# Skapa ADOBE® AIR®-program med Packager for iPhone®

## Juridiska meddelanden

Mer information om juridiska meddelanden finns i http://help.adobe.com/sv_SE/legalnotices/index.html.

# Innehåll

# Kapitel 1: Komma igång med att skapa AIR-program för iPhone

Du kan använda Adobe® Flash®-plattformsverktyg och ActionScript® 3.0-kod om du vill skapa Adobe® AIR®-program för iPhone och iPod touch. Dessa program distribueras, installeras och körs på samma sätt som andra iPhone-program.

***Obs!*** *I dokumentet används hädanefter "iPhone" som beteckning på både iPhone och iPod touch.*

Packager for iPhone® ingår i Adobe® Flash® Professional CS5. I Packager for iPhone kompileras ActionScript 3.0-bytekod till intern iPhone-programkod. iPhone-program distribueras som iPhone-programinstallationsfiler (.ipa-filer) via iTunes Store.

Du kan använda Flash Professional CS5 eller Adobe® Flash® Builder™ 4 om du vill redigera ActionScript 3.0-källinnehållet för programmet.

Använd Flash Professional CS5 för att utveckla iPhone-program.

Du behöver även ett iPhone-utvecklarcertifikat från Apple.

**Viktigt:** Läs informationen om utformning av iPhone-program innan du börjar utveckla iPhone-program. Se "Om utformning av iPhone-program" på sidan 34. Läs också om de utvecklarfiler som krävs för att skapa iPhone-program. Se "Få utvecklarfiler från Apple" på sidan 4.

## Viktiga begrepp

Innan du börjar att utveckla iPhone-program med ActionScript 3.0 är det viktigt att du förstår arbetsflödet och de tillhörande begreppen.

### Ordlista

Följande termer är viktiga att förstå när du skapar ett iPhone-program.

**Webbplatsen iPhone Dev Center** Webbplats för Apple Computer (http://developer.apple.com/iphone/) där du kan göra följande:

- Ansök om att få bli en iPhone-utvecklare.
- Hantera och skapa iPhone-utvecklingscertifikat, provisioneringsprofiler och program-ID:n (vilka definieras nedan).
- Skicka in program för App Store.

**iPhone-utvecklarcertifikat** Används för att identifiera en utvecklare för utveckling av program.

Den filen får du från Apple. Du konverterar certifikatet till en P12-certifikatfil för att signera iPhone-programmet som du skapar med hjälp av ActionScript 3.0. Se *P12-certifikatfil*.

Du behöver inget iPhone-utvecklingscertifikat för att felsöka och testa Flash Professional CS5-program på utvecklingsdatorn. Du behöver emellertid ett utvecklarcertifikat för att installera och testa programmen på en iPhone.

Utvecklarcertifikatet skiljer sig från ett distributionscertifikat, som du använder för att skapa en slutgiltig version av programmet. Du får ett disitributionscertifikat från Apple när du skapar den slutgiltiga versionen av ditt program.

**CSR-fil (certificate signing request)** En fil som innehåller personliga uppgifter och som används för att skapa ett utvecklarcertifikat. Den kallas även en CSR-fil.

**Provisioneringsprofil** En fil som låter dig testa eller distribuera ett iPhone-program. Du får filer för provisioneringsprofiler från Apple. En provisioneringsprofil tilldelas ett specifikt utvecklingscertifikat, ett program-ID och ett eller flera enhets-ID. Det finns olika typer av provisioneringsprofiler:

- **Provisioneringsprofil för utveckling** – Används för att installera en testversion av ett program på utvecklarens iPhone.
- **Provisioneringsprofil för testning** – Även kallat en ”ad hoc-provisioneringsprofil”. Används för att distribuera en testversion av programmet till flera användare (och iPhone-enheter). Med hjälp av denna provisioneringsprofil och testprogrammet kan användare testa ditt program utan att det skickas till App Store. Observera att du också kan använda en provisioneringsprofil för utvecklare för att distribuera testprogram till flera enheter.
- **Provisioneringsprofil för distribution** – Används för att skapa ett iPhone-program för att skicka ditt program till App Store.

**Program-ID** En unik sträng som identifierar ett iPhone-program (eller flera program) från en viss utvecklare. Du skapar dessa ID på webbplatsen iPhone Dev Center. I varje provisioneringsprofil finns ett associerat program-ID eller program-ID-mönster. Du använder dessa program-ID (eller mönster) vid programutvecklingen. Du använder program-ID i dialogrutan iPhone-inställningar i Flash Professional CS5 (eller i programbeskrivningsfilen).

Program-ID:n på iPhone Dev Center innehåller ett källpaket-ID följt av en källidentifierare. Källpaket-ID består av en teckensträng, som exempelvis 5RM86Z4DJM, som Apple tilldelar till program-ID:t. Källidentifieraren innehåller ett inverterat domännamn som du väljer själv. Källidentifieraren kan sluta med en asterisk (*) vilket är ett jokertecken för program-ID:n. Här visas några exempel:

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.* (ett jokertecken för progam-ID)

Det finns två typer av program-ID på iPhone Dev Center:

- Program-ID med jokertecken – På iPhone Dev Center avslutas dessa program-ID med en asterisk (*), som exempelvis i 96LPVWEASL.com.myDomain.* eller 96LPVWEASL.*. Om du har en provisioneringsprofil som använder den här typen av program-ID, kan du skapa testprogram som använder ett program-ID som matchar mönstret. Om det är ett programs program-ID kan du byta ut asterisken mot en valfri sträng med giltiga tecken. Om exempelvis 96LPVWEASL.com.example.* anges som program-ID på webbplatsen iPhone Dev Center, kan du använda com.example.foo eller com.example.bar som programmets program-ID.
- Specifika program-ID – Dessa definierar ett unikt program-ID som ska användas i ett program. På iPhone Dev Center avslutas dessa program-ID:n inte med en asterisk. Ett exempel kan vara 96LPVWEASL.com.myDomain.myApp. Om en provisioneringsprofil använder den här typen av program-ID, måste programmet matcha detta program-ID exakt. Om exempelvis 96LPVWEASL.com.example.helloWorld anges som program-ID på webbplatsen iPhone Dev Center, kan du använda com.example.foo som programmets program-ID.

När du utvecklar ett program anger du ditt program-ID i dialogrutan iPhone-inställningar i Flash Professional CS5 eller i programbeskrivningsfilen. Mer information om program-ID finns i avsnittet "Fliken Distribution" i ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14 eller i ”Ställa in iPhone-programegenskaper i programbeskrivningsfilen” på sidan 16.

**Viktigt:** När du anger ett program-ID, ska du ignorera delen med källpaket-ID i program-ID:t Om exempelvis Apple visar ditt program-ID som 96LPVWEASL.com.example.bob.myApp, ska du ignorera 96LPVWEASL och bara

använda com.example.bob.myApp som program-ID. Om Apple visar att ditt program-ID är 5RM86Z4DJM.*, ska du ignorera 5RM86Z4DJM eftersom detta är ett program-ID med jokertecken.

Du kan hitta program-ID:t (eller mönster för program-ID med jokertecken) som är associerat med en provisioneringsprofil på iPhone Dev Center (http://developer.apple.com/iphone). Gå till portalen för iPhone-utvecklingsprogram och sedan till avsnittet för provisionering.

**P12-certifikatfil** En P12-fil (en fil med filtillägget .p12) är en typ av certifikatfil (en Personal Information Exchange-fil). I Packager for iPhone används den här typen av certifikat för att skapa iPhone-program. Du konverterar utvecklarcertifikatet som du får från Apple till den här typen av certifikat.

**Unikt enhets-ID** En unik kod som identifierar en specifik iPhone. Det kallas även UDID eller enhets-ID.

## Översikt av utvecklingsarbetsflödet

När du utvecklar ett program för iPhone ska du följa dessa steg:

1. Installera Flash Professional CS5 från Adobe.
2. Installera iTunes.
3. Få utvecklarfiler från Apple. Dessa filer inkluderar utvecklarcertifikatet och provisioneringsprofiler. Se ”Få utvecklarfiler från Apple” på sidan 4.
4. Konvertera utvecklingscertifikatet till en P12-certifikatfil. Flash CS5 kräver att certifikatet är ett P12-certifikat. Se ”Få utvecklarfiler från Apple” på sidan 4.
5. Använd iTunes för att associera din provisioneringsprofil med din iPhone.
6. Skriv programmet i Flash Professional CS5.

   Det är viktigt att du förstår hur du på bästa sätt kan utforma och optimera koden för ett iPhone-program. Se ”Om utformning av iPhone-program” på sidan 34.

   Vissa ActionScript 3.0 API:er är dessutom begränsade eller stöds inte på iPhone. Se ”ActionScript 3.0-API:er som stöds för mobilenheter” på sidan 27.

   Du kan också använda Flash Builder 4.0 om du vill redigera ActionScript 3.0-kod för programmet.

   Du kan använda Flash Professional CS5 om du vill testa programmet på en utvecklingsdator.
7. Skapa ikoner och inledande skärmbilder för programmet. Varje iPhone-program innehåller en uppsättning ikoner som identifierar det för en användare. Den inledande skärmbilden visas på iPhone-skärmen under det att programmet läses in. Se ”Ikoner och inledande skärmbilder för iPhone” på sidan 12.
8. Redigera iPhone-inställningarna. Inställningarna inkluderar följande:
   - Programmets identitet (inklusive filnamnet, programnamnet, versionsnumret och programmets ID)
   - Källplatsen för programmets ikoner
   - P12-certifikatet och provisioneringsprofilen som tilldelats programmet
   - De inledande skärmproportionerna för programmet

   I Flash Professional CS5 kan du redigera dessa inställningar i dialogrutan för iPhone-inställningar. Mer information finns i ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14.

   Du kan även redigera dessa inställningar direkt i programbeskrivningsfilen. Mer information finns i ”Ställa in iPhone-programegenskaper i programbeskrivningsfilen” på sidan 16.
9. Kompilera IPA-filen med Packager for iPhone. Se ”Kompilera en installationsfil för iPhone-program (IPA-fil)” på sidan 19.

**10** Installera och testa programmet på din iPhone. Använd iTunes för att installera IPA-filen.

Vid ad hoc-distribution upprepar du de här allmänna stegen, men använder en provisioneringsprofil för testning istället för en provisioneringsprofil för utveckling. Vid den slutgiltiga distributionen av programmet upprepar du de här stegen med provisioneringsprofilen för distribution. (Se ”Ordlista” på sidan 1 för mer information om de olika typerna av provisioneringsprofiler.)

När du har skapat en distributionsversion av programmet ska du läsa anvisningarna i ”Skicka iPhone-programmet till App Store” på sidan 25.

En snabb genomgång av hur du skapar ett enkelt iPhone-program finns i ”Skapa ett ”Hello World” iPhone-program med Flash Professional CS5” på sidan 8.

# Få utvecklarverktyg från Adobe

Du måste ha Flash Professional CS5 om du vill utveckla iPhone-program med ActionScript 3.0.

**Viktigt!** Du bör uppdatera Packager for iPhone från den förhandsversion som ingick i Flash Professional CS5. Välj Hjälp > Uppdateringar i Flash Professional CS5.

Du kan även använda Flash Builder 4 för att redigera ActionScript-kod. Flash Builder 4 finns på http://www.adobe.com/se/products/flashbuilder/.

# Få utvecklarfiler från Apple

Precis som när du utvecklar andra program för iPhone måste du först få utvecklarfiler för iPhone från Apple. Du behöver ett iPhone-utvecklarcertifikat och en mobil provisioneringsprofil. Du behöver också få andra provisioneringsprofiler. Se ”Ordlista” på sidan 1 för en beskrivning av dessa filer.

***Obs!*** *Att hämta dessa filer är en viktig del av programutvecklingsprocessen. Se till att detta är gjort innan du börjar utveckla dina program. Det är inte så enkelt att hämta utvecklarfiler. Du ska därför noggrant läsa anvisningarna här och på webbplatsen iPhone Dev Center.*

## Få och arbeta med utvecklarfiler för iPhone

Du behöver ett iPhone-utvecklarcertifikat och provisioneringsprofiler från Apple. Du behöver även konvertera certifikatet till ett P12-certifikat.

### Installera iTunes

Du behöver iTunes för att installera ditt program på din iPhone. Du använder även iTunes för att avgöra enhets-ID för din iPhone. Du behöver känna till ditt enhets-ID när du ansöker om ett iPhone-utvecklarcertifikat.

### Ansöka om ett iPhone-utvecklarcertifikat och skapa en provisioneringsprofil

Om du inte redan gjort det ska du registrera dig som en iPhone-utvecklare på webbplatsen iPhone Dev Center (http://developer.apple.com/iphone/).

***Obs!*** *Du behöver inte iPhone SDK eller XCode för att utveckla AIR-program för iPhone. Du behöver inte vara en registrerad iPhone-utvecklare. Dessutom behöver du inte hämta ett utvecklarcertifikat och en provisioneringsprofil.*

**1** Logga in som iPhone-utvecklare på iPhone Dev Center med ditt konto-ID.

2 På iPhone Dev Center ansöker du om (eller köper) ett iPhone-utvecklarcertifikat.

Du kommer att få ett e-postmeddelande från Apple med en aktiveringskod för iPhone Developer Program.

3 Gå tillbaka till iPhone Dev Center. Följ instruktionerna om hur du aktiverar ditt utvecklarprogram (och ange din aktiveringskod när det behövs).

4 När aktiveringskoden godkänts går du till portalen för iPhone-utvecklingsprogram på iPhone Dev Center.

5 Skapa en CSR-fil (Certificate Signing Request). Du använder filen för att erhålla ett iPhone-utvecklingscertifikat (iPhone Development Certificate). Mer information finns i ”Skapa en CSR-fil” på sidan 6.

6 I nästa steg kommer blir du ombedd att ange enhets-ID:t (eller ett unikt enhets-ID) för din iPhone. Du får ditt unika enhets-ID från iTunes:

**a** Anslut din iPhone med en USB-kabel. Välj sedan sammanfattningsfliken för din iPhone i iTunes.

**b** När du har hämtat provisioneringsprofilen från iPhone-utvecklingscentret lägger du till den i iTunes.

**c** Klicka på det serienummer som visas. Det unika enhets-ID kommer nu att visas. Klicka på Kommando-C på en Mac eller på Ctrl+C på en Windows-dator för att kopiera det unika enhets-ID:t till urklipp.

7 Skapa och installera en provisioneringsprofil (Provisioning Profile) och ett iPhone-utvecklingscertifikat (iPhone Development Certificate).

Följ anvisningarna på iPhone Dev Center. Leta efter anvisningar i portalen för iPhone-utvecklingsprogram. Du kan använda utvecklingsprovisioneringsassistenten för att erhålla ditt utvecklingscertifikat och skapa en provisioneringsprofil.

Bortse från stegen som innehåller XCode. Du behöver inte använda XCode för att utveckla iPhone-program med Flash Professional CS5.

8 Välj Arkiv > Lägg till i biblioteket i iTunes. Välj sedan filen med provisioneringsprofilen (med filtypen mobileprovision). Synkronisera därefter din iPhone med iTunes.

Det här gör att du kan testa programmet som är associerat med denna provisioneringsprofil på din iPhone.

Om du vill kontrollera att en viss provisioneringsprofil har lagts till i iTunes, kan du försöka att lägga till den i biblioteket. Om iTunes frågar om du vill ersätta en befintlig provisioneringsprofil, kan du trycka på knappen Avbryt. (Profilen har redan installerats.) Du kan också kontrollera vilka provisioneringsprofiler som är installerade på din iPhone:

**a** Öppna inställningsprogrammet på din iPhone.

**b** Öppna kategorin Allmänt.

**c** Tryck på Profiler. De installerade provisioneringsprofilerna visas på sidan Profiler.

9 Om du inte har gjort detta hämtar du iPhone-utvecklingscertifikatfilen (en .cer-fil).

Du kan genom utvecklingsprovisioneringsassistenten ha fått en länk för att hämta den här filen. Du kan också hitta filen i certifikatsavsnittet i provisioneringsportalen på webbplatsen Apple iPhone Dev Center (http://developer.apple.com/iphone/).

10 Därefter ska du konvertera iPhone-utvecklarcertifikatet till en P12-fil. Mer information finns i ”Konvertera ett utvecklarcertifikat till en P12-fil” på sidan 7.

Du kan nu skapa ett enkelt Hello World-program. Läs mer i ”Skapa ett ”Hello World” iPhone-program med Flash Professional CS5” på sidan 8.

## Skapa en CSR-fil

Om du vill erhålla ett utvecklarcertifikat skapar du en CSR-fil (Certificate Signing Request) som du skickar på webbplatsen iPhone Dev Center.

### Skapa en CSR-fil i Mac OS

På en dator med Mac OS kan du använda programmet Nyckelhanterare för att skapa en CSR-fil. Programmet Nyckelhanterare finns i underkatalogen till Verktyg i Program-katalogen. På Nyckelhanterare-menyn väjer du Certifikatassistent > Begär ett certifikat från en certifikatsutfärdare.

1. Öppna Nyckelhanterare.
2. Välj Inställningar på Nyckelhanterare-menyn.
3. Klicka på Certifikat i dialogrutan Inställningar. Välj sedan Av för Status för Online-certifikatprotokoll och Lista över återkallade certifikat till Av. Stäng sedan dialogrutan.
4. På Nyckelhanterare-menyn väjer du Certifikatassistent > Begär ett certifikat från en certifikatsutfärdare.
5. Ange e-postadressen och namnet som motsvarar iPhone-utvecklarens konto-ID. Ange ingen e-postadress för CA. Markera Begäran är Sparat till skiva och klicka sedan på knappen Fortsätt.
6. Spara filen (CertificateSigningRequest.certSigningRequest).
7. Överför CSR-filen till Apple på webbplatsen för iPhone-utvecklare. (Se även "Ansöka om ett iPhone-utvecklarcertifikat och skapa en provisioneringsprofil".)

### Skapa en CSR-fil i Windows

För Windows-utvecklare kan det vara lättast att få iPhone-utvecklarcertifikatet på en Mac-dator. Men det är även möjligt att få ett certifikat på en Windows-dator. Först skapar du en CSR-fil (certificate signing request) med OpenSSL:

1. Installera OpenSSL på din Windows-dator. (Gå till http://www.openssl.org/related/binaries.html.)

   Du kanske även måste installera Visual C++ 2008 Redistributable-filer, som visas på hämtningssidan för OpenSSL. (Du behöver *inte* ha Visual C++ installerad på datorn.)
2. Öppna kommandoprompten i Windows och gå till OpenSSL bin-katalogen (till exempel c:\OpenSSL\bin\).
3. Skapa den personliga nyckeln genom att ange följande i kommandoraden:

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

   Spara den här personliga nyckelfilen. Du kommer att behöva den senare.

   När du använder OpenSSL ska du inte ignorera felmeddelanden. Om du får ett felmeddelande i OpenSSL kan programmet ändå generera filer. Dessa filer kommer kanske inte att vara användbara. Om du får ett felmeddelande ska du kontrollera syntaxen och köra om kommandot.
4. Skapa CSR-filen genom att ange följande i kommandoraden:

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

   Byt ut värden för e-postadress, CN (certifikatnamn) och C (land) med dina egna värden.
5. Överför CSR-filen till Apple på webbplatsen för iPhone-utvecklare. (Se även "Ansöka om ett iPhone-utvecklarcertifikat och skapa en provisioneringsprofil".)

## Konvertera ett utvecklarcertifikat till en P12-fil

Om du vill utveckla iPhone-program med Flash Professional CS5 måste du använda en P12-certifikatfil. Du skapar detta certifikat baserat på filen med Apple iPhone-utvecklarcertifikatet som du får från Apple.

### Konvertera iPhone-utvecklarcertifikatet till en P12-fil i Mac OS

När du har hämtat iPhone-certifikatet från Apple exporterar du det till ett P12-certifikatformat. Så här gör du i Mac® OS:

1. Öppna programmet Nyckelhanterare (i mappen Program/Verktygsprogram).
2. Om du inte redan lagt till certifikatet till nyckelhanteraren väljer du Arkiv > Importera. Navigera sedan till certifikatfilen (.cer-filen) som du fick från Apple.
3. Välj nyckelkategorin i Nyckelhanterare.
4. Välj den personliga nyckeln som är associerad med ditt iPhone Development Certificate.

   Den personliga nyckeln identifieras av iPhone-utvecklaren: <Förnamn> <Efternamn>, öppet certifikat som är kopplat till det.
5. Välj Arkiv > Exportera objekt.
6. Spara nyckeln i Personal Information Exchange-format (.p12).
7. Du kommer att få ett meddelande om att skapa ett lösenord som används om du försöker att importera den här nyckeln till en annan dator.

### Konvertera ett Apple-utvecklarcertifikat till en P12-fil i Windows

För att kunna utveckla iPhone-program med Flash CS5 måste du använda en P12-certifikatfil. Du skapar detta certifikat baserat på filen med Apple iPhone-utvecklarcertifikatet som du får från Apple.

1. Konvertera filen för utvecklarcertifikatet som du får från Apple till en PEM-certifikatfil. Kör följande kommandoradssats från bin-katalogen i OpenSSL:

   ```
   openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
   ```
2. Om du använder den personliga nyckeln från nyckelringen på en Mac-dator konverterar du den till en PEM-nyckel:

   ```
   openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
   ```
3. Du kan nu skapa en giltig P12-fil baserat på nyckeln och på PEM-versionen av iPhone-utvecklarcertifikatet:

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

   Om du använder en nyckel från nyckelringen i Mac OS använder du PEM-versionen som du skapade i föregående steg. Annars använder du OpenSSL-nyckeln som du skapade tidigare (i Windows).

## Hantera certifikat, enhets-ID:n, program-ID:n och provisioneringsprofiler

Du kan hantera certifikat, enhets-ID:n, program-ID:n och provisioneringsprofiler på webbplatsen iPhone Dev Center (http://developer.apple.com/iphone/). Gå till portalen för iPhone-utvecklingsprogram på webbplatsen.

- Klicka på certifikatlänken om du vill hantera dina utvecklarcertifikat. Du kan skapa, hämta och återkalla certifikat. Om du vill skapa ett certifikat måste du först skapa en CSR-fil. Se även ”Skapa en CSR-fil” på sidan 6.
- Klicka på enhetslänken för att hantera listan med enheter som ditt utvecklingsprogram kan installeras på.
- Klicka på länken för program-ID:n för att hantera dina program-ID:n. När du skapar en provisioneringsprofil kommer den att kopplas till ett program-ID.

- Klicka på länken för provisionering för att hantera provisioneringsprofiler. Du kan även använda utvecklingsprovisioneringsassistenten för att skapa provisioneringsprofiler för utveckling.
- Klicka på länken Distribution när du vill skicka ditt program till App Store eller skapa en tillfällig version av programmet. I det här avsnittet finns en länk till webbplatsen iTunes Connect, som du använder för att skicka program till App Store.

# Skapa ett ”Hello World” iPhone-program med Flash Professional CS5

**Viktigt:** Innan du skapar programmet ska du hämta de utvecklarprogram och filer som krävs. Se ”Få utvecklarverktyg från Adobe” på sidan 4 och ”Få utvecklarfiler från Apple” på sidan 4.

## Skapa ett Flash Professional CS5-projekt

Du kan skapa ett iPhone-program direkt i Flash Professional CS5:

1. Öppna Flash CS5.
2. Välj Arkiv > Nytt.
3. Välj iPhone OS.
4. Klicka på OK.

## Lägga till innehåll till programmet

Nästa steg är att lägga till texten ”Hello world!” till programmet:

1. Välj textverktyget och klicka på scenen.
2. I egenskaperna för textfältet väljer du klassisk text (inte TLF-text).

   Det här är ett enkelt program och därför används klassisk text. Om du vill använda TLF-text måste du ange flera andra inställningar. Se ”Teckensnitt och textinmatning” på sidan 37.
3. Skriv ”Hello World!” i det nya textfältet.
4. Markera textfältet med markeringsverktyget.
5. Öppna sedan egenskapsinspektören och ange följande inställningar:
   - Tecken > Familj: _sans
   - Tecken > Storlek: 50
   - Position > X: 20
   - Position > Y: 20
6. Spara filen.
7. Välj Kontroll > Testa film > I AIR Debug Launcher (mobil).

   SWF-innehållet kompileras i Flash Professional CS5 och en version av programmet visas i AIR Debug Launcher (ADL). På det här sättet får du en snabb förhandsgranskning av ditt program.

## Skapa ikoner och inledande skärmbilder för programmet

Alla iPhone-program innehåller ikoner som visas i användargränssnittet i iTunes-programmet och på iPhone-skärmen.

1 Skapa en katalog i din projektkatalog med namnet ”icons”.

2 Skapa tre PNG-filer i katalogen ”icons”. Ge dem namnen Icon29.png, Icon57.png och Icon512.png.

3 Redigera PNG-filerna till den typ av grafik som du vill ha för ditt program. Filerna måste vara 29 x 29 pixlar, 57 x 57 pixlar och 512 x 512 pixlar. För det här testet kan du helt enkelt använda enfärgade fyrkanter som grafik.

För alla iPhone-program visas en inledande bild när programmet läses in på iPhone-enheten. Du skapar den inledande bilden i en PNG-fil:

1 Skapa en PNG-fil med namnet Default.png i utvecklingens huvudkatalog. (Lägg *inte* den här filen i underkatalogen ”icons”. Se till att du ger filen namnet Default.png, med ett versalt D.)

2 Redigera filen så att den är 320 pixlar bred och 480 pixlar hög. Tills vidare kan innehållet vara en tom vit rektangel. (Du kommer att ändra detta senare.)

***Obs!*** *När du skickar ett program till Apples App Store använder du en JPG-version (inte en PNG-version) av filen med 512 pixlar. PNG-versionen använder du medan du testar utvecklingsversionerna av ett program.*

Mer information om grafiken finns i ”Ikoner och inledande skärmbilder för iPhone” på sidan 12.

## Redigera programinställningar

**Viktigt:** Om du inte redan har gjort det ska du hämta de nödvändiga utvecklarprogrammen och filerna för iPhone-utveckling. Se ”Få utvecklarfiler från Apple” på sidan 4.

I dialogrutan för iPhone-inställningar i Flash Professional CS5 anger du många grundläggande egenskaper för iPhone-programmet.

1 Välj Arkiv > iPhone-inställningar.

2 Ange följande inställningar på fliken Allmänt:

- Utdatafil: HelloWorld.ipa

  Detta är namnet på iPhone-installationsfilen som ska skapas.

- Programnamn: Hello World

  Detta är namnet på programmet som visas under programikonen på iPhone-skärmen.

- Version: 1.0

  Versionen av programmet.

- Proportioner: stående
- Helskärmsläge: markerat.
- Automatisk orientering: avmarkerat.
- Återgivning: CPU

  De övriga alternativen, GPU och Auto, använder maskinvaruacceleration för återgivning. Den här funktionen kan hjälpa till att förbättra prestandan för grafikintensiva program (till exempel spel) som har en design som utnyttjar fördelarna med maskinvaruacceleration. Mer information finns i ”Maskinvaruacceleration” på sidan 34.

- Inkluderade filer: Lägg till den inledande skärmbildsfilen (Default.png) i listan Inkluderade filer.

***Obs!*** *För detta Hello World-exempel ska du inte ändra inställningarna från dem som visas i instruktionerna. För vissa inställningar, till exempel versionsinställningen, gäller speciella restriktioner. Dessa restriktioner beskrivs i "Programinställningar för iPhone" på sidan 14.*

3 Ange följande inställningar på fliken Distribution:

- Certifikat: Bläddra och välj det .p12-certifikat som är baserat på utvecklarcertifikatet som du har fått från Apple.

  Det här certifikatet används för att signera filen. Du måste konvertera iPhone-certifikatet från Apple till .p12-format. Mer information finns i "Få utvecklarverktyg från Adobe" på sidan 4.

- Lösenord: ange lösenordet för certifikatet.
- Provisioneringsfil: Bläddra och välj den provisioneringsfil för utvecklare som du har fått från Apple. Se "Få utvecklarfiler från Apple" på sidan 4.
- Program-ID: Om det här fältet går att välja kan du ange ett program-ID som stämmer överens med det program-ID som du skickade till Apple (till exempel com.example.as3.HelloWorld).

  Med ett program-ID kan programmet identifieras.

  Om det här fältet inte går att välja är provisioneringsprofilen bunden till ett särskilt program-ID. Program-ID visas i fältet.

  Mer information om hur du anger ett program-ID finns i avsnittet "Fliken Distribution" i "Ställa in iPhone-programegenskaper i Flash Professional CS5" på sidan 14.

4 Klicka på ikonen som är 29 x 29 pixlar i ikonlistan på fliken Ikoner. Ange sedan platsen för PNG-filen med 29 x 29 pixlar som du skapade tidigare (se "Skapa ikoner och inledande skärmbilder för programmet" på sidan 9). Ange sedan PNG-filerna för ikonerna med 57 x 57 pixlar och 512 x 512 pixlar.

5 Klicka på OK.

6 Spara filen.

Mer information om programinställningar finns i "Programinställningar för iPhone" på sidan 14.

## Kompilera IPA-filen

Du kan nu kompilera IPA-installationsfilen:

1 Välj Arkiv > Publicera.

2 Klicka på OK i dialogrutan iPhone-inställningar.

Packager for iPhone genererar iPhone-programmets installationsfil HelloWorld.ipa i projektkatalogen. Det kan ta några minuter att kompilera IPA-filen.

## Installera programmet på en iPhone

Så här installerar du iPhone-programmet för att testa det på en iPhone:

1 Öppna iTunes-programmet.

2 Om du inte redan har gjort det ska du lägga till provisioneringsprofilen för programmet i iTunes. Välj Arkiv > Lägg till i biblioteket i iTunes. Välj sedan filen med provisioneringsprofilen (med filtypen mobileprovision).

När du testar programmet på din utvecklar-iPhone använder du provisioneringsprofilen för utveckling.

Sedan när du distribuerar programmet till iTunes Store använder du profilen för distribution. Om du vill göra en ad hoc-distribution av programmet (till flera enheter utan att gå via iTunes Store) använder du ad hoc-provisioneringsprofilen.

Mer information om provisioneringsprofiler finns i ”Få utvecklarfiler från Apple” på sidan 4.

3 I vissa versioner av iTunes ersätts inte programmet om samma version av programmet redan finns installerat. I så fall tar du bort programmet från din enhet och från listan över program i iTunes.

4 Dubbelklicka på IPA-filen för programmet. Det borde visas i listan över program.

5 Anslut din iPhone till USB-porten på datorn.

6 Kontrollera programfliken för enheten i iTunes och att programmet är markerat i listan över program som ska installeras.

7 Markera enheten i den vänstra listan i iTunes-programmet. Klicka sedan på Synkronisera. När synkroniseringen är slutförd visas programmet Hello World i din iPhone.

Om den nya versionen inte har installerats tar du bort programmet från din iPhone och från listan över program i iTunes och upprepar sedan proceduren. Detta kan bero på att den installerade versionen använder samma program-ID och version.

Om ett fel visas i iTunes när du försöker installera programmet ska du läsa "Felsökning av installationsprogram" i ”Installera ett iPhone-program” på sidan 22.

## Redigera grafiken för den inledande skärmen

Innan du kompilerade programmet skapade du filen Default.png (se ”Skapa ikoner och inledande skärmbilder för programmet” på sidan 9). Den här PNG-filen visas som startbild medan programmet läses in. När du testade programmet på din iPhone kanske du lade märke till den tomma skärmen som visades när programmet startades.

Du bör ändra den här bilden för att matcha startskärmen för programmet (”Hello World!”):

1 Öppna programmet på din enhet. När den första Hello World-texten visas trycker du på och håller ned hemknappen (under skärmen). Samtidigt som du håller ned hemknappen trycker du på viloläge-/strömknappen (på ovansidan av din iPhone). Då tas en skärmavbild som skickas till kamerarullen.

2 Överför bilden till utvecklingsdatorn med iPhoto eller med något annat bildöverföringsprogram. (I Mac OS kan du också använda programmet Image Capture.)

   Du kan också skicka bilden via e-post till utvecklingsdatorn:

   - Öppna bildprogrammet.
   - Öppna kamerarullen.
   - Öppna den skärmavbild som du har tagit.
   - Tryck på bilden och tryck sedan på framåtpilen längst ned till vänster. Klicka sedan på knappen E-posta bild och skicka bilden till dig själv.

3 Ersätt filen Default.png (i utvecklingskatalogen) med en PNG-version av skärmavbilden.

4 Kompilera om programmet (se ”Kompilera IPA-filen” på sidan 10) och installera om det på din iPhone.

Den nya startskärmen används nu under det att programmet läses in.

***Obs!*** *Du kan skapa vilken grafik du vill i filen Default.png, så länge som grafiken har rätt dimensioner (320 x 480 pixlar). Men det är oftast bäst att bilden Default.png matchar det inledande tillståndet för programmet.*

# Kapitel 2: Kompilera och felsöka iPhone-program

Du kan kompilera ett iPhone-program med Packager for iPhone. Packager for iPhone ingår i Flash Professional CS5.

Du kan felsöka programmet på en utvecklingsdator. Du kan också installera en felsökningsversion på din iPhone och få `trace()`-data i Flash Professional CS5.

För en genomgång av hur du skapar ett iPhone-program från början till slut, se ”Skapa ett ”Hello World” iPhone-program med Flash Professional CS5” på sidan 8.

## Ikoner och inledande skärmbilder för iPhone

Alla iPhone-program innehåller ikoner som visas i användargränssnittet i iTunes-programmet och på iPhone-skärmen.

**Ikoner för iPhone-program**

Du definierar följande ikoner för ett iPhone-program:

- En ikon med storleken 29 x 29 pixlar, som används av Spotlight-sökresultat på iPhone och iPod touch.
- En ikon med storleken 48 x 48 pixlar, som används av Spotlight-sökresultat på iPad.
- En ikon med storleken 57 x 57 pixlar, som används i hemskärmen på iPhone och iPod touch.
- En ikon med storleken 72 x 72 pixlar (valfri), som används i hemskärmen på iPad.
- En ikon med storleken 512 x 512 pixlar, som används av iTunes. PNG-filen med 512 pixlar används bara för att testa utvecklingsversioner av ditt program. När du skickar det färdiga programmet till Apple App Store skickar du 512-bilden separat som en JPG-fil. Den ingår inte i IPA-filen.

I Flash Professional CS5 lägger du till de här ikonerna på fliken Ikoner i dialogrutan iPhone-inställningar. Läs mer i ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14.

Du kan också lägga till ikonernas platser i programbeskrivningsfilen:

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

Den glänsande effekten läggs till i iPhone. Du behöver inte ha med den i din källbild. Om du vill ta bort den här glänsande standardeffekten lägger du till elementet `InfoAdditions` i programbeskrivningsfilen:

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

Se ”Ställa in iPhone-programegenskaper i programbeskrivningsfilen” på sidan 16.

**Den inledande skärmbilden (Default.png)**

För alla iPhone-program visas en inledande bild när programmet läses in på iPhone-enheten. Du skapar den inledande bilden i en PNG-fil med namnet Default.png. Skapa en PNG-fil med namnet Default.png i utvecklingens huvudkatalog. (Lägg *inte* den här filen i en underkatalog. Se till att du ger filen namnet Default.png, med ett versalt D.)

Filen Default.png är 320 pixlar bred och 480 pixlar hög, oberoende av den ursprungliga programorienteringen eller om den upptar hela skärmen eller inte.

Om den ursprungliga programorienteringen är liggande, ska du använda samma dimensioner som används i ett stående program: 320 pixlar bred och 480 pixlar hög. Du ska emellertid rotera grafik 90° moturs i PNG-filen. Den vänstra sidan av PNG-grafiken motsvarar den övre delen av iPhone-skärmen i liggande läge. (Mer information om hur du ställer in den inledande skärmbildens orientering i programmet finns i ”Programinställningar för iPhone” på sidan 14.)

För ett program som inte upptar hela skärmen kommer de 20 översta pixlarna av standardbilden att ignoreras. Statusfältet i iPhone visas över en 20 pixlar bred rektangel högst upp i standardbilden. I ett program med liggande orientering motsvarar det här området den 20 pixlar breda rektangeln till vänster i filen Default.png (vilken visas högst upp i liggande läge). I program med stående orientering motsvarar detta den översta 20 pixlar breda rektangeln i filen Default.png.

För de flesta program ska bilden Default.png motsvara programmets startbild. Så här tar du en skärmavbild av startskärmen i ditt program:

1. Öppna programmet på din iPhone. När den första skärmen med användargränssnitt visas trycker du och håller ned hemknappen (nedanför skärmen). Samtidigt som du håller ned hemknappen trycker du på viloläge-/strömknappen (på ovansidan av enheten). Då tas en skärmavbild som skickas till kamerarullen.
2. Överför bilden till utvecklingsdatorn med iPhoto eller med något annat bildöverföringsprogram. (I Mac OS kan du också använda programmet Image Capture.)

   Du kan också skicka bilden via e-post till utvecklingsdatorn:

   - Öppna bildprogrammet.
   - Öppna kamerarullen.
   - Öppna den skärmavbild som du har tagit.
   - Tryck på bilden och tryck sedan på framåtpilen längst ned till vänster. Klicka sedan på knappen E-posta bild och skicka bilden till dig själv.

***Obs!*** *Du kan skapa vilken grafik du vill i filen Default.png, så länge som grafiken har rätt dimensioner. Men det är oftast bäst att bilden Default.png matchar det inledande tillståndet för programmet.*

Lägg inte in någon text i bilden Default.png om programmet ska översättas till flera språk. Default.png är en statisk bild och texten kan inte anpassas för andra språk.

Se till att du i Flash Professional CS5 lägger till filen Default.png i listan Inkluderade filer i dialogrutan iPhone-inställningar. Läs mer i ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14.

Kom ihåg att referera till den här filen i listan över tillgångar som ska inkluderas när du kompilerar med PFI-programmet i kommandoraden. Se även ”Skapa en installationsfil för ett iPhone-program från kommandoraden” på sidan 20.

# Programinställningar för iPhone

Programinställningarna innehåller:

- Programnamnet
- IPA-filnamnet
- Programversion
- Den inledande skärmorienteringen för programmet och om skärmorienteringen ska roteras automatiskt när iPhone roteras
- Om den inledande bilden ska visas över hela skärmen eller inte
- Information om programmets ikoner
- Information om maskinvaruacceleration

Du kan ändra programinställningarna i Flash Professional CS5.

Du kan även redigera dem i programbeskrivningsfilen. Programbeskrivningsfilen är en XML-fil som innehåller programinställningarna.

## Ställa in iPhone-programegenskaper i Flash Professional CS5

I dialogrutan för iPhone-inställningar i Flash Professional CS5 kan du ange många grundläggande egenskaper för iPhone-programmet.

Så här öppnar du dialogrutan iPhone-inställningar:

❖ Välj Arkiv > iPhone-inställningar.

**Fliken Allmänt**

Fliken Allmänt innehåller följande iPhone-relaterade inställningar:

- Utdatafil – Namnet på programmet som visas under programikonen på iPhone-skärmen. Använd inte ett plustecken (+) i utdatafilsnamnet.
- Programnamn – Namnet på programmet som visas under programikonen på iPhone-skärmen. Använd inte ett plustecken (+) i programnamnet.
- Version – Detta hjälper användarna att avgöra vilken version av ditt program som de har installerat. Versionen som används som CFBundleVersion i iPhone-programmet. Detta måste följa formatet nnnnn[.nn[.nn]] där n är en siffra mellan 0-9 och klammerparenteser anger valfria komponenter, till exempel 1, 1.0 eller 1.0.1. iPhone-versioner får endast innehålla siffror och decimalpunkter. iPhone-versioner kan högst innehålla två decimalpunkter.
- Proportioner – De inledande skärmproportionerna för programmet (stående eller liggande).
- Helskärmsläge – Om programmet använder helskärmsläge eller om statusfältet i iPhone ska visas.
- Automatisk orientering – Markera det här programmet om du vill visa programmets visningsinnehåll i en ny orientering när iPhone-orientering ändras.

  För att få bästa resultat när du använder autoorientering ska du lägga till ActionScript-kod för att ställa in egenskapen `align` på scenen enligt följande:

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

- Återgivning – Hur visningsobjekt ska återges på iPhone-skärmen:
  - CPU – Programmet använder processorn för att återge alla visningsobjekt. Ingen maskinvaruacceleration används.
  - GPU – Programmet använder grafikprocessorn i iPhone-enheten för att komponera bitmappar.
  - Auto – Den här funktionen har inte implementerats.

  Mer information finns i ”Maskinvaruacceleration” på sidan 34.
- Inkluderade filer – Lägg till alla filer och kataloger till paketet i iPhone-programmet. Den viktigaste SWF-filen och programbeskrivningsfilen inkluderas som standard. Lägg till eventuella obligatoriska resurser i listan Inkluderade filer. Lägg till den inledande skärmbildsfilen (Default.png) i listan Inkluderade filer.

**Fliken Distribution**

Fliken Distribution innehåller signerings- och kompileringsinställningar för programmet:

- Digital iPhone-signatur – Ange en P12-certifikatfil och lösenordet för certifikatet. Du måste konvertera iPhone-certifikatet från Apple till .p12-format. Mer information finns i ”Få utvecklarfiler från Apple” på sidan 4.
- Provisioneringsfil – Peka på provisioneringsfilen för programmet, som du fått från Apple. Mer information finns i ”Få utvecklarfiler från Apple” på sidan 4.
- Program-ID – Program-ID används för att identifiera ditt program. Om provisioneringsfilen är knuten till ett specifikt program-ID, ställs värdet i det här fältet in automatiskt med Flash Professional CS5 och du kan inte ändra det. I andra fall tillåter provisioneringsprofiler att du använder flera (jokertecken) program-ID:n. Ange ett program-ID som stämmer överens med wildcardmönstret för det program-ID som du skickade till Apple:
  - Om ditt program-ID från Apple är com.myDomain.*, kommer program-ID:t i dialogrutan iPhone-inställningar att börja med com.myDomain. (till exempel com.myDomain.myApp eller com.myDomain.app22).
  - Om ditt program-ID från Apple är *, kan program-ID:t i dialogrutan iPhone-inställningar endast bestå av en sträng med giltiga tecken.

  Du kan hitta program-ID:t (eller mönster för program-ID med jokertecken) som är associerat med din provisioneringsprofil på iPhone Dev Center (http://developer.apple.com/iphone). Gå till portalen för iPhone-utvecklingsprogram och sedan till avsnittet för provisionering.

  **Viktigt:** Ignorera tecknen framför program-ID:t från Apple. Apple kallar den här strängen för källpaket-ID. Om exempelvis Apple visar ditt program-ID som 96LPVWEASL.com.example.bob.myApp, ska du ignorera 96LPVWEASL och bara använda com.example.bob.myApp som program-ID. Om Apple visar att ditt program-ID är 5RM86Z4DJM.*, ska du ignorera 5RM86Z4DJM eftersom detta är ett program-ID med jokertecken.
- Typ av iPhone-distribution:
  - Snabbpublicering för enhetstestning – Välj det här alternativet om du snabbt vill kompilera en version av programmet för att testa det på din utvecklar-iPhone.
  - Snabbpublicering för enhetsfelsökning – Välj det här alternativet om du snabbt vill kompilera en felsökningsversion av programmet för att testa det på din utvecklar-iPhone. Med det här alternativet kan felsökningsprogrammet i Flash Professional CS5 ta emot `trace()`-data från iPhone-programmet. (Se ”Felsöka ett iPhone-program” på sidan 23.)
  - Distribution – Ad hoc – Välj det här alternativet om du vill skapa ett program för ad hoc-distribution. Se Apples utvecklingscenter för iPhone
  - Distribution – Apple App Store – Välj det här alternativet om du vill skapa en slutgiltig version av IPA-filen för distribution till Apple App Store.

**Fliken Ikoner**

Använd fliken Ikoner för att ange platserna för ikonbilderna med storlekarna 29 x 29 pixlar, 48 x 48 pixlar, 57 x 57 pixlar, 72 x 72 pixlar och 512 x 512 pixlar. Se ”Ikoner och inledande skärmbilder för iPhone” på sidan 12.

***Obs!*** *Alternativ för 48 x 48-pixelbilden och 72 x 72-pixelbilden finns inte i den version av Packager for iPhone Preview som ingår i Flash Professional CS5. Välj Hjälp > Uppdateringar i Flash Professional CS5 för att lägga till de här alternativen.*

## Ställa in iPhone-programegenskaper i programbeskrivningsfilen

Programbeskrivningsfilen är en XML-fil som innehåller egenskaper för hela programmet, till exempel namn, version, copyright och andra inställningar.

I Flash Professional CS5 genereras en programbeskrivningsfil utifrån inställningarna i dialogrutan iPhone-inställningar. Du kan också redigera programbeskrivningsfilen i en textredigerare. I Flash Professional namnges programbeskrivningsfilen genom att ”-app.xml” läggs till projektets namn. Programbeskrivningsfilen för till exempel ett HelloWorld-projekt får namnet HelloWorld-app.xml. Redigera programbeskrivningsfilen om du vill ange inställningar som inte stöds i dialogrutan iPhone-inställningar i Flash Professional CS5. Du kan till exempel definiera `InfoAdditions`-elementet om du vill ange inställningar för info.Plist till programmet.

**Viktigt:** Du ska inte redigera programbeskrivningsfilen medan dialogrutan Flash Professional CS5 är öppen. Spara ändringar i programbeskrivningsfilen innan du öppnar dialogrutan för iPhone-inställningar.

Här är ett exempel på en programbeskrivningsfil:

```
<?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>
```

Här följer mer information om inställningarna i den här programbeskrivningsfilen:

- I elementet `<application>` krävs namnutrymmet AIR 2.0 när du skapar iPhone-program:

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

- Elementet `<id>`:

  **<id>com.example.as3.HelloWorld</id>** Det program-ID som ger programmet en unik identifiering. Det rekommenderade formatet är en punktavgränsad sträng med "omvänd DNS", till exempel `"se.foretag.Programnamn"`. I kompileringen används det här värdet som källpaket-ID för iPhone-programmet.

  Om provisioneringsfilen är knuten till ett specifikt program-ID ska du använda detta program-ID i detta element. Ignorera de tecken som Apple tilldelar i början av program-ID:t (kallas källpaket-ID). Om exempelvis program-ID:t för provisioneringsprofilen är 96LPVWEASL.com.example.bob.myApp, ska du använda com.example.bob.myApp som program-ID i programbeskrivningsfilen.

  Om provisioneringsprofilen tillåter flera (jokertecken) program-ID:n, kommer dess program-ID att avslutas med en asterisk (till exempel 5RM86Z4DJM.*). Ange ett program-ID som stämmer överens med jokerteckenmönstret för det program-ID som du skickade till Apple:

  - Om ditt program-ID från Apple är com.myDomain.*, kommer program-ID i programbeskrivningsfilen att börja med com.myDomain. Du kan ange ett program-ID som till exempel com.myDomain.myApp eller com.myDomain.app22.
  - Om ditt program-ID från Apple är *, kan program-ID i programbeskrivningsfilen bestå av en sträng med giltiga tecken.

  Du kan hitta program-ID:t (eller mönster för program-ID med jokertecken) som är associerat med din provisioneringsprofil på iPhone Dev Center (http://developer.apple.com/iphone). Gå till portalen för iPhone-utvecklingsprogram och sedan till avsnittet för provisionering.

  **Viktigt:** Ignorera tecknen framför program-ID:t från Apple. Apple kallar den här strängen för källpaket-ID. Om exempelvis Apple visar att ditt program-ID är 5RM86Z4DJM.*, ska du ignorera 5RM86Z4DJM eftersom detta är ett program-ID med jokertecken. Om exempelvis Apple visar att ditt program-ID är 96LPVWEASL.com.example.bob.myApp, ska du ignorera 96LPVWEASL och bara använda com.example.bob.myApp som program-ID.

- Elementet `<filename>`:

  **<filename>HelloWorld</filename>** Namnet på iPhone-installationsfilen. Använd inte ett plustecken (+) i filnamnet.

- Elementet `<name>`:

  **<name>Hello World</name>** Namnet på programmet som det visas i iTunes-programmet och på iPhone-skärmen. Använd inte ett plustecken (+) i namnet.

- Elementet `<version>`:

  **<version>1.0</version>** Detta hjälper användarna att avgöra vilken version av ditt program som de installerar. Versionen som används som CFBundleVersion i iPhone-programmet. Detta måste följa formatet nnnnn[.nn[.nn]] där n är en siffra mellan 0-9 och klammerparenteser anger valfria komponenter, till exempel 1, 1.0 eller 1.0.1. iPhone-versioner får endast innehålla siffror och decimalpunkter. iPhone-versioner kan högst innehålla två decimalpunkter.

- Elementet `<initialWindow>` innehåller följande underordnade element som anger egenskaperna för hur programmet visas i inledningsskedet:

  **<content>HelloWorld.swf</content>** Detta identifierar SWF-rotfilen som ska kompileras i iPhone-programmet.

**<visible>true</visible>** Den här inställningen krävs.

**<fullScreen>true</fullScreen>** Det här anger att programmet använder hela iPhone-skärmen.

**<aspectRatio>portrait</aspectRatio>** Det här anger att den inledande skärmbilden för programmet har stående orientering (istället för liggande). Observera att filen Default.png som definierar det inledande fönstret för programmet bör vara 320 pixlar brett och 480 pixlar högt, oavsett inställning. (Se ”Ikoner och inledande skärmbilder för iPhone” på sidan 12.)

**<autoOrients>true</autoOrients>** (Valfritt) Det här anger att innehållets orientering i programmet automatiskt ska omorienteras när enheten ändrar fysisk orientering. Standardvärdet är `true`. Du kan avbryta automatisk orientering genom att anropa metoden `preventDefault()` för en `orientationChanging`-händelse som skickats av scenobjektet. Mer information finns i Ställa in och identifiera skärmorientering.

För att få bästa resultat när du använder autoorientering ska du ställa in egenskapen `align` på scenen enligt följande:

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

**<renderMode>gpu</renderMode>** (Valfritt) Det återgivningsläge som används av programmet. Det finns tre möjliga inställningar:

- `cpu` – Programmet använder processorn för att återge alla visningsobjekt. Ingen maskinvaruacceleration används.
- `gpu` – Programmet använder grafikprocessorn i iPhone-enheten för att komponera bitmappar.
- `auto` – Den här funktionen har inte implementerats.

Mer information finns i ”Maskinvaruacceleration” på sidan 34.

- Elementet `<profiles>`:

  **<profiles>mobileDevice</profiles>** Begränsar programmet till att kompileras i profilen för mobila enheter. Den här profilen stöder för närvarande endast iPhone-program. Det finns tre profiler som stöds:

  - `desktop` – Ett AIR-skrivbordsprogram.
  - `extendedDesktop` – Ett AIR-skrivbordsprogram med stöd för API:t för inbyggda processer.
  - `mobileDevice` – Ett AIR-program för en mobil enhet. För närvarande är iPhone den enda mobila enheten som stöds.

  Genom att begränsa programmet till en särskild profil förhindras det från att kompileras till andra profiler. Om du inte anger någon profil kan du kompilera ett program för vilken profil som helst. Du kan ange flera profiler genom att ange dem var och en, avgränsade med blanksteg, i elementet `<profiles>`.

  Se till att du inkluderar `mobileDevice` bland profilerna som stöds (eller lämna `<profiles>`-elementet tomt).

- Elementet `<icon>` innehåller följande underordnade element som anger de ikoner som används i programmet:

  **<image29x29>icons/icon29.png</image29x29>** Den här bilden används för Spotlight-sökresultat.

  **<image48x48>icons/icon48.png</image48x48>** Den här bilden används för Spotlight-sökresultat på iPad.

  **<image57x57>icons/icon57.png</image57x57>** Den här bilden används i hemskärmen på iPhone och iPod touch.

  **<image72x72>icons/icon72.png</image72x72>** Den här bilden används i hemskärmen på iPad.

  **<image512x512>icons/icon512.png</image512x512>** Den här bilden används i iTunes-programmet.

I verktyget Packager for iPhone används 29-, 57- och 512-ikonerna som refereras i programbeskrivningsfilen. Verktyget kopierar dem till filer med namnen Icon-Small.png, Icon.png respektive iTunesArtwork. Om du vill undvika den här kopieringen kan du paketera filerna direkt. Paketera dem direkt genom att placera dem i katalogen med programbeskrivningsfilen och ange de korrekta namnen och sökvägarna.

Bilden med 512 pixlar är endast till för intern testning. När du skickar ett program till Apple skickar du bilden med 512 pixlar separat. Den ingår inte i IPA-filen. Se till att du kan vara säker på att bilden med 512 pixlar ser bra ut i iTunes innan du skickar den.

- Elementet `<iPhone>` innehåller följande underordnade element som anger iPhone-specifika inställningar:

  **<InfoAdditions></InfoAdditions>** innehåller de underordnade element som anger nyckelvärdepar som används som inställningar för Info.plist för programmet:

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

  I detta exempel anger värdena statusfältets format i programmet och att programmet inte kräver ständig trådlös nätverksåtkomst.

  InfoAdditions-inställningarna anges i en `CDATA`-tagg.

  För iPad-stöd inkluderar du nyckelvärdesinställningar för `UIDeviceFamily`. Inställningen `UIDeviceFamily` är en array med strängar. Varje sträng definierar enheter som stöds. Inställningen `<string>1</string>` definierar stöd för iPhone och iPod touch. Inställningen `<string>2</string>` definierar stöd för iPad. Inställningen `<string>3</string>` definierar stöd för tvOS. Om du bara anger en av dessa strängar stöds bara den enhetsgruppen. Följande sträng begränsar till exempel stödet till iPad:

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

  Följande anger stöd för båda enhetsgrupperna (iPhone/iPod touch och iPad):

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

  Mer information om andra Info.plist-inställningar finns i Apples dokumentation för utvecklare.

# Kompilera en installationsfil för iPhone-program (IPA-fil)

Använd Packager for iPhone för att kompilera ett ActionScript 3.0-program till en IPA-installationsfil.

## Skapa en installationsfil för iPhone-program med Packager for iPhone som ingår i Flash Professional CS5

Så här använder du Packager for iPhone som finns i Flash Professional CS5:

1 Välj Arkiv > Publicera.

2 Kontrollera att du har angett värden för alla inställningar i dialogrutan iPhone-inställningar. Kontrollera att du har valt rätt alternativ på fliken Distribution. Läs mer i ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14.

3 Klicka på knappen Publicera.

Installationsfilen (IPA-filen) för iPhone-programmet genereras av Packager for iPhone. Det kan ta några minuter att kompilera IPA-filen.

Du kan även köra Packager for iPhone från kommandoraden. Se även ”Skapa en installationsfil för ett iPhone-program från kommandoraden” på sidan 20.

## Skapa en installationsfil för ett iPhone-program från kommandoraden

Du kan köra Packager for iPhone från kommandoraden. Packager for iPhone konverterar SWF-filens bytekod och andra källfiler till ett internt iPhone-program.

1 Öppna kommandoskalet eller terminalen och gå till projektmappen för iPhone-programmet.

2 Använd sedan pfi-verktyget för att skapa IPA-filen med följande syntax:

```
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
```

Ändra referensen `pfi` så att hela sökvägen till pfi-programmet finns med. Pfi-programmet installeras i underkatalogen pfi/bin i installationskatalogen för Flash Professional CS5.

Välj det `-target`-alternativ som bäst stämmer överens med den typ av iPhone-program som du vill skapa:

- `-target ipa-test` – Välj det här alternativet om du snabbt vill kompilera en version av programmet för att testa det på din utvecklar-iPhone.
- `-target ipa-debug` – Välj det här alternativet om du vill kompilera en felsökningsversion av programmet för att testa det på din utvecklar-iPhone. Med det här alternativet kan du använda en felsökningssession för att ta emot `trace()`-utdata från iPhone-programmet.

  Du kan inkludera ett av följande `-connect`-alternativ (`CONNECT_OPTIONS`) för att ange IP-adressen till den utvecklingsdator som felsökaren körs på:

  - `-connect` – Programmet försöker ansluta till en felsökningssession på den utvecklingsdator som används för att kompilera programmet.
  - `-connect IP_ADDRESS` – Programmet försöker ansluta till en felsökningssession på datorn med den angivna IP-adressen. Till exempel:

    ```
    -target ipa-debug -connect 192.0.32.10
    ```

  - `-connect HOST_NAME` – Programmet försöker ansluta till en felsökningssession på datorn med det angivna värdnamnet. Till exempel:

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

  ***Obs!*** *Alternativet `-connect` finns inte i Packager for iPhone Preview som ingår i Flash Professional CS5. Uppdatera Packager for iPhone genom att välja Hjälp > Uppdateringar i Flash Professional CS5.*

Alternativet `-connect` är valfritt. Om det inte anges försöker det skapade felsökningsprogrammet inte att ansluta till någon felsökningssession.

Om ett anslutningsförsök till en felsökningssession misslyckas visas en dialogruta i programmet, där användaren uppmanas ange IP-adressen till felsökningsvärden. Ett anslutningsförsök kan misslyckas om enheten inte är ansluten till wifi. Det kan också inträffa om enheten är ansluten, men inte finnas innanför felsökningsvärdens brandvägg.

Du hittar mer information i ”Felsöka ett iPhone-program” på sidan 23.

Du kan även inkludera alternativet `renderingdiagnostics` för att aktivera diagnostikfunktionen för grafikprocessoråtergivning. Mer information finns i ”Felsöka med hjälp av diagnostikfunktionen för grafikprocessoråtergivning” i ”Felsöka ett iPhone-program” på sidan 23.

- `-target ipa-ad-hoc` – Välj det här alternativet om du vill skapa ett program för ad hoc-distribution. Se Apples utvecklingscenter för iPhone
- `-target ipa-app-store` – Välj det här alternativet om du vill skapa en slutgiltig version av IPA-filen för distribution till Apple App Store.

  Ersätt *`PROFILE_PATH`* med sökvägen till filen för provisioneringsprofilen för ditt program. Mer information om provisioneringsprofiler finns i ”Få utvecklarfiler från Apple” på sidan 4.

  Ersätt *`SIGNING_OPTIONS`* med en referens till ditt iPhone-utvecklarcertifikat och lösenord. Använd följande syntax:

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

  Ersätt *P12_FILE_PATH* med sökvägen till P12-certifikatfilen. Ersätt *PASSWORD* med certifikatlösenordet. (Se exemplet nedan.) Mer information om P12-certifikatfilen finns i ”Konvertera ett utvecklarcertifikat till en P12-fil” på sidan 7.

  Ersätt *`APP_DESCRIPTOR`* med en referens till programbeskrivningsfilen.

  Ersätt *`SOURCE_FILES`* med en referens till SWF-huvudfilen för ditt projekt följt av alla eventuella andra resurser som du vill ha med. Inkludera sökvägarna till alla ikonfiler som du definierade i dialogrutan för programinställningarna i Flash CS5 eller i en egen programbeskrivningsfil. Lägg dessutom till den inledande skärmbildsgrafiken med filen Default.png.

Se följande exempel:

```
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"
```

I exemplet används följande för att kompilera filen HelloWorld.ipa:

- Ett särskilt PKCS#12-certifikat med certifikatlösenordet dfb7VKL19
- Programbeskrivningsfilen HelloWorld-app.xml
- En HelloWorld.swf-källfil
- En särskild Default.png och ikonfiler

Pfi-programmet kompilerar programmet med hjälp av programbeskrivningsfilen, SWF-filen och andra resurser till en IPA-fil.

På Mac OS kan du använda ett certifikat som finns lagrat i nyckelkedjan genom att lägga till följande alternativ i pfi-kommandot:

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

Ersätt *ALIAS_NAME* med det alias för det certifikat som du vill använda. När du pekar på ett certifikat som finns lagrat i Mac-nyckelkedjan, anger du alias i stället för att peka på en certifikatfilsplats.

# Installera ett iPhone-program

Om du vill installera utvecklingsprogrammet på en iPhone, lägger du till provisioneringsprofilen på iPhone och installerar sedan programmet på iPhone.

### Lägga till provisioneringsprofilen på iPhone

Så här lägger du till provisioneringsprofilen på iPhone:

1 Välj Arkiv > Lägg till i biblioteket i iTunes. Välj sedan filen med provisioneringsprofilen (med filtypen mobileprovision).

   Kontrollera att din iPhone har lagts till provisioneringsprofilen. Du hanterar provisioneringsprofiler på webbplatsen iPhone Dev Center (http://developer.apple.com/iphone/). Gå till portalen för iPhone-utvecklingsprogram på webbplatsen. Klicka på enhetslänken för att hantera listan med enheter som ditt utvecklingsprogram kan installeras på. Klicka på länken för provisionering för att hantera provisioneringsprofiler.

2 Anslut din iPhone till datorn och synkronisera.

Mer information om hur du erhåller en provisioneringsprofil finns i ”Få utvecklarfiler från Apple” på sidan 4.

### Installera programmet

Du kan installera utvecklingsprogrammet på samma sätt som du installerar andra IPA-filer:

1 Om du tidigare installerat en version av programmet ska du först ta bort det från enheten och från programlistan i iTunes.

2 Lägg till programmet i iTunes på något av följande sätt:

   - Välj (i iTunes) Lägg till i bibliotek på Arkiv-menyn. Markera sedan IPA-filen och klicka på knappen Öppna.
   - Dubbelklicka på IPA-filen.
   - Dra IPA-filen till iTunes-biblioteket.

3 Anslut din iPhone till USB-porten på datorn.

4 Kontrollera programfliken för enheten i iTunes och att programmet är markerat i listan över program som ska installeras.

5 Synkronisera din iPhone.

### Felsökning av installationsprogram

Om ett fel visas i iTunes när du försöker installera programmet ska du kontrollera följande:

- Kontrollera att enhets-ID finns i provisioneringsprofilen.
- Kontrollera att provisioneringsprofilen är installerad genom att dra den till iTunes eller välja Arkiv > Lägg till i bibliotek.

Kontrollera dessutom att programmets program-ID överstämmer med Apples program-ID.

- Om Apples program-ID är com.myDomain.*, måste programbeskrivningsfilens program-ID eller program-ID i dialogrutan iPhone-inställningar börja med com.myDomain (till exempel com.myDomain.anythinghere).

- Om Apples program-ID är com.myDomain.myApp, måste programbeskrivningsfilens program-ID eller Flash Profession CS5-gränssnittet vara com.myDomain.myApp.
- Om Apples program-ID är *, kan programbeskrivningsfilens program-ID eller Flash Profession CS5-gränssnittet vara vad som helst.

Du ställer in programmets program-ID i dialogrutan iPhone-inställningar i Flash Professional CS5 eller i programbeskrivningsfilen.

# Felsöka ett iPhone-program

Du felsöker programmet på utvecklingsdatorn medan programmet körs i ADL. Du kan också felsöka programmet på iPhone.

Viss AIR-funktionalitet som inte stöds i iPhone finns dock tillgänglig för testning av program där ADL används (på utvecklingsdatorn). Tänk på dessa skillnader när du testar innehållet på datorn. Mer information finns i ”ActionScript 3.0-API:er som inte stöds för mobilenheter” på sidan 27.

### Felsökning av program på utvecklingsdatorn

Så här felsöker du programmet på utvecklingsdatorn med Flash Professional CS5:

❖ Välj Felsök > Felsök filmen > I AIR Debug Launcher (mobil).

Du kan också felsöka programmet genom att anropa ADL från kommandoraden. Så här ser syntaxen ut:

```
adl -profile mobileDevice appDescriptorFile
```

Ersätt *`appDescriptorFile`* med sökvägen till programbeskrivningsfilen.

Glöm inte att inkludera alternativet `-profile mobileDevice`.

### Felsöka programmet på en iPhone

Så här felsöker du programmet på en iPhone:

1. Kompilera programmet med felsökningsstöd:
   - I Flash Professional CS5 kompilerar du med inställningen ”Snabbpublicering för enhetsfelsökning”. (Läs mer i ”Skapa en installationsfil för iPhone-program med Packager for iPhone som ingår i Flash Professional CS5” på sidan 20.)
   - Använd PFI-programmet på kommandoraden för att kompilera programmet med alternativet `target ipa-debug`. (Se även ”Skapa en installationsfil för ett iPhone-program från kommandoraden” på sidan 20.)
2. Installera programmet på en iPhone.
3. På iPhone aktiverar du nätverksanslutningen och ansluter till samma nätverk som det där utvecklingsdatorn finns.
4. Starta en felsökningssession på utvecklingsdatorn. Välj Felsök > Starta fjärrfelsökning > ActionScript 3.0 i Flash Professional CS5.
5. Kör programmet på iPhone-enheten.

   I felsökningsversionen av programmet anger du IP-adressen för utvecklingsdatorn när du blir ombedd att göra det. Ange IP-adressen och peka på knappen OK. Så här tar du reda på IP-adressen för utvecklingsdatorn:
   - I Mac OS väljer du Systeminställningar i Apple-menyn. Klicka på nätverksikonen i fönstret Systeminställningar. I fönstret med nätverksinställningar visas IP-adressen.

- I Windows startar du en kommandosession och kör kommandot `ipconfig`.

I felsökningssession visas eventuella `trace()`-utdata från programmet.

I Flash Professional CS5 finns alla felsökningsfunktioner för felsökning av program som är installerade på iPhone, med bland annat brytpunktskontroll, stegning igenom kod och variabelövervakning.

**Felsöka med hjälp av diagnostikfunktionen för grafikprocessoråtergivning**

Med diagnostikfunktionen för grafikprocessoråtergivning kan du se hur programmet använder maskinvaruacceleration (för program som använder grafikprocessoråtergivning). Om du vill använda den här funktionen kompilerar du programmet med PFI-verktyget i kommandoraden och inkluderar alternativet `-renderingdiagnostics`:

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

Flaggan `-renderingdiagnostics` måste komma direkt efter flaggan `-package`.

Diagnostikfunktionen för grafikprocessoråtergivning visar färgade rektanglar för alla visningsobjekt:

- Blått – Visningsobjektet är inte en bitmapp och är inte cachat som en bitmapp, och det håller på att återges.

  Om blått visas upprepade gånger för ett visningsobjekt som inte ändras kan det beror på att det överlappar visningsobjekt som rör sig. Visningsobjektet kan till exempel vara en bakgrund för visningsobjekt som rör sig. Det kan då vara en bra idé att cacha visningsobjektet som en bitmapp.

  Om blått visas för ett objekt som du anser ska cachas kan det bero på att objektet använder en effekt som grafikprocessorn inte kan hantera. Bland de effekterna finns vissa blandningslägen, färgomvandlingar, egenskapen `scrollRect` samt masker.

  Programmet visar även blått om visningsobjekt som överförs till grafikprocessorn överskrider minnesgränserna.

  Det loggas ett meddelande för varje blå rektangel. De här meddelandena matas ut tillsammans med övriga `trace()`- och felsökningsmeddelanden.

- Grönt – Visningsobjektet är en bitmapp eller är cachat som en bitmapp, och det håller på att överföras till grafikprocessorn för första gången.

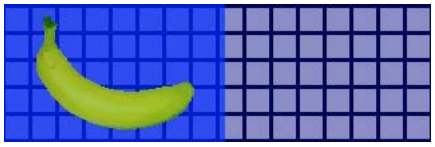
  Om grönt visas upprepade gånger för ett visningsobjekt återskapar koden i programmet visningsobjektet. Detta kan till exempel hända om tidslinjen återgår till en bildruta som skapar visningsobjektet. I så fall kan det vara en bra idé att ändra innehållet, så att identiska objekt inte återskapas.

- Rött – Visningsobjektet är en bitmapp eller är cachat som en bitmapp, och det håller på att överföras till grafikprocessorn igen.

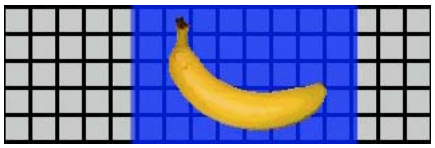
  Rött visas varje gång ett sådant visningsobjekt ändras på ett sätt som innebär att programmet måste återge bitmappprepresentationen igen. Ett 2D-objekt som inte har egenskapen `cacheAsBitmapMatrix` angiven återges till exempel igen när det skapas eller roteras. Dessutom sker en ny återgivning även när underordnade visningsobjekt flyttas eller ändras.

Varje färgad rektangel försvinner efter fyra skärmomritningscykler, förutsatt att orsaken till färgläggningen inte inträffar igen under de cyklerna. Om det inte finns några förändringar på skärmen ändras färgläggningen inte.

Som exempel kan vi ta ett bitmappsvisningsobjekt (en banan) framför en vektorbakgrund som inte cachas som en bitmapp. När bananen först återges blir den grön. När bakgrunden först återges blir den blå:

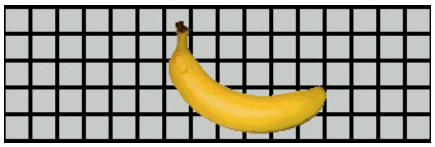


När bananen rör sig måste processorn återge bakgrunden igen, vilket medför att bakgrunden får en blå skuggning:



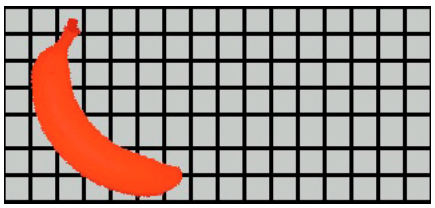
Den blå skuggningen över bakgrunden motsvarar omritade områden som måste skickas till grafikprocessorn.

Om bakgrunden däremot cachas som en bitmapp visar diagnostikfunktionen för grafikprocessoråtergivning ingen färgning när bananen rör sig:



Diagnostikfunktionen visar inga färgtoner eftersom grafikprocessorn har bakgrundsbitmappen. Grafikprocessorn kan skapa bananen med bakgrund utan processorn.

Anta nu att bananen är ett 2D-visningsobjekt som inte har egenskapen `cacheAsBitmapMatrix` angiven. Så snart visningsobjektet roteras (eller skalas) visar diagnostikfunktionen för återgivning rött. Detta anger att programmet måste överföra en ny version av visningsobjektet till grafikprocessorn:



# Skicka iPhone-programmet till App Store

Så här skickar du programmet till App Store:

1 Hämta ett distributionscertifikat och en provisioneringsprofil från webbplatsen iPhone Dev Center (http://developer.apple.com/iphone/).

   Ett distributionscertifikat har namnet "iPhone Developer: XXX.cer", där XXX är ditt namn.

   Mer information finns i ”Få utvecklarfiler från Apple” på sidan 4.

2 Konvertera distributionscertifikatet till en P12-fil.

Mer information finns i ”Konvertera ett utvecklarcertifikat till en P12-fil” på sidan 7.

3 Kompilera programmet med hjälp av P12-filen och provisioneringsprofilen.

Använd P12-filen som du skapade utifrån distributionscertifikatet. Använd det program-ID som är kopplat till provisioneringsprofilen för distributionen.

Mer information finns i ”Kompilera en installationsfil för iPhone-program (IPA-fil)” på sidan 19.

4 Skicka programmet till webbplatsen iPhone Dev Center (http://developer.apple.com/iphone/).

***Viktigt!*** *Apple kräver att du använder Apple-programmet Application Loader för att överföra program till deras App Store. Apple publicerar bara Application Loader för Mac OS X. Även om du kan utveckla AIR-program för iPhone med en Windows-dator måste du ha tillgång till en Mac med OS X (version 10.5.3 eller senare) för att kunna skicka programmet till App Store. Du kan hämta Application Loader från Apple iOS Dev Center.*

# Kapitel 3: ActionScript 3.0-API:er som stöds för mobilenheter

När du skapar AIR-program för mobilenheter kan du använda samma ActionScript 3.0-API:er som finns för utveckling av skrivbordsprogram för Adobe Flash Player 10.1 och AIR 2. Det finns dock vissa undantag och tillägg.

## ActionScript 3.0-API:er som inte stöds för mobilenheter

Vissa ActionScript 3.0-API:er stöds inte för program som körs i mobilenhetens profil (till exempel program som körs på en iPhone).

När du använder samma ActionScript-kod för att utveckla till flera olika profiler (till exempel för datorer och mobila enheter) kan du använda kod för att testa om en API stöds eller inte. Klassen NativeWindow stöds till exempel inte i iPhone-program. (iPhone-program kan inte använda eller skapa inbyggda fönster.) Om du vill testa om ett program körs på en profil som stöder inbyggda fönster (till exempel för en dator), kontrollerar du egenskapen `NativeWindow.isSupported`.

Följande lista visar vilka API:er som inte stöds i mobilenhetens profil. I tabellen visas också egenskaper som du kan kontrollera för att avgöra om ett program körs på en plattform med stöd för ett API.

| **API** | **Egenskap att testa** |
|---|---|
| Accessibility | Capabilities.hasAccessibility |
| Camera | Camera.isSupported |
| DatagramSocket | DatagramSocket.isSupported |
| DNSResolver | DNSResolver.isSupported |
| DockIcon | NativeApplication.supportsDockIcon |
| DRMManager | DRMManager.isSupported |
| EncryptedLocalStore | EncryptedLocalStore.isSupported |
| HTMLLoader | HTMLLoader.isSupported |
| LocalConnection | LocalConnection.isSupported |
| Microphone | Microphone.isSupported |
| NativeApplication.exit() | — |
| NativeApplication.menu | NativeApplication.supportsMenu |
| NativeApplication.isSetAsDefaultApplication() | NativeApplication.supportsDefaultApplication |
| NativeApplication.startAtLogin | NativeApplication.supportsStartAtLogin |
| NativeMenu | NativeMenu.isSupported |
| NativeProcess | NativeProcess.isSupported |
| NativeWindow | NativeWindow.isSupported |

| API | Egenskap att testa |
|---|---|
| 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 |

Du kan inte skriva HTML- eller JavaScript-baserade AIR-program för mobilenhetens profil.

Vissa ActionScript 3.0-klasser stöds endast delvis:

**Fil**
iPhone-program har endast åtkomst till programkatalogen och lagringskatalogen för programmet. Du kan också anropa `File.createTempFile()` och `File.createTempDirectory()`. Om du anropar en åtgärd för att få åtkomst till en annan katalog (till exempel en läs- eller skrivmetod för FileStream) inträffar ett IOError-undantag.

I iPhone-program stöds inte inbyggda dialogrutor för filbläddring, till exempel en sådan som fås genom metoden `File.browseForOpen()`.

**Loader**
I ett iPhone-program kan du inte använda metoden `Loader.load()`. Med metoden `Loader.load()` kan du emellertid inte köra ActionScript-kod i SWF-innehåll. Du kan använda resurser i SWF-filen (till exempel filmklipp, bilder, teckensnitt och ljud i biblioteket). Du kan även använda metoden `Loader.load()` för att läsa in bildfiler.

**Video**
Endast Sorensen-video och ON2 VP6-video stöds i AIR-program på iPhone.

Du kan använda metoden `navigateToURL()` för att öppna en H.264-video utanför programmet. Som `request`-parameter skickar du ett URLRequest-objekt med en URL som pekar till videon. Videon öppnas i videospelaren på iPhone-enheten.

**Textfält**
Det finns vissa begränsningar för teckensnitt och andra textfältsinställningar för iPhone. Se ”Teckensnitt och textinmatning” på sidan 37.

**API:er som inte stöds och felsökning med ADL**
Viss AIR-funktionalitet som inte stöds i iPhone finns dock tillgänglig för testning av program där ADL används (på utvecklingsdatorn). Var uppmärksam på dessa skillnader när du testar innehåll med ADL.

Denna funktionalitet innehåller följande program för video- och ljudavkodning: Speex (ljud), H.264/AVC (video) och AAC (ljud). Dessa avkodningsprogram är inte tillgängliga för AIR-program som körs på iPhone. De kommer emellertid att fungera normalt på en dator.

Stöd för hjälpmedel och skärmläsare fungerar i ADL på Windows. Dessa API:er stöds emellertid inte på iPhone.

RTMPE-protokollet fungerar normalt när det används från ADL på datorn. Däremot fungerar inte ett NetConnection som försöker ansluta till iPhone med RTMPE-protokollet.

Klassen Loader fungerar utan ytterligare begränsningar när innehållet körs med ADL. När den körs på iPhone kommer försök att läsa in SWF-innehåll med ActionScript-bytekod att resultera i ett felmeddelande.

Shader-instanser körs i ADL. Pixel Bender-bytekod på iPhone kan inte tolkas och skuggningar har ingen grafisk effekt.

Mer information finns i ”Felsöka ett iPhone-program” på sidan 23.

# ActionScript-API:er för AIR-program för mobilenheter

Följande API:er är bara tillgängliga i AIR-program på mobilenheter. De fungerar för tillfället inte i Flash Player eller skrivbordsversioner av AIR.

**API för skärmorientering**
Med API:t för skärmorientering kan du arbeta med orienteringen för scenen och för iPhone-enheten:

- `Stage.autoOrients` – Anger om programmet är inställt på att scenen automatiskt ska ändra orientering när enheten roteras. Den här egenskapen anges till `true` om alternativet Automatisk orientering markeras i dialogrutan iPhone-inställningar i Flash Professional CS5. (Du kan också ange elementet `autoOrients` till `true` i programbeskrivningsfilen.) Se ”Programinställningar för iPhone” på sidan 14. Du kan avbryta automatisk omorientering genom att lägga till en `orientationChanging`-händelseavlyssnare för stage-objektet. Automatisk omorientering avbryts om metoden `preventDefault()` anropas för händelseobjektet.

  För att få bästa resultat när du använder autoorientering ska du ställa in egenskapen `align` på scenen enligt följande:

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

- `Stage.deviceOrientation` – Enhetens fysiska orientering. Värden för den här egenskapen definieras av klassen StageOrientation.
- `Stage.orientation` – Scenens aktuella orientering. Värden för den här egenskapen definieras av klassen StageOrientation.
- `Stage.supportsOrientationChange` – Ange till `true` på iPhone-enheten och till `false` i ett AIR-program.
- `Stage.setOrientation()` – Anger scenens orientering. Den här metoden har en parameter, som är en sträng som definierar den nya scenens orientering. Konstanterna i StageOrientation-klassen definierar tänkbara värden för parametern.
- StageOrientation – Anger värden för scenens orientering. `StageOrientation.ROTATED_RIGHT` indikerar exempelvis en scen som är roterad åt höger relativt till standardorienteringen för enheten.
- StageOrientationEvent – Anger händelser som skickas av scenen när orienteringen av scenen ändras. Den här händelsen uppstår när användaren vrider på iPhone-enheten. Det finns två typer av händelser. Scenen skickar händelsen `orientationChanging` när enhetens fysiska orientering ändras. Om du vill undvika att scenen ändrar orientering anropar du metoden `preventDefault()` för händelseobjektet `orientationChanging`. Scenen skickar händelsen `orientationChange` när ändringen av scenens orientering har slutförts.

API:t för skärmorientering är för tillfället bara användbart i AIR-program på mobilenheter. Om ett mobilt AIR-program och ett AIR-program för stationära datorer delar samma källkod använder du egenskapen `Stage.supportsOrientationChange` för att kontrollera om API:t stöds.

I följande exempel visas vad som ska hända när användaren roterar enheten:

```
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;
}
```

I det här exemplet används kommentarer istället för kod för ändrad scenorientering.

Du ändrar orientering på scenen genom att anropa metoden `setOrientation()` för Stage-objektet. Att ställa in orienteringen är en asynkron åtgärd. Du kontrollerar när orienteringen är klar genom att slutföra avlyssningen av händelsen `orientationChange`. Med följande kod visas hur du ställer in scenen för högerhänt orientering:

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

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

När scenen roterar ändras dess storlek och Stage-objektet skickar en `resize`-händelse. Du kan ändra storlek på visningsobjekt, och ändra deras positioner på scenen, som svar på händelsen `resize`.

#### NativeApplication.systemIdleMode och SystemIdleMode

Egenskapen `NativeApplication.systemIdleMode` använder du för att förhindra att en iPhone försätts i vänteläge. Standard är att en iPhone försätts i vänteläge när inte någonting vidrör skärmen under en tidsperiod. Vänteläget kan resultera i att skärmen tonas ned. Det kan också leda till att iPhone kommer att låsa sig. Den här egenskapen kan ha ett av två möjliga värden:

- `SystemIdleMode.NORMAL` – Det normala vänteläget används för iPhone.
- `SystemIdleMode.KEEP_AWAKE` – Programmet försöker hindra att iPhone försätts i vänteläge.

Den här funktionen stöds bara på mobilenheter. Den stöds inte i AIR-program som körs i operativsystemet på en stationär dator. I ett program som körs på en stationär dator har egenskapen `NativeApplication.systemIdleMode` ingen effekt.

I följande kod visas hur vänteläget för iPhone inaktiveras:

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

### CameraRoll

Med klassen CameraRoll kan du lägga till en bild i iPhones filmrulle. Metoden `addBitmapData()` använder du för att lägga till en bild i iPhones filmrulle. Metoden har en parameter, `bitmapData`. Den här parametern är BitmapData-objektet som innehåller bilden som ska läggas in i filmrullen.

CameraRoll-funktionen stöds bara på mobilenheter. Den stöds inte i AIR-program som körs i operativsystemet på en stationär dator. Om du under körningen vill kontrollera att programmet har stöd för CameraRoll-funktionen ska du kontrollera den statiska egenskapen `CameraRoll.supportsAddBitmapData`.

Sedan du anropat metoden `addBitmapData()` skickas en av dessa båda händelser från CameraRoll-objektet:

- `complete` – Operationen slutfördes på ett korrekt sätt.
- `error`– Det har uppstått ett fel. Det kanske inte finns tillräckligt med utrymme på iPhone för att kunna lagra bilden.

Med följande kod läggs en bild av scenen (en skärmdump) in i filmrullen:

```
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

Egenskapen `cacheAsBitmapMatrix` är ett Matrix-objekt som definierar hur ett visningsobjekt ska återges när `cacheAsBitmap` är `true`. I programmet används den här matrisen som en omformningsmatris när bitmappsversionen av visningsobjektet återges.

När `cacheAsBitmapMatrix` är inställt kommer programmet att behålla en cache-lagrad bitmappsbild som är återgiven med matrisen, i stället för med visningsmatrisen. (Visningsmatrisen har värdet för `transform.concatenatedMatrix` i visningsobjektet.) Om matrisen inte överensstämmer med visningsmatrisen, kommer bitmappen att skalförändras och roteras om det skulle behövas.

Ett visningsobjekt där `cacheAsBitmapMatrix` är inställt kommer endast att återges när värdet för `cacheAsBitmapMatrix` ändras. Bitmappen skalförändras eller roteras för att anpassas mot visningsmatrisen.

Både processor- och grafikprocessoråtergivning tjänar på att egenskapen `cacheAsBitmapMatrix` används, men vanligtvis är fördelarna för grafikprocessoråtergivningen större.

***Obs!*** *Om du vill använda maskinvaruacceleration ska du ställa in Återgivning till GPU på fliken Allmänt i dialogrutan iPhone-inställningar i Flash Professional CS5. (Du kan också ange att egenskapen* `renderMode` *ska vara* `gpu` *i programbeskrivningsfilen.)*

I följande exempelkod används en ej omformad bitmappsrepresentation av visningsobjektet:

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

I följande kod används en bitmappsrepresentation som matchar den aktuella återgivningen:

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

Vanligtvis är det identitetsmatrisen (`new Matrix()`) eller `transform.concatenatedMatrix` som används. Du kan emellertid använda en annan matris, till exempel en nedskalad matris, för att ladda upp en annan bitmapp till grafikprocessorn. I följande exempel används en `cacheAsBitmapMatrix`-matris som är skalförändrad med 0,5 på x- och y-axlarna. Bitmappsobjektet som används i grafikprocessorn är mindre, emellertid ändras storleken i grafikprocessorn så att det matchar egenskapen `transform.matrix` i visningsobjekt:

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

Generellt gäller att du bör välja en matris som omformar visningsobjektet till den storlek så som det kommer att visas i programmet. Om till exempel programmet visar bimappsversionen av sprite nedskalad till hälften, ska du använda en matris som skalas ned till hälften. Om sprite visas större än de aktuella dimensionerna i programmet, ska du använda en matris som skalas upp med samma faktor.

Det finns en praktisk begränsning för storleken på visningsobjekt för vilken egenskapen `cacheAsBitmapMatrix` är inställd. Begränsningen är 1020 gånger 1020 pixlar. Det finns även en praktisk begränsning för antalet pixlar för alla visningsobjekt som har egenskapen `cacheAsBitmapMatrix` inställd. Begränsningen är fyra miljoner pixlar.

Det finns mycket att tänka på när du använder `cacheAsBitmapMatrix` och maskinvaruacceleration. Det är viktigt att veta vilka visningsobjekt som ska ha egenskapen inställd och vilka som inte ska ha den. Viktig information om hur du använder den här egenskapen finns i ”Maskinvaruacceleration” på sidan 34.

Du kan använda diagnostikfunktionen för grafikprocessoråtergivning för att analysera grafikprocessoranvändningen i felsökningsversioner av ditt program. Mer information finns i ”Felsöka ett iPhone-program” på sidan 23.

**Nätverkskommentarer**

Om du använder följande URL-schema med funktionen `nativigateToURL()` kommer ett dokument att öppnas i ett externt program:

| URL-schema | Resultat av anrop till nativeToURL() | Exempel |
|---|---|---|
| mailto: | Öppnar ett nytt meddelande i e-postprogrammet. | `str = "mailto:test@example.com";`<br>`var urlReq:URLReq = new URLRequest(str);`<br>`navigateToURL(urlReq);` |
| sms: | Öppnar ett meddelande i textmeddelandeprogrammet. | `str = "sms:1-415-555-1212";`<br>`var urlReq:URLReq = new URLRequest(str);`<br>`navigateToURL(urlReq);` |
| tel: | Ringer upp ett telefonnummer med telefonen (om användaren godkänner det). | `str = "tel:1-415-555-1212";`<br>`var urlReq:URLReq = new URLRequest(str);`<br>`navigateToURL(urlReq);` |

Ett iPhone-program kan förlita sig på installerade självsignerade rotcertifikat för serverautentisering under en säker transaktion, till exempel vid en https-förfrågan. En server ska inte bara skicka bladcertifikatet utan även alla mellanliggande certifikatskedjor till rotcertifikatet.

# ActionScript 3.0-API:er av särskilt intresse för mobilprogramutvecklare

Följande ActionScript 3.0-API:er definierar funktioner som är användbara på mobilenheter.

### Accelerometer API

Följande klasser låter programmet få händelser från enhetens accelerometer:

- Accelerometer
- AccelerometerEvent

Mer information finns i Indata från accelerometer.

### Geolocation API

Följande klasser låter programmet få händelser från enhetens positionssensor:

- Geolocation
- GeolocationEvent

Mer information finns i Geolocation.

### Pekhändelser och gest-API:er

Följande klasser låter programmet ta emot pekhändelser och pekrörelser:

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

Mer information finns i Pekhändelser och gester.

# Kapitel 4: Om utformning av iPhone-program

Bearbetningshastigheten och skärmstorleken på en iPhone ger särskilda utformnings- och kodningssituationer. Många utformningssituationer är emellertid vanliga vid all programutveckling eller utveckling av program för mobila enheter.

Mer information om programoptimering finns i Optimera mobilt innehåll för Flash-plattformen. Detta dokument innehåller många förslag på hur du kan optimera prestandan för mobilt innehåll, Flash Player-innehåll, AIR-innehåll och ActionScript-baserat innehåll i allmänhet. De flesta av dessa förslag gäller också AIR-program som utvecklats för iPhone.

**Viktigt:** Många av dessa designöverväganden och optimeringstekniker är viktiga att ta hänsyn till vid utveckling av program som ska fungera bra på en iPhone.

## Maskinvaruacceleration

Du kan använda OpenGL ES 1.1-maskinvaruacceleration om du vill förbättra grafiska prestanda i vissa program. För spel och andra program med animerade visningsobjekt kan maskinvaruacceleration vara en fördel. För program som använder maskinvaruacceleration avlastas processorn från den grafiska bearbetningen som i stället utförs av grafikprocessorn i iPhone-enheten, vilket kan förbättra prestandan betydligt.

När du utformar ett program för att använda grafikprocessorn är det viktigt att du följer regler som garanterar att innehållet accelereras effektivt.

Om du vill använda maskinvaruacceleration ska du ställa in Återgivning till GPU på fliken Allmänt i dialogrutan iPhone-inställningar i Flash Professional CS5. Du kan i programbeskrivningsfilen också ange att egenskapen`renderMode` ska vara `gpu`.

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

Se även ”Ställa in iPhone-programegenskaper i Flash Professional CS5” på sidan 14 och ”Ställa in iPhone-programegenskaper i programbeskrivningsfilen” på sidan 16.

Det finns fyra klasser med visningsobjekt som kan återges snabbt med maskinvaruacceleration om deras innehåll ändras sällan:

- Bitmappsobjekt
- 2D-visningsobjekt vars `cacheAsBitmap`-egenskap är inställd på `true` och som kan ha egenskapen `cacheAsBitmapMatrix` angiven (se nedan)
- 3D-visningsobjekt (d.v.s. objekt som har egenskapen `z` angiven)
- Visningsobjekt som har en heltäckande fyllning och har kanter som är justerade mot pixlarna på skärmen.

Vektorbaserade objekt återges när en annan sprite animeras över eller under dem. Detta innebär att även andra objekt i bak- eller förgrunden till en animering ska tillhöra en av dessa kategorier.

För visningsobjekt där `cacheAsBitmap` är `true`, kommer en inställning av `cacheAsBitmapMatrix` att medföra att grafikprocessorn kommer att använda bitmappen som är ett resultat av omformningsmatrisen. I grafikprocessorn används bitmappsrepresentationen även om objektet roteras eller skalförändras. Grafikprocessorn kan sammanställa och animera den här bitmappen snabbare än vad processorn kan rita om ett vektoråtergivet objekt.

Om du bara anger `cacheAsBitmap` som `true` medför det att visningsobjektet (och eventuella underordnade objekt) cachas. Visningsobjektet ritas inte om när nya områden visas eller hela den kombinerade bilden översätts.

När ett visningsobjekts `cacheAsBitmapMatrix`-egenskap är inställd, kan en representation av visningsobjektet skapas i programmet även om objektet inte visas. I programmet skapas en cache-lagrad representation av visningsobjektet när nästa bildruta startas. När du sedan lägger till visningsobjektet på scenen kommer det att återges snabbt i programmet. Dessutom kan objektet snabbt animeras, roteras och skalförändras i programmet. För objekt som roteras eller skalförändras ska du inte ställa in egenskapen `cacheAsBitmap` utan att samtidigt ställa in egenskapen `cacheAsBitmapMatrix`.

Programmet kan även utföra alfaomformningar snabbt på ett objekt som cachas som en bitmapp. Det är emellertid bara `alpha`-värden mellan 0 och 1,0 som stöds för maskinvaruaccelererade alfaomformningar. Det motsvarar en inställning på mellan 0 och 256 för `colorTransform.alphaMultiplier`.

Ange inte egenskapen `cacheAsBitmap` som `true` för objekt som uppdateras ofta, till exempel textfält.

Visningsobjekt vars bildinnehåll ändras ofta passar oftast dåligt för grafikprocessoråtergivning. Detta gäller särskilt äldre enheter med mindre kraftfulla grafikprocessorer. Eftersom det krävs avsevärd processorkapacitet för att överföra grafiken till grafikprocessorn kan CPU-återgivning vara ett bättre val.

Strukturera om visningsobjekt som innehåller underordnade visningsobjekt, som rör sig i förhållande till det överordnade objektet. Ändra dem så att de underordnade visningsobjekten blir jämställda med de överordnade. På så sätt får vart och ett av dem en egen bitmapprepresentation. Varje visningsobjekt kan dessutom röra sig i förhållande till de andra, utan att nya bilder måste överföras till grafikprocessorn.

Ställ in `true` för egenskapen `cacheAsBitmap` på den högsta nivån i visningslistan där underordnade visningsobjekt inte animeras. Med andra ord, ange den för visningsobjektbehållare som inte innehåller några rörliga delar. Ange den inte för de underordnade visningsobjekten. Gör det *inte* för ett sprite som innehåller andra visningsobjekt som animeras.

När du ställer in `z`-egenskapen för ett visningsobjekt, kommer en cache-lagrad bitmappsrepresentation alltid att användas i programmet. Dessutom kommer, sedan du ställt in `z`-egenskapen för ett visningsobjekt, programmet att använda den cache-lagrade bitmappsrepresentationen även om du roterar eller skalförändrar objektet. I programmet används inte egenskapen `cacheAsBitmapMatrix` för visningsobjekt med en inställd `z`-egenskap. Samma regler kommer att gälla när du ställer in tredimensionella egenskaper för visningsobjekt, inklusive egenskaperna `rotationX`, `rotationY`, `rotationZ` och `transform.matrix3D`.

Ställ inte in egenskaperna `scrollRect` eller `mask` för en visningsobjektsbehållare som har innehåll som du vill använda för maskinvaruacceleration. När du ställer in dessa egenskaper inaktiveras maskinvaruacceleration för visningsobjektsbehållaren och dess underordnade objekt. Ett alternativ till att ange egenskapen `mask` är att skapa ett lager med ett maskvisningsobjekt ovanpå det visningsobjekt som ska maskeras.

Det finns gränser för storleken på de visningsobjekt som är tillgängliga för maskinvaruacceleration. På äldre enheter är gränsen högst 1 024 pixlar för både bredd och höjd. På nyare enheter är gränsen högst 2 048 pixlar. Du kan använda diagnostikverktyget för grafikprocessoråtergivning för att testa prestanda på en enhet.

Grafikprocessorn kan även använda RAM-minnet i iPhone för att lagra bitmappsbilder. Då används minst den mängd minne som behövs för bitmappsbilderna.

Minnesallokeringen för grafikprocessorn är potenser av två för varje dimension av bitmappsbilden. Grafikprocessorn kan till exempel reservera minne i storlekarna 512 x 1024 eller 8 x 32. Detta innebär att en bild på 9 x 15 pixlar tar lika mycket minne i anspråk som en bild på 16 x 16 pixlar. För cachade visningsobjekt kan du använda dimensioner som ligger närmre potenser av två (men inte mer) för varje riktning. Det är till exempel mer effektivt att använda ett visningsobjekt som är 32 x 16 pixlar än ett som är 33 x 17 pixlar.

Förlita dig inte på en storleksförändrad scen för att förminska resurser som har storleksanpassats för andra plattformar (till exempel datorn). Du ska i stället använda egenskapen `cacheAsBitmapMatrix` eller ändra storlek på resursen innan den publiceras för iPhone. 3D-objekt ignoreras i egenskapen `cacheAsBitmapMatrix` när en ytbild cache-lagras. Det är av den anledningen bättre att ändra storleken på visningsobjekt före publiceringen om de ska återges på en 3D-yta.

Det finns anledning att vara observant när det gäller maskinvaruacceleration och RAM-användning. Medan minnet fylls upp kommer operativsystemet i iPhone att frigöra minne från andra iPhone-program som körs. När bearbetningen pågår i dessa program och minne frigörs, kan en tävlan med ditt program om CPU-resurser äga rum. Detta kan tillfälligt minska prestandan i ditt program. Du bör därför testa dina program på äldre enheter eftersom de har betydligt mindre tillgängligt minne för den process som körs.

När du felsöker ett program på iPhone kan du aktivera diagnostikfunktionen för grafikprocessoråtergivning. Med den här funktionen kan du enklare se hur programmet använder grafikprocessoråtergivning. Mer information finns i ”Felsöka med hjälp av diagnostikfunktionen för grafikprocessoråtergivning” i ”Felsöka ett iPhone-program” på sidan 23.

Mer information om hur du använder egenskapen `cacheAsBitmapMatrix`finns i avsnittet "DisplayObject.cacheAsBitmapMatrix” i ”ActionScript-API:er för AIR-program för mobilenheter” på sidan 29.

# Andra sätt att förbättra prestanda för visningsobjekt

Maskinvaruacceleration kan ge snabbare grafiska prestanda för vissa klasser av visningsobjekt. Här följer några tips på hur du kan maximera grafiska prestanda:

- Försök att begränsa antalet objekt som visas på scenen. Det tar tid att återge ett objekt och kombinera det med andra objekt som placerats intill det.

  När du inte längre behöver visa ett visningsobjekt anger du egenskapen `visible` för objektet till `false` eller tar bort det från scenen (`removeChild()`). Det räcker inte med att bara ange egenskapen `alpha` för objektet till 0.
- Undvik blandningslägen i allmänhet och lagerblandningslägen i synnerhet. Använd normalt blandningsläge när det är möjligt.
- Visningsobjektfilter är datormässigt kostsamma. Använd dem med måtta. Du kan till exempel använda ett par filter på en introduktionsskärm om du vill. Men undvik att använda filter på många objekt eller på objekt som är animerade, eller när du måste använda en högre bildrutehastighet.
- Undvik övergångsformer.
- Undvik att använda klippning.
- Om det är möjligt anger du parametern `repeat` till `false` när du anropar metoden `Graphic.beginBitmapFill()`.
- Undvik att rita för mycket. Använd bakgrundsfärgen som bakgrund. Använd inte stora former i olika lager ovanpå varandra. Varje pixel som måste ritas utgör en belastning. Detta gäller särskilt visningsobjekt som inte är maskinvaruaccelererade.

- Undvik former med långa, tunna spetsar, kanter som korsar sig själva eller många detaljer vid kanterna. Dessa former tar längre tid att återge än visningsobjekt med jämna kanter. Detta gäller särskilt visningsobjekt som inte är maskinvaruaccelererade.
- Gör bitmappar som är ungefär $2^n$ gånger $2^m$ bitar stora, men inte större. Dimensionerna måste inte vara av tvåpotens, men bör vara i närheten av tvåpotens, men inte större. En bild med 31 x 15 pixlar återges till exempel snabbare än en bild med 33 x 17 pixlar. (31 och 15 är precis under tvåpotenserna 2: 32 och 16.) Sådana bilder använder även minne mer effektivt.
- Begränsa storleken på visningsobjekt till 1 024 x 1 024 pixlar (eller 2 048 x 2 048 på nyare enheter).

# Informationstäthet

Den fysiska storleken på skärmen på mobila enheter är mindre än en datorskärm, men pixeltätheten är högre. En skarpare text ser bättre ut, men glyferna måste ha en minimal fysisk storlek om de ska vara läsliga.

Mobila enheter används ofta i rörelse och under svaga ljusförhållanden. Fundera över hur mycket information som det är realistiskt att visa på skärmen om den ska vara läsbar. Det kan vara mindre än det som visas på en datorskärm med samma pixelmått.

Använd en typografisk hierarki för att framhäva den viktiga informationen. Använd teckenstorlekar, vikt, placering och avstånd för att uttrycka hierarkin för de olika elementen i användargränssnittet. Du kan använda en eller flera referenser på varje nivå i hierarkin. Använd dessa konsekvent i hela programmet. Ett sätt kan vara med hjälp av utrymme (indrag, radavstånd, placering), ett annat kan vara grafiskt (storlek, form, färg på typsnitt). Använd gärna flera olika sätt samtidigt för att se till att hierarkin är klar och tydlig. Men försök att inte använda fler än tre olika sätt för varje grupperingsnivå.

Försök att förenkla etiketter och nödvändig förklaringstext. Du kan till exempel använda exempeltext i textfält för att exemplifiera det avsedda innehållet så att du slipper använda separata etiketter.

# Teckensnitt och textinmatning

För bästa resultat bör du använda enhetsteckensnitt. Följande teckensnitt är till exempel enhetsteckensnitt på en iPhone:

- Serif: Times New Roman, Georgia och _serif
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma och _sans
- Fast bredd: Courier New, Courier och _typewriter

Använd teckensnitt som är 14 pixlar eller större.

Använd enhetsteckensnitt för redigerbara textfält. Enhetsteckensnitt i textfält återges också snabbare än inbäddade teckensnitt.

Använd inte understruken text för indatafält. Ställ inte heller in textjusteringar i textfältet. Indatatextfält i iPhone har endast stöd för vänsterjusterad text (standard).

Om du använder inställningen för TLF-text för ett textfält i Flash Professional CS5, ska du stänga av RSL-biblioteket i standardlänkningen i ActionScript 3.0-inställningarna. Annars kommer inte programmet att fungera på iPhone-enheten eftersom den då försöker att använda SWF-filen för RSL-biblioteket:

1 Välj Arkiv > Publiceringsinställningar.

2 I dialogrutan Publiceringsinställningar klickar du på Flash-fliken.

3 Klicka på knappen Skript till höger om listrutan Skript (ActionScript 3.0).

4 Klicka på fliken Bibliotekssökväg.

5 Välj Sammanfogad i kod i listrutan Standardlänkning.

Den kan även vara en bra idé att implementera alternativ till inmatning i textfält. Om du till exempel vill att användaren ska ange ett numeriskt värde behöver du inte använda ett textfält. Du kan istället använda två knappar för att höja eller sänka ett värde.

Var medveten om det utrymme som används av det virtuella tangentbordet. När det virtuella tangentbordet aktiveras (till exempel när en användare trycker på ett textfält) anpassar programmet scenens position. Den automatiska ändringen av positionen ser till att det markerade textfältet för inmatning är synligt:

- Ett textfält högst upp på scenen flyttas överst på det synliga scenområdet. (Det synliga scenområdet är mindre för att även rymma det virtuella tangentbordet.)
- Ett textfält längst ned på scenen förblir längst ned på det nya scenområdet.
- Ett textfält på någon annan del av scenen flyttas till den vertikala mittpositionen på scenen.

När användaren klickar i ett textfält för att redigera det (och det virtuella tangentbordet visas), skickar objektet TextField en `focusIn`-händelse. Du kan lägga till en händelseavlyssnare för den här händelsen för att flytta textfältet.

Vid textfält med endast en rad visas en raderingsknapp (till höger om texten) när användaren redigerar texten. Den här knappen visas emellertid inte om textfältet är för nätt tilltaget.

Efter redigering av textfält med en rad stänger användaren det virtuella tangentbordet genom att peka på Klar på tangentbordet.

Efter redigering av textfält med flera rader stänger användaren det virtuella tangentbordet genom att peka utanför textfältet. Detta leder till att fokus tas bort från textfältet. Se till att du i din design tar med områden utanför textfältet när virtuella tangentbord visas. Om textfältet är för stort kommer kanske inga andra områden att kunna visas.

Om du använder vissa andra Flash Professional CS5-komponenter kan du undvika att fokus tas bort från ett textfält. Dessa komponenter är avsedda att användas för datorer, där detta fokusbeteende är önskvärt. En sådan komponent är TextArea-komponenten. När den är i fokus (och redigeras) kan du inte ta bort fokus från den genom att klicka på ett annat visningsobjekt. Du kan även placera andra Flash Professional CS5-komponenter på scenen för att förhindra att fokus ändras från textfältet som redigeras.

Förlita dig inte på tangentbordshändelser. I SWF-innehåll som är utformat för webben kan till exempel tangentbordet användas för att låta användaren kontrollera programmet. Men på en iPhone visas det virtuella tangentbordet endast när användaren redigerar ett textfält. För iPhone-program skickas endast tangentbordshändelser när det virtuella tangentbordet är synligt.

# Spara programmets status

Programmet kan avslutas när som helst (till exempel om det ringer på telefonen). Det kan därför vara bra att spara programmets status varje gång den ändras. Du kan till exempel spara inställningarna i en fil eller i en databas i lagringskatalogen för programmet. Du kan även spara data i ett lokalt delat objekt. Du kan sedan återställa programmets status när det startas om. Om ett telefonsamtal avbryter ett program, kommer programmet att starta om när samtalet är slut.

Förlita dig inte på att objektet NativeApplication skickar en `exiting`-händelse när programmet avslutas. Det kanske inte gör det.

# Ändringar av skärmorientering

Innehåll kan visas på iPhone-enheten med stående eller liggande orientering. Fundera över hur du vill att ditt program ska hantera ändringar av skärmorienteringen. Mer information finns i Identifiera ändringar av skärmorienteringen.

# Träffytor

Tänk på storleken på träffytorna när du utformar knappar och andra element i användargränssnittet som användaren kan trycka på. Gör dessa element tillräckligt stora för att lätt kunna aktiveras med ett finger på pekskärmen. Se också till att du har tillräckligt med utrymme mellan träffytorna. Träffytorna bör vara mellan 44 och 57 pixlar stora.

# Minnesallokering

Allokering av färska minnesblock är kostsamt. Det kan göra att program eller prestanda blir långsammare vid animeringar eller interaktivitet eftersom skräpsamlingen aktiveras.

Försök att återvinna objekt när det går istället för att göra dig av med objekt och skapa nya.

Tänk på att vektorobjekt förbrukar ofta mindre minne än arrayer. Se även Vector-klassen och Array-klassen.

Mer information om minneshantering finns i Spara minneskapacitet.

# Rit-API

Försök att undvika rit-API:t i ActionScript (klassen Graphics) när du skapar grafik. Om du använder rit-API:t skapas objekt dynamiskt på scenen och sedan renderas de till rastreraren. Om det är möjligt gör du istället så att dessa objekt är statiska när de skapas på scenen.

Objekt som skapats med rit-API, och som anropas flera gånger, förstörs och återskapas varje gång ActionScript körs. Statiska objekt behålls däremot i minnet under olika tidsperioder.

# Händelsebubbling

För djupt inkapslade visningsobjektbehållare kan händelsebubbling vara kostsamt. Du kan minska den här kostnaden genom att hantera händelsen helt i målobjektet och sedan anropa metoden `stopPropagation()` för händelseobjektet. När du anropar den här metoden förhindrar du att händelsen bubblar upp. Om du anropar den här metoden innebär det också att händelsen inte mottas av de överordnade objekten.

Du kan skapa relaterade fördelar genom att göra inkapslingen av visningsobjektet plattare och därmed undvika långa händelsekedjor.

Registrera MouseEvent-händelser i stället för TouchEvent-händelser när det är möjligt. För MouseEvent-händelser används mindre processorkapacitet än för TouchEvent-händelser.

Ange att egenskaperna `mouseEnabled` och `mouseChildren` ska vara `false` när det är möjligt.

# Optimera videoprestandan

Du optimerar videouppspelning på mobila enheter genom att se till att så lite som möjligt av andra aktiviteter pågår i programmet samtidigt som videon spelas upp. Detta gör att så mycket processorkapacitet som möjligt lämnas till videoavkodning och återgivning.

Försök undvika att ha ActionScript-kod som körs medan videon spelas upp. Du bör dessutom undvika att ha kod som körs ofta med en intervalltimer eller på tidslinjen.

Undvik att rita om visningsobjekt som inte tillhör videon. Du bör speciellt tänka på att undvika visningsobjekt som överlappar videoområdet. Det gäller även om de är dolda under videon. Det kommer ändå att ritas om och förbruka bearbetningsresurser. Du kan till exempel använda enkla former för positionsindikatorn och uppdatera den endast några gånger per sekund och inte i varje bildruta. Låt inte videokontrollerna överlappa videoområdet, utan placera dem direkt under. Om du har en videobuffertanimering ska du inte dölja den bakom videon när den inte används utan i stället göra den osynlig.

# Flex- och Flash-komponenter

Många Flex- och Flash-komponenter är utformade för att användas i program på stationära datorer. De här komponenterna, särskilt visningskomponenterna, kan vara långsamma på mobilenheter. Utöver detta kan sådana komponenter dessutom ha olämpliga interaktionsmodeller för mobilenheter.

Adobe arbetar med att utveckla en mobiloptimerad version av Flex-ramverket. Du hittar mer information på http://labs.adobe.com/technologies/flex/mobile/.

# Minska programmets filstorlek

Här följer några tips på hur du kan minska filstorleken på IPA-filer:

- Kontrollera att bakgrundsbitmappar har rätt storlek (inte är större än vad de behöver vara).
- Kontrollera om några extra teckensnitt bäddats in.
- Titta på PNG-resurser för alfakanaler och ta bort dem om de inte behövs. Använd ett verktyg som exempelvis PNG crunch för att minska storleken på PNG-resurser.
- Konvertera om det är möjligt PNG-resurser till JPG-resurser.
- Överväg att komprimera ljudfiler (genom att använda en lägre bithastighet).
- Ta bort resurser som inte används.