

Création d'applications Adobe AIR à l'aide de Packager for iPhone

Informations juridiques

Vous trouverez des informations juridiques à l'adresse http://help.adobe.com/fr_FR/legalnotices/index.html.

Sommaire

Chapitre 1 : Création d'applications AIR pour iPhone

Concepts importants	1
Obtention d'outils de développement auprès d'Adobe	4
Obtention de fichiers de développement auprès d'Apple	5
Création d'une application iPhone Hello World dans Flash Professional CS5	8

Chapitre 2 : Compilation et débogage d'applications iPhone

Icône et images de l'écran initial d'un iPhone	13
Paramètres de l'application iPhone	15
Compilation d'un fichier d'installation d'application iPhone (IPA)	21
Installation d'une application iPhone	24
Débogage d'une application iPhone	25
Envoi de l'application iPhone à l'App Store	28

Chapitre 3 : Prise en charge des périphériques mobiles dans les API d'ActionScript 3.0

API d'ActionScript 3.0 non prises en charge sur les périphériques mobiles	29
API d'ActionScript propres aux applications AIR mobiles	31
API d'ActionScript 3.0 particulièrement adaptées aux développeurs d'applications mobiles	36

Chapitre 4 : Considérations liées à la conception d'applications iPhone

Accélération matérielle	37
Autres techniques d'optimisation des performances des objets d'affichage	40
Densité des informations	40
Polices et entrée de texte	41
Enregistrement de l'état de l'application	42
Changements d'orientation de l'écran	42
Cibles tactiles	43
Affectation de la mémoire	43
API de dessin	43
Propagation vers le haut (bubbling) des événements	43
Optimisation des performances vidéo	44
Composants Flex et Flash	44
Réduction de la taille d'un fichier d'application	44

Chapitre 1 : Création d'applications AIR pour iPhone

Vous pouvez utiliser les outils de la plate-forme Adobe® Flash® et le code ActionScript® 3.0 pour créer des applications Adobe® AIR® pour iPhone et iPod touch. Ces applications sont distribuées, installées et exécutées comme toute autre application iPhone.

Remarque : dans ce document, « iPhone » se rapporte à l'iPhone et à l'iPod touch.

L'outil Packager for iPhone® est intégré à Adobe® Flash® Professional CS5. Packager for iPhone compile le bytecode ActionScript 3.0 en code d'application iPhone natif. Les applications iPhone sont distribuées sous forme de fichiers d'installation d'application iPhone (fichiers .ipa) via l'iTunes Store.

Vous disposez de Flash Professional CS5 ou d'Adobe® Flash® Builder™ 4 pour modifier le contenu ActionScript 3.0 source associé à l'application.

Pour développer des applications iPhone, faites appel à Flash Professional CS5.

Vous devez également obtenir des certificats de développement iPhone auprès d'Apple.

Important : avant de développer des applications iPhone, passez en revue les informations relatives à la conception d'applications pour l'iPhone (voir « [Considérations liées à la conception d'applications iPhone](#) » à la page 37).

Familiarisez-vous également avec les fichiers de développement requis pour créer une application iPhone (voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5).

Concepts importants

Vous devez impérativement assimiler les concepts et flux de travaux associés au développement d'une application iPhone à l'aide d'ActionScript 3.0 avant de démarrer toute tâche correspondante.

Glossaire

Il est impératif d'assimiler les termes ci-dessous lors de la création d'une application iPhone.

Site du centre des développeurs iPhone Site Web d'Apple Computer (<http://developer.apple.com/iphone/>) permettant d'effectuer les opérations suivantes :

- Demander à devenir un développeur iPhone
- Gérer et créer des certificats de développement, des profils de configuration et des ID d'application (définis ci-dessous) iPhone
- Envoyer des applications destinées à l'App Store

Certificat de développement iPhone Permet d'identifier un développeur dans le cadre du développement d'applications.

Le fichier correspondant vous est transmis par Apple. Vous convertissez ce certificat en fichier de certificat P12 pour signer l'application iPhone créée à l'aide d'ActionScript 3.0 (voir *Fichier de certificat P12*).

Il est inutile de disposer d'un certificat de développement iPhone si vous souhaitez uniquement déboguer et tester des applications Flash Professional CS5 sur l'ordinateur de développement. En revanche, pour installer et tester l'application sur un iPhone, vous devez impérativement disposer d'un certificat de développement.

Le certificat de développement est différent d'un certificat de distribution, qui permet de créer une version définitive de l'application. Vous obtenez un certificat de distribution auprès d'Apple lorsque vous créez une version définitive de l'application.

Demande de signature de certificat Fichier contenant les informations personnelles requises pour générer un certificat de développement. Il porte également le nom de fichier CSR.

Profil de configuration Fichier permettant de tester ou distribuer une application iPhone. Vous obtenez les fichiers du profil de configuration auprès d'Apple. Un profil de configuration est affecté à un certificat de développement déterminé, un ID d'application et un ou plusieurs ID de périphérique. Plusieurs types de profils de configuration sont proposés :

- **Profil de configuration pour le développement d'applications** : permet d'installer une version de test d'une application sur l'iPhone du développeur.
- **Profil de configuration pour le test d'applications** : également appelé profil de configuration ad-hoc. Permet de distribuer une version de test de l'application à plusieurs utilisateurs (et périphériques iPhone). S'ils disposent de ce profil de configuration et de l'application de test, les utilisateurs sont en mesure de tester l'application sans qu'elle ne soit envoyée à l'App Store. Remarque : un profil de configuration pour le développement d'applications permet également de distribuer des applications de test sur plusieurs périphériques.
- **Profil de configuration pour la distribution d'applications** : permet de créer une application iPhone à envoyer à l'App Store.

ID d'application Chaîne unique qui identifie une ou plusieurs applications iPhone créées par un développeur donné. Vous créez les ID d'application sur le site du centre des développeurs iPhone. A chaque profil de configuration correspond un ID d'application ou un modèle d'ID d'application. Cet ID (ou modèle d'ID) d'application est utilisé lors du développement d'application. Vous le définissez dans la boîte de dialogue Paramètres de l'iPhone de Flash Professional CS5 (ou dans le fichier descripteur d'application).

Dans le centre des développeurs iPhone, les ID d'application contiennent un identifiant d'application (préfixe) suivi d'un identifiant d'application (suffixe). L'identifiant d'application (préfixe) est une chaîne de caractères, telle que 5RM86Z4DJM, affectée par Apple à l'ID d'application. L'identifiant d'application (suffixe) contient une chaîne de nom de domaine inversé que vous sélectionnez. Il se termine parfois par un astérisque (*), qui représente un ID d'application contenant un caractère générique. Exemples :

- 5RM86Z4DJM.com.exemple.helloWorld
- 96LPVWEASL.com.exemple.* (ID d'application contenant un caractère générique)

Le centre des développeurs iPhone propose deux types d'ID d'application :

- ID d'application contenant un caractère générique : dans le centre des développeurs iPhone, ces ID d'application se terminent par un astérisque (*) (96LPVWEASL.com.monDomaine.* ou 96LPVWEASL.*, par exemple). Si un profil de configuration fait appel à ce type d'ID d'application, vous pouvez générer des applications de test dont l'ID d'application correspond au modèle défini. Pour générer l'ID de l'application, vous pouvez remplacer l'astérisque par toute chaîne de caractères valides. Ainsi, si le centre des développeurs iPhone stipule l'ID d'application 96LPVWEASL.com.exemple.*, vous pouvez utiliser com.exemple.foo ou com.exemple.bar en tant qu'ID de l'application.

- ID d'application spécifique : une application utilise alors un ID d'application unique. Dans le centre des développeurs iPhone, ces ID d'application ne se terminent pas par un astérisque. Exemple : 96LPVWEASL.com.monDomaine.monApp. Si un profil de configuration fait appel à ce type d'ID d'application, les applications doivent correspondre exactement à l'ID d'application indiqué. Ainsi, si le centre des développeurs iPhone stipule l'ID d'application 96LPVWEASL.com.exemple.helloWorld, vous devez impérativement utiliser l'ID com.exemple.foo pour l'application.

Lorsque vous développez une application, vous stipulez son ID dans la boîte de dialogue Paramètres d'iPhone de Flash Professional CS5 ou dans le fichier descripteur de l'application. Pour plus d'informations sur les ID d'application, voir la section « Onglet Déploiement » de « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15 ou « [Définition des propriétés d'une application iPhone dans le fichier descripteur correspondant](#) » à la page 17.

Important : lorsque vous stipulez l'ID d'application, ignorez la section identifiant d'application (préfixe) de l'ID d'application. Ainsi, si Apple affecte l'ID 96LPVWEASL.com.exemple.jean.monApp à l'application, ignorez 96LPVWEASL. En d'autres termes, utilisez l'ID d'application com.exemple.jean.monApp. Si Apple stipule l'ID d'application 5RM86Z4DJM.*, ignorez 5RM86Z4DJM, qui correspond à un ID d'application contenant un caractère générique.

Vous trouverez l'ID d'application (ou le modèle d'ID d'application contenant un caractère générique) associé à un profil de configuration dans le centre des développeurs iPhone (<http://developer.apple.com/iphone>). Accédez au portail iPhone Developer Program, puis à la section Provisioning.

Fichier de certificat P12 Un fichier P12 (doté de l'extension .p12) est un type de fichier de certificat, soit un fichier d'échange d'informations personnelles. L'outil Packager for iPhone fait appel à ce type de certificat pour créer une application iPhone. Vous convertissez le certificat de développement reçu d'Apple en fichier de certificat P12.

ID unique de périphérique Code unique qui identifie un iPhone donné, également appelé UDID ou ID de périphérique.

Présentation du flux de travaux de développement

Procédez comme suit pour développer une application iPhone :

- 1 Installez Flash Professional CS5 d'Adobe.
- 2 Installez iTunes.
- 3 Procurez-vous les fichiers de développement auprès d'Apple. Parmi ces fichiers figurent le certificat de développement et les profils de configuration (voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5).
- 4 Convertissez le certificat de développement en fichier de certificat P12. Flash CS5 requiert que le certificat soit au format P12 (voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5).
- 5 Associez le profil de configuration à l'iPhone via iTunes.
- 6 Programmez l'application dans Flash Professional CS5.

Vous devez impérativement maîtriser les recommandations de bonne pratique relatives à la conception et à l'optimisation du code d'une application iPhone (voir « [Considérations liées à la conception d'applications iPhone](#) » à la page 37).

Par ailleurs, certaines API d'ActionScript 3.0 ne sont que partiellement prises en charge sur l'iPhone, tandis que d'autres ne le sont pas du tout (voir « [Prise en charge des périphériques mobiles dans les API d'ActionScript 3.0](#) » à la page 29).

Vous pouvez également modifier le code ActionScript 3.0 de l'application par le biais de Flash Builder 4.0.

Flash Professional CS5 permet de tester l'application sur l'ordinateur de développement.

- 7 Créez les icônes et les graphiques de l'écran initial de l'application. Chaque application iPhone contient un jeu d'icônes qui permet aux utilisateurs de l'identifier. L'iPhone affiche l'image de l'écran initial lors du chargement du programme (voir « [Icône et images de l'écran initial d'un iPhone](#) » à la page 13).
- 8 Modifiez les paramètres de l'iPhone. Parmi ces paramètres figurent notamment :
 - Identité de l'application (nom de fichier, nom de l'application, numéro de version et ID d'application)
 - Emplacement des graphiques de l'icône source de l'application
 - Certificat P12 et profil de configuration affecté à l'application
 - Format initial de l'application

Dans Flash Professional CS5, vous pouvez modifier ces paramètres dans la boîte de dialogue Paramètres d'iPhone. Pour plus d'informations, voir « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15.

Vous pouvez également modifier ces paramètres directement dans le fichier descripteur de l'application. Pour plus d'informations, voir « [Définition des propriétés d'une application iPhone dans le fichier descripteur correspondant](#) » à la page 17.

- 9 Compilez le fichier IPA à l'aide de l'outil Packager for iPhone. (voir « [Compilation d'un fichier d'installation d'application iPhone \(IPA\)](#) » à la page 21).

- 10 Installez et testez l'application sur l'iPhone. Installez le fichier IPA via iTunes.

Pour effectuer une distribution ad hoc, répétez ce processus général, mais substituez un profil de configuration pour le test d'applications au profil de configuration pour le développement d'applications. Pour effectuer la distribution définitive de l'application, répétez cette procédure en utilisant le profil de configuration pour la distribution d'applications. (Pour plus d'informations sur les divers types de profils de configuration, voir le « [Glossaire](#) » à la page 1.)

Une fois la version de distribution de l'application générée, voir « [Envoi de l'application iPhone à l'App Store](#) » à la page 28 pour plus d'informations.

Pour accéder à un didacticiel rapide consacré à la création d'une application iPhone de base, voir « [Création d'une application iPhone Hello World dans Flash Professional CS5](#) » à la page 8.

Obtention d'outils de développement auprès d'Adobe

Pour développer des applications iPhone en ActionScript 3.0, vous devez disposer de Flash Professional CS5.

Important : vous devez mettre à jour la version de Packager for iPhone à partir de la version d'évaluation incluse dans Flash Professional CS5. Dans Flash Professional CS5, sélectionnez Aide > Mises à jour.

Vous pouvez également modifier le code ActionScript avec Flash Builder 4. Flash Builder 4 est disponible à l'adresse suivante : <http://www.adobe.com/fr/products/flashbuilder/>.

Obtention de fichiers de développement auprès d'Apple

Pour développer toute application iPhone, vous devez au préalable vous procurer des fichiers de développement iPhone auprès d'Apple. Vous devez en particulier obtenir un certificat de développement iPhone et un profil de configuration mobile, ainsi que d'autres profils de configuration. Pour connaître la définition de ces fichiers, voir le « [Glossaire](#) » à la page 1.

***Remarque :** obtenir ces fichiers représente une étape importante du processus de développement d'application. Veillez à la compléter avant de développer l'application. L'obtention des fichiers de développement n'a rien de simple. Lisez attentivement ces instructions, ainsi que les instructions figurant sur le site du centre des développeurs iPhone d'Apple.*

Obtention et utilisation des fichiers de développement iPhone

Vous devez obtenir un certificat de développement iPhone et des profils de configuration auprès d'Apple. Vous devez également convertir le certificat en certificat P12.

Installation d'iTunes

Vous devez disposer d'iTunes pour installer l'application sur l'iPhone. iTunes permet aussi d'identifier l'ID de périphérique de l'iPhone. Vous devez également connaître l'ID de périphérique lorsque vous demandez un certificat de développement iPhone.

Demande de certificat de développement iPhone et création d'un profil de configuration

Le cas échéant, enregistrez-vous en tant que développeur iPhone homologué sur le centre des développeurs iPhone d'Apple (<http://developer.apple.com/iphone/>).

***Remarque :** il est inutile de disposer de XCode ou du kit SDK d'iPhone pour développer des applications AIR pour l'iPhone. Vous devez être un développeur iPhone homologué. Vous devez également obtenir un certificat de développement et un profil de configuration.*

- 1 Connectez-vous au centre des développeurs iPhone en entrant votre ID de compte de développeur iPhone.
- 2 Sur le site des développeurs iPhone, demandez (et achetez) un certificat de développement iPhone.
Vous recevrez un E-mail d'Apple contenant le code d'activation du programme pour développeurs iPhone.
- 3 Retournez au centre des développeurs iPhone. Suivez les instructions d'activation du programme pour développeurs et entrez le code d'activation lorsque vous y êtes invité.
- 4 Lorsque le code d'activation est accepté, accédez au portail iPhone Developer Program du centre des développeurs iPhone.
- 5 Créez un fichier de demande de signature de certificat. Vous utiliserez ce fichier pour obtenir un certificat de développement iPhone. Pour plus d'informations, voir « [Génération d'une demande de signature de certificat](#) » à la page 6.
- 6 Dans le cadre de l'étape suivante, vous serez invité à entrer l'ID de périphérique (ou l'UDID, ID unique de périphérique) de l'iPhone. Vous pouvez vous procurer l'UDID auprès d'iTunes :
 - a Connectez l'iPhone par le biais d'un câble USB. Dans iTunes, sélectionnez l'onglet résumé associé à l'iPhone.
 - b Une fois le profil de configuration téléchargé du site du centre des développeurs iPhone, ajoutez-le à iTunes.
 - c Cliquez ensuite sur le numéro de série affiché. La valeur UDID est maintenant affichée. Cliquez sur Commande-C sous Mac ou Contrôle-C sous Windows pour copier l'UDID dans le Presse-papiers.

7 Créez et installez un profil de configuration et un certificat de développement iPhone.

Suivez les instructions affichées dans le centre des développeurs iPhone. Recherchez des instructions sur le portail iPhone Developer Program. Le cas échéant, faites appel à l'assistant Development Provisioning Assistant pour obtenir un certificat de développement et créer un profil de configuration.

Ignorez les étapes associées à XCode. Il est inutile d'utiliser XCode pour développer des applications iPhone dans Flash Professional CS5.

8 Dans iTunes, sélectionnez Fichier > Ajouter le fichier à la bibliothèque. Sélectionnez ensuite le fichier du profil de configuration (dont l'extension correspond à mobileprovision). Synchronisez à présent l'iPhone avec iTunes.

Vous pouvez ainsi tester l'application associée au profil de configuration sur l'iPhone.

Pour vérifier qu'un profil de configuration donné a été ajouté à iTunes, vous pouvez tenter de l'intégrer à la bibliothèque. Si vous êtes invité par iTunes à remplacer un profil de configuration existant, vous pouvez appuyer sur le bouton Annuler. Le profil est en effet déjà installé. Vous pouvez également vérifier les profils de configuration installés sur l'iPhone :

- a Ouvrez l'application Réglages sur l'iPhone.
- b Ouvrez la catégorie Général.
- c Touchez Profil. La page Profil recense les profils de configuration installés.

9 Le cas échéant, téléchargez le fichier du certificat de développement iPhone (fichier .cer).

L'assistant Development Provisioning vous a peut-être indiqué un lien pour télécharger ce fichier. Vous trouverez également ce fichier à la section Certificats du portail Provisioning du site du centre des développeurs iPhone d'Apple (<http://developer.apple.com/iphone/>).

10 Vous devez ensuite convertir le certificat de développement iPhone en fichier P12. Pour plus d'informations, voir « [Conversion d'un certificat de développement en fichier P12](#) » à la page 7.

Vous pouvez maintenant créer une application Hello World simple. Voir « [Création d'une application iPhone Hello World dans Flash Professional CS5](#) » à la page 8.

Génération d'une demande de signature de certificat

Pour obtenir un certificat de développement, vous générez un fichier de demande de signature de certificat, que vous envoyez au site du centre des développeurs iPhone d'Apple.

Génération d'une demande de signature de certificat sous Mac OS

Sous Mac OS, vous disposez de l'application Trousseau d'accès pour générer une demande de signature de code. L'application Trousseau d'accès réside dans le sous-répertoire Utilitaires du répertoire Applications. Dans le menu Trousseau d'accès, sélectionnez Assistant de certification > Demander un certificat à une autorité de certificat.

- 1 Ouvrez Trousseau d'accès.
- 2 Dans le menu Trousseau d'accès, sélectionnez Préférences.
- 3 Dans la boîte de dialogue Préférences, cliquez sur Certificats. Désactivez ensuite les options Protocole d'état des cert. en ligne et Liste des certificats révoqués. Fermez la boîte de dialogue.
- 4 Dans le menu Trousseau d'accès, sélectionnez Assistant de certification > Demander un certificat à une autorité de certificat.
- 5 Entrez l'adresse électronique et le nom correspondant à votre ID de compte de développeur iPhone. N'entrez pas d'adresse électronique CA. Sélectionnez l'option d'enregistrement sur disque, puis cliquez sur le bouton Continuer.

- 6 Enregistrez le fichier (CertificateSigningRequest.certSigningRequest).
- 7 Téléchargez le fichier CSR sur le [site du centre des développeurs iPhone d'Apple](#). (Voir « Demande de certificat de développement iPhone et création d'un profil de configuration ».)

Génération d'une demande de signature de certificat sous Windows

Il est recommandé aux développeurs Windows d'obtenir le certificat de développement iPhone sur un ordinateur Mac. Il leur est toutefois possible d'obtenir ce certificat sous Windows. Commencez par créer une demande de signature de certificat (fichier CSR) par le biais d'OpenSSL en procédant comme suit :

- 1 Installez OpenSSL sur l'ordinateur Windows. (Accédez à <http://www.openssl.org/related/binaries.html>.)

Il sera peut-être nécessaire d'installer également les fichiers redistribuables Visual C++ 2008, recensés sur la page de téléchargement OpenSSL. (Il est toutefois *inutile* d'installer Visual C++ sur l'ordinateur.)

- 2 Ouvrez une session de commande Windows et accédez au répertoire bin d'OpenSSL (c:\OpenSSL\bin\, par exemple).
- 3 Créez la clé privée en entrant le texte ci-dessous sur la ligne de commande :

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

Enregistrez cette clé privée. Vous en aurez besoin ultérieurement.

Lorsque vous utilisez OpenSSL, tenez compte de tous les messages d'erreur. Même si OpenSSL génère un message d'erreur, il est possible que la sortie de fichiers continue. Ces fichiers risquent toutefois d'être inutilisables. Si des messages d'erreur sont générés, vérifiez la syntaxe et exécutez à nouveau la commande.

- 4 Créez le fichier CSR en entrant le texte ci-dessous sur la ligne de commande :

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

Remplacez l'adresse électronique, la valeur CN (nom du certificat) et la valeur C (pays) par vos coordonnées.

- 5 Téléchargez le fichier CSR sur le [site du centre des développeurs iPhone d'Apple](#). (Voir « Demande de certificat de développement iPhone et création d'un profil de configuration ».)

Conversion d'un certificat de développement en fichier P12

Pour développer des applications iPhone dans Flash Professional CS5, vous devez disposer d'un certificat P12. Vous générez ce certificat à partir du fichier de certificat de développement iPhone envoyé par Apple.

Conversion du certificat de développement iPhone en fichier P12 sous Mac OS

Une fois le certificat iPhone téléchargé d'Apple, exportez-le au format P12. Procédez comme suit sous Mac® OS :

- 1 Ouvrez l'application Trousseau d'accès (qui réside dans le dossier Applications/Utilitaires).
- 2 Si vous n'avez pas encore ajouté le certificat au trousseau, sélectionnez Fichier > Importer. Accédez ensuite au fichier de certificat (fichier .cer) que vous avez obtenu d'Apple.
- 3 Sélectionnez la catégorie Clés dans Trousseau d'accès.
- 4 Sélectionnez la clé privée associée au certificat de développement iPhone.
La clé privée est identifiée par le certificat public du développeur iPhone : <Prénom> <Nom> auquel elle est associée.
- 5 Sélectionnez Fichier > Exporter des éléments.
- 6 Enregistrez la clé au format de fichier Echange d'informations personnelles (.p12).

7 Vous êtes invité à créer un mot de passe utilisé lorsque vous essayez d'importer la clé sur un autre ordinateur.

Conversion d'un certificat de développement Apple en fichier P12 sous Windows

Pour développer des applications iPhone dans Flash CS5, vous devez disposer d'un certificat P12. Vous générez ce certificat à partir du fichier de certificat de développement iPhone envoyé par Apple.

1 Convertissez le certificat de développement reçu d'Apple en certificat PEM. Exécutez l'instruction de ligne de commande suivante à partir du répertoire bin d'OpenSSL :

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

2 Si vous utilisez la clé privée extraite du trousseau sur un ordinateur Mac, convertissez-la en clé PEM :

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

3 Vous pouvez maintenant générer un fichier P12 valide basé sur la clé et la version PEM du certificat de développement iPhone :

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

Si vous utilisez une clé issue du trousseau Mac OS, utilisez la version PEM générée à l'étape précédente. Si tel n'est pas le cas, utilisez la clé OpenSSL générée précédemment (sous Windows).

Gestion des certificats, ID de périphérique, ID d'application et profils de configuration

Vous pouvez gérer les certificats, ID de périphérique, ID d'application et profils de configuration sur le site du centre des développeurs iPhone d'Apple (<http://developer.apple.com/iphone/>). Accédez au portail iPhone Developer Program du site.

- Cliquez sur le lien Certificates pour gérer les certificats de développement. Vous pouvez créer, télécharger ou révoquer un certificat. Pour créer un certificat, vous devez générer au préalable une demande de signature de certificat (voir « Génération d'une demande de signature de certificat » à la page 6).
- Cliquez sur le lien Devices pour gérer la liste des périphériques sur lesquels peut être installée l'application de développement.
- Cliquez sur le lien App IDs pour gérer les ID d'application. Lorsque vous créez un profil de configuration, il est lié à un ID d'application.
- Cliquez sur le lien Provisioning pour gérer les profils de configuration. Vous disposez également de l'assistant Development Provisioning Assistant pour créer des profils de configuration pour le développement d'applications.
- Cliquez sur le lien Distribution pour envoyer l'application à l'App Store ou en créer une version Ad Hoc. Cette section contient un lien vers le site iTunes Connect, qui permet d'envoyer une application à l'App Store.

Création d'une application iPhone Hello World dans Flash Professional CS5

Important : avant de créer l'application, téléchargez les fichiers et applications pour développeur requis. Pour plus d'informations, voir « Obtention d'outils de développement auprès d'Adobe » à la page 4 et « Obtention de fichiers de développement auprès d'Apple » à la page 5.

Création d'un projet Flash Professional CS5

Pour créer une application iPhone directement dans Flash Professional CS5, procédez comme suit :

- 1 Ouvrez Flash CS5.
- 2 Sélectionnez Fichier > Nouveau.
- 3 Sélectionnez iPhone OS.
- 4 Cliquez sur le bouton OK.

Ajout de contenu à l'application

Ajoutez ensuite le texte « Hello world! » à l'application en procédant comme suit :

- 1 Sélectionnez l'outil Texte et cliquez sur la scène.
- 2 Dans les paramètres Propriétés du champ de texte, sélectionnez Texte classique (et non Texte TLF).
Etant donné qu'il s'agit d'une application de base, l'option Texte classique est appropriée. Pour utiliser l'option Texte TLF, vous devez activer quelques autres paramètres (voir « [Polices et entrée de texte](#) » à la page 41).
- 3 Dans le nouveau champ de texte, saisissez « Hello World! ».
- 4 Sélectionnez le champ de texte à l'aide de l'outil Sélection.
- 5 Ouvrez ensuite l'inspecteur Propriétés et définissez les paramètres suivants :
 - Caractère > Famille : -sans
 - Caractère > Taille : 50
 - Position > X2: 20
 - Position > Y : 20
- 6 Enregistrez le fichier.
- 7 Sélectionnez Contrôle > Tester l'animation > dans l'application de débogage du lanceur AIR (Mobile).
Flash Professional CS5 compile le contenu SWF et affiche une version de l'application dans l'application de débogage du lanceur AIR (ADL). Vous obtenez ainsi un bref aperçu de l'application.

Création des icônes et des graphiques de l'écran initial de l'application

Toutes les applications iPhone contiennent des icônes affichées dans leur interface utilisateur et sur l'écran de l'iPhone.

- 1 Créez un répertoire au sein du répertoire de projet et affectez-lui le nom « icons ».
- 2 Créez trois fichiers PNG dans le répertoire icons, Icon29.png, Icon57.png et Icon512.png.
- 3 Modifiez les fichiers PNG pour créer des graphiques adaptés à l'application. Les fichiers doivent mesurer 29 pixels sur 29 pixels, 57 pixels sur 57 pixels et 512 pixels sur 512 pixels. Dans le cadre de ce test, vous pouvez vous contenter de carrés de couleur unie.

Toute application iPhone affiche une image initiale lors de son chargement sur l'iPhone. Vous définissez cette image initiale dans un fichier PNG, comme suit :

- 1 Dans le répertoire de développement principal, créez un fichier PNG appelé Default.png. (Ne placez *pas* ce fichier dans le sous-répertoire icons et respectez la casse du nom du fichier.)
- 2 Modifiez le fichier de sorte que ses dimensions correspondent à 320 pixels de large sur 480 pixels de haut. Pour le moment, contentez-vous d'un contenu composé d'un rectangle blanc uni, que vous modifierez ultérieurement.

Remarque : lorsque vous envoyez une application à l'App Store d'Apple, vous utilisez une version JPG (et non PNG) du fichier de 512 pixels. La version PNG est réservée au test des versions de développement d'une application.

Pour plus d'informations sur ces graphiques, voir « [Icône et images de l'écran initial d'un iPhone](#) » à la page 13.

Modification des paramètres de l'application

Important : le cas échéant, téléchargez les fichiers et applications nécessaires au développement d'applications iPhone (voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5).

La boîte de dialogue Paramètres d'iPhone de Flash Professional CS5 permet de définir diverses propriétés de base de l'application iPhone.

1 Sélectionnez Fichier > Paramètres d'iPhone.

2 Sur l'onglet Général, définissez les paramètres suivants :

- Fichier de sortie : HelloWorld.ipa

Nom du fichier d'installation iPhone à générer.

- Nom de l'application : Hello World

Nom de l'application affiché sous l'icône de l'application sur l'iPhone.

- Version : 1.0

Version de l'application.

- Format : portrait

- Activez l'option Plein écran.

- Désactivez l'option Orientation auto.

- Rendu : CPU

Les autres options, GPU et Auto, font appel à l'accélération matérielle pour le rendu. Cette fonction permet d'optimiser les performances des applications utilisant intensivement les graphiques (les jeux, par exemple) conçues pour exploiter l'accélération matérielle. Pour plus d'informations, voir « [Accélération matérielle](#) » à la page 37.

- Fichiers inclus : ajoutez le fichier contenant les graphiques de l'écran initial (Default.png) à la liste Fichiers inclus.

Remarque : dans le cadre de cet exemple Hello World, ne modifiez pas les paramètres définis dans les instructions. Certains paramètres, tels que Version, sont sujets à des restrictions spécifiques. Ces restrictions sont décrites dans « [Paramètres de l'application iPhone](#) » à la page 15.

3 Sur l'onglet Déploiement, définissez les paramètres suivants :

- Certificat : recherchez et sélectionnez le certificat P12 basé sur le certificat de développement obtenu auprès d'Apple.

Ce certificat permet de signer le fichier. Vous devez convertir le certificat iPhone d'Apple au format .p12. Pour plus d'informations, voir « [Obtention d'outils de développement auprès d'Adobe](#) » à la page 4.

- Mot de passe : entrez le mot de passe associé au certificat.

- Fichier de configuration : recherchez et sélectionnez le fichier de configuration de développement obtenu auprès d'Apple (voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5).

- ID de l'application : si vous pouvez sélectionner ce champ, entrez l'ID d'application fourni à Apple (com.example.as3.HelloWorld, par exemple).

L'ID identifie de manière unique une application.

S'il est impossible de sélectionner le champ, le profil de configuration est lié à un ID d'application donné. L'ID de l'application est alors affiché dans le champ.

Pour plus d'informations sur la définition d'un ID d'application, voir la section « Onglet Déploiement » de « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15.

- 4 Dans la liste Icônes de l'onglet Icônes, cliquez sur Icône 29 x 29. Indiquez ensuite l'emplacement du fichier PNG de 29 x 29 pixels créé précédemment (voir « [Création des icônes et des graphiques de l'écran initial de l'application](#) » à la page 9). Indiquez ensuite les fichiers PNG associés à l'icône de 57 x 57 pixels et à l'icône de 512 x 512 pixels.
- 5 Cliquez sur le bouton OK.
- 6 Enregistrez le fichier.

Pour plus d'informations sur les paramètres de l'application, voir « [Paramètres de l'application iPhone](#) » à la page 15.

Compilation du fichier IPA

Vous pouvez maintenant compiler le fichier d'installation IPA, comme suit :

- 1 Sélectionnez Fichier > Publier.
- 2 Dans la boîte de dialogue Paramètres de l'iPhone, cliquez sur le bouton OK.

Le gestionnaire de package pour applications iPhone génère le fichier d'installation de l'application iPhone, HelloWorld.ipa, dans le répertoire du projet. La compilation du fichier IPA prend parfois quelques minutes.

Installation de l'application sur l'iPhone

Pour installer l'application sur un iPhone à des fins de test :

- 1 Ouvrez l'application iTunes.
- 2 Le cas échéant, ajoutez à iTunes le profil de configuration associé à l'application. Dans iTunes, sélectionnez Fichier > Ajouter le fichier à la bibliothèque. Sélectionnez ensuite le fichier du profil de configuration (dont le type de fichier correspond à mobileprovision).

Utilisez à présent le profil de configuration pour le développement d'applications pour tester l'application sur l'iPhone.

Lors de la distribution ultérieure de l'application sur l'iTunes Store, vous utiliserez le profil de distribution. Pour procéder à une distribution ad hoc de l'application (en d'autres termes, pour la distribuer sur plusieurs périphériques sans passer par l'iTunes Store), utilisez le profil de configuration ad hoc.

Pour plus d'informations sur les profils de configuration, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.

- 3 Certaines versions d'iTunes ne remplacent pas l'application si une version identique de l'application est déjà installée. Dans ce cas de figure, supprimez l'application du périphérique et de la liste d'applications dans iTunes.
- 4 Double-cliquez sur le fichier IPA associé à l'application. Elle devrait apparaître dans la liste d'applications.
- 5 Connectez l'iPhone au port USB de l'ordinateur.
- 6 Dans iTunes, vérifiez sur l'onglet Application associé au périphérique que l'application est sélectionnée dans la liste d'applications à installer.

7 Sélectionnez le périphérique dans la liste de gauche d'applications. Cliquez ensuite sur le bouton Synchroniser. Une fois la synchronisation terminée, l'application Hello World apparaît sur l'iPhone.

Si la nouvelle version n'est pas installée, supprimez-la de l'iPhone et de la liste d'applications dans iTunes, puis répétez la procédure. Ce cas de figure se produit parfois si la version actuellement installée utilise le même ID et le même numéro.

Si iTunes affiche un message d'erreur lorsque vous essayez d'installer l'application, voir « Résolution des incidents d'installation d'une application » dans « [Installation d'une application iPhone](#) » à la page 24.

Modification des graphiques de l'écran initial

Avant de compiler l'application, vous avez créé le fichier Default.png (voir « [Création des icônes et des graphiques de l'écran initial de l'application](#) » à la page 9). Ce fichier PNG contient l'image de démarrage affichée lors du chargement de l'application. Lorsque vous avez testé l'application sur l'iPhone, peut-être avez-vous remarqué l'écran vide au démarrage.

Remplacez cette image par l'écran de démarrage de l'application (« Hello World! ») en procédant comme suit :

- 1 Ouvrez l'application sur le périphérique. Lorsque la première occurrence du texte « Hello World » apparaît, appuyez sur le bouton principal (figurant sous l'écran) et maintenez-le enfoncé. Tout en maintenant appuyé le bouton principal, appuyez sur le bouton Marche/Veille figurant dans la partie supérieure de l'iPhone. Vous effectuez ainsi une capture d'écran, qui est envoyée à Pellicule.
- 2 Transférez l'image sur l'ordinateur de développement via iPhoto ou toute autre application adaptée. (Sous Mac OS, vous disposez également de l'application Transfert d'images.)

Vous pouvez aussi envoyer la photo par E-mail à l'ordinateur de développement, comme suit :

- Ouvrez l'application Photos.
- Ouvrez Pellicule.
- Ouvrez la capture d'écran que vous avez effectuée.
- Touchez l'image, puis le bouton fléché figurant dans l'angle inférieur gauche. Cliquez ensuite sur le bouton Envoyer par courrier et envoyez l'image à votre propre adresse électronique.

3 Remplacez le fichier Default.png, qui réside dans le répertoire de développement, par une version PNG de la capture d'écran effectuée.

4 Recompilez l'application (voir « [Compilation du fichier IPA](#) » à la page 11) et installez-la à nouveau sur l'iPhone.

L'application utilise à présent le nouvel écran de démarrage au chargement.

Remarque : sous réserve de respecter les dimensions requises (320 x 480 pixels), libre à vous de créer n'importe quel graphique pour le fichier Default.png. Il est toutefois préférable que l'image issue du fichier Default.png corresponde à l'état initial de l'application.

Chapitre 2 : Compilation et débogage d'applications iPhone

Vous pouvez compiler une application iPhone à l'aide de Packager for iPhone. Packager for iPhone est intégré à Flash Professional CS5.

Vous pouvez déboguer l'application sur l'ordinateur de développement. Vous pouvez également installer une version de débogage sur l'iPhone et recevoir les résultats `trace()` dans Flash Professional CS5.

Pour apprendre à créer de toutes pièces une application iPhone, voir « [Création d'une application iPhone Hello World dans Flash Professional CS5](#) » à la page 8.

Icône et images de l'écran initial d'un iPhone

Toutes les applications iPhone contiennent des icônes qui s'affichent dans l'interface utilisateur de l'application iTunes et sur l'écran de l'iPhone.

Icônes d'une application iPhone

Vous définissez les icônes suivantes pour une application iPhone :

- Icône de 29 x 29 pixels : les résultats de la recherche Spotlight sur l'iPhone et l'iPod touch utilisent cette icône.
- Icône de 48 x 48 pixels : utilisée par les résultats de la recherche Spotlight sur l'iPad.
- Icône de 57 x 57 pixels : les écrans d'accueil de l'iPhone et de l'iPod touch affichent cette icône.
- Icône de 72 x 72 pixels (facultative) : utilisée par l'écran d'accueil de l'iPad.
- Icône de 512 x 512 pixels : affichée par iTunes. Le fichier PNG de 512 pixels est réservé au test des versions de développement de l'application. Lorsque vous envoyez l'application définitive à l'App Store d'Apple, vous transmettez séparément l'image de 512 pixels au format JPG. Elle n'est pas incluse dans l'IPA.

Dans Flash Professional CS5, ajoutez ces icônes sur l'onglet Icônes de la boîte de dialogue Paramètres d'iPhone. Voir « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15.

Vous pouvez également ajouter l'emplacement des icônes dans le fichier descripteur d'application :

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

Étant donné que l'iPhone ajoute un effet de lueur jaune à une icône, il est inutile de l'intégrer à l'image source. Pour désactiver cet effet de lueur jaune activé par défaut, ajoutez le texte ci-dessous à l'élément `InfoAdditions` du fichier descripteur d'application :


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

Voir « [Définition des propriétés d'une application iPhone dans le fichier descripteur correspondant](#) » à la page 17.

Graphiques de l'écran initial (Default.png)

Toute application iPhone affiche une image initiale lors de son chargement sur l'iPhone. Vous définissez cette image initiale dans un fichier PNG appelé Default.png. Dans le répertoire de développement principal, créez un fichier PNG appelé Default.png. (Ne placez *pas* ce fichier dans un sous-répertoire et respectez la casse du nom du fichier.)

Le fichier Default.png mesure 320 pixels de large sur 480 pixels de haut, quelle que soit l'orientation initiale de l'application et qu'elle soit affichée en plein écran ou non.

Si l'orientation initiale de l'application correspond à paysage, sélectionnez les dimensions utilisées par une application affichée en portrait (à savoir 320 pixels de large sur 480 pixels de haut). Faites toutefois pivoter les illustrations de 90° dans le sens inverse des aiguilles d'une montre dans le fichier PNG. Le côté gauche des graphiques PNG correspond au haut de l'écran de l'iPhone en mode paysage. (Pour plus d'informations sur la définition de l'orientation initiale d'une application, voir « [Paramètres de l'application iPhone](#) » à la page 15.)

Si une application n'est pas affichée en plein écran, les 20 pixels placés dans le haut de l'image par défaut sont ignorés. La barre d'état de l'iPhone est superposée au rectangle de 20 pixels de large situé dans le haut de l'image par défaut. Dans une application affichée en mode paysage, cette zone correspond au rectangle de gauche de 20 pixels de large du fichier Default.png (affiché dans le haut en mode paysage). Dans une application affichée en mode portrait, cette zone correspond au rectangle supérieur de 20 pixels de large du fichier Default.png.

Dans la plupart des applications, l'image Default.png doit correspondre à l'écran de démarrage de l'application. Pour capturer l'écran initial de l'application :

- 1 Ouvrez l'application sur l'iPhone. Lorsque le premier écran de l'interface utilisateur apparaît, appuyez sur le bouton principal (figurant sous l'écran) et maintenez-le enfoncé. Tout en maintenant enfoncé le bouton principal, appuyez sur le bouton Marche/Veille figurant dans la partie supérieure du périphérique. Vous effectuez ainsi une capture d'écran, qui est envoyée à Pellicule.
- 2 Transférez l'image sur l'ordinateur de développement via iPhoto ou toute autre application adaptée. (Sous Mac OS, vous disposez également de l'application Transfert d'images.)

Vous pouvez aussi envoyer la photo par E-mail à l'ordinateur de développement, comme suit :

- Ouvrez l'application Photos.
- Ouvrez Pellicule.
- Ouvrez la capture d'écran que vous avez effectuée.
- Touchez l'image, puis le bouton fléché figurant dans l'angle inférieur gauche. Cliquez ensuite sur le bouton Envoyer par courrier et envoyez l'image à votre propre adresse électronique.

Remarque : *Sous réserve de respecter les dimensions requises, libre à vous de créer n'importe quel graphique pour le fichier Default.png. Il est toutefois préférable que l'image issue du fichier Default.png corresponde à l'état initial de l'application.*

N'incluez pas de texte dans l'image Default.png si l'application est localisée en plusieurs langues. Étant donné que le fichier Default.png est statique, le texte ne serait pas traduit.

Dans Flash Professional CS5, veillez à ajouter le fichier Default.png à la liste Fichiers inclus dans la boîte de dialogue Paramètres de l'iPhone. Voir « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15.

Lors de l'exécution de la compilation à l'aide de l'application PFI sur la ligne de commande, veillez à faire référence à ce fichier dans la liste des actifs inclus. Voir « [Création d'un fichier d'installation d'une application iPhone depuis la ligne de commande](#) » à la page 22.

Paramètres de l'application iPhone

Parmi les paramètres de l'application figurent :

- Nom de l'application
- Nom du fichier IPA
- Version de l'application
- Orientation initiale de l'écran de l'application ; un paramètre indique également si l'écran subit automatiquement une rotation lorsque l'utilisateur fait pivoter l'iPhone
- Vue initiale en plein écran ou non
- Informations relatives aux icônes de l'application
- Informations relatives à l'accélération matérielle

Vous pouvez modifier les paramètres de l'application dans Flash Professional CS5.

Vous pouvez également les modifier dans le fichier descripteur de l'application. Le fichier descripteur de l'application est un fichier XML qui contient les paramètres de l'application.

Définition des propriétés d'une application iPhone dans Flash Professional CS5

La boîte de dialogue Paramètres de l'iPhone de Flash Professional CS5 permet de définir un grand nombre de propriétés de base de l'application iPhone.

Pour ouvrir la boîte de dialogue Paramètres de l'iPhone :

- ❖ Sélectionnez Fichier > Paramètres de l'iPhone.

Onglet Général

L'onglet Général contient les paramètres iPhone suivants :

- Fichier de sortie : nom de l'application affiché sous l'icône de l'application sur l'iPhone. Le nom du fichier de sortie ne doit pas comporter de signe plus (+).
- Nom de l'application : nom de l'application affiché sous l'icône de l'application sur l'iPhone. Le nom de l'application ne doit pas comporter de signe plus (+).
- Version : permet à l'utilisateur de déterminer la version de l'application qu'il installe. Cette version fait office de valeur CFBundleVersion de l'application iPhone. Vous devez l'exprimer dans un format similaire à nnnnn[.nn[.nn]], où n correspond à un chiffre compris entre 0 et 9 et les crochets représentent des composants facultatifs tels que 1, 1.0 ou 1.0.1. Les versions d'une application iPhone ne doivent comporter que des chiffres et des séparateurs décimaux (deux au plus).

- Format : format initial de l'application (portrait ou paysage).
- Plein écran : indique si l'application occupe la totalité de l'écran ou si la barre d'état de l'iPhone est affichée.
- Orientation auto : permet de réorienter le contenu affiché de l'application lorsque l'iPhone pivote.

Si vous activez l'orientation automatique, ajoutez un code ActionScript pour définir la propriété `align` de la scène comme suit afin d'optimiser les résultats :

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

- Rendu : indique le mode de rendu des objets d'affichage sur l'iPhone, à savoir :
 - CPU : l'application fait appel à l'unité centrale pour effectuer le rendu de tous les objets d'affichage. Aucune accélération matérielle n'est utilisée.
 - GPU : l'application fait appel au processeur graphique de l'iPhone pour créer des images bitmap composites.
 - Auto : fonctionnalité non mise en œuvre actuellement.

Pour plus d'informations, voir « [Accélération matérielle](#) » à la page 37.

- Fichiers inclus : ajoute tous les fichiers et répertoires à mettre en package à l'application iPhone. Le fichier SWF principal et le fichier descripteur d'application sont inclus par défaut. Ajoutez tout autre actif requis à la liste Fichiers inclus. Veillez à ajouter le fichier contenant les graphiques de l'écran initial (Default.png) à la liste Fichiers inclus.

Onglet Déploiement

L'onglet Déploiement comprend les paramètres de signature et de compilation de l'application :

- Signature numérique iPhone : indiquez un fichier de certificat P12 et le mot de passe correspondant. Vous devez convertir le certificat iPhone d'Apple au format .p12. Pour plus d'informations, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.
- Fichier de configuration : pointe vers le fichier de configuration associé à l'application, obtenu auprès d'Apple. Pour plus d'informations, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.
- ID app. : ID d'application qui identifie de manière unique l'application. Si le fichier de configuration est lié à un ID d'application donné, Flash Professional CS5 définit ce champ et vous ne pouvez pas le modifier. Dans le cas contraire, le profil de configuration gère plusieurs ID d'application (contenant un caractère générique). Entrez un ID d'application correspondant au modèle d'ID d'application à caractère générique fourni à Apple :
 - Si l'ID d'application d'Apple est `com.monDomaine.*`, l'ID d'application figurant dans la boîte de dialogue Paramètres de l'iPhone doit débuter par `com.monDomaine` (`com.monDomaine.monApp` ou `com.monDomaine.app22`, par exemple).
 - Si l'ID d'application Apple est `*`, l'ID d'application figurant dans la boîte de dialogue Paramètres de l'iPhone peut correspondre à n'importe quelle chaîne de caractères valides.

Vous trouverez l'ID d'application (ou le modèle d'ID d'application à caractère générique) d'Apple associé à un profil de configuration dans le centre des développeurs iPhone (<http://developer.apple.com/iphone>). Accédez au portail iPhone Developer Program, puis à la section Provisioning.

Important : ignorez les premiers caractères de l'ID d'application d'Apple. Apple appelle cette chaîne l'identifiant d'application (préfixe). Ainsi, si Apple affecte l'ID `96LPVWEASL.com.exemple.jean.monApp` à l'application, ignorez `96LPVWEASL`. En d'autres termes, utilisez l'ID d'application `com.exemple.jean.monApp`. Si Apple stipule l'ID d'application `5RM86Z4DJM.*`, ignorez `5RM86Z4DJM`, qui correspond à un ID d'application contenant un caractère générique.

- Type de déploiement iPhone :
 - Publication rapide pour test de périphériques : cette option permet de compiler rapidement une version de l'application en vue de la tester sur l'iPhone de développement.
 - Publication rapide pour débogage de périphériques : cette option permet de compiler rapidement une version de débogage de l'application en vue de la tester sur l'iPhone de développement. Si vous activez cette option, le débogueur de Flash Professional CS5 peut recevoir les résultats `trace()` de l'application iPhone (voir « [Débogage d'une application iPhone](#) » à la page 25).
 - Déploiement - Ad hoc : cette option permet de créer une application destinée à un déploiement ad hoc (voir le centre des développeurs iPhone d'Apple).
 - Déploiement - App Store d'Apple : cette option permet de créer une version définitive du fichier IPA à déployer sur l'App Store d'Apple.

Onglet Icônes

Dans l'onglet Icônes, spécifiez l'emplacement de l'image d'icône de 29 x 29 pixels, de l'image d'icône de 48 x 48 pixels, de l'image d'icône de 57 x 57 pixels, de l'image d'icône de 72 x 72 pixels et de l'image d'icône de 512 x 512 pixels. (voir « [Icône et images de l'écran initial d'un iPhone](#) » à la page 13).

Remarque : les icônes de 48 x 48 pixels et de 72 x 72 pixels ne sont pas incluses dans la version d'évaluation de Packager for iPhone Preview de Flash Professional CS5. Dans Flash Professional CS5, sélectionnez Aide > Mises à jour pour ajouter ces options.

Définition des propriétés d'une application iPhone dans le fichier descripteur correspondant

Le fichier descripteur d'une application est un fichier XML qui contient les propriétés de l'application entière, tels que son nom, sa version, le copyright et d'autres paramètres.

Flash Professional CS5 génère un fichier descripteur d'application basé sur les paramètres que contient la boîte de dialogue Paramètres d'iPhone. Toutefois, vous pouvez également modifier le fichier descripteur d'application dans un éditeur de texte. Flash Professional définit le nom du fichier descripteur d'application en ajoutant « -app.xml » au nom du projet. Ainsi, le fichier descripteur d'application associé au projet HelloWorld porte le nom HelloWorld-app.xml. Modifiez le fichier descripteur d'application si vous souhaitez définir des paramètres non pris en charge dans la boîte de dialogue Paramètres de l'iPhone de Flash Professional CS5. Vous pouvez, par exemple, définir l'élément `InfoAdditions` pour stipuler les paramètres `info.Plist` associés à l'application.

Important : ne modifiez pas le fichier descripteur d'application tant que la boîte de dialogue de Flash Professional CS5 est ouverte. Enregistrez les modifications dans le fichier descripteur d'application avant d'ouvrir la boîte de dialogue Paramètres de l'iPhone.

Exemple de fichier descripteur d'application :

```
<?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>
```

Informations complémentaires relatives aux paramètres du fichier descripteur d'application :

- Dans l'élément `<application>`, l'espace de noms d'AIR 2.0 est requis pour créer des applications iPhone :

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

- Élément `<id>` :

`<id>com.example.as3.HelloWorld</id>` L'ID d'application identifie de manière unique l'application. Le format recommandé correspond à une chaîne de style DNS inversé dont les éléments sont séparés par un point, telle que « com.company.AppName ». Le compilateur utilise cette valeur à titre d'ID de test de composant pour l'application iPhone.

Si le fichier de configuration est lié à un ID d'application donné, utilisez ce dernier dans l'élément. Ignorez les caractères affectés par Apple au début de l'ID d'application Apple (appelés identifiant d'application (préfixe)). Ainsi, si l'ID d'application associé au profil de configuration est 96LPVWEASL.com.exemple.jean.monApp, com.exemple.jean.monApp fait office d'ID d'application dans le fichier descripteur d'application.

Si le profil de configuration prend en charge plusieurs ID d'application (à caractères génériques), son ID d'application se termine par un astérisque (5RM86Z4DJM.*, par exemple). Entrez un ID d'application correspondant au modèle d'ID d'application à caractère générique fourni à Apple :

- Si l'ID d'application d'Apple est com.monDomaine.*, l'ID d'application figurant dans le fichier descripteur d'application doit débuter par com.monDomaine. Vous pouvez définir un ID d'application tel que com.monDomaine.monApp ou com.monDomaine.app22.

- Si l'ID d'application Apple est *, l'ID d'application figurant dans le fichier descripteur d'application peut contenir n'importe quelle chaîne de caractères valides.

Vous trouverez l'ID d'application (ou le modèle d'ID d'application à caractère générique) d'Apple associé à un profil de configuration dans le centre des développeurs iPhone (<http://developer.apple.com/iphone>). Accédez au portail iPhone Developer Program, puis à la section Provisioning.

Important : ignorez les premiers caractères de l'ID d'application d'Apple. Apple appelle cette chaîne l'identifiant d'application (préfixe). Ainsi, si Apple stipule l'ID d'application 5RM86Z4DJM.*, ignorez 5RM86Z4DJM, qui correspond à un ID d'application contenant un caractère générique. Si Apple stipule l'ID d'application 96LPVWEASL.com.exemple.jean.monApp, ignorez 96LPVWEASL. En d'autres termes, utilisez l'ID d'application com.exemple.jean.monApp.

- Élément `<filename>` :
`<filename>HelloWorld</filename>` Nom affecté au fichier d'installation iPhone. Le nom du fichier ne doit pas comporter de signe plus (+).
- Élément `<name>` :
`<name>Hello World</name>` Nom de l'application affiché dans l'application iTunes et sur l'iPhone. Le nom ne doit pas comporter de signe plus (+).
- Élément `<version>` :
`<version>1.0</version>` Permet à l'utilisateur de déterminer la version de l'application qu'il installe. Cette version fait office de valeur `CFBundleVersion` de l'application iPhone. Vous devez l'exprimer dans un format similaire à `nnnn[.nn[.nn]]`, où `n` correspond à un chiffre compris entre 0 et 9 et les crochets représentent des composants facultatifs tels que 1, 1.0 ou 1.0.1. Les versions d'une application iPhone ne doivent comporter que des chiffres et des séparateurs décimaux (deux au plus).
- L'élément `<initialWindow>` contient les éléments enfant suivants, qui indiquent les propriétés de l'aspect initial de l'application :
`<content>HelloWorld.swf</content>` Identifie le fichier SWF racine à compiler pour générer l'application iPhone.
`<visible>>true</visible>` Paramètres obligatoire.
`<fullScreen>>true</fullScreen>` Indique que l'application occupe la totalité de l'écran de l'iPhone.
`<aspectRatio>portrait</aspectRatio>` Indique que le format initial de l'application est en mode portrait (plutôt que paysage). Notez que le fichier `Default.png` qui définit la fenêtre initiale de l'application doit mesurer 320 pixels de large sur 480 pixels de haut, quelle que soit la valeur de ce paramètre (voir « [Icône et images de l'écran initial d'un iPhone](#) » à la page 13).
`<autoOrients>true</autoOrients>` (facultatif) Indique si l'orientation du contenu de l'application est automatiquement modifiée lorsque le périphérique pivote. La valeur par défaut est `true`. Pour annuler l'orientation automatique, appelez la méthode `preventDefault()` d'un événement `orientationChanging` distribué par l'objet `Stage`. Pour plus d'informations, voir [Définition et détection de l'orientation d'un écran](#).
Si vous activez l'orientation automatique, définissez la propriété `align` de la scène comme suit pour optimiser les résultats :

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

`<renderMode>gpu</renderMode>` (facultatif) Mode de rendu utilisé par l'application. Trois paramètres sont pris en charge, à savoir :

- `cpu` : l'application fait appel à l'unité centrale pour effectuer le rendu de tous les objets d'affichage. Aucune accélération matérielle n'est utilisée.
- `gpu` : l'application fait appel au processeur graphique de l'iPhone pour créer des images bitmap composites.
- `auto` : fonctionnalité non mise en œuvre actuellement.

Pour plus d'informations, voir « [Accélération matérielle](#) » à la page 37.

- Élément `<profiles>` :

`<profiles>mobileDevice</profiles>` Impose la compilation de l'application en fonction du profil de périphérique mobile. Ce profil ne prend actuellement en charge que les applications iPhone. Trois profils sont pris en charge :

- `desktop` : application AIR de bureau.
- `extendedDesktop` : application AIR de bureau qui prend en charge l'API de processus native.
- `mobileDevice` : application AIR destinée à un périphérique mobile. iPhone est l'unique périphérique mobile actuellement pris en charge.

En associant l'application à un profil donné, vous interdisez sa compilation en fonction d'autres profils. Si vous ne stipulez pas de profil, il est possible de compiler l'application en fonction de n'importe quel profil. Pour stipuler plusieurs profils, recensez-les en séparant chaque élément de la liste par un espace dans l'élément `<profiles>`.

Veillez à inclure `mobileDevice` en tant que profil pris en charge (ou ne renseignez pas l'élément `<profiles>`).

- L'élément `<icon>` contient les éléments enfant suivants, qui stipulent les icônes utilisées par l'application :

`<image29x29>icons/icon29.png</image29x29>` Image utilisée dans les résultats de la recherche Spotlight.

`<image48x48>icons/icon48.png</image48x48>` Image utilisée dans les résultats de la recherche Spotlight sur l'iPad.

`<image57x57>icons/icon57.png</image57x57>` Image utilisée dans les écrans d'accueil de l'iPhone et de l'iPod touch.

`<image72x72>icons/icon72.png</image72x72>` Image utilisée dans l'écran d'accueil de l'iPad.

`<image512x512>icons/icon512.png</image512x512>` Image utilisée dans l'application iTunes.

L'outil Packager for iPhone fait appel aux icônes 29, 57 et 512 référencées dans le fichier descripteur d'application. Il les copie dans des fichiers appelés respectivement `Icon-Small.png`, `Icon.png` et `iTunesArtwork`. Pour éviter l'exécution de la copie, vous pouvez mettre directement en package les fichiers. Pour ce faire, placez-les dans le répertoire contenant le fichier descripteur d'application et indiquez les noms et chemins appropriés.

L'image 512 est réservée aux tests internes. Lorsque vous envoyez une application à Apple, l'image 512 est transmise séparément. Elle n'est pas incluse dans l'IPA. Stipulez-la pour vous assurer que l'image 512 vous convient avant de l'envoyer.

- L'élément `<iPhone>` contient les éléments enfant suivants, qui stipulent les paramètres propres à l'iPhone :

`<InfoAdditions></InfoAdditions>` Contient les éléments enfants qui stipulent les paires clé-valeur faisant office de paramètres `Info.plist` dans l'application :

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

Dans cet exemple, les valeurs définies déterminent le style de la barre d'état de l'application et stipulent que cette dernière ne nécessite pas un accès Wi-Fi permanent.

Les paramètres InfoAdditions sont entourés d'une balise CDATA.

Pour une prise en charge de l'iPad, incluez des paramètres clé-valeur pour `UIDeviceFamily`. Le paramètre `UIDeviceFamily` est un tableau de chaînes. Chaque chaîne définit les périphériques pris en charge. Le paramètre `<string>1</string>` définit la prise en charge de l'iPhone et de l'iPod touch. Le paramètre `<string>2</string>` définit la prise en charge de l'iPad. Le paramètre `<string>3</string>` définit la prise en charge de tvOS. Si vous spécifiez uniquement l'un de ces paramètres, seule cette famille de périphériques est prise en charge. Par exemple, le paramètre suivant limite la prise en charge à l'iPad :

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

Le paramètre suivant prend en charge les deux familles de périphériques (iPhone/iPod touch et iPad) :

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

Pour plus d'informations sur d'autres paramètres Info.plist, voir la documentation Apple destinée aux développeurs.

Compilation d'un fichier d'installation d'application iPhone (IPA)

À l'aide de l'outil Packager for iPhone, compilez une application ActionScript 3.0 pour obtenir un fichier d'installation IPA.

Création d'un fichier d'installation d'application iPhone à l'aide de l'outil Packager for iPhone Preview intégré à Flash Professional CS5

Pour utiliser l'outil Packager for iPhone intégré à Flash Professional CS5 :

- 1 Sélectionnez Fichier > Publier.
- 2 Dans la boîte de dialogue Paramètres de l'iPhone, vérifiez que tous les paramètres ont été renseignés. Assurez-vous que les options appropriées sont sélectionnées dans l'onglet Déploiement. Voir « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15.
- 3 Cliquez sur le bouton Publier.

Le gestionnaire de package pour applications iPhone génère le fichier du programme d'installation de l'application iPhone (IPA). La compilation du fichier IPA prend parfois quelques minutes.

Vous pouvez également exécuter l'outil Packager for iPhone depuis la ligne de commande. Voir « [Création d'un fichier d'installation d'une application iPhone depuis la ligne de commande](#) » à la page 22.

Création d'un fichier d'installation d'une application iPhone depuis la ligne de commande

Vous pouvez exécuter le gestionnaire de package pour applications iPhone depuis la ligne de commande. Le gestionnaire de package pour applications iPhone convertit le pseudo-code issu du fichier SWF et d'autres fichiers source en application iPhone native.

- 1 Ouvrez une interface de commande ou un terminal et accédez au dossier de projet contenant l'application iPhone.
- 2 Créez ensuite le fichier IPA à l'aide de l'outil pfi en appliquant la syntaxe suivante :

```
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
```

Modifiez la référence `pfi` de sorte à inclure le chemin d'accès complet à l'application `pfi`. L'application `pfi` est installée dans le sous-répertoire `pfi/bin` du répertoire d'installation de Flash Professional CS5.

Sélectionnez l'option `-target` correspondant au type d'application iPhone à créer :

- `-target ipa-test` : cette option permet de compiler rapidement une version de l'application en vue de la tester sur l'iPhone de développement.
- `-target ipa-debug` : cette option permet de compiler une version de débogage de l'application en vue de la tester sur l'iPhone de développement. Cette option permet de recevoir la sortie `trace()` issue de l'application iPhone dans le cadre d'une session de débogage.

Vous pouvez inclure l'une des options `-connect` suivantes (`CONNECT_OPTIONS`) pour spécifier l'adresse IP de l'ordinateur de développement qui exécute le débogueur :

- `-connect` : l'application tentera de se connecter à une session de débogage sur l'ordinateur de développement utilisé pour compiler l'application.
- `-connect IP_ADDRESS` : l'application tentera de se connecter à une session de débogage sur l'ordinateur dont l'adresse IP a été spécifiée. Exemple :

```
-target ipa-debug -connect 192.0.32.10
```
- `-connect HOST_NAME` : l'application tentera de se connecter à une session de débogage sur l'ordinateur dont le nom d'hôte a été spécifié. Exemple :

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

Remarque : l'option `-connect` n'est pas incluse dans l'utilitaire *Packager for iPhone Preview* fourni avec *Flash Professional CS5*. Mettez à jour *Packager for iPhone* en sélectionnant *Aide > Mises à jour dans Flash Professional CS5*.

L'option `-connect` est facultative. Si elle n'est pas spécifiée, l'application de débogage résultante ne tentera pas de se connecter à un débogueur hébergé.

En cas d'échec d'une tentative de connexion à une session de débogage, l'application affiche une boîte de dialogue qui invite l'utilisateur à saisir l'adresse IP de l'ordinateur qui héberge le débogueur. Une tentative de connexion peut échouer si le périphérique n'est pas connecté au réseau Wi-Fi. Elle peut également échouer si le périphérique est connecté, mais pas derrière le pare-feu de l'ordinateur de débogage hôte.

Pour plus d'informations, voir « [Débogage d'une application iPhone](#) » à la page 25.

Vous pouvez également inclure l'option `-renderingdiagnostics` pour activer la fonction de diagnostic du rendu basé sur le processeur graphique. Pour plus d'informations, voir la rubrique « Débogage avec l'outil de diagnostic du rendu sur GPU » à la section « [Débogage d'une application iPhone](#) » à la page 25.

- `-target ipa-ad-hoc` : cette option permet de créer une application destinée à un déploiement ad hoc (voir le centre des développeurs iPhone d'Apple).
- `-target ipa-app-store` : cette option permet de créer une version définitive du fichier IPA à déployer sur l'App Store d'Apple.

Remplacez `PROFILE_PATH` par le chemin du profil de configuration de l'application. Pour plus d'informations sur les profils de configuration, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.

Remplacez l'élément `SIGNING_OPTIONS` de sorte à faire référence au certificat de développement iPhone et au mot de passe correspondant. Utilisez la syntaxe suivante :

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

Remplacez `P12_FILE_PATH` par le chemin du fichier de certificat P12. Remplacez `PASSWORD` par le mot de passe associé au certificat (voir l'exemple ci-dessous). Pour plus d'informations sur le fichier du certificat P12, voir « [Conversion d'un certificat de développement en fichier P12](#) » à la page 7.

Remplacez l'élément `APP_DESCRIPTOR` de sorte à faire référence au fichier descripteur de l'application.

Remplacez l'élément `SOURCE_FILES` de sorte à faire référence au principal fichier SWF du projet, suivi de tout autre actif à inclure. Incluez les chemins d'accès à tous les fichiers d'icône définis dans la boîte de dialogue des paramètres de l'application de Flash CS5 ou dans un fichier descripteur d'application personnalisé. Ajoutez également le fichier contenant les graphiques de l'écran initial, `Default.png`.

Passez en revue l'exemple suivant :

```
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 compile un fichier `HelloWorld.ipa` en faisant appel aux éléments suivants :

- Un certificat PKCS#12 donné qui utilise le mot de passe de certificat `dfb7VKL19`
- Le fichier descripteur d'application `HelloWorld-app.xml`
- Un fichier `HelloWorld.swf` source
- Un fichier `Default.png` et des fichiers d'icône déterminés

L'application `pfi` compile l'application pour générer un fichier IPA en fonction du fichier descripteur d'application, du fichier SWF et des autres actifs.

Sous Mac OS, vous disposez d'un certificat stocké dans le trousseau en ajoutant les options suivantes à la commande `pfi` :

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

Remplacez `ALIAS_NAME` par l'alias du certificat à utiliser. Si vous pointez vers un certificat stocké dans le trousseau Mac, vous spécifiez l'alias au lieu de pointer vers un emplacement de fichier de certificat.

Installation d'une application iPhone

Pour installer l'application de développement sur un iPhone, vous ajoutez le profil de configuration sur ce dernier, puis vous y installez l'application.

Ajout du profil de configuration à l'iPhone

Pour ajouter le profil de configuration à l'iPhone :

- 1 Dans iTunes, sélectionnez Fichier > Ajouter le fichier à la bibliothèque. Sélectionnez ensuite le fichier du profil de configuration (dont l'extension correspond à mobileprovision).

Assurez-vous que l'iPhone est ajouté au profil de configuration. Vous pouvez gérer les profils de configuration sur le site du centre des développeurs iPhone d'Apple (<http://developer.apple.com/iphone/>). Accédez au portail iPhone Developer Program du site. Cliquez sur le lien Devices pour gérer la liste des périphériques sur lesquels peut être installée l'application de développement. Cliquez sur le lien Provisioning pour gérer les profils de configuration.

- 2 Connectez l'iPhone à l'ordinateur et synchronisez-le.

Pour plus d'informations sur l'obtention d'un profil de configuration, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.

Installation de l'application

Vous installez l'application de développement comme tout autre fichier IPA :

- 1 Si vous avez précédemment installé une version de l'application, supprimez-la du périphérique et de la liste d'applications dans iTunes.
- 2 Procédez comme suit, au choix, pour ajouter l'application dans iTunes :
 - Dans le menu Fichier d'iTunes, sélectionnez la commande Ajouter le fichier à la bibliothèque. Sélectionnez ensuite le fichier IPA et cliquez sur le bouton Ouvrir.
 - Double-cliquez sur le fichier IPA.
 - Faites glisser le fichier IPA vers la bibliothèque iTunes.
- 3 Connectez l'iPhone au port USB de l'ordinateur.
- 4 Dans iTunes, vérifiez sur l'onglet Application associé au périphérique que l'application est sélectionnée dans la liste d'applications à installer.
- 5 Synchronisez l'iPhone.

Résolution des incidents d'installation d'une application

Si iTunes affiche un message d'erreur lorsque vous essayez d'installer l'application, procédez comme suit :

- Vérifiez que l'ID du périphérique est ajouté au profil de configuration.
- Pour vérifier que le profil de configuration est installé, vous pouvez le faire glisser vers iTunes ou faire appel à la commande Fichier > Ajouter le fichier à la bibliothèque.

Vérifiez également que l'ID de l'application est identique à l'ID d'application d'Apple :

- Si l'ID d'application d'Apple est com.monDomaine.*, l'ID d'application figurant dans le fichier descripteur d'application ou dans la boîte de dialogue Paramètres de l'iPhone doit commencer par com.monDomaine (com.monDomaine.touteValeur, par exemple).
- Si l'ID d'application d'Apple est com.monDomaine.monApp, l'ID d'application figurant dans le fichier descripteur d'application ou l'interface utilisateur Flash Professional CS5 doit correspondre à com.monDomaine.monApp.

- Si l'ID d'application d'Apple est *, l'ID d'application figurant dans le fichier descripteur d'application ou l'interface utilisateur de Flash Professional CS5 peut correspondre à n'importe quelle valeur.

Vous définissez l'ID de l'application dans la boîte de dialogue Paramètres d'iPhone de Flash Professional CS5 ou dans le fichier descripteur de l'application.

Débogage d'une application iPhone

Vous pouvez déboguer l'application sur l'ordinateur de développement si elle s'exécute dans ADL. Vous pouvez également la déboguer sur l'iPhone.

Certaines fonctionnalités AIR qui ne sont pas prises en charge par l'iPhone sont néanmoins disponibles lorsque vous testez une application avec ADL (sur l'ordinateur de développement). Tenez compte de ces différences lorsque vous testez le contenu sur le bureau. Pour plus d'informations, voir « [API d'ActionScript 3.0 non prises en charge sur les périphériques mobiles](#) » à la page 29.

Débogage de l'application sur l'ordinateur de développement

Pour déboguer l'application sur l'ordinateur de développement avec Flash Professional CS5 :

- ❖ Sélectionnez Débogage > Déboguer l'animation > dans l'application de débogage du lanceur AIR (mobile).

Vous pouvez également déboguer l'application en appelant ADL à partir de la ligne de commande. La syntaxe requise est la suivante :

```
adl -profile mobileDevice appDescriptorFile
```

Remplacez *appDescriptorFile* par le chemin du fichier descripteur de l'application.

Veillez à inclure l'option `-profile mobileDevice`.

Débogage de l'application sur l'iPhone

Pour déboguer l'application sur l'iPhone :

- 1 Compilez l'application en prenant en charge le débogage :
 - Dans Flash Professional CS5, compilez l'application en activant le paramètre « Publication rapide pour le débogage du périphérique ». (Voir « [Création d'un fichier d'installation d'application iPhone à l'aide de l'outil Packager for iPhone Preview intégré à Flash Professional CS5](#) » à la page 21.)
 - Utilisez l'application PFI en ligne de commande pour compiler l'application avec l'option `target ipa-debug`. (Voir « [Création d'un fichier d'installation d'une application iPhone depuis la ligne de commande](#) » à la page 22.)
- 2 Installez l'application sur l'iPhone.
- 3 Sur l'iPhone, activez le mode Wi-Fi et connectez-vous au même réseau que l'ordinateur de développement.
- 4 Démarrez une session de débogage sur l'ordinateur de développement. Dans Flash Professional CS5, sélectionnez Débogage > Commencer la session de débogage à distance > ActionScript 3.0.
- 5 Exécutez l'application sur l'iPhone.

La version de débogage de l'application vous invite à entrer l'adresse IP de l'ordinateur de développement. Entrez l'adresse IP et touchez le bouton OK. Pour obtenir l'adresse IP de l'ordinateur de développement :

- Sous Mac OS, sélectionnez Préférences Système depuis le menu Pomme. Dans la fenêtre Préférences Système, cliquez sur l'icône de réseau. La fenêtre Préférences Réseau contient l'adresse IP.

- Sous Windows, ouvrez une session de ligne de commande et exécutez la commande `ipconfig`.

La session de débogage affiche toute sortie `trace()` issue de l'application.

Lors du débogage d'une application installée sur l'iPhone, Flash Professional CS5 prend en charge toutes les fonctions de débogage, y compris le contrôle du point d'arrêt, le défilement du code et la supervision des variables.

Débogage avec l'outil de diagnostic du rendu sur GPU

La fonction de diagnostic du rendu sur GPU permet de comprendre la façon dont l'application utilise l'accélération matérielle (pour les applications qui utilisent le mode de rendu sur GPU). Pour utiliser cette fonction, compilez l'application à l'aide de l'outil PFI en ligne de commande, puis incluez l'option `-renderingdiagnostics` :

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

L'indicateur `-renderingdiagnostics` doit suivre directement l'indicateur `-package`.

La fonction de diagnostic du rendu sur GPU affiche des rectangles colorés pour tous les objets d'affichage :

- Bleu : l'objet d'affichage n'est pas un bitmap ou n'est pas mis en cache sous forme de bitmap, et il est en cours de rendu.

Si un rectangle bleu apparaît plusieurs fois pour un objet d'affichage qui ne change pas, il est possible que ce dernier soit lié d'une manière ou d'une autre à des objets d'affichage en mouvement. Par exemple, l'objet d'affichage peut être l'arrière-plan d'objets d'affichage en mouvement. Il est peut-être souhaitable de mettre en cache l'objet d'affichage sous forme de bitmap.

Si un rectangle bleu apparaît pour un objet d'affichage qui, selon vous, devrait être mis en cache, il est possible que l'objet utilise un effet que le processeur graphique ne peut pas appliquer. Ces effets peuvent être les suivants : certains modes de fusion, des transformations de couleur, la propriété `scrollRect` et des masques.

L'application affiche également des rectangles bleus si les objets d'affichage chargés sur le processeur graphique dépassent les limites de mémoire.

L'application génère des messages pour chaque rectangle bleu. Elle publie ces messages avec des messages `trace()` et des messages de débogage.

- Vert : l'objet d'affichage est un bitmap ou mis en cache sous forme de bitmap et il est chargé pour la première fois sur le processeur graphique.

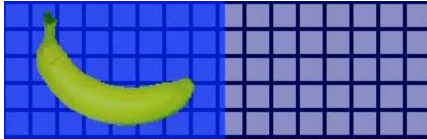
Si un rectangle vert apparaît plusieurs fois pour un objet d'affichage, le code dans l'application est en train de recréer l'objet d'affichage. C'est notamment le cas lorsque le scénario revient à une image qui crée l'objet d'affichage. Il est peut-être souhaitable de modifier le contenu pour éviter de créer des objets identiques.

- Rouge : l'objet d'affichage est un bitmap ou mis en cache sous forme de bitmap et il est à nouveau chargé sur le processeur graphique.

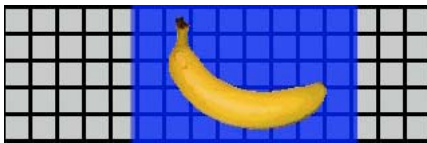
Un rectangle rouge apparaît dans le cas où chaque changement d'un objet d'affichage implique un nouveau rendu de la représentation sous forme de bitmap de la part de l'application. Par exemple, un objet 2D dont la propriété `cacheAsBitmapMatrix` n'est pas définie est à nouveau rendu lorsque l'utilisateur le fait pivoter ou le met à l'échelle. C'est aussi le cas lorsque des objets d'affichage enfants bougent ou changent.

Chaque rectangle perd sa couleur après quatre cycles de retraçage, à condition que la coloration ne se produise plus durant ces cycles. Toutefois, si aucun changement ne se produit à l'écran, les couleurs de diagnostic ne changent pas.

Prenons par exemple un objet d'affichage bitmap (une banane) devant un arrière-plan de vecteurs non mis en cache sous forme de bitmap. Lors du premier rendu de la banane, celle-ci est de couleur verte. Lors du premier rendu de l'arrière-plan, celui-ci est de couleur bleue :

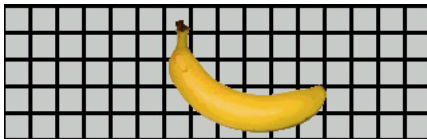


Au fur et à mesure que la banane bouge, l'unité centrale doit effectuer un nouveau rendu de l'arrière-plan, ce qui provoque l'apparition d'une trame bleue sur l'arrière-plan :



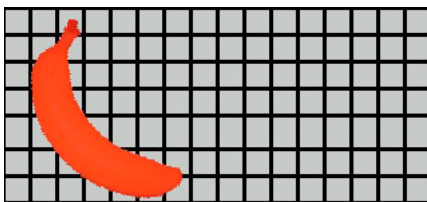
La trame bleue sur l'arrière-plan indique les zones retracées devant être envoyées au processeur graphique.

Néanmoins, si l'arrière-plan est mis en cache sous forme de bitmap, lorsque la banane bouge, la fonction de diagnostic du rendu n'affiche aucune couleur :



La fonction de diagnostic n'affiche aucune couleur, car le processeur graphique retient l'image bitmap de l'arrière-plan. Le processeur graphique peut créer une épreuve couleurs de la banane avec l'arrière-plan sans avoir recours à l'unité centrale.

Supposons que la banane est un objet 2D dont la propriété `cacheAsBitmapMatrix` n'est pas définie. Dès que l'utilisateur fait pivoter (ou met à l'échelle) l'objet d'affichage, la fonction de diagnostic du rendu s'affiche en rouge. Cela indique que l'application doit charger une nouvelle version de l'objet d'affichage sur le processeur graphique :



Envoi de l'application iPhone à l'App Store

Pour envoyer l'application à l'App Store :

- 1 Procurez-vous un certificat de distribution et un profil de configuration sur le site Web du centre des développeurs iPhone (<http://developer.apple.com/iphone/>).

Un certificat de distribution porte le nom « iPhone Developer: XXX.cer » où XXX correspond à votre nom.

Pour plus d'informations, voir « [Obtention de fichiers de développement auprès d'Apple](#) » à la page 5.

- 2 Convertissez le certificat de distribution en fichier P12.

Pour plus d'informations, voir « [Conversion d'un certificat de développement en fichier P12](#) » à la page 7.

- 3 Compilez l'application avec le fichier P12 et le profil de configuration.

Utilisez le fichier P12 que vous avez généré à partir du certificat de distribution. Spécifiez l'ID d'application associé au profil de configuration pour la distribution d'applications.

Pour plus d'informations, voir « [Compilation d'un fichier d'installation d'application iPhone \(IPA\)](#) » à la page 21.

- 4 Envoyez l'application au site Web du centre des développeurs iPhone (<http://developer.apple.com/iphone/>).

Important : Apple requiert que vous utilisiez le programme Application Loader d'Apple pour télécharger une application sur l'App Store. Apple publie Application Loader pour Mac OS X uniquement. Bien que vous puissiez développer une application AIR pour l'iPhone sur un ordinateur Windows, vous devez disposer d'un ordinateur qui exécute OS X (version 10.5.3 ou ultérieure) pour envoyer l'application à l'App Store. Le programme Application Loader est disponible dans le centre de développement iOS d'Apple.

Chapitre 3 : Prise en charge des périphériques mobiles dans les API d'ActionScript 3.0

Lorsque vous créez des applications AIR mobiles, vous disposez des API d'ActionScript 3.0 associées au développement d'applications de bureau Adobe Flash Player 10.1 et AIR 2. Certaines API ne sont toutefois pas gérées, tandis que d'autres sont réservées aux applications mobiles.

API d'ActionScript 3.0 non prises en charge sur les périphériques mobiles

Certaines API d'ActionScript 3.0 ne sont pas prises en charge pour les applications qui s'exécutent dans le profil du périphérique mobile (telles les applications qui tournent sur l'iPhone).

Si vous utilisez le même code ActionScript pour développer des applications relevant de profils divers (bureau et mobile, par exemple), testez à l'aide d'un code la prise en charge d'une API. La classe `NativeWindow` n'est, par exemple, pas prise en charge par les applications iPhone, puisque ces dernières ne peuvent ni utiliser, ni créer de fenêtres natives. Pour s'assurer qu'une application s'exécute sur un profil qui prend en charge les fenêtres natives (le profil de bureau, par exemple), vérifiez la propriété `NativeWindow.isSupported`.

Le tableau ci-dessous recense les API qui ne sont pas prises en charge par le profil du périphérique mobile. Il dresse également la liste des propriétés qui permettent de déterminer si une application s'exécute sur une plate-forme prenant en charge une API.

API	Propriété permettant de tester la prise en charge
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

API	Propriété permettant de tester la prise en charge
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
PDF support	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

Il est impossible de programmer des applications AIR en HTML et JavaScript pour le profil du périphérique mobile.

Certaines classes d'ActionScript 3.0 ne sont que partiellement prises en charge :

Fichier

Les applications iPhone ne peuvent accéder qu'au répertoire de stockage et au répertoire d'une application. Vous pouvez également appeler `File.createTempFile()` et `File.createTempDirectory()`. Appeler une opération pour accéder à un autre répertoire (une méthode de lecture ou d'écriture `FileStream`, par exemple) entraîne le renvoi d'une exception `IOError`.

Les applications iPhone ne prennent pas en charge les boîtes de dialogue de navigateur de fichiers natives, telle la boîte de dialogue fournie par la méthode `File.browseForOpen()`.

Loader

Une application iPhone permet d'utiliser la méthode `Loader.load()`. Vous ne pouvez toutefois pas exécuter un code ActionScript dans un contenu SWF chargé à l'aide de la méthode `Loader.load()`. Il est cependant possible d'utiliser les actifs du fichier SWF (tels que les clips, les images, les polices et les sons de la bibliothèque). Vous pouvez également charger des fichiers d'image à l'aide de la méthode `Loader.load()`.

Vidéo

Une application AIR qui s'exécute sur l'iPhone ne prend en charge que les codecs vidéo Sorensen et ON2 VP6.

Vous disposez de la méthode `navigateToURL()` pour ouvrir une vidéo H.264 hors de l'application. Transmettez un objet `URLRequest` associé à une URL pointant vers la vidéo en tant que paramètre `request`. La vidéo démarre dans le lecteur vidéo de l'iPhone.

Champs de texte

Les polices et autres paramètres des champs de texte sont sujets à des restrictions sur l'iPhone (voir « [Polices et entrée de texte](#) » à la page 41).

API non prises en charge et débogage avec ADL

Certaines fonctionnalités AIR qui ne sont pas prises en charge par l'iPhone sont néanmoins disponibles lorsque vous testez une application avec ADL (sur l'ordinateur de développement). Tenez compte de ces différences lorsque vous testez un contenu avec ADL.

Cette fonctionnalité comprend les codecs vidéo et audio suivants : Speex (audio), H.264/AVC (vidéo) et AAC (audio). Ces codecs ne sont pas gérés par les applications AIR qui s'exécutent sur l'iPhone. Ils continuent toutefois à fonctionner normalement sur le bureau.

Sous Windows, la prise en charge des fonctions d'accessibilité et des logiciels de lecture d'écran est assurée dans ADL. Ces API ne sont toutefois pas prises en charge sur l'iPhone.

Le protocole RTMPE fonctionne normalement à partir d'ADL sur le bureau. Toutefois, un objet `NetConnection` qui essaie de se connecter par le biais du protocole RTMPE échoue sur l'iPhone.

La classe `Loader` fonctionne sans restriction complémentaire lorsqu'un contenu est exécuté avec ADL. Toutefois, lors d'une exécution sur l'iPhone, les tentatives de chargement de contenu SWF qui inclut un pseudo-code ActionScript entraînent la génération d'un message d'erreur.

Les occurrences de Shader s'exécutent dans ADL. Toutefois, le pseudo-code Pixel Bender n'est pas interprété et les shaders n'ont aucun effet graphique sur l'iPhone.

Pour plus d'informations, voir « [Débogage d'une application iPhone](#) » à la page 25.

API d'ActionScript propres aux applications AIR mobiles

Les API suivantes sont disponibles dans les applications AIR uniquement sur les périphériques mobiles. Elles ne fonctionnent actuellement pas dans Flash Player ou les versions de bureau d'AIR.

API d'orientation de l'écran

L'API d'orientation de l'écran permet de manipuler l'orientation de la scène et de l'iPhone :

- `Stage.autoOrients` : indique si l'application est définie de façon à réorienter la scène automatiquement lorsque le périphérique pivote. Cette propriété est définie sur `true` lorsque l'option Orientation auto est sélectionnée dans la boîte de dialogue Paramètres de l'iPhone de Flash Professional CS5. (Vous pouvez également définir l'élément `autoOrients` sur `true` dans le fichier descripteur d'application.) Pour plus d'informations, voir « [Paramètres de l'application iPhone](#) » à la page 15. Vous pouvez annuler la réorientation automatique en ajoutant un écouteur d'événement `orientationChanging` à l'objet `Stage`. L'appel de la méthode `preventDefault()` de cet objet d'événement annule la réorientation automatique.

Si vous activez l'orientation automatique, définissez la propriété `align` de la scène comme suit pour optimiser les résultats :

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

- `Stage.deviceOrientation` : orientation physique du périphérique. La classe `StageOrientation` définit les valeurs de cette propriété.
- `Stage.orientation` : orientation actuelle de la scène. La classe `StageOrientation` définit les valeurs de cette propriété.
- `Stage.supportsOrientationChange` : définie sur `true` sur l'iPhone, sur `false` dans une application AIR.

- `Stage.setOrientation()` : définit l'orientation de la scène. Cette méthode gère un paramètre unique, à savoir une chaîne qui définit la nouvelle orientation de la scène. Les constantes de la classe `StageOrientation` définissent les diverses valeurs du paramètre.
- `StageOrientation` : définit les valeurs d'orientation de la scène. Ainsi, `StageOrientation.ROTATED_RIGHT` indique une scène qui pivote vers la droite relativement à l'orientation par défaut du périphérique.
- `StageOrientationEvent` : définit les événements distribués par la scène en cas de changement d'orientation de l'écran. Cet événement se produit lorsque l'utilisateur fait pivoter l'iPhone. Il existe deux types d'événement. La scène distribue l'événement `orientationChanging` lorsque le périphérique pivote. Pour désactiver la réorientation de la scène, appelez la méthode `preventDefault()` de l'objet d'événement `orientationChanging`. La scène distribue l'événement `orientationChange` une fois la réorientation de la scène terminée.

L'API d'orientation de l'écran ne présente actuellement un intérêt que dans les applications AIR qui s'exécutent sur les périphériques mobiles. Si une application AIR mobile et une application AIR de bureau partagent le code source, utilisez la propriété `Stage.supportsOrientationChange` pour vérifier si l'API est prise en charge.

L'exemple suivant indique comment réagir à la rotation du périphérique par l'utilisateur :

```
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;
    }
}
```

Dans cet exemple, des commentaires remplacent le code fonctionnel en cas d'orientations diverses de la scène.

Pour modifier l'orientation de la scène, appelez la méthode `setOrientation()` de l'objet `Stage`. La définition de l'orientation est une opération asynchrone. Pour vérifier si l'orientation est terminée, écoutez l'événement `orientationChange`. Le code suivant indique comment activer l'orientation vers la droite de la scène :

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

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

Lorsque la scène pivote, elle est redimensionnée et l'objet Stage distribue un événement `resize`. Vous pouvez redimensionner et repositionner sur la scène les objets d'affichage suite à la distribution de l'événement `resize`.

NativeApplication.systemIdleMode et SystemIdleMode

La propriété `NativeApplication.systemIdleMode` permet de désactiver le mode Veille de l'iPhone. Par défaut, un iPhone passe en mode Veille s'il ne se produit aucune interaction avec l'écran tactile pendant une période donnée. Le mode Veille risque d'entraîner l'assombrissement de l'écran. Il active également parfois l'activation du mode Verrouillage. Vous pouvez définir cette propriété sur deux valeurs, au choix :

- `SystemIdleMode.NORMAL` : l'iPhone adopte le comportement standard en mode Veille.
- `SystemIdleMode.KEEP_AWAKE` : l'application essaye de désactiver le mode Veille de l'iPhone.

Cette fonctionnalité est prise en charge uniquement sur les périphériques mobiles. Elle n'est pas prise en charge dans les applications AIR qui s'exécutent sur des systèmes d'exploitation de bureau. Dans une application qui s'exécute sur le bureau, le réglage de la propriété `NativeApplication.systemIdleMode` est sans effet.

Le code suivant illustre la désactivation du mode Veille de l'iPhone :

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

CameraRoll

La classe `CameraRoll` permet d'ajouter une image au dossier Pellicule de l'iPhone. La méthode `addBitmapData()` ajoute une image au dossier Pellicule de l'iPhone. Elle ne gère qu'un seul paramètre, `bitmapData`. Ce paramètre correspond à l'objet `BitmapData` contenant l'image à ajouter au dossier Pellicule.

La fonctionnalité `CameraRoll` est prise en charge uniquement sur les périphériques mobiles. Elle n'est pas prise en charge dans les applications AIR qui s'exécutent sur des systèmes d'exploitation de bureau. Pour vérifier à l'exécution si l'application prend en charge la fonction `CameraRoll`, vérifiez la propriété `CameraRoll.supportsAddBitmapData` statique.

Une fois la méthode `addBitmapData()` appelée, l'objet `CameraRoll` distribue l'un des deux événements suivants :

- `complete` : indique que l'opération a abouti.
- `error` : indique qu'il s'est produit une erreur. Il est, par exemple, possible que l'iPhone ne dispose pas de suffisamment d'espace libre pour stocker l'image.

Le code suivant ajoute une image de la scène (une capture d'écran) au dossier Pellicule :

```
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 propriété `cacheAsBitmapMatrix` est un objet `Matrix` qui définit le mode de rendu d'un objet d'affichage si `cacheAsBitmap` est défini sur `true`. Cette matrice fait office de matrice de transformation lors du rendu de la version bitmap de l'objet d'affichage par l'application.

Lorsque la propriété `cacheAsBitmapMatrix` est définie, l'application conserve une image bitmap en mémoire cache dont le rendu est basé sur cette matrice au lieu de la matrice d'affichage. (La matrice d'affichage est la valeur de la propriété `transform.concatenatedMatrix` de l'objet d'affichage.) Si cette matrice ne correspond pas à la matrice d'affichage, l'image bitmap est mise à l'échelle et subit une rotation si besoin est.

Un objet d'affichage dont la propriété `cacheAsBitmapMatrix` est définie n'est rendu que si la valeur de `cacheAsBitmapMatrix` change. L'image bitmap est mise à l'échelle ou subit une rotation conformément à la matrice d'affichage.

Si les deux types de rendu (processeur graphique et unité centrale) exploitent la propriété `cacheAsBitmapMatrix`, l'impact de cette dernière est généralement plus important sur le rendu basé sur le processeur graphique.

Remarque : Pour utiliser l'accélération matérielle, définissez l'option *Rendu sur GPU* dans l'onglet *Général* de la boîte de dialogue *Paramètres de l'iPhone de Flash Professional CS5*. (Vous pouvez également définir la propriété `renderMode` sur `gpu` dans le fichier descripteur d'application.)

Le code suivant utilise par exemple une représentation d'image bitmap non transformée de l'objet d'affichage :

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

Le code suivant utilise une représentation d'image bitmap qui correspond au rendu en cours :

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

En règle générale, la matrice d'identité (`new Matrix()`) ou `transform.concatenatedMatrix` est suffisante. Vous disposez toutefois d'une autre matrice, telle la matrice de réduction d'échelle, pour envoyer par téléchargement une image bitmap différente au processeur graphique. Ainsi, l'exemple suivant applique une matrice `cacheAsBitmapMatrix` correspondant à une mise à l'échelle de 0,5 sur les axes x et y. L'objet bitmap utilisé par le processeur graphique est de taille inférieure. Ce dernier ajuste toutefois sa taille en fonction de la propriété `transform.matrix` de l'objet d'affichage :

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

En règle générale, sélectionnez une matrice qui transforme l'objet d'affichage de sorte à obtenir la taille requise dans l'application. Par exemple, si l'application affiche la version bitmap du sprite à laquelle a été appliquée une mise à l'échelle de 0,5, utilisez une matrice qui produit une mise à l'échelle de 0,5. Si l'application affiche le sprite dans une taille supérieure aux dimensions en cours, utilisez une matrice dont le facteur de mise à l'échelle est identique.

La taille des objets d'affichage dont vous définissez la propriété `cacheAsBitmapMatrix` est limitée pour des raisons pratiques (1 020 pixels sur 1 020 pixels au plus). Le nombre total de pixels de tous les objets d'affichage dont la propriété `cacheAsBitmapMatrix` est définie est limité pour des raisons pratiques (environ quatre millions de pixels au plus).

Vous devez tenir d'un grand nombre de facteurs lorsque vous utilisez `cacheAsBitmapMatrix` et l'accélération matérielle. Il est important d'identifier les objets d'affichage pour lesquels cette propriété doit être définie. Pour plus d'informations sur l'utilisation de cette propriété, voir « [Accélération matérielle](#) » à la page 37.

Vous pouvez utiliser la fonction de diagnostic de rendu sur GPU pour diagnostiquer l'utilisation du processeur graphique dans les versions de débogage de votre application. Pour plus d'informations, voir « [Débogage d'une application iPhone](#) » à la page 25.

Notes relatives à la mise en réseau

L'utilisation des modèles d'URL suivants avec la fonction `navigateToURL()` entraîne l'ouverture d'un document dans une application externe :

Modèle d'URL	Résultat de l'appel de <code>navigateToURL()</code>	Exemple
mailto :	Ouvre un nouveau message dans l'application de courrier électronique.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	Ouvre un message dans l'application SMS.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	Compose un numéro sur le téléphone (avec l'accord de l'utilisateur).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Une application iPhone peut se fier à des certificats racine auto-signés installés pour authentifier un serveur au cours d'une transaction sécurisée, telle une requête https. Un serveur devrait envoyer non seulement le certificat feuille, mais également tous les certificats intermédiaires liés au certificat racine.

API d'ActionScript 3.0 particulièrement adaptées aux développeurs d'applications mobiles

Les API d'ActionScript 3.0 suivantes définissent une fonctionnalité qui présente un intérêt particulier sur les périphériques mobiles.

API d'accéléromètre

Les classes suivantes permettent à l'application d'extraire des événements de l'accéléromètre du périphérique :

- Accelerometer
- AccelerometerEvent

Pour plus d'informations, voir [Saisie via un accéléromètre](#).

API de géolocalisation

Les classes suivantes permettent à l'application d'extraire des événements du capteur de localisation du périphérique :

- Geolocation
- GeolocationEvent

Pour plus d'informations, voir [Géolocalisation](#).

API tactile, multipoint et de mouvement

Les classes suivantes permettent à l'application de recevoir des événements tactiles et de mouvement :

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

Pour plus d'informations, voir [Saisie tactile, multipoint et par mouvement](#).

Chapitre 4 : Considérations liées à la conception d'applications iPhone

La vitesse de traitement et la taille de l'écran de l'iPhone exigent un codage et une conception spécialement adaptés. La plupart des considérations de conception s'appliquent toutefois à toutes les applications, qu'elles s'exécutent ou non sur un périphérique mobile.

Pour plus d'informations sur l'optimisation des applications, voir [Optimisation des performances pour la plate-forme Adobe Flash](#). Ce document comprend un grand nombre de suggestions relatives à l'optimisation des performances des contenus mobiles, Flash Player et AIR, ainsi que des contenus ActionScript en général. La plupart de ces suggestions s'appliquent également aux applications AIR pour l'iPhone.

Important : un grand nombre de ces techniques d'optimisation et considérations liées à la conception jouent un rôle déterminant dans le développement d'applications pour iPhone performantes.

Accélération matérielle

L'accélération matérielle OpenGL ES 1.1 permet d'améliorer les performances graphiques de certaines applications. L'accélération matérielle s'avère souvent un atout pour les jeux et autres applications dans lesquelles les objets d'affichage sont animés. Les applications qui font appel à l'accélération matérielle peuvent confier certains processus graphiques incombant normalement à l'unité centrale au processeur graphique de l'iPhone, d'où une amélioration considérable des performances.

Lorsque vous configurez une application de façon à ce qu'elle utilise le processeur graphique, vous devez suivre certaines règles pour vous assurer que l'accélération matérielle du contenu est correctement effectuée via le processeur graphique.

Pour utiliser l'accélération matérielle, définissez l'option Rendu sur GPU dans l'onglet Général de la boîte de dialogue Paramètres de l'iPhone de Flash Professional CS5. Vous pouvez également définir la propriété `renderMode` sur `gpu` dans le fichier descripteur d'application :

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

Voir « [Définition des propriétés d'une application iPhone dans Flash Professional CS5](#) » à la page 15 et « [Définition des propriétés d'une application iPhone dans le fichier descripteur correspondant](#) » à la page 17.

Il existe quatre classes d'objets d'affichage pouvant être rendus rapidement via l'accélération matérielle si leur contenu ne change pas fréquemment :

- Objets Bitmap
- Objets d'affichage 2D dont la propriété `cacheAsBitmap` est définie sur `true` et dont la propriété `cacheAsBitmapMatrix` peut ou non être définie (voir ci-dessous)
- Objets d'affichage 3D (dont la propriété `z` est définie)
- Objets d'affichage qui possèdent un remplissage rectangulaire uni dont les bords sont alignés sur les pixels à l'écran.

Le rendu des objets vectoriels est actualisé lorsqu'un autre sprite est animé au-dessus ou au-dessous d'eux. Par conséquent, tout objet servant de toile de fond ou de premier plan à une animation devrait également appartenir à l'une de ces catégories.

Pour les objets d'affichage dont la propriété `cacheAsBitmap` est définie sur `true`, définir `cacheAsBitmapMatrix` force le processeur graphique à utiliser le bitmap résultant de la transformation matricielle. Le processeur graphique utilise la représentation sous forme de bitmap même si l'objet subit une rotation ou une mise à l'échelle. Il peut en effet composer et animer ce bitmap considérablement plus rapidement que l'unité centrale ne peut actualiser l'affichage d'un objet au rendu vectoriel.

Si vous définissez la propriété `cacheAsBitmap` sur `true`, l'objet d'affichage (et tous ses enfants) est mis en cache. L'objet d'affichage n'est pas redessiné lorsque de nouvelles régions sont exposées ou lorsque l'intégralité de l'image combinée est translaturée.

Lorsque la propriété `cacheAsBitmapMatrix` d'un objet d'affichage est définie, l'application peut représenter ce dernier même s'il n'est pas visible. L'application crée une représentation en mémoire cache de l'objet d'affichage au début de l'image suivante. Lorsque vous ajoutez ensuite l'objet d'affichage à la scène, l'application effectue rapidement son rendu. L'application peut également animer, faire pivoter ou mettre à l'échelle rapidement l'objet. Pour les objets qui pivotent ou se mettent à l'échelle, ne définissez pas la propriété `cacheAsBitmap` sans définir également la propriété `cacheAsBitmapMatrix`.

Par ailleurs, l'application peut rapidement effectuer des transformations alpha sur un objet mis en cache sous forme de bitmap. Toutefois, seules les valeurs alpha comprises entre 0 et 1.0 sont prises en charge pour les transformations alpha effectuées par accélération matérielle. Cela correspond à un paramètre `colorTransform.alphaMultiplier` compris entre 0 et 256.

Ne définissez pas la propriété `cacheAsBitmap` sur `true` pour les objets fréquemment mis à jour, tels que les champs de texte.

Les objets d'affichage dont le contenu graphique change fréquemment ne conviennent généralement pas au rendu sur GPU. C'est notamment le cas sur les anciens périphériques dotés de processeurs graphiques moins puissants. Il est donc plus judicieux d'opter pour un rendu basé sur l'unité centrale plutôt que d'avoir à charger les images sur le processeur graphique.

Restructurez les objets d'affichage qui contiennent des objets d'affichage enfants qui se déplacent en fonction d'un objet parent. Modifiez-les de sorte que les objets d'affichage enfants deviennent enfants de l'objet parent. Vous vous assurez ainsi que chacun d'eux dispose de sa propre représentation sous forme de bitmap. Par ailleurs, chaque objet d'affichage sera capable de se déplacer par rapport aux autres sans avoir à charger de nouvelles images sur le processeur graphique.

Définissez la propriété `cacheAsBitmap` sur `true` au niveau le plus élevé de la liste d'affichage où les objets d'affichage enfant ne sont pas animés. En d'autres termes, définissez cette propriété pour les conteneurs d'objets d'affichage qui ne comportent pas d'éléments animés. Ne la définissez pas pour les objets d'affichage enfant. *Évitez également de la définir pour les sprites qui contiennent d'autres objets d'affichage animés.*

Lorsque vous définissez la propriété `z` d'un objet d'affichage, l'application utilise toujours une représentation bitmap en mémoire cache. Par ailleurs, une fois la propriété `z` d'un objet d'affichage définie, l'application fait appel à la représentation bitmap en mémoire cache, même si vous faites pivoter ou mettez à l'échelle l'objet. L'application n'utilise pas la propriété `cacheAsBitmapMatrix` pour les objets d'affichage dont la propriété `z` est définie. Les mêmes règles s'appliquent lorsque vous définissez les propriétés de tout objet d'affichage en trois dimensions, notamment les propriétés `rotationX`, `rotationY`, `rotationZ` et `transform.matrix3D`.

Ne définissez pas la propriété `scrollRect` ou `mask` d'un conteneur d'objet d'affichage qui comporte un contenu pour lequel vous souhaitez utiliser l'accélération matérielle. En définissant ces propriétés, vous désactivez en effet l'accélération matérielle pour le conteneur d'objet d'affichage et ses objets enfant. Au lieu de définir la propriété `mask`, superposez un objet d'affichage de masque à l'objet d'affichage que vous souhaitez masquer.

La taille des objets d'affichage disponibles pour l'accélération matérielle est limitée. Sur les anciens périphériques, la taille maximale est de 1 024 pixels ou moins en largeur et en hauteur. Sur les périphériques plus récents, la limite est de 2 048 pixels. Pour pouvez utiliser l'outil de diagnostic du rendu sur GPU pour tester les performances sur un périphérique.

Le processeur graphique mémorise également des images bitmap dans la RAM de l'iPhone. Il sollicite au moins autant de mémoire que la quantité requise par les images bitmap.

Le processeur graphique fait appel aux allocations de mémoire exprimées sous la forme de valeurs à la puissance 2 pour chaque dimension de l'image bitmap. Il peut par exemple réserver de la mémoire de taille 512 x 1 024 ou 8 x 32. Une image de 9 x 15 pixels consomme ainsi autant de mémoire qu'une image de 16 x 16 pixels. Pour les objets d'affichage mis en cache, vous pouvez utiliser les dimensions proches des puissances de 2 (mais pas au-delà) dans chaque direction. Il est ainsi plus efficace d'utiliser un objet d'affichage de 32 x 16-pixels qu'un objet d'affichage de 33 x 17 pixels.

Ne comptez pas sur le redimensionnement de la scène pour réduire la taille des actifs dimensionnés pour d'autres plates-formes (le bureau, par exemple). Faites de préférence appel à la propriété `cacheAsBitmapMatrix` ou redimensionnez les actifs avant la publication de l'application l'iPhone. Les objets 3D ignorent la propriété `cacheAsBitmapMatrix` lors de la mise en mémoire cache d'une image de surface. C'est pourquoi il est préférable de redimensionner les objets d'affichage avant la publication de l'application s'ils sont appelés à être rendus sur une surface 3D.

Il est nécessaire de parvenir à un juste équilibre entre les avantages de l'accélération matérielle et l'utilisation de la RAM. Au fur et à mesure que la mémoire se remplit, iPhone OS demande à d'autres applications iPhone natives en cours d'exécution de libérer de la mémoire. Lorsque les applications traitent cette notification et libèrent de la mémoire, elles risquent de rivaliser avec votre application pour disposer de cycles de l'unité centrale. Les performances de l'application risquent alors de se dégrader temporairement. Veillez à tester l'application sur des périphériques plus anciens, susceptibles de consacrer considérablement moins de mémoire au processus en cours d'exécution.

Lors du débogage de l'application sur l'iPhone, il est possible d'activer la fonction de diagnostic du rendu sur GPU. Cette fonction vous aide à comprendre comment votre application utilise le rendu sur GPU. Pour plus d'informations, voir la rubrique « [Débogage avec l'outil de diagnostic du rendu sur GPU](#) » à la section « [Débogage d'une application iPhone](#) » à la page 25.

Pour plus d'informations sur l'utilisation de la propriété `cacheAsBitmapMatrix`, voir la section « [DisplayObject.cacheAsBitmapMatrix](#) » d'« [API d'ActionScript propres aux applications AIR mobiles](#) » à la page 31.

Autres techniques d'optimisation des performances des objets d'affichage

L'accélération matérielle peut optimiser les performances graphiques de certaines classes d'objets d'affichage. Vous trouverez ci-dessous quelques conseils d'amélioration des performances graphiques :

- Essayez de limiter le nombre d'éléments visibles sur la scène. Le rendu de chaque élément et son impact sur les éléments qui l'entourent prennent un certain temps.
Lorsqu'il est superflu d'afficher un objet d'affichage, définissez la propriété `visible` correspondante sur `false` ou supprimez-le de la scène par le biais de `removeChild()`. Ne vous contentez pas de définir la propriété `alpha` correspondante sur `0`.
- Évitez les modes de fusion en général et le mode de fusion des calques en particulier. Dans la mesure du possible, utilisez le mode de fusion normal.
- Les filtres d'objet d'affichage se traduisent par une charge de calcul élevée. Réservez-les aux cas où ils sont absolument nécessaires. L'utilisation de quelques filtres dans un écran d'introduction, par exemple, ne présente généralement pas de problème. Évitez toutefois d'appliquer des filtres à un grand nombre d'objets, à des objets en cours d'animation ou lorsque l'utilisation d'une cadence élevée s'impose.
- Évitez les formes associées à un morphing.
- Évitez l'écrêtage.
- Dans la mesure du possible, définissez le paramètre `repeat` sur `false` lorsque vous appelez la méthode `Graphic.beginBitmapFill()`.
- Ne dessinez pas inutilement. Appliquez la couleur d'arrière-plan en tant qu'arrière-plan. Ne superposez pas les formes de grande taille. A chaque pixel à dessiner correspond une charge de traitement. Cela est particulièrement vrai pour les objets d'affichage qui ne prennent pas en charge l'accélération matérielle.
- Évitez les formes à pointes longues et fines, les bords qui se recoupent ou un volume élevé de détails le long des bords. Le rendu de ces formes est en effet plus long que celui des objets d'affichage aux bords lisses. Cela est particulièrement vrai pour les objets d'affichage qui ne prennent pas en charge l'accélération matérielle.
- Créez des bitmaps dont la taille est inférieure à $2^n \times 2^m$ bits, mais proche de cette valeur. Si les dimensions ne sont pas nécessairement exprimées sous la forme 2 à la puissance n , elles ne doivent ni s'en éloigner, ni dépasser cette valeur. Ainsi, le rendu d'une image de 31×15 pixels est plus rapide que celui d'une image de 33×17 pixels. (32 et 16 sont des puissances de 2 , et 31 et 15 sont des valeurs légèrement inférieures.) Ces images utilisent en outre plus efficacement la mémoire.
- Limitez la taille des objets d'affichage à $1\,024 \times 1\,024$ pixels (ou $2\,048 \times 2\,048$ sur les périphériques les plus récents).

Densité des informations

La taille physique de l'écran des périphériques mobiles est inférieure à celle d'un ordinateur de bureau, bien que la densité des pixels soit supérieure. Si un affichage plus net du texte est convivial, les glyphes doivent atteindre une taille physique minimale pour assurer leur lisibilité.

Les périphériques mobiles sont souvent utilisés en cours de déplacement et dans des environnements mal éclairés. Tenez compte du volume d'informations qu'il est possible d'afficher sans perte de lisibilité. Cette valeur est probablement inférieure à celle d'un écran d'ordinateur de bureau aux dimensions identiques en pixels.

Appliquez une hiérarchie typographique pour mettre en évidence les informations importantes. Exprimez l'importance relative des éléments de l'interface utilisateur par le biais de la taille de police, de l'épaisseur, du positionnement et du pas. Vous pouvez utiliser un ou plusieurs repères à chaque niveau de la hiérarchie. Appliquez ces repères avec cohérence dans l'ensemble de l'application. Ce repère peut être d'ordre spatial (retrait, interligne, positionnement) ou graphique (taille, style, couleur des caractères). L'utilisation de repères redondants permet souvent d'exprimer clairement une hiérarchie. Evitez toutefois d'utiliser plus de trois repères par niveau de groupement.

Essayez de simplifier les libellés et explications textuelles requis. Insérez, par exemple, un exemple de saisie dans un champ de texte afin de suggérer le contenu tout en évitant un libellé distinct.

Polices et entrée de texte

Utilisez de préférence les polices de périphérique. L'iPhone utilise par exemple les polices de périphérique suivantes :

- Serif : Times New Roman, Georgia et `_serif`
- Sans-serif : Helvetica, Arial, Verdana, Trebuchet, Tahoma et `_sans`
- Largeur fixe : Courier New, Courier et `_typewriter`

Utilisez des polices dont la taille est supérieure ou égale à 14 pixels.

Utilisez les polices de périphérique pour les champs de texte modifiables. Les polices de périphérique s'affichent également plus rapidement dans les champs de texte que les polices incorporées.

N'utilisez pas de texte souligné dans les champs de texte de saisie. Ne définissez pas non plus l'alignement des champs de texte. Sur l'iPhone, les champs de texte de saisie ne gèrent que l'alignement à gauche (valeur par défaut).

Si vous faites appel au paramètre Texte TLF pour un champ de texte dans Flash Professional CS5, désactivez la bibliothèque partagée à l'exécution dans la liaison par défaut dans Paramètres d'ActionScript 3.0. Si vous ne désactivez pas ce paramètre, l'application ne fonctionne pas sur l'iPhone, car elle essaie d'exécuter le fichier SWF issu de la bibliothèque partagée à l'exécution :

- 1 Sélectionnez Fichier > Paramètres de publication.
- 2 Dans la boîte de dialogue Paramètres de publication, cliquez sur l'onglet Flash.
- 3 Cliquez sur le bouton Script figurant sur la droite de la liste déroulante Script (ActionScript 3.0).
- 4 Cliquez sur l'onglet Chemin d'accès à la bibliothèque.
- 5 Dans la liste déroulante Liaison par défaut, sélectionnez Fusionné dans le code.

Envisagez de substituer d'autres solutions aux champs de texte de saisie. Il est, par exemple, inutile d'inviter l'utilisateur à entrer une valeur numérique dans un champ de texte. Proposez deux boutons pour augmenter ou réduire la valeur.

Tenez compte de la place qu'occupe le clavier virtuel. Lorsque le clavier virtuel est activé (si l'utilisateur touche un champ de texte, par exemple), l'application ajuste la position de la scène. Le repositionnement automatique assure la visibilité du champ de texte de saisie sélectionné :

- Un champ de texte figurant dans le haut de la scène se déplace vers le haut de la zone de scène visible. (La taille de la zone de scène visible est inférieure en raison de la présence du clavier virtuel.)
- Un champ de texte figurant dans le bas de la scène demeure au bas de la nouvelle zone de scène.
- Un champ de texte figurant dans une autre partie de la scène se déplace vers le centre vertical de la scène.

Lorsque l'utilisateur clique sur un champ de texte pour le modifier (et que le clavier virtuel est affiché), l'objet `TextField` distribue un événement `focusIn`. Vous pouvez associer un écouteur à cet événement pour repositionner le champ de texte.

Un champ de texte d'une seule ligne comprend un bouton Effacer (situé sur la droite du texte) lorsque l'utilisateur modifie le texte. Ce bouton Effacer ne s'affiche toutefois pas si le champ de texte est trop étroit.

Après avoir modifié le contenu d'un champ de texte d'une seule ligne, l'utilisateur ferme le clavier virtuel en appuyant brièvement sur la touche Terminé du clavier.

Après avoir modifié le contenu d'un champ de texte de plusieurs lignes, l'utilisateur ferme le clavier virtuel en appuyant brièvement en dehors du champ de texte. Le focus quitte alors le champ de texte. Assurez-vous que le champ de texte est entouré d'un espace dans l'application lorsque le clavier virtuel est affiché. Si la taille du champ de texte est trop élevée, celui-ci masque toutes les autres zones.

L'utilisation de certains composants Flash Professional CS5 interdit parfois le déplacement du focus à partir d'un champ de texte. Ces composants (`TextArea`, par exemple) sont conçus pour être utilisés sur un ordinateur de bureau, où ce comportement est souhaitable. Lorsqu'un champ de texte reçoit le focus (et qu'il est en cours de modification), il est impossible de déplacer ce dernier en cliquant sur un autre objet d'affichage. L'ajout d'autres composants Flash Professional CS5 sur la scène risque également d'interdire le déplacement du focus à partir du champ de texte en cours de modification.

Ne vous fiez pas aux événements de clavier. Un contenu SWF adapté au Web fait, par exemple, appel au clavier pour permettre à l'utilisateur de contrôler l'application. Toutefois, sur l'iPhone, le clavier virtuel n'est affiché que lorsque l'utilisateur modifie un champ de texte. Une application iPhone ne distribue un événement de clavier que lorsque le clavier virtuel est affiché.

Enregistrement de l'état de l'application

L'application risque de se fermer à tout moment (lorsque le téléphone sonne, par exemple). Envisagez d'enregistrer l'état de l'application à chaque fois qu'il est modifié. Vous pouvez, par exemple, enregistrer les paramètres dans un fichier ou une base de données qui réside dans le répertoire de stockage de l'application. Vous pouvez également enregistrer les données dans un objet partagé local. Il est ensuite possible de restaurer l'état de l'application lorsqu'elle redémarre. Si un appel téléphonique interrompt une application, elle redémarre au terme de l'appel.

Ne vous fiez pas à la distribution d'un événement `exitting` par l'objet `NativeApplication` lorsque l'application se ferme, car elle n'est pas systématique.

Changements d'orientation de l'écran

Un contenu iPhone peut être visualisé en mode portrait ou paysage. Pensez à la manière dont l'application gérera les changements d'orientation de l'écran. Pour plus d'informations, voir [Détection d'un changement d'orientation de l'écran](#).

Cibles tactiles

Tenez compte de la taille des cibles tactiles lorsque vous concevez les boutons et autres éléments de l'interface utilisateur touchés par l'utilisateur. Assurez-vous que la taille de ces éléments est suffisamment élevée pour que l'utilisateur puisse facilement les activer du doigt sur un écran tactile. Veillez également à ménager un espace suffisant entre cibles tactiles. Les cibles tactiles devraient mesurer entre 44 et 57 pixels.

Affectation de la mémoire

L'affectation de blocs de mémoire inutilisés se traduit par une augmentation de la charge de traitement. Elle risque de ralentir l'application ou d'affecter les performances pendant une animation ou une interaction, en raison du déclenchement du nettoyage de mémoire.

Dans la mesure du possible, recyclez les objets au lieu de vous débarrasser d'un objet existant, puis d'en créer un nouveau.

N'oubliez pas que les objets vectoriels consomment généralement moins de mémoire que les tableaux. Voir [Comparaison des classes Vector et Array](#).

Pour plus d'informations sur l'utilisation de la mémoire, voir [Conservation de mémoire](#).

API de dessin

Dans la mesure du possible, évitez de créer des graphiques par le biais de l'API de dessin d'ActionScript (classe Graphics). L'utilisation de l'API de dessin crée des objets sur la scène en mode dynamique, puis effectue leur rendu dans l'utilitaire de pixellisation. Créez de préférence ces objets en mode statique sur la scène lors de la phase de création.

Lorsqu'ils sont régulièrement appelés, les objets créés par le biais d'API de dessin sont détruits et recréés à chaque exécution d'ActionScript. Les objets statiques, en revanche, résident en mémoire au terme d'un scénario.

Propagation vers le haut (bubbling) des événements

Dans un conteneur d'objets d'affichage imbriqué à un grand nombre de niveaux, la propagation vers le haut des événements risque de solliciter une importante puissance de traitement. Pour réduire l'impact de cette opération, gérez chaque aspect de l'événement dans l'objet cible, puis appelez la méthode `stopPropagation()` de l'objet d'événement. En appelant cette méthode, vous désactivez la propagation vers le haut de l'événement. L'appel de cette méthode a également pour conséquence que les objets parent ne reçoivent pas l'événement.

La réduction des niveaux d'imbrication de l'objet d'affichage permet d'éviter les longues chaînes d'événements, d'où une charge de traitement réduite.

Dans la mesure du possible, inscrivez-vous aux événements `MouseEvent` au lieu des événements `TouchEvent`. Les événements `MouseEvent` sollicitent une charge de traitement inférieure à celle des événements `TouchEvent`.

Dans la mesure du possible, définissez les propriétés `mouseEnabled` et `mouseChildren` sur `false`.

Optimisation des performances vidéo

Pour optimiser la lecture vidéo mobile, limitez les activités exécutées en parallèle dans l'application. Les processus de décodage et de rendu de la vidéo peuvent ainsi solliciter autant que possible l'unité centrale.

Veillez à ce qu'aucun code ActionScript ne s'exécute pendant la lecture de la vidéo (ou réduisez-le autant que possible). Dans la mesure du possible, évitez tout code exécuté à fréquence régulière ou dans le scénario.

Évitez autant que possible de redessiner les objets d'affichage non vidéo. Évitez en particulier de redessiner les objets d'affichage qui recoupent la zone vidéo, même s'ils sont masqués sous la vidéo. Ils seront en effet redessinés et solliciteront des ressources de traitement. Attribuez par exemple une forme simple à l'indicateur de position et actualisez ce dernier deux fois par seconde uniquement plutôt qu'à chaque image. Placez les contrôles vidéo directement sous la zone vidéo afin qu'ils ne la recouvrent pas. Si une animation met en mémoire tampon la vidéo, ne la masquez pas derrière la vidéo lorsqu'elle n'est pas utilisée. Définissez-la sur invisible.

Composants Flex et Flash

De nombreux composants Flex et Flash sont conçus pour être utilisés dans des applications de bureau. Il est possible que ces composants, et notamment les composants d'affichage, s'exécutent plus lentement sur les périphériques mobiles. Outre une exécution plus lente, les composants de bureau peuvent disposer de modèles d'interaction qui ne sont pas adaptés aux périphériques mobiles.

Adobe développe actuellement une version mobile de la structure d'application de Flex. Pour plus d'informations, voir <http://labs.adobe.com/technologies/flex/mobile/>.

Réduction de la taille d'un fichier d'application

Voici quelques conseils permettant de réduire la taille du fichier IPA :

- Vérifiez les bitmaps d'arrière-plan pour vous assurer que leur taille est correcte (en d'autres termes, qu'elle n'est pas supérieure à la taille requise).
- Assurez-vous qu'aucune police supplémentaire n'est incorporée.
- Vérifiez si les actifs PNG contiennent des canaux alpha et supprimez-les s'ils ne sont pas nécessaires. Utilisez un utilitaire tel que PNG crunch pour réduire la taille des actifs PNG.
- Dans la mesure du possible, convertissez les actifs PNG en actifs JPG.
- Envisagez de compresser les fichiers son (en utilisant une vitesse de transmission inférieure).
- Supprimez tout actif inutilisé.