiche su schermo, il colore della diagnostica non cambia.

Ad esempio, considerate un oggetto di visualizzazione bitmap (una banana) davanti a uno sfondo vettoriale che non è memorizzato nella cache come una bitmap. La prima volta che viene eseguito il rendering della banana, questa viene colorata di verde. La prima volta che viene eseguito il rendering dello sfondo, questo viene colorato di blu:

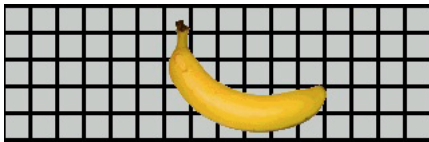


Mentre la banana si sposta, la CPU deve eseguire nuovamente il rendering dello sfondo; questo causa l'ombreggiatura blu per apparire sopra lo sfondo:



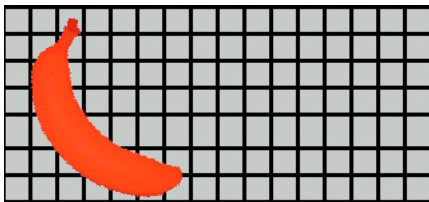
L'ombreggiatura blu sopra lo sfondo riflette le regioni ridisegnate che devono essere inviate alla GPU.

Tuttavia, se lo sfondo è memorizzato nella cache come una bitmap, quando lo banana si sposta, la funzione di diagnostica del rendering non visualizza nessuna tinta colorata:



La funzione di diagnostica non visualizza nessuna tinta colorata poiché la GPU conserva la bitmap di sfondo. La GPU può comporre la banana con lo sfondo senza coinvolgere la CPU.

Supponete che la banana sia un oggetto di visualizzazione 2D la cui proprietà `cacheAsBitmapMatrix` non è impostata. Ogni volta che l'oggetto di visualizzazione viene ruotato (o ridimensionato), la funzione di diagnostica del rendering visualizza rosso. Questo indica che l'applicazione deve caricare una nuova versione dell'oggetto di visualizzazione nella GPU:



Invio dell'applicazione iPhone ad App Store

Per inviare l'applicazione ad App Store:

- 1 Ottenete un certificato di distribuzione e un profilo di provisioning dal sito Web iPhone Dev Center (<http://developer.apple.com/iphone/>).
Un certificato di distribuzione è denominato “iPhone Developer: XXX.cer”, dove XXX è il vostro nome.
Per ulteriori informazioni, vedete “[Come ottenere file per sviluppatori da Apple](#)” a pagina 4.
- 2 Convertite il certificato di distribuzione in un file P12.
Per ulteriori informazioni, vedete “[Conversione di un certificato per sviluppatori in un file P12](#)” a pagina 7.
- 3 Compilate l'applicazione utilizzando il file P12 e il profilo di provisioning.
Utilizzate il file P12 creato basato sul certificato di distribuzione. Utilizzate l'ID applicazione associato al profilo di provisioning di distribuzione.

Per ulteriori informazioni, vedete “[Compilazione di un file del programma di installazione dell'applicazione iPhone \(IPA\)](#)” a pagina 21.

- 4 Inviare l'applicazione al sito Web iPhone Dev Center (<http://developer.apple.com/iphone/>).

Importante: Apple richiede l'uso del programma Apple Application Loader per caricare applicazioni su App Store. Apple pubblica Application Loader solo per Mac OS X e quindi, anche se potete sviluppare un'applicazione AIR per iPhone usando un computer Windows, dovete avere accesso a un computer con sistema operativo OS X (versione 10.5.3 o successiva) per inviarla all'App Store. Il programma Application Loader può essere scaricato dall'Apple iOS Developer Center.

Capitolo 3: Supporto API di ActionScript 3.0 per dispositivi mobili

Durante la creazione di applicazioni AIR per dispositivi mobili, potete utilizzare le stesse API di ActionScript 3.0 disponibili per lo sviluppo di applicazioni desktop Adobe Flash Player 10.1 e AIR 2. Esistono, tuttavia, eccezioni e aggiunte.

API di ActionScript 3.0 non supportate nei dispositivi mobili

Alcune API di ActionScript 3.0 non sono disponibili per applicazioni in esecuzione nel profilo per dispositivi mobili (ad esempio, applicazioni in esecuzione sull'iPhone).

Se utilizzate lo stesso codice ActionScript per sviluppare più profili (ad esempio, desktop e mobile), utilizzate codice per provare se un'API è supportata. Ad esempio, la classe `NativeWindow` non è supportata nelle applicazioni iPhone. (Le applicazioni iPhone non possono utilizzare o creare finestre native.) Per provare se un'applicazione è in esecuzione su un profilo che supporta finestre native (ad esempio, profilo desktop), controllate la proprietà `NativeWindow.isSupported`.

Nella tabella seguente sono elencate le API che non sono supportate nel profilo per dispositivi mobili. Inoltre, contiene un elenco delle proprietà che è possibile controllare per determinare quando un'applicazione è in esecuzione su una piattaforma che offre il supporto per un'API.

API	Prova per supporto
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

API	Prova per supporto
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
PDF support	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

Non potete scrivere applicazioni AIR basate su HTML e JavaScript per il profilo per dispositivi mobili.

Alcune classi ActionScript 3.0 sono supportate solo parzialmente:

File

Le applicazioni iPhone possono accedere solo alla directory dell'applicazione e alla directory di archiviazione dell'applicazione. Potete anche chiamare i metodi `File.createTempFile()` e `File.createTempDirectory()`. La chiamata di un'operazione per accedere a un'altra directory (ad esempio, un metodo di lettura o scrittura `FileStream`) determina un'eccezione `IOError`.

Le applicazioni iPhone non supportano finestre di dialogo del browser dei file nativi, ad esempio quella fornita dal metodo `File.browseForOpen()`.

Loader

In un'applicazione iPhone, non potete utilizzare il metodo `Loader.load()` e nemmeno eseguire codice ActionScript nel contenuto SWF caricato con il metodo `Loader.load()`. Potete, tuttavia, utilizzare risorse nel file SWF (ad esempio, clip filmato, immagini, caratteri e audio nella libreria), nonché utilizzare il metodo `Loader.load()` per caricare file immagine.

Video

Solo i video Sorensen e ON2 VP6 sono supportati all'interno di un'applicazione AIR nell'iPhone.

Potete utilizzare il metodo `navigateToURL()` per aprire un video H.264 esternamente all'applicazione. Come parametro `request`, passate un oggetto `URLRequest` con un URL che fa riferimento al video. Il video viene avviato nel lettore video dell'iPhone.

Campi di testo

Esistono delle limitazioni per i caratteri e altre impostazioni per i campi di testo sull'iPhone. Vedete [“Caratteri e immissione di testo”](#) a pagina 40.

API non supportate e debug utilizzando ADL

Alcune funzionalità AIR che non sono supportate sull'iPhone sono ancora disponibili quando si esegue il test di un'applicazione utilizzando ADL (sul computer di sviluppo). Tenete presente queste differenze quando eseguite il test del contenuto utilizzando ADL.

Questa funzionalità include i codec video e audio seguenti: Speex (audio), H.264/AVC (video) e AAC (audio). Questi codec non sono disponibili nelle applicazioni AIR in esecuzione sull'iPhone. Tuttavia, continuano a funzionare normalmente sul desktop.

L'accessibilità e il supporto screen reader funziona in ADL per Windows. Tuttavia, queste API non sono supportate sull'iPhone.

Il protocollo RTMPE funziona normalmente se utilizzato da ADL sul desktop. Tuttavia, un NetConnection non è in grado di connettersi utilizzando il protocollo RTMPE sull'iPhone.

La classe Loader funziona senza limitazioni aggiuntive quando il contenuto viene eseguito con ADL. Tuttavia, quando viene eseguita sull'iPhone, il tentativo di caricare contenuto SWF che contiene bytecode ActionScript determina un messaggio di errore.

Le istanze Shader vengono eseguite in ADL. Tuttavia, sull'iPhone il bytecode Pixel Bender non viene interpretato e gli shader non hanno effetti grafici.

Per ulteriori informazioni, vedete [“Debug di un'applicazione iPhone”](#) a pagina 25.

API di ActionScript specifiche per applicazioni AIR per dispositivi mobili

Le API seguenti sono disponibili solo in applicazioni AIR su dispositivi mobili. Al momento non possono essere utilizzate nelle versioni Flash Player o desktop di AIR.

API di orientamento dello schermo

L'API di orientamento dello schermo consente di utilizzare l'orientamento dello stage e dell'iPhone:

- `Stage.autoOrients`: indica se l'applicazione è impostata per fare in modo che lo stage si riorienta automaticamente quando il dispositivo viene ruotato. Questa proprietà è impostata su `true` quando nella finestra di dialogo delle impostazioni iPhone di Flash Professional CS5 è selezionata l'opzione di orientamento automatico. (Potete anche impostare l'elemento `autoOrients` su `true` nel file descrittore dell'applicazione.) Vedete [“Impostazioni applicazione iPhone”](#) a pagina 15. Potete annullare il riorientamento automatico aggiungendo un listener dell'evento `orientationChanging` per l'oggetto Stage. La chiamata al metodo `preventDefault()` di questo oggetto evento annulla il riorientamento automatico.

Quando utilizzate l'orientamento automatico, per risultati ottimali impostate la proprietà `align` dello Stage sui valori seguenti:

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

- `Stage.deviceOrientation`: indica l'orientamento fisico del dispositivo. La classe `StageOrientation` definisce i valori per questa proprietà.
- `Stage.orientation`: indica l'orientamento attuale dello stage. La classe `StageOrientation` definisce i valori per questa proprietà.
- `Stage.supportsOrientationChange`: impostato su `true` sull'iPhone e su `false` in un'applicazione AIR.

- `Stage.setOrientation()`: consente di impostare l'orientamento dello stage. Questo metodo dispone di un solo parametro, che è una stringa che definisce il nuovo orientamento dello stage. Le costanti nella classe `StageOrientation` definiscono i possibili valori del parametro.
- `StageOrientation`: definisce i valori di orientamento dello stage. Ad esempio, `StageOrientation.ROTATED_RIGHT` indica uno stage ruotato a destra rispetto all'orientamento predefinito del dispositivo.
- `StageOrientationEvent`: definisce eventi inviati dallo Stage quando l'orientamento dello schermo cambia. Questo evento si verifica quando l'utente ruota l'iPhone. Esistono due tipi di eventi. Lo Stage invia l'evento `orientationChanging` quando il dispositivo viene ruotato. Per impedire che lo stage venga riorientato, chiamate il metodo `preventDefault()` dell'oggetto evento `orientationChanging`. Lo Stage invia l'evento `orientationChange` dopo che lo stage è stato riorientato.

Attualmente, l'API di orientamento dello schermo è utile solo nelle applicazioni AIR su dispositivi mobili. Se un'applicazione AIR per dispositivi mobili e un'applicazione AIR desktop condividono il codice di origine, utilizzate la proprietà `Stage.supportsOrientationChange` per verificare se l'API è supportata.

Nell'esempio seguente viene mostrato come rispondere alla rotazione del dispositivo da parte dell'utente:

```
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;
    }
}
```

In questo esempio, in caso di orientamenti stage diversi, sono presenti commenti anziché codice funzionante.

Potete cambiare l'orientamento dello stage chiamando il metodo `setOrientation()` dell'oggetto `Stage`. L'impostazione dell'orientamento è un'operazione asincrona. Potete verificare quando l'orientamento è completato intercettando l'evento `orientationChange`. Il codice seguente mostra come impostare lo stage sull'orientamento destro:

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

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

Mentre lo Stage ruota si ridimensiona e l'oggetto Stage invia un evento `resize`. Potete ridimensionare e riposizionare gli oggetti di visualizzazione sullo Stage in risposta all'evento `resize`.

NativeApplication.systemIdleMode e SystemIdleMode

La proprietà `NativeApplication.systemIdleMode` impedisce che l'iPhone entri nella modalità di inattività. Per impostazione predefinita, un iPhone entra nella modalità di inattività se lo schermo sensibile non viene utilizzato per un certo periodo di tempo. La modalità di inattività può causare la riduzione della luminosità dello schermo. Inoltre, può causare l'attivazione della modalità di blocco. Questa proprietà può essere impostata su uno di due valori:

- `SystemIdleMode.NORMAL`: l'iPhone segue il normale comportamento della modalità di inattività.
- `SystemIdleMode.KEEP_AWAKE`: l'applicazione tenta di impedire che l'iPhone entri nella modalità di inattività.

Questa funzionalità è supportata solo sui dispositivi mobili e non nelle applicazioni AIR in esecuzione su sistemi operativi desktop. In un'applicazione eseguita sul desktop, l'impostazione della proprietà `NativeApplication.systemIdleMode` non ha alcun effetto.

Nel codice seguente viene mostrato come disattivare la modalità di inattività dell'iPhone:

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

CameraRoll

La classe `CameraRoll` consente di aggiungere un'immagine al rullino fotografico dell'iPhone. Il metodo `addBitmapData()` aggiunge un'immagine al rullino fotografico dell'iPhone. Il metodo dispone di un solo parametro, `bitmapData`, che rappresenta l'oggetto `BitmapData` contenente l'immagine da aggiungere al rullino fotografico.

La funzionalità `CameraRoll` è supportata solo sui dispositivi mobili e non nelle applicazioni AIR in esecuzione su sistemi operativi desktop. Per verificare in fase di runtime se l'applicazione supporta la funzionalità `CameraRoll`, controllate la proprietà `CameraRoll.supportsAddBitmapData` statica.

Dopo aver chiamato il metodo `addBitmapData()`, l'oggetto `CameraRoll` invia uno di due eventi:

- `complete`: l'operazione è terminata correttamente.
- `error`: si è verificato un errore. Ad esempio, è possibile che la quantità di memoria disponibile sull'iPhone non sia sufficiente per memorizzare l'immagine.

Il codice seguente aggiunge un'immagine dello stage (un'acquisizione schermo) al rullino fotografico:

```
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

La proprietà `cacheAsBitmapMatrix` è un oggetto `Matrix` che definisce in che modo viene eseguito il rendering di un oggetto di visualizzazione quando `cacheAsBitmap` è impostata su `true`. L'applicazione utilizza questa matrice come una matrice di trasformazione quando esegue il rendering della versione bitmap dell'oggetto di visualizzazione.

Se `cacheAsBitmapMatrix` è impostata, l'applicazione conserva un'immagine della bitmap memorizzata nella cache di cui effettua il rendering utilizzando questa matrice, anziché la matrice di visualizzazione. (La matrice di visualizzazione è il valore della proprietà `transform.concatenatedMatrix` dell'oggetto di visualizzazione.) Se questa matrice non corrisponde alla matrice di visualizzazione, la bitmap viene ridimensionata e ruotata in base alle esigenze.

Per un oggetto di visualizzazione con impostata la proprietà `cacheAsBitmapMatrix`, il rendering viene eseguito solo quando il valore di `cacheAsBitmapMatrix` cambia. La bitmap viene ridimensionata e ruotata come appropriato in modo che sia conforme alla matrice di visualizzazione.

I rendering basati su CPU e GPU traggono entrambi vantaggio dall'utilizzo della proprietà `cacheAsBitmapMatrix`, anche se i benefici per il rendering GPU sono maggiori.

Nota: Per utilizzare l'accelerazione hardware, impostate il Rendering su GPU nella scheda Generali della finestra di dialogo Impostazioni iPhone in Flash Professional CS5. (In alternativa, impostate la proprietà `renderMode` su `gpu` nel file descrittore dell'applicazione.)

Ad esempio, il codice seguente utilizza una rappresentazione bitmap non trasformata dell'oggetto di visualizzazione:

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

Il codice seguente utilizza una rappresentazione bitmap che corrisponde al rendering corrente:

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

In genere, la matrice dell'identità (`new Matrix()`) o `transform.concatenatedMatrix` è sufficiente. Tuttavia, potete utilizzare un'altra matrice, ad esempio una matrice ridotta, per caricare una bitmap diversa nella GPU. L'esempio seguente applica una matrice `cacheAsBitmapMatrix` che è ridotta di un fattore 0,5 sugli assi x e y. Le dimensioni dell'oggetto bitmap utilizzato dalla GPU sono inferiori, tuttavia la GPU regola le sue dimensioni in modo che corrispondano alla proprietà `transform.matrix` dell'oggetto di visualizzazione:

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

In generale, scegliete una matrice che trasformi l'oggetto di visualizzazione nelle dimensioni con cui appare nell'applicazione. Ad esempio, se l'applicazione visualizza la versione bitmap dello sprite ridotta della metà, utilizzate la matrice che riduce della metà. Se l'applicazione visualizza lo sprite più grande rispetto alle sue dimensioni correnti, utilizzate una matrice che ingrandisce di quel fattore.

Per le dimensioni degli oggetti di visualizzazione per i quali la proprietà `cacheAsBitmapMatrix` è impostata esiste un limite pratico di 1020 x 1020 pixel. Per il numero totale di pixel di tutti gli oggetti di visualizzazione per i quali la proprietà `cacheAsBitmapMatrix` è impostata esiste un limite pratico di circa quattro milioni di pixel.

L'utilizzo della proprietà `cacheAsBitmapMatrix` e dell'accelerazione hardware richiede diverse considerazioni. È importante sapere quali oggetti di visualizzazione devono avere la proprietà impostata e quali no. Per informazioni importanti sull'utilizzo di questa proprietà, vedete [“Accelerazione hardware”](#) a pagina 37.

Potete utilizzare la funzione di diagnostica del rendering GPU per diagnosticare l'uso della GPU nelle build di debug dell'applicazione. Per ulteriori informazioni, vedete [“Debug di un'applicazione iPhone”](#) a pagina 25.

Note sulla connettività di rete

L'uso degli schemi URL seguenti con la funzione `navigateToURL()` causa l'apertura di un documento in un'applicazione esterna:

Schema URL	Risultato delle chiamate a <code>navigateToURL()</code>	Esempio
mailto:	Apre un nuovo messaggio nell'applicazione di posta elettronica.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	Apre un messaggio nell'applicazione dei messaggi di testo.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	Compone un numero di telefono (con approvazione utente).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Un'applicazione iPhone potrebbe basarsi su certificati principali autofirmati per l'autenticazione server durante una transazione protetta, ad esempio una richiesta https. Un server deve inviare non solo il certificato foglia ma anche tutti i certificati intermedi che si concatenano al certificato principale.

API di ActionScript 3.0 di particolare interesse per gli sviluppatori di applicazioni per dispositivi mobili

Le API di ActionScript 3.0 seguenti definiscono le funzionalità utili su dispositivi mobili.

API accelerometro

Le classi sottostanti consentono all'applicazione di ricevere eventi dall'accelerometro del dispositivo:

- Accelerometer
- AccelerometerEvent

Per ulteriori informazioni, vedete [Input accelerometro](#).

API di localizzazione geografica

Le classi sottostanti consentono all'applicazione di ricevere eventi dal sensore di posizione del dispositivo:

- Geolocation
- GeolocationEvent

Per ulteriori informazioni, vedete [Localizzazione geografica](#).

API tocco, multitocco e gestuali

Le classi sottostanti consentono all'applicazione di ricevere eventi touch e gesture:

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

Per ulteriori informazioni, vedete [Input tocco, multitocco e gestuale](#).

Capitolo 4: Considerazioni sulla progettazione di applicazioni iPhone

La velocità di elaborazione e le dimensioni schermo dell'iPhone si prestano a considerazioni di codifica e progettazione speciali. Tuttavia, molte delle considerazioni di progettazione sono comuni allo sviluppo di tutte le applicazioni o alle applicazioni per dispositivi mobili.

Per ulteriori informazioni sull'ottimizzazione delle applicazioni, vedete [Ottimizzazione del contenuto per la piattaforma Flash](#). Questo documento include numerosi suggerimenti per l'ottimizzazione delle prestazioni del contenuto per dispositivi mobili, contenuto Flash Player, contenuto AIR e contenuto basato su ActionScript in generale. La maggior parte di questi suggerimenti si applicano anche alle applicazioni AIR per l'iPhone.

Importante: molte di queste considerazioni sulla progettazione e tecniche di ottimizzazione sono essenziali nello sviluppo di applicazioni in grado di funzionare correttamente sull'iPhone.

Accelerazione hardware

Potete utilizzare l'accelerazione hardware OpenGL ES 1.1 per migliorare le prestazioni grafiche in alcune applicazioni. Giochi e altre applicazioni in cui oggetti di visualizzazione sono animati possono trarre vantaggio dall'accelerazione hardware. Le applicazioni che utilizzano l'accelerazione hardware possono trasferire alcuni processi grafici dalla CPU alla GPU dell'iPhone, con considerevole miglioramento delle prestazioni.

Quando si progetta un'applicazione per utilizzare GPU è importante seguire le regole che garantiscono che il contenuto sia effettivamente accelerato GPU.

Per utilizzare l'accelerazione hardware, impostate il Rendering su GPU nella scheda Generali della finestra di dialogo Impostazioni iPhone in Flash Professional CS5. Potete anche impostare la proprietà `renderMode` su `gpu` nel file descrittore dell'applicazione:

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

Vedete [“Impostazione delle proprietà dell'applicazione iPhone in Flash Professional CS5”](#) a pagina 15 e [“Impostazione delle proprietà dell'applicazione iPhone nel file descrittore dell'applicazione”](#) a pagina 17.

Esistono quattro classi di oggetti di visualizzazione che possono eseguire rapidamente il rendering nell'accelerazione hardware se il loro contenuto cambia raramente:

- Oggetti bitmap
- Oggetti di visualizzazione 2D per i quali la proprietà `cacheAsBitmap` è impostata su `true` e la cui proprietà è facoltativamente impostata su `cacheAsBitmapMatrix` (vedete di seguito)
- Oggetti di visualizzazione 3D (ovvero, la cui proprietà `z` è impostata)
- Oggetti di visualizzazione dotati di un singolo riempimento rettangolare a tinta unita e bordi che si allineano con i pixel sullo schermo.

Gli oggetti vettoriali vengono nuovamente sottoposti a rendering ogni volta che un altro sprite si anima sopra o sotto di essi. Pertanto, qualsiasi oggetto che serve come sfondo o primo piano a un'animazione deve anche appartenere a una di queste categorie.

Per gli oggetti di visualizzazione per i quali la proprietà `cacheAsBitmap` è impostata su `true`, l'impostazione `cacheAsBitmapMatrix` determina l'utilizzo della bitmap che deriva dalla trasformazione della matrice. La GPU utilizza la rappresentazione bitmap anche se l'oggetto viene ruotato o ridimensionato. La GPU può creare e animare questa bitmap molto più rapidamente di quanto la CPU non possa ridisegnare un oggetto riprodotto vettorialmente.

La sola impostazione di `cacheAsBitmap` su `true` causa la memorizzazione nella cache di un oggetto di visualizzazione (e di eventuali elementi secondari). L'oggetto di visualizzazione non viene ridisegnato quando le nuove regioni vengono esposte o quando l'intero grafico combinato viene tradotto.

Quando la proprietà `cacheAsBitmapMatrix` dell'oggetto di visualizzazione è impostata, l'applicazione può creare una rappresentazione dell'oggetto di visualizzazione anche quando questo non è visibile. L'applicazione crea una rappresentazione memorizzata nella cache dell'oggetto di visualizzazione all'inizio del fotogramma successivo. Quando poi aggiungete l'oggetto di visualizzazione allo stage, l'applicazione effettua rapidamente il rendering. Inoltre, l'applicazione può rapidamente animare, ruotare o ridimensionare l'oggetto. Per oggetti che vengono ruotati o ridimensionati, non impostate la proprietà `cacheAsBitmap` senza impostare anche la proprietà `cacheAsBitmapMatrix`.

L'applicazione può anche eseguire rapidamente trasformazioni alfa su un oggetto memorizzato nella cache come una bitmap. Tuttavia, solo i valori `alpha` compresi tra 0 e 1.0 sono supportati per le trasformazioni alfa accelerate via hardware. Questo corrisponde a un'impostazione `colorTransform.alphaMultiplier` compresa tra 0 e 256.

Non impostate la proprietà `cacheAsBitmap` su `true` per oggetti aggiornati frequentemente, ad esempio i campi di testo.

Gli oggetti di visualizzazione il cui contenuto grafico viene aggiornato spesso sono in genere candidati non ideali per il rendering GPU, in particolare su dispositivi non recenti dotati di GPU meno potenti. L'overhead del caricamento di grafica nella GPU può consigliare l'utilizzo del rendering CPU.

Ristrutturate gli oggetti di visualizzazione contenenti oggetti di visualizzazione secondari che si spostano rispetto a quello principale. Modificateli in modo che gli oggetti di visualizzazione secondari diventino parenti di quello principale. Questo garantisce che ciascuno abbia la propria rappresentazione bitmap. Inoltre, ogni oggetto di visualizzazione sarà in grado di muoversi in relazione agli altri senza che sia necessario caricare nuova grafica nella GPU.

Impostate la proprietà `cacheAsBitmap` su `true` al livello più alto dell'elenco di visualizzazione dove gli oggetti di visualizzazione secondari non si animano. In altre parole, impostatela per contenitori oggetti di visualizzazione che non contengono parti in movimento. Non impostatela sugli oggetti di visualizzazione subordinati. *Non* impostatela per sprite contenenti altri oggetti di visualizzazione che si animano.

Quando impostate la proprietà `z` di un oggetto di visualizzazione, l'applicazione utilizza sempre una rappresentazione bitmap memorizzata nella cache. Inoltre, dopo che avete impostato la proprietà `z` di un oggetto di visualizzazione, l'applicazione utilizza la rappresentazione bitmap memorizzata nella cache, anche se ruotate o ridimensionate l'oggetto. L'applicazione non utilizza la proprietà `cacheAsBitmapMatrix` per oggetti di visualizzazione la cui proprietà `z` è impostata. Le stesse regole sono valide quando impostate le proprietà di qualsiasi oggetto di visualizzazione tridimensionale, comprese le proprietà `rotationX`, `rotationY`, `rotationZ` e `transform.matrix3D`.

Non impostate la proprietà `scrollRect` o `mask` di un contenitore di oggetti di visualizzazione per il quale desiderate utilizzare l'accelerazione hardware. L'impostazione di queste proprietà disabilita l'accelerazione hardware per il contenitore oggetti di visualizzazione e i suoi oggetti secondari. In alternativa all'impostazione della proprietà `mask`, stratificate un oggetto di visualizzazione utilizzato come maschera sull'oggetto di visualizzazione che viene mascherato.

Esistono dei limiti alle dimensioni degli oggetti di visualizzazione disponibili per l'accelerazione hardware. Nei dispositivi meno recenti, il limite è al massimo 1024 pixel per la larghezza e l'altezza. Nei dispositivi più recenti, il limite è al massimo 2048 pixel. Potete utilizzare lo strumento di diagnostica del rendering GPU per provare le prestazioni su un dispositivo.

La GPU utilizza anche la RAM dell'iPhone per archiviare immagini bitmap. Viene utilizzata la quantità di memoria minima necessaria per le immagini bitmap.

La GPU utilizza allocazioni di memoria che sono potenze di 2 per ogni dimensione dell'immagine bitmap. Ad esempio, la GPU può riservare memoria in dimensioni di 512 x 1024 o 8 x 32. Pertanto un'immagine di 9 x 15 pixel richiede la stessa quantità di memoria di un'immagine di 16 x 16 pixel. Per oggetti di visualizzazione memorizzati nella cache, potete utilizzare le dimensioni più vicine alla potenza di 2 (ma non superiori) in ogni direzione. Ad esempio, è più efficiente utilizzare un oggetto di visualizzazione da 32 x 16 pixel anziché uno da 33 x 17 pixel.

Non basatevi su uno Stage ridimensionato per ridurre in scala risorse che sono state dimensionate per altre piattaforme (ad esempio, il desktop). Piuttosto, utilizzate la proprietà `cacheAsBitmapMatrix`, oppure ridimensionate le risorse prima della pubblicazione per l'iPhone. Gli oggetti 3D ignorano la proprietà `cacheAsBitmapMatrix` quando si esegue la memorizzazione nella cache di un'immagine superficie. Per questo motivo, è meglio ridimensionare gli oggetti di visualizzazione prima della pubblicazione se non ne verrà eseguito il rendering su un superficie 3D.

Esiste un compromesso tra i vantaggi dell'accelerazione hardware e l'utilizzo della RAM. Man mano che la memoria si riempie, il sistema operativo dell'iPhone segnala ad altre applicazioni iPhone native in esecuzione di liberare memoria. Poiché le applicazioni elaborano questa notifica e operano per liberare memoria, è possibile che entrino in competizione con l'applicazione per le risorse di CPU. Questo può causare un peggioramento temporaneo delle prestazioni dell'applicazione. Accertatevi di provare l'applicazione nei dispositivi più vecchi in quanto potrebbero disporre di una quantità di memoria notevolmente inferiore per il processo in esecuzione.

Quando eseguite il debug dell'applicazione sull'iPhone, potete abilitare la funzione di diagnostica del rendering GPU. Questa funzione consente di vedere in che modo l'applicazione usa il rendering GPU. Per ulteriori informazioni, vedete "Debug con la diagnostica del rendering GPU" in "[Debug di un'applicazione iPhone](#)" a pagina 25.

Per informazioni sull'utilizzo della proprietà `cacheAsBitmapMatrix`, vedete la sezione "DisplayObject.cacheAsBitmapMatrix" in "[API di ActionScript specifiche per applicazioni AIR per dispositivi mobili](#)" a pagina 31.

Altri modi per migliorare le prestazioni degli oggetti di visualizzazione

L'accelerazione hardware può velocizzare le prestazioni grafiche in alcune classi di oggetti di visualizzazione. Di seguito vengono forniti alcuni suggerimenti su come ottimizzare le prestazioni grafiche:

- Provate a limitare il numero di elementi visibili sullo stage. Il rendering e la composizione di ciascun elemento richiede del tempo quando intorno sono presenti altri elementi.

Quando non è più necessario visualizzare un oggetto di visualizzazione, impostate la sua proprietà `visible` su `false` o rimuovetelo dallo stage (`removeChild()`). Non impostate semplicemente la sua proprietà `alpha` su 0.

- Evitate metodi di fusione in generale e i metodi di fusione Livello in particolare. Quando possibile, utilizzate il metodo di fusione normale.
- I filtri oggetti di visualizzazione sono onerosi in termini di risorse di calcolo. Utilizzateli sporadicamente. Ad esempio, l'uso di pochi filtri in una schermata iniziale è considerato accettabile. Tuttavia, evitate di utilizzare filtri su molti oggetti o su oggetti in fase di animazione o quando dovete utilizzare una frequenza fotogrammi elevata.
- Evitate forme di trasformazione.
- Evitate di utilizzare il ritaglio.

- Se possibile, impostate il parametro `repeat` su `false` durante la chiamata del metodo `Graphic.beginBitmapFill()`.
- Non eccedete nel disegno. Utilizzate il colore di sfondo come sfondo. Non sovrapponete forme di grandi dimensioni una sopra l'altra. Ad ogni pixel da disegnare è associato un costo, in particolare per gli oggetti di visualizzazione che non dispongono di hardware accelerato.
- Evitate forme con lunghe punte sottili, bordi che si auto intersecano o molti dettagli fini lungo i bordi. Il rendering di queste forme richiede più tempo rispetto a quello degli oggetti di visualizzazione con bordi arrotondati, in particolare per gli oggetti di visualizzazione che non dispongono di hardware accelerato.
- Create bitmap le cui dimensioni siano prossime, ma non inferiori, a $2^n \times 2^m$ bit. Le dimensioni devono essere prossime alla potenza di 2, senza che siano uguali o superiori. Ad esempio, il rendering di un'immagine 31x15 pixel viene eseguito più rapidamente rispetto a un'immagine 33x17 pixel. (31 e 15 sono appena inferiori alle potenze di 2: 32 e 16.) Tali immagini utilizzano inoltre la memoria in maniera più efficiente.
- Limitate le dimensioni degli oggetti di visualizzazione a 1024 x 1024 pixel (o 2048 x 2048 nei dispositivi più recenti).

Densità di informazione

Le dimensioni fisiche dello schermo dei dispositivi mobili sono inferiori a quelle del desktop, sebbene la densità dei pixel sia maggiore. Il testo nitido è piacevole da guardare, ma la dimensione fisica dei glifi deve essere minima per essere leggibile.

I dispositivi mobili sono spesso utilizzati durante gli spostamenti e in condizioni di scarsa illuminazione. Considerate la quantità di informazioni che è possibile visualizzare realisticamente sullo schermo in maniera leggibile. Potrebbe essere inferiore a quella visualizzata su uno schermo desktop avente le stesse dimensioni in pixel.

Utilizzate la gerarchia tipografica per evidenziare informazioni importanti. Utilizzate dimensione, spessore, posizionamento e spaziatura carattere per esprimere l'importanza relativa degli elementi dell'interfaccia utente. Potete utilizzare uno o più cue point ad ogni livello della gerarchia. Applicare questi cue point in maniera uniforme nell'applicazione. Un cue point può essere spaziale (rientro, interlinea, posizionamento) o grafico (dimensione, stile, colore del carattere tipografico). L'applicazione di cue point ridondanti può essere un modo efficace per accertarsi che la gerarchia sia espressa chiaramente. Tuttavia, provate a utilizzare un massimo di tre cue point per ogni livello di raggruppamento.

Provate a semplificare le etichette e il testo descrittivo richiesto. Ad esempio, utilizzate input di esempio nel campo di testo per suggerire il contenuto ed evitate un'etichetta separata.

Caratteri e immissione di testo

Per ottimizzare l'aspetto, utilizzate i caratteri dispositivo. Di seguito, ad esempio, sono elencati i caratteri dispositivo sull'iPhone:

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

Utilizzate caratteri di 14 pixel o più grandi.

Utilizzate caratteri dispositivo per i campi di testo modificabili. I caratteri dispositivo nei campi di testo vengono inoltre riprodotti più rapidamente dei caratteri incorporati.

Non utilizzate testo sottolineato per campi di testo di input. Inoltre, non impostate l'allineamento per il campo di testo. I campi di testo di input sull'iPhone supportano solo l'allineamento a sinistra (l'impostazione predefinita).

Se utilizzate l'impostazione Testo TLF per un campo di testo Flash Professional CS5, disattivate la libreria condivisa in runtime nel concatenamento predefinito nelle Impostazioni di ActionScript 3.0. In caso contrario, l'applicazione non funziona sull'iPhone poiché cerca di utilizzare il file SWF della libreria condivisa in runtime:

- 1 Selezionate File > Impostazioni pubblicazione.
- 2 Nella finestra di dialogo Impostazioni pubblicazione, fate clic sulla scheda Flash.
- 3 Fate clic sul pulsante Script a destra dell'elenco a discesa Script (ActionScript 3.0).
- 4 Fate clic sulla scheda Percorso di libreria.
- 5 Nell'elenco a discesa Concatenamento predefinito, selezionate Merged Into Code (Unito nel codice).

Valutate la possibilità di implementare delle alternative all'utilizzo dei campi di testo di input. Ad esempio, per fare in modo che l'utente immetta un valore numerico, non è necessario un campo di testo. Potete fornire due pulsanti per aumentare o diminuire il valore.

Tenete presente lo spazio utilizzato dalla tastiera virtuale. Quando la tastiera virtuale è attivata (ad esempio, quando un utente tocca un campo di testo), l'applicazione regola la posizione dello stage. Il riposizionamento automatico garantisce che il campo di testo di input selezionato sia visibile:

- Un campo di testo nella parte superiore dello stage si sposta in cima all'area dello stage visibile. (L'area dello stage visibile è più piccola per contenere la tastiera virtuale.)
- Un campo di testo nella parte inferiore dello stage rimane in fondo alla nuova area dello stage.
- Un campo di testo in un'altra parte dello stage viene spostata nel centro verticale dello stage.

Quando l'utente fa clic su un campo di testo per modificarlo (e la tastiera virtuale è visualizzata), l'oggetto TextField invia un evento `focusIn`. Potete aggiungere un listener di eventi per questo evento per riposizionare il campo di testo.

Un campo di testo a riga singola include un pulsante di cancellazione (a destra del testo) quando l'utente modifica il testo. Tuttavia, il pulsante di cancellazione non viene visualizzato se il campo di testo è troppo stretto.

Dopo la modifica del testo in un campo di testo a riga singola, l'utente rimuove la tastiera virtuale toccando il tasto Fine sulla tastiera.

Dopo la modifica del testo in un campo di testo a più righe, l'utente rimuove la tastiera virtuale toccando l'esterno del campo di testo. Questa operazione disattiva il campo di testo. Verificate che il design includa area al di fuori del campo di testo quando la tastiera virtuale è visualizzata. Se il campo di testo è troppo grande, è possibile che non venga visualizzata nessuna altra area.

L'utilizzo di alcuni componenti Flash Professional CS5 può impedire la disattivazione di un campo di testo. Questi componenti sono progettati per l'utilizzo sulle macchine desktop, dove questo comportamento di attivazione è utile. Uno di questi componenti è TextArea. Quando è attivato (e in fase di modifica), non potete disattivarlo facendo clic su un altro oggetto di visualizzazione. L'inserimento di altri componenti Flash Professional CS5 sullo stage può altresì impedire la disattivazione del campo di testo in fase di modifica.

Non basatevi sugli eventi di tastiera. Ad esempio, il contenuto SWF progettato per il Web utilizza la tastiera per consentire all'utente di controllare l'applicazione. Tuttavia, sull'iPhone, la tastiera virtuale è presente solo quando l'utente modifica un campo di testo. Un'applicazione iPhone invia eventi di tastiera solo quando la tastiera virtuale è presente.

Salvataggio dello stato dell'applicazione

L'applicazione potrebbe chiudersi in qualsiasi istante (ad esempio, quando squilla il telefono). Si consiglia di salvare lo stato dell'applicazione ad ogni modifica. Ad esempio, potete salvare le impostazioni in un file o un database nella directory di archiviazione dell'applicazione. In alternativa, potete salvare i dati in un oggetto condiviso locale. Potete quindi ripristinare lo stato dell'applicazione quando viene nuovamente avviata. Un'applicazione interrotta da una telefonata viene automaticamente riavviata al termine della chiamata.

Non basatevi sull'invio di un evento `exiting` da parte dell'oggetto `NativeApplication` quando l'applicazione viene chiusa, in quanto questa operazione potrebbe non essere eseguita.

Modifica dell'orientamento dello schermo

È possibile visualizzare il contenuto dell'iPhone con orientamento verticale o orizzontale. Valutate il modo in cui l'applicazione reagisce alle variazioni dell'orientamento dello schermo. Per ulteriori informazioni, vedete [Rilevamento dei cambiamenti orientamento schermo](#).

Destinazioni di tocco

Valutate le dimensioni delle destinazioni di tocco durante la progettazione di pulsanti e altri elementi dell'interfaccia utente che vengono toccati dall'utente. Si consiglia di rendere questi elementi sufficientemente larghi da poter essere comodamente attivati con un dito su uno schermo sensibile. Inoltre, accertarsi che lo spazio tra le destinazioni sia sufficiente. La dimensione delle destinazioni deve essere di circa 44 x 57 pixel.

Allocazione di memoria

L'operazione di allocazione di nuovi blocchi di memoria è costoso e inoltre potrebbe rallentare l'applicazione o causare ritardi di prestazione durante l'animazione o l'interazione quando viene attivato il `garbage collection`.

Provate a riciclare gli oggetti quando possibile, anziché eliminarli e crearne di nuovi.

Tenete presente che gli oggetti vettoriali richiedono meno memoria degli array. Vedete [Classe `Vector` e classe `Array` a confronto](#).

Per ulteriori informazioni sull'utilizzo di memoria, vedete [Risparmiare memoria](#).

API di disegno

Provate a non utilizzare le API di disegno `ActionScript` (la classe `Graphics`) per creare grafici. L'uso delle API di disegno crea oggetti dinamicamente sullo stage, quindi ne esegue il rendering nel sistema di rasterizzazione. Se possibile, create invece questi oggetti sullo stage in maniera statica al momento della creazione.

Gli oggetti creati utilizzando API di disegno, se chiamati ripetutamente, vengono distrutti e ricreati ogni volta che si esegue `ActionScript`. Tuttavia, gli oggetti statici risiedono in memoria mediante linee temporali diverse.

Bubbling degli eventi

Per un contenitore dell'oggetto di visualizzazione altamente nidificato, il bubbling degli eventi può essere costoso. Potete ridurre questa spesa gestendo gli eventi completamente nell'oggetto target e chiamando il metodo `stopPropagation()` dell'oggetto evento. La chiamata di questo metodo impedisce il bubbling dell'evento e inoltre significa che gli oggetti principali non ricevono l'evento.

Potete realizzare guadagni correlati riducendo la nidificazione dell'oggetto di visualizzazione.

Se possibile, effettuate la registrazione per gli eventi `MouseEvent` anziché gli eventi `TouchEvent`. Gli eventi `MouseEvent` utilizzano meno carico di lavoro del processore rispetto agli eventi `TouchEvent`.

Se possibile, impostate le proprietà `mouseEnabled` e `mouseChildren` su `false`.

Ottimizzazione delle prestazioni video

Per ottimizzare la riproduzione dei video per dispositivi mobili, verificate che le altre applicazioni siano chiuse durante la riproduzione del video. Questo consente ai processi di rendering e di decodifica del video di utilizzare tutte le risorse di CPU necessarie.

Durante la riproduzione del video verificate che la quantità di codice `ActionScript` in esecuzione sia minima o nulla. Evitate codice che viene eseguito a intervalli frequenti o sulla timeline.

Riducete al minimo l'aggiornamento di oggetti di visualizzazione non video. Evitate in particolare l'aggiornamento di oggetti di visualizzazione che si intersecano con l'area video. Questo vale anche se tali oggetti sono nascosti sotto il video in quanto verranno ancora ridisegnati e utilizzano risorse di elaborazione. Ad esempio, utilizzate forme semplici per l'indicatore di posizione e aggiornatelo solo un paio di volte al secondo anziché ad ogni fotogramma. Fate in modo che i controlli video non si sovrappongano con l'area video; inseriteli direttamente sotto. Se disponete di un'animazione di bufferizzazione video, non nascondetela dietro il video quando non è utilizzata; impostatela invece in modo da renderla invisibile.

Componenti Flex e Flash

Numerosi componenti Flex e Flash sono progettati per l'utilizzo nelle applicazioni desktop. Questi componenti, in particolare quelli di visualizzazione, può essere eseguiti più lentamente sui dispositivi mobili. Oltre a questo, i componenti desktop potrebbero disporre di modelli di interazioni non appropriati per i dispositivi mobili.

Adobe sta sviluppando una versione del framework Flex ottimizzata per i dispositivi mobili. Per ulteriori informazioni, vedete <http://labs.adobe.com/technologies/flex/mobile/>.

Riduzione delle dimensioni del file dell'applicazione

Di seguito sono riportati alcuni suggerimenti su come ridurre la dimensione del file IPA:

- Controllate le bitmap di sfondo per verificare che siano delle dimensioni corrette (non più grandi del necessario).
- Controllate se vengono incorporati font aggiuntivi.

Considerazioni sulla progettazione di applicazioni iPhone

- Ricercate eventuali canali alfa nelle risorse PNG e rimuoveteli se non sono necessari. Utilizzate un'utilità come PNG crunch per ridurre la dimensione delle risorse PNG.
- Convertite le risorse PNG in risorse JPG, se possibile.
- Valutate la possibilità di comprimere i file audio (utilizzando una velocità di trasferimento più bassa)
- Rimuovete eventuali risorse non utilizzate.