

Packager for iPhone® ile ADOBE® AIR® Uygulamaları oluřturma

Yasal bildirimler

Yasal bildirimler için bkz. http://help.adobe.com/tr_TR/legalnotices/index.html.

İçindekiler

Bölüm 1: iPhone için AIR uygulamaları oluşturmaya başlarken

Önemli kavramlar	1
Adobe geliştirici araçlarını edinme	4
Apple geliştirici dosyalarını edinme	4
Flash Professional CS5 ile bir Hello World iPhone uygulaması oluşturma	8

Bölüm 2: iPhone uygulamalarını derleme ve hata ayıklama

iPhone simgesi ve başlangıç ekran görüntüleri	12
iPhone uygulama ayarları	14
iPhone uygulaması yükleyici dosyasını (IPA) derleme	19
Bir iPhone uygulaması yükleme	21
Bir iPhone uygulamasında hata ayıklama	23
iPhone uygulamanızı App Store'a gönderme	25

Bölüm 3: Mobil aygıtlar için ActionScript 3.0 API desteği

Mobil aygıtlarda desteklenmeyen ActionScript 3.0 API'leri	27
Mobil AIR uygulamalarına özgü ActionScript API'leri	29
Mobil uygulama geliştiricilerini özellikle ilgilendiren ActionScript 3.0 API'leri	33

Bölüm 4: iPhone uygulama tasarımı

Donanım hızlandırma	35
Ekran nesnesi performansını arttırmanın diğer yolları	37
Bilgi yoğunluğu	38
Fontlar ve metin girdisi	38
Uygulama durumunu kaydetme	39
Ekran yönlendirme değişiklikleri	40
Vuruş hedefleri	40
Bellek ayırma	40
Çizim API'si	40
Olay köpürmesi	40
Video performansını en iyi hale getirme	41
Flex ve Flash bileşenleri	41
Uygulama dosyası boyutunu azaltma	41

Bölüm 1: iPhone için AIR uygulamaları oluşturmaya başlarken

Adobe® Flash® Platform araçlarını ve ActionScript® 3.0 kodunu iPhone ve iPod Touch için Adobe® AIR® uygulamaları oluşturmada kullanabilirsiniz. Bu uygulamaların dağıtımını, kurulumunu ve çalıştırılması diğer iPhone uygulamaları gibidir.

Not: Bu belgenin geri kalanında iPhone ve iPod Touch aygıtlarından birlikte “iPhone” olarak bahsedilecektir.

Packager for iPhone®, Adobe® Flash® Professional CS5'e dahildir. Packager for iPhone, tüm ActionScript bayt kodlarını yerel iPhone uygulama kodları halinde derler. iPhone uygulamaları iPhone uygulama yükleyici dosyaları (.ipa dosyaları) halinde iTunes Mağazasından dağıtılır.

Uygulamanın kaynak ActionScript 3.0 içeriğini düzenlemek için Flash Professional CS5'i veya Adobe® Flash® Builder™ 4'ü kullanabilirsiniz.

iPhone uygulamalarını geliştirmek için, Flash Professional CS5 kullanın.

Bunun yanında Apple iPhone geliştirici sertifikalarını da edinmeniz gerekir.

Önemli: iPhone uygulamaları geliştirmeden önce, iPhone için uygulama tasarlama hakkındaki bilgileri inceleyin. Bkz. “iPhone uygulama tasarımı” sayfa 35. Aynı zamanda, bir iPhone uygulaması oluşturmak için gereken geliştirici dosyaları hakkında bilgi edinin. Bkz. “Apple geliştirici dosyalarını edinme” sayfa 4.

Önemli kavramlar

ActionScript 3.0 ile bir iPhone uygulaması geliştirmeden önce gerekli kavramları ve iş akışını öğrenmek gerekir.

Terimler Sözlüğü

Bir iPhone uygulaması oluşturmadan önce aşağıdaki kavramlar öğrenilmelidir.

iPhone Dev Center sitesi (<http://developer.apple.com/iphone/>) adresindeki Apple Computer web sitesinde aşağıdakileri yapabilirsiniz:

- iPhone geliştiricisi olmak için başvurabilirsiniz.
- iPhone geliştirme sertifikaları, temel hazırlık profilleri ve uygulama kimlikleri yönetebilir ve oluşturabilirsiniz (aşağıda tanımlanmışlardır).
- Uygulama Deposu için uygulama gönderebilirsiniz.

iPhone geliştirme sertifikası Uygulama geliştirme amacıyla geliştiricinin tanımlanmasında kullanılır.

Bu dosyayı Apple'dan edirsiniz. ActionScript 3.0 kullanarak oluşturduğunuz iPhone uygulamasını imzalamak için bu sertifikayı bir P12 sertifikasına dönüştürsünüz. Bkz. *P12 sertifika dosyası*.

Geliştirme bilgisayarındaki Flash Professional CS5 uygulamalarında hata ayıklamak ve test yapmak için bir iPhone geliştirme sertifikası gerekmez. Ancak, uygulamayı bir iPhone aygıtına kurmak ve test etmek için geliştirme sertifikası gerekir.

Geliştirme sertifikası, uygulamanızın son sürümünü oluşturmak için kullandığınız dağıtım sertifikasından farklıdır. Uygulamanızın son sürümünü oluşturduğunuzda Apple size bir dağıtım sertifikası verir.

Sertifika imzalama talebi Bir geliştirme sertifikası oluşturmada kullanılan kişisel bilgileri içeren dosya. Ayrıca CSR dosyası olarak da bilinir.

Ön hazırlık profili Bir iPhone uygulamasını test etmenize veya dağıtmanıza imkan veren dosyadır. Ön hazırlık profili dosyalarını Apple'dan edebilirsiniz. Ön hazırlık profili belli bir geliştirme sertifikasına, bir uygulama kimliğine ve bir ya da daha fazla aygıt kimliğine atanır. Farklı ön hazırlık profili tipleri mevcuttur:

- **Geliştirme ön hazırlık profili**—Geliştiricinin iPhone aygıtına uygulamanın test sürümünü kurmaya yarar.
- **Test ön hazırlık profili**—Aynı zamanda geçici bir ön hazırlık profili olarak da bilinir. Birden fazla kullanıcıya (ve iPhone ünitesine) uygulamanın test sürümünü dağıtmaya yarar. Bu ön hazırlık profili ve test uygulaması ile kullanıcılar uygulamanızı Uygulama Deposuna sunulmadan önce test edebilir. Not: test uygulamalarını birden fazla aygıtta dağıtmak için bir geliştirme ön hazırlık profili de kullanabilirsiniz.
- **Dağıtım ön hazırlık profili**—Uygulamanızı Uygulama Deposuna sunmak için bir iPhone uygulaması oluşturmaya yarar.

Uygulama Kimliği Belli bir geliştiricinin iPhone uygulamasını (veya uygulamalarını) tanımlayan benzersiz bir dizedir. iPhone Dev Center sitesinde uygulama kimliği oluşturursunuz. Her temel hazırlık profilinin ilişkili bir uygulama kimliği ve uygulama kimliği deseni vardır. Bu uygulama kimliğini (veya deseni) bir uygulama geliştirirken kullanabilirsiniz. Uygulama kimliğini Flash Professional CS5 iPhone Ayarları iletişim kutusunda (veya uygulama tanımlayıcısı dosyasında) kullanırsınız.

iPhone Dev Center'daki uygulama kimlikleri bir paket tanımlayıcı tarafından takip edilen paket çekirdek kimliği içerir. Paket çekirdek kimliği, Apple'ın Uygulama Kimliğine atadığı, 5RM86Z4DJM gibi bir karakter dizesidir. Paket tanımlayıcısı sizin seçtiğiniz ters bir etki alanı adı dizesi içerir. Paket tanımlayıcısı bir joker uygulama kimliğini gösteren yıldız karakteri (*) ile bitebilir. Örnekler şu şekildedir:

- 5RM86Z4DJM.com.ornek.helloWorld
- 96LPVWEASL.com.ornek.* (joker uygulama kimliği)

iPhone Dev Center'da iki tür uygulama kimliği vardır:

- **Joker uygulama kimlikleri**—iPhone Dev Center'da, bu uygulama kimlikleri yıldız işareti (*) ile biter, örn. 96LPVWEASL.com.etkiAlanım.* veya 96LPVWEASL.*. Bu tür uygulama kimliğini kullanan temel hazırlık profili ile, desen ile eşleşen uygulama kimliği kullanan test uygulamaları oluşturabilirsiniz. Uygulamanın uygulama kimliği için, yıldız işaretini geçerli karakterlerden oluşan herhangi bir dizeyle değiştirebilirsiniz. Örneğin, iPhone Dev Center sitesi uygulama kimliği olarak 96LPVWEASL.com.ornek.* ögesinibelirtiyorsa, uygulamanın uygulama kimliği olarak com.ornek.foo veya com.ornek.bar ögesini kullanabilirsiniz.
- **Belirli uygulama kimlikleri**—Bunlar uygulamada kullanılacak eşsiz bir uygulama kimliğini tanımlar. iPhone Dev Center'da bu uygulama kimlikleri yıldız işareti ile bitmez. 96LPVWEASL.com.etkiAlanım.benimUygulamam buna bir örnektir. Bu tür uygulama kimliğini kullanan temel hazırlık profili ile, uygulamalar uygulama kimliğiyle tam olarak eşleşmelidir. Örneğin, iPhone Dev Center sitesi uygulama kimliği olarak 96LPVWEASL.com.ornek.helloWorld ögesini belirtmişse, uygulamanın uygulama kimliği olarak com.ornek.foo ögesini kullanmanız gerekir.

Uygulamanızı geliştirirken, uygulama kimliğini Flash Professional CS5'te iPhone ayarları iletişim kutusunda veya uygulama açıklayıcı dosyasında belirtirsiniz. Uygulama kimlikleriyle ilgili daha fazla bilgi için, "[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)" sayfa 14 başlığında "Dağıtım sekmesi" bölümüne veya "[Uygulama açıklayıcı dosyasında iPhone uygulama özelliklerini ayarlama](#)" sayfa 16 başlığına bakın.

Önemli: Uygulama kimliği belirtilirken, uygulama kimliğinin paket çekirdeği kimliği kısmını göz ardı edin. Örneğin, Apple uygulama kimliğinizi 96LPVWEASL.com.ornek.bob.uygulamam olarak listelerse, 96LPVWEASL kısmını göz

ardı edip uygulama kimliği olarak com.ornek.bob.uygulamam kısmını kullanın. Apple uygulama kimliğinizi 5RM86Z4DJM.* olarak listelerse, 5RM86Z4DJM kısmını göz ardı edin—bu bir joker uygulama kimliğidir.

Bir temel hazırlık profiliyle ilişkili uygulama kimliğini (veya joker uygulama kimliği desenini) iPhone Dev Center'da bulabilirsiniz (<http://developer.apple.com/iphone>). iPhone Geliştirici Programı Portalı'na gidin ve ardından Temel Hazırlık bölümüne gidin.

P12 sertifika dosyası P12 dosyası (.p12 uzantılı bir dosya) bir tür sertifika dosyasıdır (bir Kişisel Bilgi Alışverişi dosyası). Packager for iPhone, bir iPhone uygulaması geliştirmek için bu tür sertifikaları kullanır. Apple'dan aldığınız geliştirici sertifikasını bu sertifika şekline dönüştürsünüz.

Benzersiz Aygıt Kimliği Belli bir iPhone aygıtını tanımlayan benzersiz kod. Aynı zamanda UUID veya aygıt kimliği olarak da bilinir.

Geliştirme iş akışına genel bakış

iPhone için bir uygulama geliştirirken, şu adımları izlersiniz:

- 1 Adobe Flash Professional CS5'i yükleyin
- 2 iTunes programını yükleyin.
- 3 Apple geliştirici dosyalarını edinin. Bu dosyalar geliştirici sertifikası ile ön hazırlık profillerini içerir. Bkz. "[Apple geliştirici dosyalarını edinme](#)" sayfa 4.
- 4 Geliştirme sertifikasını bir P12 sertifikasına dönüştürün. Flash CS5 için sertifikanın bir P12 sertifikası olması gerekir. Bkz. "[Apple geliştirici dosyalarını edinme](#)" sayfa 4.
- 5 Ön hazırlık profilinizi iPhone ile ilişkilendirmek için iTunes programını kullanın.
- 6 Uygulamayı Flash Professional CS5'te yazın.

Bir iPhone uygulaması için kod tasarımı ve en iyileştirmesinde kullanılacak en iyi uygulamalar bilinmelidir. Bkz. "[iPhone uygulama tasarımı](#)" sayfa 35.

Ayrıca, bazı ActionScript 3.0 API'leri sınırlıdır veya iPhone aygıtında desteklenmez. Bkz. "[Mobil aygıtlar için ActionScript 3.0 API desteği](#)" sayfa 27.

Uygulamanın ActionScript 3.0 kodunu düzenlemek için Flash Builder 4.0 da kullanabilirsiniz.

Flash Professional CS5'i uygulamayı geliştirme bilgisayarında test etmek için kullanabilirsiniz.

- 7 Uygulama için simge resmi ve başlangıç ekranı resmi oluşturun. Her iPhone uygulamasının onu kullanıcılara tanıtan bir dizi simgesi vardır. iPhone, program yüklenirken başlangıç ekranı görüntüsünü gösterir. Bkz. "[iPhone simgesi ve başlangıç ekran görüntüleri](#)" sayfa 12.
- 8 iPhone ayarlarını düzenleyin. Bu ayarlar şunları içerir:

- Uygulamanın kimliği (dosya adı, uygulama adı, sürüm numarası ve uygulama kimliği dahil)
- Uygulama kaynak simgesinin konumu
- Uygulamaya atanan P12 sertifikası ve ön hazırlık profili
- Uygulamanın başlangıçtaki boyut oranı

Flash Professional CS5'te, bu ayarları iPhone Ayarları iletişim kutusunda düzenleyebilirsiniz. Ayrıntılar için bkz. "[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)" sayfa 14.

Bu ayarları uygulama açıklayıcı dosyasında doğrudan da ayarlayabilirsiniz. Daha fazla bilgi için bkz. "[Uygulama açıklayıcı dosyasında iPhone uygulama özelliklerini ayarlama](#)" sayfa 16.

9 Packager for iPhone kullanarak IPA dosyasını derleyin. Bkz. “iPhone uygulaması yükleyici dosyasını (IPA) derleme” sayfa 19.

10 Uygulamayı iPhone aygıtına yükleyip test edin. IPA dosyasını yüklemek için iTunes programını kullanın.

Geçici dağıtım için bu genel işlemi tekrar edin, fakat geliştirme ön hazırlık profili yerine test ön hazırlık profili kullanın. Uygulamanın son dağıtımı için dağıtım ön hazırlık profilini kullanarak bu işlemi tekrar edin. (Ön hazırlık profillerinin farklı tipleri hakkında daha fazla bilgi edinmek için bkz. “Terimler Sözlüğü” sayfa 1.)

Uygulamanızın dağıtım sürümünü oluşturduğunuzda “iPhone uygulamanızı App Store'a gönderme” sayfa 25 bölümündeki talimatlara bakın.

Basit bir iPhone uygulaması oluşturmayla ilgili hızlı bir eğitim için, bkz. “Flash Professional CS5 ile bir Hello World iPhone uygulaması oluşturma” sayfa 8.

Adobe geliştirici araçlarını edinme

ActionScript 3.0 kullanarak iPhone uygulamaları geliştirmek için, Flash Professional CS5'e ihtiyacınız vardır.

Önemli: Packager for iPhone sürümünü, Flash Professional CS5'teki önizleme sürümünden güncellemelisiniz. Flash Professional CS5'te Yardım > Güncellemeler'i seçin.

ActionScript kodunu düzenlemek için Flash Builder 4'ü de kullanabilirsiniz. Flash Builder 4 şu adreste mevcuttur: <http://www.adobe.com/products/flashbuilder/>.

Apple geliştirici dosyalarını edinme

iPhone için geliştirilen her uygulamada olduğu gibi, ilk olarak Apple iPhone geliştirici dosyalarını edinmek zorundasınız. Bir iPhone geliştirici sertifikası ve mobil ön hazırlık profili edinmeniz gerekiyor. Ayrıca diğer ön hazırlık profillerini de edinmeniz gerekiyor. Bu dosyaların tanımları için bkz. “Terimler Sözlüğü” sayfa 1.

Not: Bu dosyaları edinmek, uygulama geliştirme işleminin önemli bir parçasıdır. Uygulamanızı geliştirmeden önce bu işlemi tamamladığınızdan emin olun. Geliştirme dosyalarını edinmek basit bir işlem değildir. Bu talimatları ve Apple iPhone Dev Center web sitesindeki talimatları dikkatle okuyun.

iPhone geliştirici dosyalarını edinme ve bunlarla çalışma

Apple'dan bir iPhone geliştirici sertifikası ve ön hazırlık profilleri edinmelisiniz. Ayrıca, sertifikayı bir P12 sertifikasına dönüştürmelisiniz.

iTunes programını yükleme

Uygulamanızı iPhone aygıtına yüklemek için iTunes programı gerekir. Ayrıca, iTunes programını iPhone aygıtının aygıt kimliğini belirlemek için de kullanabilirsiniz. Bir iPhone geliştirici sertifikası için başvurduğunuzda aygıt kimliğini bilmeniz gerekir.

iPhone geliştirici sertifikası için başvurma ve bir temel hazırlık profili oluşturma

Hala yapmadıysanız, Apple iPhone Dev Center sitesinde iPhone geliştiricisi olmak için kaydolun (<http://developer.apple.com/iphone/>).

Not: iPhone'a yönelik AIR uygulamaları geliştirmek için iPhone SDK'ye veya XCode'a ihtiyacınız yoktur. Kayıtlı bir iPhone geliştiricisi olmanız gerekir. Ayrıca bir geliştirici sertifikası ve temel hazırlık profili edinmeniz gerekir.

- 1 iPhone geliştiricisi hesap kimliğinizi kullanarak iPhone Dev Center'da oturum açın.
 - 2 iPhone Dev Center'da, iPhone geliştirici sertifikası için başvuru yapın (ve satın alın).
Apple size iPhone Geliştirici Programı aktivasyon kodunu içeren bir e-posta gönderir.
 - 3 iPhone Dev Center'a geri dönün. Geliştirici programınızın etkinleştirilmesiyle ilgili talimatları takip edin (ve istendiğinde aktivasyon kodunuzu girin).
 - 4 Aktivasyon kodunuz kabul edildiğinde, iPhone Dev Center'ın iPhone geliştirici Programı Portalı kısmına gidin.
 - 5 Sertifika imzalama isteği dosyası oluşturun. Bu dosyayı iPhone Geliştirme Sertifikası almak için kullanacaksınız. Talimatlar için bkz. “Bir sertifika imzalama isteği oluşturma” sayfa 6.
 - 6 Sonraki adımda iPhone'unuz için Aygıt Kimliği (veya Benzersiz Aygıt Kimliği) belirtmeniz istenir. UDID'i (Benzersiz Aygıt Kimliği) iTunes'dan edinebilirsiniz.
 - a iPhone aygıtını bir USB kabloyla bağlayın. Ardından, iTunes programında iPhone özet sekmesini seçin.
 - b iPhone geliştirici merkezinin sitesinden temel hazırlık profilini indirdiğinizde, iTunes'a ekleyin.
 - c Sonra gösterilen Seri Numarasını tıklatın. UDID şimdi görüntülenmektedir. UDID'yi panoya kopyalamak için; Mac'te, Command ve C tuşuna aynı anda basın veya Windows'ta Ctrl ve C tuşuna aynı anda basın.
 - 7 Bir temel hazırlık profili ve iPhone geliştirme sertifikası oluşturarak yükleyin.
iPhone Dev Center'daki talimatları izleyin. iPhone Geliştirme Programı Portalı bölümünde talimatlara bakın. Geliştirme sertifikanızı almak ve temel hazırlık profilinizi oluşturmak için Geliştirme Temel Hazırlığı Asistanı'nı kullanmak isteyebilirsiniz.
XCode içeren adımları yok sayın. Flash Professional CS5 kullanarak iPhone uygulamaları geliştirmek için XCode kullanmanız gerekmez.
 - 8 iTunes'ta File (Dosya) > Add To Library (Kütüphaneye Ekle) seçimini yapın. Ardından, temel hazırlık profili dosyasını seçin (dosya adı uzantısı olarak mobileprovision uzantısına sahiptir). Ardından iPhone'unuzu iTunes ile senkronize edin.
Böylece bu ön hazırlık profiliyle ilgili uygulamayı iPhone aygıtınızda test edebilirsiniz.
Belli bir ön hazırlık profilinin iTunes'a eklendiğini doğrulamak için, bu profili kütüphaneye eklemeyi deneyebilirsiniz. iTunes sizden mevcut bir ön hazırlık profilini yenisiyle değiştirmenizi isterse, Cancel (İptal) düğmesine basabilirsiniz. (Profil yüklenmiş durumdadır.) Ayrıca, iPhone aygıtına yüklü ön hazırlık profillerini kontrol edebilirsiniz:
 - a iPhone'da Ayarlar uygulamasını açın.
 - b Genel kategorisini açın.
 - c Profilleri tıklatın. Profiller sayfası yüklü ön hazırlık profillerini listeler.
 - 9 Bunu yapmadıysanız, iPhone geliştirme sertifikası dosyasını (.cer dosyası) indirin.
Geliştirme Temel Hazırlık Yardımcısı bu dosyayı indirmeniz için bir bağ temin etmiş olabilir. Dosyayı ayrıca Apple iPhone Dev Center web sitesinin Provisioning Portal'ının (Temel Hazırlık Portalı) Certificates (Sertifikalar) bölümünde de bulabilirsiniz (<http://developer.apple.com/iphone/>).
 - 10 Daha sonra, iPhone geliştirici sertifikasını bir P12 dosyasına dönüştüreceksiniz. Talimatlar için bkz. “Bir geliştirici sertifikasını P12 dosyasına dönüştürme” sayfa 7.
- Artık basit bir Hello World uygulaması oluşturabilirsiniz. Bkz. “Flash Professional CS5 ile bir Hello World iPhone uygulaması oluşturma” sayfa 8.

Bir sertifika imzalama isteği oluşturma

Bir geliştirici sertifikası almak için, Apple iPhone Dev Center sitesine gönderilen bir sertifika imzalama isteği dosyası oluşturursunuz.

Mac OS'de bir sertifika imzalama isteği oluşturma

Mac OS'de, bir kod imzalama isteği oluşturmak için Keychain Access uygulamasını kullanabilirsiniz. Keychain Access uygulaması, Applications (Uygulamalar) dizininin Utilities (Yardımcı Programlar) adlı alt dizinindedir. Keychain Access menüsünde, Certificate Authority (Sertifika Yetkilisi) bölümünden Certificate Assistant (Sertifika Asistanı) > Request a Certificate (Bir Sertifika İste) öğesini seçin.

- 1 Keychain Access'i açın.
- 2 Keychain Access menüsünde Preferences (Tercihler) öğesini seçin.
- 3 Preferences (Tercihler) iletişim kutusunda Certificates (Sertifikalar) öğesini tıklatın. Ardından Online Certificate Status Protocol (Çevrimiçi Sertifika Durumu Protokolü) ve Certificate Revocation List (Sertifika İptal Listesi) öğelerini kapalı duruma getirin. İletişim kutusunu kapatın.
- 4 Keychain Access menüsünde, Certificate Authority (Sertifika Yetkilisi) bölümünden Certificate Assistant (Sertifika Asistanı) > Request a Certificate (Bir Sertifika İste) öğesini seçin.
- 5 iPhone geliştirici hesabı kimliğinizle eşleşen e-posta adresini ve adı girin. CA e-posta adresini girmeyin. Request is Saved to Disk (İstek Diske Kaydedildi) öğesini seçin ve ardından Continue (Devam Et) düğmesine basın.
- 6 Dosyayı kaydedin (CertificateSigningRequest.certSigningRequest).
- 7 CSR dosyasını [adresindeki Apple iPhone geliştiricisi sitesine](#) yükleyin. (Bkz "iPhone geliştirici sertifikası için başvurma ve bir temel hazırlık profili oluşturma".)

Windows'ta bir sertifika imzalama isteği oluşturma

Windows geliştiricileri için iPhone geliştirici sertifikasını bir Mac bilgisayarında edinmek en kolay yol olabilir. Ancak, bir Windows bilgisayarında da sertifika almak mümkündür. İlk olarak, OpenSSL kullanarak bir sertifika imzalama talebi (CSR dosyası) oluşturursunuz:

- 1 OpenSSL'i Windows bilgisayarınıza yükleyin. (<http://www.openssl.org/related/binaries.html> adresine gidin.)
Ayrıca Open SSL indirme sayfasında listelenen Visual C++ 2008 Yeniden Dağıtılabılır dosyalarını da yüklemeniz gerekebilir. (Visual C++ uygulamasının bilgisayarınızda yüklü olması *gerekmez*.)
- 2 OpenSSL sepet dizinine bir Windows komut oturumu ve CD açın (c:\OpenSSL\bin\ gibi).
- 3 Komut satırına şunları girerek özel anahtar oluşturun:

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

Bu özel anahtar dosyasını kaydedin. Daha sonra kullanılacaktır.

OpenSSL'yi açarken hata mesajlarını göz ardı etmeyin. OpenSSL bir hata mesajı oluşturursa, hala dosya çıktısı oluşturabilir. Ancak, bu dosyalar kullanılabilir durumda olmayabilir. Hata görürseniz, söz diziminizi kontrol edin ve komutu tekrardan çalıştırın.
- 4 Komut satırına şunları girerek CSR dosyasını oluşturun:

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

E-posta adresini, CN (sertifika adı) ve C (ülke) değerlerini kendinizinkilerle değiştirin.
- 5 CSR dosyasını [adresindeki Apple iPhone geliştiricisi sitesine](#) yükleyin. (Bkz "iPhone geliştirici sertifikası için başvurma ve bir temel hazırlık profili oluşturma".)

Bir geliştirici sertifikasını P12 dosyasına dönüştürme

Flash Professional CS5 kullanarak iPhone uygulamaları geliştirmek için bir P12 sertifika dosyası kullanmalısınız. Bu sertifikayı Apple'dan aldığınız Apple iPhone geliştirici sertifika dosyasına göre oluşturursunuz.

iPhone geliştirici sertifikasını Mac OS'de P12 dosyasına dönüştürme

Apple iPhone sertifikasını indirdiğinizde, P12 sertifikası biçimine aktarın. Mac® OS'ta bunu yapmak için:

- 1 Keychain Access uygulamasını açın (Applications/Utilities (Uygulamalar/Yardımcı Programlar) klasöründe)
- 2 Anahtarlığa sertifikayı eklemediyseniz, File (Dosya) > Import (İçe Aktar) öğesini seçin. Daha sonra Apple'dan aldığınız sertifika dosyasına (.cer dosyası) gidin.
- 3 Daha sonra Keychain Access'te Keys (Anahtarlar) kategorisini seçin.
- 4 iPhone Geliştirme Sertifikasıyla ilgili özel anahtarı seçin.
Özel anahtar iPhone Geliştiricisi tarafından belirlenir: <Adı> <Soyadı> onunla eşleştirilen genel sertifika.
- 5 File (Dosya) > Export Items (Öğeleri Dışa Aktar) seçimini yapın.
- 6 Anahtarınızı Kişisel Bilgi Alışverişi (.p12) dosya biçiminde kaydedin.
- 7 Sizden bu anahtarı başka bir bilgisayara aktarmaya çalıştığınızda kullanılacak bir şifre oluşturmanız istenir.

Apple geliştirici sertifikasını Windows'ta P12 dosyasına dönüştürme

Flash CS5 kullanarak iPhone uygulamaları geliştirmek için bir P12 sertifika dosyası kullanmalısınız. Bu sertifikayı Apple'dan aldığınız Apple iPhone geliştirici sertifika dosyasına göre oluşturursunuz.

- 1 Apple'dan aldığınız geliştirici sertifika dosyasını bir PEM sertifika dosyasına dönüştürün. OpenSSL bölme dizininden aşağıdaki komut satırını çalıştırın:

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

- 2 Bir Mac bilgisayarındaki anahtarlıktan özel anahtar kullanıyorsanız, bunu bir PEM anahtarına dönüştürün:

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

- 3 Artık anahtara ve iPhone geliştirici sertifikasının PEM sürümüne göre geçerli bir P12 dosyası oluşturabilirsiniz:

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

Mac OS anahtarlığından bir anahtar kullanıyorsanız, bir önceki adımda oluşturduğunuz PEM sürümünü kullanın. Aksi takdirde, daha önce oluşturduğunuz (Windows'ta) OpenSSL anahtarını kullanın.

Sertifikaları, aygıt kimliklerini, uygulama kimliklerini ve temel hazırlık profillerini yönetme

Apple iPhone Dev Center sitesinde sertifikaları, aygıt kimliklerini, uygulama kimliklerini ve temel hazırlık profillerini yönetebilirsiniz (<http://developer.apple.com/iphone/>). Sitenin iPhone Developer Program Portal (iPhone Geliştirici Programı Portalı) bölümüne gidin.

- Geliştirme sertifikalarınızı yönetmek için Certificates (Sertifikalar) bağına tıklayın. Bir sertifika oluşturabilir, indirebilir veya iptal edebilirsiniz. Sertifika oluşturmak için, öncelikle bir sertifika imzalama isteği oluşturmanız gerekir. Bkz. “Bir sertifika imzalama isteği oluşturma” sayfa 6
- Geliştirme uygulamanızın yüklenebileceği aygıtların listesini yönetmek için Aygıtlar bağına tıklayın.
- Uygulama kimliklerinizi yönetmek için Uygulama Kimlikleri bağına tıklayın. Bir temel hazırlık profili oluşturduğunuzda, bu bir uygulama kimliğine bağlıdır.

- Temel hazırlık profillerinizi yönetmek için Temel Hazırlık bağına tıklayın. Ayrıca geliştirme temel hazırlık profilleri oluşturmak için Geliştirme Temel Hazırlık Asistanı'nı kullanabilirsiniz.
- Uygulamanızı Uygulama Deposu'na göndermek istediğinizde veya uygulamanızın Ad Hoc sürümünü oluşturmak istediğinizde Dağıtım bağına tıklayın. Bu bölümün, Uygulama Deposu'na uygulama göndermek için kullandığınız iTunes Connect sitesine bağı vardır.

Flash Professional CS5 ile bir Hello World iPhone uygulaması oluşturma

Önemli: Uygulamayı oluşturmadan önce, gereken geliştirici uygulamalarını ve dosyalarını indirin. Bkz. "[Adobe geliştirici araçlarını edinme](#)" sayfa 4 ve "[Apple geliştirici dosyalarını edinme](#)" sayfa 4.

Bir Flash Professional CS5 projesi oluşturma

Bir iPhone uygulamasını doğrudan Flash Professional CS5'te oluşturabilirsiniz:

- 1 Flash CS5'i açın.
- 2 Dosya > Yeni seçimini belirleyin.
- 3 iPhone OS ögesini seçin.
- 4 Tamam düğmesini tıklayın.

Uygulamaya içerik ekleme

Ardından, uygulamaya "Hello world!" metnini ekleyin:

- 1 Metin Aracını seçin ve sahneyi tıklayın.
- 2 Metin alanına ait Özellikler ayarlarında, Klasik Metin'i (TLF Metni değil) seçin.
Bu basit bir uygulamadır ve Klasik Metin yeterlidir. TLF Metnini kullanmak için bazı başka ayarlar uygulamanız gerekir. Bkz. "[Fontlar ve metin girdisi](#)" sayfa 38.
- 3 Yeni metin alanına "Hello World!" yazın
- 4 Seçim Aracıyla metin alanını seçin.
- 5 Sonra Özellikler denetçisini açarak aşağıdaki ayarları yapın:
 - Karakter > Aile: _sans
 - Karakter > Boyut: 50
 - Konum > X: 20
 - Konum > Y: 20
- 6 Dosyayı kaydedin.
- 7 Kontrol Et > Test Filmi > AIR Hata Ayıklama Başlatıcısı'nda (Mobil) ögesini seçin.
Flash Professional CS5, SWF içeriğini derleyerek AIR Hata Ayıklama Başlatıcısı'nda (ADL) uygulamanın bir sürümünü gösterir. Bu size uygulamanın hızlı bir önizlemesini verir.

Uygulama için simge resmi ve başlangıç ekranı resmi oluşturma

Tüm iPhone uygulamaları iTunes uygulamasının kullanıcı arabiriminde ve iPhone ekranında görünen simgelere sahiptir.

- 1 Proje dizininde bir dizin oluşturarak simgeler adını verin.
- 2 Simgeler dizininde üç adet PNG dosyası oluşturun. Bunları Simge29.png, Simge57.png ve Simge512.png. olarak adlandırın.
- 3 Uygulamanız için uygun resmi oluşturmak üzere PNG dosyalarını düzenleyin. Dosyalar 29x29 piksel, 57x57 piksel ve 512x512 piksel olmalıdır. Bu test için resim olarak düz renkli kareler kullanabilirsiniz.

Tüm iPhone uygulamaları yüklenirken bir açılış resmi görüntüler. Açılış resmini bir PNG dosyasında tanımlarsınız:

- 1 Ana geliştirme dizininde Default.png adlı bir PNG dosyası oluşturun. (Bu dosyayı simgeler alt dizinine koymayın. Dosyayı baş harfi büyük olacak şekilde Default.png olarak adlandırdığınızdan emin olun.
- 2 Dosyayı genişliği 320 piksel, yüksekliği 480 piksel olacak şekilde düzenleyin. Şimdilik içerik düz beyaz bir dikdörtgen olabilir. (Bunu daha sonra değiştirirsiniz.)

Not: Apple Uygulama Deposuna bir uygulama gönderirken 512 piksellik dosyanın JPG (PNG değil) versiyonunu kullanırsınız. Bir uygulamanın geliştirme sürümlerini test ederken ise PNG versiyonunu kullanırsınız.

Bu grafikler hakkında ayrıntılı bilgi edinmek için, bkz. “[iPhone simgesi ve başlangıç ekran görüntüleri](#)” sayfa 12.

Uygulama ayarlarını düzenleme

Önemli: Hala yapmadıysanız, gerekli geliştirici uygulamalarını ve iPhone geliştirme dosyalarını indirin. Bkz. “[Apple geliştirici dosyalarını edinme](#)” sayfa 4.

Flash Professional CS5 iPhone Ayarları iletişim kutusunda iPhone uygulamasının birçok temel özelliğini tanımlayın.

- 1 Dosya > iPhone İşletim Sistemi Ayarları seçimini yapın.
- 2 Genel sekmesinde aşağıdaki ayarları yapın:
 - Çıktı dosyası: HelloWorld.ipa
Bu oluşturulacak iPhone yükleyici dosyasının dosya adıdır.
 - Uygulama adı: Hello World
Bu, uygulamanın iPhone'daki uygulama simgesi altında gösterilen adıdır.
 - Sürüm: 1.0
Uygulamanın sürümü.
 - Boyut oranı: portre
 - Tam Ekran: seçili.
 - Otomatik Yönlendirme: seçili değil.
 - Görüntüleme: CPU
Diğer seçenekler GPU ve Otomatik, görüntüleme için donanım hızlandırma kullanır. Bu özellik, donanım hızlandırmasından faydalanmak üzere tasarlanmış grafik üzerine yoğunlaşan uygulamaların (oyunlar gibi) performansının iyileştirilmesine yardımcı olabilir. Daha fazla bilgi edinmek için bkz. “[Donanım hızlandırma](#)” sayfa 35.
 - Dahil edilen dosyalar: Başlangıç ekranı resim dosyasını (Default.png) Dahil Edilen Dosyalar listesine ekleyin.

Not: Bu Hello World örneği için, bu talimatlarda verilen ayarlarda değişiklik yapmayın. Sürüm ayarı gibi bazı ayarlar belirli kısıtlamalar içerir. Bu kısıtlamalar “iPhone uygulama ayarları” sayfa 14 başlığında açıklanmıştır.

3 Konuşlandırma sekmesinde aşağıdaki ayarları yapın:

- Sertifika: Apple'dan edindiğiniz geliştirici sertifikasına göre .p12 sertifikasını arayıp seçin.

Bu sertifika dosyayı imzalamak için kullanılır. Apple iPhone sertifikasını .p12 biçimine dönüştürmelisiniz. Daha fazla bilgi edinmek için bkz. “[Adobe geliştirici araçlarını edinme](#)” sayfa 4.

- Şifre: sertifikanın şifresini girin.
- Ön hazırlık dosyası: Apple'dan edindiğiniz geliştirici ön hazırlık dosyasını arayıp seçin. Bkz. “[Apple geliştirici dosyalarını edinme](#)” sayfa 4.
- Uygulama Kimliği: Bu alan seçilebilir özellikteyse, Apple'a verdiğiniz uygulama kimliğiyle eşleşen bir uygulama kimliği girebilirsiniz (com.örnek.as3.HelloWorld gibi).

Uygulama kimliği, uygulamanızı diğerlerinden ayırır.

Alan seçilebilir özellikte değilse, ön hazırlık profili belli bir uygulama kimliğine bağlı demektir. Uygulama kimliği bu alanda gösterilir.

Uygulama Kimliği belirtme konusunda ayrıntılar için “[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)” sayfa 14 başlığının “Dağıtım sekmesi” bölümüne bakın.

- ### 4 Simgeler sekmesinde, Simgeler listesindeki Simge 29 x 29'u tıklatın. Sonra, daha önce oluşturduğunuz 29 x 29 piksel PNG dosyasının konumunu belirtin (bkz. “[Uygulama için simge resmi ve başlangıç ekranı resmi oluşturma](#)” sayfa 9). Ardından 57 x 57 piksel simge ve 512 x 512 piksel simge için PNG dosyalarını belirtin.

- ### 5 Tamam düğmesini tıklatın.

- ### 6 Dosyayı kaydedin.

Uygulama ayarları hakkında ayrıntılı bilgi edinmek için bkz. “[iPhone uygulama ayarları](#)” sayfa 14.

IPA dosyasını derleme

Artık IPA yükleyici dosyasını derleyebilirsiniz:

- 1 Dosya > Yayınla seçeneğini belirleyin.
- 2 iPhone Ayarlar iletişim kutusunda, Tamam düğmesini tıklatın.

iPhone Paketleyicisi proje dizininde iPhone uygulama yükleyici dosyası olan HelloWorld.ipa dosyasını oluşturur. IPA dosyasını derlemek birkaç dakika sürebilir.

Uygulamayı iPhone aygıtına yükleme

iPhone uygulamasını iPhone aygıtında test etmek üzere yüklemek için:

- 1 iTunes uygulamasını açın.
- 2 Bunu zaten yaptıysanız, bu uygulamanın ön hazırlık profilini iTunes'a ekleyin. iTunes'ta File (Dosya) > Add File To Library (Kütüphaneye Dosya Ekle) öğesini seçin. Ardından, ön hazırlık profili dosyasını seçin (dosya tipi mobil ön hazırlıktır).

Şimdilik, uygulamayı geliştirici iPhone aygıtında test etmek için, geliştirme ön hazırlık profilini kullanın.

Daha sonra uygulamayı iTunes Mağazasına dağıtırken, dağıtım profilini kullanın. Uygulamayı geçici olarak (iTunes Mağazasından geçirmeden birden fazla aygıt) dağıtmak için geçici ön hazırlık profilini kullanın.

Ön hazırlık profilleri hakkında daha fazla bilgi edinmek için bkz. “[Apple geliştirici dosyalarını edinme](#)” sayfa 4.

- 3 Bazı iTunes sürümleri uygulamanın aynı versiyonu zaten yüklüyse uygulamayı yenisiyle değiştirmez. Bu durumda, uygulamayı aygıtın ve iTunes içerisindeki uygulamalar listesinden silin.
- 4 Uygulamaya ait IPA dosyasını çift tıklayın. Bu dosya uygulamalar listesinde görünmelidir.
- 5 iPhone aygıtını bilgisayarın USB bağlantı noktasına bağlayın.
- 6 iTunes'ta aygıtın Uygulama sekmesini kontrol edin ve uygulamanın yüklenecek uygulamalar listesinde seçili olduğundan emin olun.
- 7 iTunes uygulamasının sol taraftaki listesinden aygıtı seçin. Ardından Sync (Senkr) düğmesine basın. Senkr tamamlandığında, Hello World uygulaması iPhone aygıtınızda görünür.

Yeni sürüm yüklü değilse, iPhone aygıtınızdan ve iTunes'taki uygulamalar listesinden silin ve ardından bu işlemi yineleyin. O anda yüklü olan sürüm aynı uygulama kimliği ve sürümüne sahipse bu durum söz konusu olabilir.

Uygulamanızı yüklemeye çalışırken iTunes bir hata görüntülerse, “[Bir iPhone uygulaması yükleme](#)” sayfa 21 bölümündeki “Uygulama yükleme sorunlarını giderme” kısmına bakın.

Başlangıç ekranı grafiğini düzenleme

Uygulamayı derlemenizden önce bir Default.png dosyası oluşturduanız (bkz. “[Uygulama için simge resmi ve başlangıç ekranı resmi oluşturma](#)” sayfa 9). Bu PNG dosyası, uygulama yüklenirken başlangıç görüntüsü olarak görev yapar. Uygulamayı iPhone aygıtınızda test ederken başlangıçtaki bu boş ekranı fark etmiş olabilirsiniz.

Bu resmi, uygulamanın (“Hello World!”) başlangıç ekranıyla eşleşecek şekilde değiştirmelisiniz:

- 1 Uygulamayı aygıtınızda açın. İlk “Hello World” metni görüldüğünde, Home (Ana Sayfa) düğmesini (ekranın altında) basılı tutun. Home (Ana Sayfa) düğmesini basılı tutarken Power/Sleep (Güç/Uyku) düğmesine basın (iPhone menüsünün en üstünde). Bu bir ekran görüntüsü alır ve bu görüntüyü Film Rulosu'na gönderir.
- 2 iPhoto veya başka bir fotoğraf aktarma uygulamasından fotoğraf aktararak görüntüyü geliştirme bilgisayarınıza aktarın. (Mac OS'ta Image Capture (Resim Yakalama) uygulamasını da kullanabilirsiniz.)

Ayrıca fotoğrafı e-posta ile geliştirme bilgisayarınıza gönderebilirsiniz:

- Fotoğraflar uygulamasını açın.
- Kamera Kaydını açın.
- Yakaladığınız ekran görüntüsü resmini açın.
- Resmi tıklayın ve ardından sol alt köşedeki “ileri” (ok) düğmesini tıklayın. Sonra Email Photo (Fotoğrafi E-posta ile Gönder) seçeneğini tıklayarak resmi kendinize gönderin.

- 3 Default.png dosyasını (geliştirme dizininde) yakalanan ekran resminin bir PNG sürümüyle değiştirin.

- 4 Uygulamayı yeniden derleyin (bkz. “[IPA dosyasını derleme](#)” sayfa 10) ve iPhone aygıtınıza tekrar yükleyin.

Artık uygulama yüklenirken yeni başlangıç ekranını kullanır.

Not: Doğru boyutlarda olduğu sürece (320'ye 480 piksel), Default.png dosyası için dilediğiniz resmi oluşturabilirsiniz. Ancak, çoğunlukla ideal olan Default.png resmini uygulamanın ilk haliyle eşleştirmektir.

Bölüm 2: iPhone uygulamalarını derleme ve hata ayıklama

Packager for iPhone kullanarak bir iPhone uygulamasını derleyebilirsiniz. Packager for iPhone, Flash Professional CS5'e dahildir.

Geliştirme bilgisayarında uygulamanın hataları ayıklanabilir. Ayrıca iPhone aygıtına bir hata ayıklama sürümü yüklenerek Flash Professional CS5'te izleme() çıktısı alınabilir.

Başlangıçtan sona kadar bir iPhone uygulamasının nasıl oluşturulacağıyla ilgili bir eğitim için bkz. "[Flash Professional CS5 ile bir Hello World iPhone uygulaması oluşturma](#)" sayfa 8.

iPhone simgesi ve başlangıç ekran görüntüleri

Tüm iPhone uygulamaları iTunes uygulamasının kullanıcı arabiriminde ve iPhone ekranında görünen simgelere sahiptir.

iPhone uygulama simgeleri

Bir iPhone uygulaması için aşağıdaki simgeleri belirlersiniz:

- 29x29 piksel simge—iPhone ve iPod touch'da Spotlight arama sonuçları bu simgeyi kullanır.
- 48x48 piksel simge: iPad'de Spotlight araması sonuçları bu simgeyi kullanır.
- 57x57 piksel simge—iPhone ve iPod touch ana ekranları bu simgeyi gösterir.
- 72x72 piksel simge (isteğe bağlı): iPad ana ekranı bu simgeyi gösterir.
- 512'ye 512-piksel simgesi—iTunes bu simgeyi görüntüler. 512-piksel PNG dosyası yalnızca uygulamanızın geliştirme sürümlerini test etmek için kullanılır. Son uygulamayı Apple Uygulama Deposu'na gönderdiğinizde, 512 görüntüsünü ayrı olarak, bir JPG dosyası biçiminde gönderirsiniz. Bu dosya IPA'da yer almaz.

Flash Professional CS5'te bu simgeleri iPhone Ayarları iletişim kutusunun Simgeler sekmesine ekleyin. Bkz. "[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)" sayfa 14.

Simgelerin konumlarını da uygulama açıklayıcı dosyasına ekleyebilirsiniz:

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

iPhone simgeye parlak bir efekt verir. Bunu kaynak resme dahil etmeniz gerekmez. Bu varsayılan parlak efekti kaldırmak için, aşağıdakini uygulama tanımlayıcı dosyasındaki InfoAdditions ögesine ekleyin:

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

Bkz. “Uygulama açıklayıcı dosyasında iPhone uygulama özelliklerini ayarlama” sayfa 16.

Başlangıç ekranı resmi (Default.png)

Tüm iPhone uygulamaları yüklenirken bir açılış resmi görüntüler. Açılış resmini Default.png adlı bir PNG dosyasında tanımlarsınız. Ana geliştirme dizininde Default.png adlı bir PNG dosyası oluşturun. (Bu dosyayı bir alt dizine koymayın. Dosyayı baş harfi büyük olacak şekilde Default.png olarak adlandırdığımızdan emin olun.)

Default.png dosyası, uygulamanın başlangıç yönüne veya tam ekran olup olmasına bağlı olmadan 320 piksel genişliğinde ve 480 piksel uzunluğundadır.

Uygulamanızın başlangıç yönü yataysa, dikey bir uygulamanın kullandığı boyutlarla aynı boyutu kullanın: 320 piksel genişlik ve 480 piksel yükseklik. Ancak, PNG dosyasında resmi saat yönünün tersine 90° derece döndürün. PNG resminin sol yanı yatay modda iPhone ekranının üstüne denk gelir. (Uygulamanın başlangıç yönünü ayarlama hakkında daha fazla bilgi edinmek için bkz. “iPhone uygulama ayarları” sayfa 14.)

Tam ekran olmayan bir uygulama için varsayılan görüntünün resminin üstteki 20 pikseli göz ardı edilir. iPhone durum çubuğunu varsayılan görüntünün en üstünde 20 piksel genişliğindeki dikdörtgenin üzerinde görüntüler. Yatay yönlü bir uygulamada, bu bölge Default.png dosyasının (yatay modda en üstte görüntülenen dosya) sol tarafta bulunan 20 piksel genişliğindeki dikdörtgenine karşılık gelir. Dikey yönlendirme uygulamasında, bu bölge Default.png dosyasının 20 piksel genişliğindeki dikdörtgenidir.

Çoğu uygulama için, Default.png görüntüsü uygulamanın başlangıç ekranı ile eşleşmelidir. Uygulamanızın başlangıç ekranının bir ekran görüntüsünü almak için:

- 1 Uygulamanızı iPhone aygıtında açın. Kullanıcı arabiriminin ilk ekranı görüldüğünde, Home (Ana Sayfa) düğmesini (ekranın altında) basılı tutun. Home (Ana Sayfa) düğmesini basılı tutarken Power/Sleep (Güç/Uyku) düğmesine basın (aygıt menüsünün en üstünde). Bu bir ekran görüntüsü alır ve bu görüntüyü Film Rulosu'na gönderir.
- 2 iPhoto veya başka bir fotoğraf aktarma uygulamasından fotoğraf aktararak görüntüyü geliştirme bilgisayarınıza aktarın. (Mac OS'ta Image Capture (Resim Yakalama) uygulamasını da kullanabilirsiniz.)

Ayrıca fotoğrafı e-posta ile geliştirme bilgisayarınıza gönderebilirsiniz:

- Fotoğraflar uygulamasını açın.
- Kamera Kaydını açın.
- Yakaladığınız ekran görüntüsü resmini açın.
- Resmi tıklatın ve ardından sol alt köşedeki “ileri” (ok) düğmesini tıklatın. Sonra Email Photo (Fotoğrafi E-posta ile Gönder) seçeneğini tıklatarak resmi kendinize gönderin.

Not: Doğru boyutlarda olduğu sürece, Default.png dosyası için istediğiniz resmi oluşturabilirsiniz. Ancak, çoğunlukla ideal olan Default.png resmini uygulamanın ilk haliyle eşleştirmektir.

Uygulamanız birden çok dile yerleştirilmişse Default.png görüntüsüne metin eklemeyin. Default.png sabittir ve metin, diğer diller ile eşleşmez.

Flash Professional CS5'te, Default.png dosyasını iPhone ayarları iletişim kutusundaki Dahil Edilen Dosyalar listesine eklemeyi unutmayın. Bkz. “Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama” sayfa 14

Komut satırındaki PFI uygulamasını kullanarak derleme yaparken, dahil edilen varlıklar listesinde bu dosyaya başvurduğunuzdan emin olun. Bkz. “Komut satırından bir iPhone uygulama yükleyici dosyası oluşturma” sayfa 20.

iPhone uygulama ayarları

Uygulama ayarlarına şunlar dahildir:

- Uygulama adı
- IPA dosyası adı
- Uygulamanın sürümü
- Uygulamanın başlangıç ekranı yönlendirmesi ve iPhone döndürüldüğünde ekran yönlendirmesinin otomatik olarak döndürülüp döndürülmediği
- Başlangıç görünümünün tam ekran olup olmadığı
- Uygulamanın simgeleri hakkında bilgi
- Donanım hızlandırma hakkında bilgi

Uygulama ayarlarını Flash Professional CS5'te düzenleyebilirsiniz.

Bu ayarları uygulama açıklayıcı dosyasından da ayarlayabilirsiniz. Uygulama açıklayıcı dosyası, uygulama ayarlarını içeren bir XML dosyasıdır.

Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama

Flash Professional CS5 iPhone Ayarları iletişim kutusunda iPhone uygulamasının birçok temel özelliğini tanımlayabilirsiniz.

iPhone Ayarları iletişim kutusunu açmak için:

- ❖ Dosya > iPhone Ayarları seçimini yapın.

Genel sekmesi

Genel sekmesi iPhone ile ilgili aşağıdaki ayarları içerir:

- Çıktı adı—Uygulamanın iPhone uygulama simgesi altında gösterilen adı. Çıktı dosya adına bir artı işareti (+) karakteri dahil etmeyin.
- Uygulama adı—Uygulamanın iPhone uygulama simgesi altında gösterilen adı. Uygulama adına bir artı işareti (+) karakteri dahil etmeyin.
- Sürüm—Kullanıcıların uygulamanın hangi sürümünü yüklediklerini belirlemelerine yardımcı olur. Bu sürüm iPhone uygulamasının CFBundleVersion sürümü olarak kullanılır. n 0 ile 9 arasında bir rakamı ve parantezler isteğe bağlı bileşenleri ifade etmek üzere nnnnn[.nn[.nn]] gibi bir biçimde olmalıdır; örneğin, 1, 1.0 veya 1.0.1. iPhone sürümleri yalnızca basamak ve ondalık noktaları içermelidir. iPhone sürümleri en fazla iki ondalık noktası içerebilir.
- Boyut oranı—Uygulamanın başlangıçtaki boyut oranı (dikey veya yatay).
- Tam ekran—Uygulamanın tam ekran kullanması veya iPhone durum çubuğunu göstermesi.
- Otomatik yönlendirme—iPhone yeniden yönlendirildiğinde uygulamanın görüntüleme içeriklerinin yeniden yönlendirilmesini sağlamak için bu uygulamayı seçin.

Otomatik yönlendirme kullanırken en iyi sonucu almak için Sahne Alanı'nın `align` özelliğini şu şekilde ayarlamak için ActionScript kodu ekleyin:

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

- Görüntüleme—Ekran nesnelерinin iPhone aygıtında nasıl görüntülendiğidir:
 - CPU—Uygulama tüm ekran objelerini görüntülemek için CPU kullanır. Donanım hızlandırma kullanılmaz.
 - GPU—Uygulama bitmap'leri birleştirmek için iPhone GPU kullanır.
 - Otomatik—Bu özellik uygulamaya konmamıştır.Daha fazla bilgi edinmek için bkz. “Donanım hızlandırma” sayfa 35.
- Dahil edilen dosyalar—Tüm dosya ve dizinleri iPhone uygulamasındaki pakete ekleme. Ana SWF dosyası ve uygulama açıklayıcısı dosyası varsayılan olarak dahildir. Gerekli olan diğer varlıkları Dahil Olan Dosyalar listesine ekleyin. Başlangıç ekranı resim dosyasını (Default.png) Dahil Edilen Dosyalar listesine eklemeyi unutmayın.

Konuşlandırma sekmesi

Konuşlandırma sekmesi, uygulamanın imzalama ve derleme ayarlarını içerir:

- iPhone dijital imza—Bir P12 sertifika dosyası ve sertifika için şifre belirtin. Apple iPhone sertifikasını .p12 biçimine dönüştürmelisiniz. Daha fazla bilgi edinmek için bkz. “Apple geliştirici dosyalarını edinme” sayfa 4.
- Ön hazırlık dosyası—Bu uygulama için Apple'dan edindiğiniz ön hazırlık dosyasını gösterin. Daha fazla bilgi edinmek için bkz. “Apple geliştirici dosyalarını edinme” sayfa 4.
- Uygulama kimliği—Uygulama kimliği uygulamanızı benzersiz bir şekilde tanımlar. Temel hazırlık dosyası belirli bir uygulama kimliğine bağlıysa, Flash Professional CS5 bu alanı ayarlar ve siz düzenleyemezsiniz. Aksi takdirde, temel hazırlık profili birden fazla (joker) uygulamaya kimliğine izin verir. Apple'a verdiğiniz uygulama kimliği joker deseniyle eşleşen bir uygulama kimliği verin.
 - Apple uygulama kimliğinizin com.etkiAlanım.* ise, iPhone Ayarları iletişim kutusundaki uygulama kimliği com.etkiAlanım ile başlamalıdır. (com.etkiAlanım.uygulamam veya com.etkiAlanım.app22 gibi).
 - Apple uygulama kimliğiniz * ise, iPhone Ayarları iletişim kutusundaki Uygulama Kimliği geçerli karakterlerden oluşan herhangi bir dize olabilir.

Bir temel hazırlık profiliyle ilişkili Apple uygulama kimliğini (veya joker uygulama kimliği desenini) iPhone Dev Center'da bulabilirsiniz (<http://developer.apple.com/iphone>). iPhone Geliştirici Programı Portalı'na gidin ve ardından Temel Hazırlık bölümüne gidin.

Önemli: Apple uygulama kimliğinin önündeki karakterleri göz ardı edin. Apple bu dizeyi Paket Çekirdeği Kimliği olarak adlandırır. Örneğin, Apple uygulama kimliğinizi 96LPVWEASL.com.ornek.bob.uygulamam olarak listelerse, 96LPVWEASL kısmını göz ardı edip uygulama kimliği olarak com.ornek.bob.uygulamam kısmını kullanın. Apple uygulama kimliğinizi 5RM86Z4DJM.* olarak listelerse, 5RM86Z4DJM kısmını göz ardı edin—bu bir joker uygulama kimliğidir.

- iPhone konuşlandırma tipi:
 - Aygıt testi için hızlı yayınlama—Uygulamanın geliştirici iPhone aygıtında test edilecek bir sürümünü çabucak derlemek için bu seçeneği işaretleyin.
 - Aygıt hata ayıklaması için hızlı yayınlama—Uygulamanın geliştirici iPhone aygıtında test edilecek bir hata ayıklama sürümünü çabucak derlemek için bu seçeneği işaretleyin. Bu seçenekle Flash Professional CS5 hata ayıklayıcı, iPhone uygulamasından izleme() çıktısı alabilir. (Bkz. “Bir iPhone uygulamasında hata ayıklama” sayfa 23.)
 - Konuşlandırma - Geçici—Geçici olarak konuşlandırılacak bir uygulama oluşturmak için bunu seçin. Bkz. Apple iPhone geliştirici merkezi
 - Konuşlandırma - Apple Uygulama Deposu—IPA dosyasının Apple Uygulama Deposuna konuşlandırılacak son sürümünü oluşturmak için bunu seçin.

Simgeler sekmesi

Simgeler sekmesinde 29 x 29 piksel simge görüntüsü, 48 x 48 piksel simge görüntüsü, 57 x 57 piksel simge görüntüsü, 72 x 72 piksel simge görüntüsü ve 512 x 512 piksel simge görüntüsü. Bkz. [“iPhone simgesi ve başlangıç ekran görüntüleri”](#) sayfa 12.

Not: 48 x 48 piksel ve 72 x 72 piksel seçenekleri, Flash Professional CS5'e dahil olan Packager for iPhone önizlemede yoktur. Bu seçenekleri eklemek için Flash Professional'da Yardım > Güncellemeler'e gidin.

Uygulama açıklayıcı dosyasında iPhone uygulama özelliklerini ayarlama

Uygulama açıklayıcı dosyası, adı, sürümü, telif hakkı ve diğer ayarlar gibi uygulamanın bütününün özelliklerini içeren bir XML dosyasıdır.

Flash Professional CS5, iPhone Ayarlar iletişim kutusundaki ayarlara göre bir uygulama açıklayıcı dosyası oluşturur. Ancak, uygulama açıklayıcı dosyasını bir metin düzenleyicide de düzenleyebilirsiniz. Flash Professional uygulama açıklayıcı dosyasını proje adına “-app.xml” ekleyerek adlandırır. Örneğin, HelloWorld projesinin uygulama açıklayıcı dosyası HelloWorld-app.xml'dir. Flash Professional CS5 iPhone Ayarlar iletişim kutusunda desteklenmeyen ayarlar tanımlamak isterseniz uygulama açıklayıcı dosyasını düzenleyin. Örneğin, uygulamanın info.Plist ayarlarını tanımlamak için InfoAdditions ögesini tanımlayabilirsiniz.

Önemli: Flash Professional CS5 iletişim kutusu açıkken uygulama açıklayıcı dosyasını düzenlemeyin. iPhone Ayarları iletişim kutusunu açmadan önce, değişiklikleri uygulama açıklayıcı dosyasına kaydedin.

Bu örnek bir uygulama tanımlayıcı dosyasıdır:

```
<?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>
```

Bunlar da uygulama açıklayıcı dosyasındaki ayarlara ilişkin ayrıntılardır:

- <application> ögesinde iPhone uygulamaları oluşturmak için AIR 2.0 ad alanı gerekir:

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

- <id> ögesi:

<id>com.ornek.as3.HelloWorld</id> Uygulama kimliği uygulamanızı diğerlerinden ayırır. Tavsiye edilen biçim, "com.company.AppName" gibi nokta sınırlı ve ters DNS stili olan bir dizedir. Derleyici bu değeri iPhone uygulamasının paket kimliği olarak kullanır.

Temel hazırlık dosyası belirli bir uygulama kimliğine bağlıysa, o uygulama kimliğini bu ögede kullanın. Apple'ın Apple uygulama kimliğinin (paket çekirdeği kimliği olarak da bilinir) başına atadığı karakterleri göz ardı edin. Örneğin, temel hazırlık profilinin uygulama kimliği 96LPVWEASL.com.ornek.bob.uygulamam ise, uygulama açıklayıcısı dosyasında uygulama kimliği olarak com.ornek.bob.uygulamam ögesini kullanın.

Temel hazırlık profili birden fazla (joker) uygulama kimliğinin kullanılmasına izin veriyorsa, uygulama kimliği yıldız işareti ile biter (5RM86Z4DJM.* gibi). Apple'a verdiğiniz uygulama kimliği joker deseniyle eşleşen bir uygulama kimliği sağlayın.

- Apple uygulama kimliğinizin com.etkiAlanim.* ise, uygulama açıklayıcısı dosyasındaki uygulama kimliği com.etkiAlanim ile başlamalıdır. com.etkiAlanim.uygulamam veya com.etkiAlanim.app22 gibi bir uygulama kimliği belirtebilirsiniz.
- Apple uygulama kimliğiniz * ise, uygulama açıklayıcısı dosyasındaki Uygulama Kimliği geçerli karakterlerden oluşan herhangi bir dize olabilir.

Bir temel hazırlık profiliyle ilişkili Apple uygulama kimliğini (veya joker uygulama kimliği desenini) iPhone Dev Center'da bulabilirsiniz (<http://developer.apple.com/iphone>). iPhone Geliştirici Programı Portalı'na gidin ve ardından Temel Hazırlık bölümüne gidin.

Önemli: Apple uygulama kimliğinin önündeki karakterleri göz ardı edin. Apple bu dizeyi Paket Çekirdeği Kimliği olarak adlandırır. Örneğin, Apple uygulama kimliğinizi 5RM86Z4DJM.* olarak listeliyorsa 5RM86Z4DJM kısmını göz ardı edin—bu bir joker uygulama kimliğidir. Apple uygulama kimliğinizi 96LPVWEASL.com.example.bob.uygulamam olarak listeliyorsa, 96LPVWEASL kısmını göz ardı edin—uygulama kimliği olarak com.example.bob.uygulamam kısmını kullanın.

- <filename> ögesi:

<filename>HelloWorld</filename> iPhone yükleyici dosyası için kullanılan ad. Dosya adına bir artı işareti (+) karakteri dahil etmeyin.

- <name> ögesi:

<name>Hello World</name> Uygulamanın iTunes uygulamasında ve iPhone aygıtında gösterilen adı. Ada bir artı işareti (+) karakteri dahil etmeyin.

- <version> ögesi:

<version>1.0</version> Kullanıcıların uygulamanızın hangi sürümünü yüklediklerini belirlemelerine yardımcı olur. Bu sürüm iPhone uygulamasının CFBundleVersion sürümü olarak kullanılır. n 0 ile 9 arasında bir rakamı ve parantezler isteğe bağlı bileşenleri ifade etmek üzere nnnnn[.nn[.nn]] gibi bir biçimde olmalıdır; örneğin, 1, 1.0 veya 1.0.1. iPhone sürümleri yalnızca basamak ve ondalık noktaları içermelidir. iPhone sürümleri en fazla iki ondalık noktası içerebilir.

- <initialWindow> ögesi uygulamanın başlangıç görünümüne ait özellikleri belirten şu alt öğeleri içerir:

<content>HelloWorld.swf</content> iPhone uygulamasına derlenecek kök SWF dosyasını tanımlar.

<visible>true</visible> Bu zorunlu bir ayardır.

`<fullScreen>true</fullScreen>` Uygulamanın iPhone aygıtının tüm ekranını kullandığını belirtir.

`<aspectRatio>portrait</aspectRatio>` Uygulamanın ilk boyut oranının dikey (yatay yerine) olduğunu belirtir). Bu ayar ne olursa olsun, uygulamanın başlangıç penceresini tanımlamada kullanılan Default.png dosyası 320 piksel genişlik ve 480 piksel yüksekliğe sahip olmalıdır. (Bkz. “iPhone simgesi ve başlangıç ekran görüntüleri” sayfa 12.)

`<autoOrients>true</autoOrients>` (İsteğe bağlı) Aygıt kendi fiziksel yönlendirmesini değiştirdiğinde uygulamadaki içerik yönlendirmesinin de kendiliğinden değişip değişmediğini belirtir. Varsayılan olarak true değerindedir. Otomatik yönlendirmeyi bir `orientationChanging` olayının Sahne nesnesiyle gönderilen `preventDefault()` yöntemini kullanarak iptal edebilirsiniz. Daha fazla bilgi için bkz. [Ekran yönlendirmesini ayarlama ve tespit etme](#).

Otomatik yönlendirme kullanırken en iyi sonucu almak için Sahne Alanı'nın `align` özelliğini şu şekilde ayarlayın:

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

`<renderMode>gpu</renderMode>` (İsteğe bağlı) Uygulamanın kullandığı görüntüleme modu. Üç tip ayar mümkündür:

- `cpu`—Uygulama tüm ekran objelerini görüntülemek için CPU kullanır. Donanım hızlandırma kullanılmaz.
- `gpu`—Uygulama bitmap'leri birleştirmek için iPhone GPU kullanır.
- `otomatik`—Bu özellik uygulamaya konmamıştır.

Daha fazla bilgi edinmek için bkz. “Donanım hızlandırma” sayfa 35.

- `<profiles>` ögesi:

`<profiles>mobileDevice</profiles>` Uygulamanın mobil aygıt profiline derlenmesini sınırlandırır. Bu profil şu anda sadece iPhone uygulamalarını desteklemektedir. Üç adet profil desteklenmektedir:

- `desktop`—Bir masaüstü AIR uygulaması.
- `extendedDesktop`—Yerel işlem API desteğine sahip bir masaüstü AIR uygulaması.
- `mobileDevice`—Mobil aygıtlar için bir AIR uygulaması. Şu anda iPhone desteklenen tek mobil aygıttır.

Uygulamayı belli bir profile sınırlandırmak, diğer profillere derlenmesini önler. Herhangi bir profil belirtmezseniz, uygulamayı bu profillerden herhangi birine derleyebilirsiniz. `<profiles>` ögesinde her birini boşluklarla ayırarak listeleme yoluyla birden fazla profil belirleyebilirsiniz.

`mobileDevice` profilini desteklenen bir profil olarak eklemeyi unutmayın (ya da `<profiles>` ögesini boş bırakın).

- `<icon>` ögesi uygulama tarafından kullanılan simgeleri belirten şu alt öğeleri içerir:

`<image29x29>icons/icon29.png</image29x29>` Spotlight arama sonuçlarında bu resim kullanılır.

`<image48x48>icons/icon48.png</image48x48>` iPad'de Spotlight arama sonuçlarında bu resim kullanılır.

`<image57x57>icons/icon57.png</image57x57>` iPhone ve iPod Touch ana ekranında bu resim kullanılır.

`<image72x72>icons/icon72.png</image72x72>` iPad ana ekranında bu resim kullanılır.

`<image512x512>icons/icon512.png</image512x512>` iTunes uygulamasında bu resim kullanılır.

Packager for iPhone aracı, uygulama açıklayıcı dosyasında belirtilen 29, 57 ve 512 simgelerini kullanır. Araç bunları sırasıyla Icon-Small.png, Icon.png ve iTunesArtwork adlı dosyalara kopyalar. Bu kopyalamayı önlemek için bu dosyaları doğrudan paketleyebilirsiniz. Bu dosyaları uygulama açıklayıcı dosyasını içeren dizine yerleştirerek doğrudan paketleyin ve doğru adları ve yolları listeyin.

512 resmi yalnızca dahili testler içindir. Bir uygulamayı Apple'a gönderirken 512 resmini ayrıca gönderirsiniz. Bu dosya IPA'da yer almaz. 512 resminizi göndermeden önce iTunes'ta iyi görünmesini sağlamak için belirtin.

- <iPhone> ögesi iPhone aygıtına özel ayarları belirten aşağıdaki alt öğeleri içerir:

<InfoAdditions></InfoAdditions> uygulamanın Info.plist ayarları olarak kullanılacak tuş değeri çiftlerini belirten alt öğeler içerir:

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

Bu örnekte değerler uygulamanın durum çubuğu stilini belirlerken uygulamanın sürekli bir Wi-Fi erişimi gerektirmediğini ifade eder.

InfoAdditions ayarları bir CDATA etiketine eklenmiştir.

iPad desteği için UIDeviceFamily için tuş değeri ayarlarını ekleyin. UIDeviceFamily ayarı bir dize dizisidir. Her dize desteklenen aygıtları tanımlar. <string>1</string> ayarı iPhone ve iPod Touch için desteği tanımlar. <string>2</string> ayarı iPad için desteği tanımlar. <string>3</string> ayarı tvOS için desteği tanımlar. Bu dizelerden yalnızca birini belirtirseniz, yalnızca bu aygıt ailesi desteklenir. Örneğin, aşağıdaki ayar sınırları iPad'i destekler:

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

Aşağıdaki kümeler iki aygıt ailesini de destekler (iPhone/iPod Touch ve iPad):

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

Diğer Info.plist ayarları hakkında bilgi edinmek için bkz. Apple geliştirici belgeleri.

iPhone uygulaması yükleyici dosyasını (IPA) derleme

Bir ActionScript 3.0 uygulamasını IPA yükleyici dosyasına derlemek için Packager for iPhone kullanın.

Flash Professional CS5'te bulunan Packager for iPhone uygulamasını kullanarak bir iPhone uygulaması yükleyici dosyası oluşturma

Flash Professional CS5'e dahil olan Packager for iPhone uygulamasını kullanmak için:

- 1 Dosya > Yayınla seçeneğini belirleyin.
- 2 iPhone Ayarlar iletişim kutusunda mutlaka tüm ayarlara değer verin. Konuşlandırma sekmesinde doğru seçenekleri seçtiğinizden emin olun. Bkz. "[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)" sayfa 14
- 3 Yayınla düğmesini tıklatın.

iPhone Paketleyicisi iPhone uygulama yükleyici (IPA) dosyasını oluşturur. IPA dosyasını derlemek birkaç dakika sürebilir.

Packager for iPhone uygulamasını komut satırından da çalıştırabilirsiniz. Bkz. “[Komut satırından bir iPhone uygulama yükleyici dosyası oluşturma](#)” sayfa 20.

Komut satırından bir iPhone uygulama yükleyici dosyası oluşturma

iPhone Paketleyicisini komut satırından çalıştırabilirsiniz. iPhone Paketleyicisi SWF dosyası bayt kodunu ve diğer kaynak dosyalarını yerel bir iPhone uygulamasına dönüştürür.

1 Bir komut kabuğu veya terminal açarak iPhone uygulamasının proje klasörüne gidin.

2 Ardından aşağıdaki sözdizimini kullanarak IPA dosyasını oluşturmak için pfi aracını kullanın:

```
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
```

pfi uygulamasının tam yolunu dahil etmek için başvuruyu pfi olarak değiştirin. Pfi uygulaması, Flash Professional CS5 yükleme dizininin pfi/lib alt dizininde yüklüdür

Oluşturmak istediğiniz iPhone uygulaması türüne uygun olan -target seçeneğini belirleyin:

- -target ipa-test—Uygulamanın geliştirici iPhone aygıtınızda test edilecek sürümünü hızlı bir şekilde derlemek için bunu seçin.
- -target ipa-debug—Uygulamanın geliştirici iPhone aygıtınızda test edilecek bir hata ayıklama sürümünü hızlı bir şekilde derlemek için bunu seçin. Bu seçenekle, iPhone uygulamasından trace() çıktısı almak için bir hata ayıklama oturumu kullanabilirsiniz.

Hata ayıklayıcısını çalıştıran geliştirici bilgisayarın IP adresini belirtmek için aşağıdaki -connect seçeneklerinden (CONNECT_OPTIONS) birini kullanabilirsiniz:

- -connect—Uygulama, uygulamayı derlemek için kullanılan geliştirici bilgisayarıda bir hata ayıklama oturumuna bağlanmaya çalışır.
- -connect IP_ADDRESS—Uygulama belirtilen IP adresine sahip bilgisayardaki bir hata ayıklama oturumuna bağlanmaya çalışır. Örneğin:

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

- -connect IP_ADDRESS—Uygulama belirtilen ana makine adına sahip bilgisayardaki bir hata ayıklama oturumuna bağlanmaya çalışır. Örneğin:

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

Not: -connect seçeneği Flash Professional CS5'deki Packager for iPhone Önizlemesi'nde yoktur. Packager for iPhone'u Flash Professional CS5'te Yardım > Güncellemeler'i seçerek güncelleyin.

-connect seçeneği isteğe bağlıdır. Belirtilmezse, sonuç olarak ortaya çıkan hata ayıklama uygulaması ana makinedeki bir hata ayıklayıcıya bağlanmaya çalışmaz.

Bir hata ayıklama bağlantısı girişimi başarısız olursa, uygulama kullanıcıya hata ayıklama ana makinesinin IP adresini girmesini istediği bir iletişim kutusu sunar. Aygıt wifi'a bağlı değilse bir bağlantı girişimi başarısız olabilir. Aygıt bağlıysa ancak hata ayıklayan ana makinenin güvenlik duvarının arkasında değilse de oluşabilir.

Daha fazla bilgi için bkz. “[Bir iPhone uygulamasında hata ayıklama](#)” sayfa 23.

GPU görüntü oluşturma tanılama özelliğini etkinleştirmek için -renderingdiagnostics seçeneğini de ekleyebilirsiniz. Daha fazla bilgi için “[Bir iPhone uygulamasında hata ayıklama](#)” sayfa 23 bölümünde “GPU görüntü oluşturma tanılama ile hata ayıklama” kısmına bakın.

- `-hedef ipa-geçici`—Geçici olarak konuşlandırılacak bir uygulama oluşturmak için bunu seçin. Bkz. Apple iPhone geliştirici merkezi
- `-hedef ipa-uygulama-deposu`—IPA dosyasının Apple Uygulama Deposuna konuşlandırılacak son sürümünü oluşturmak için bunu seçin.

`PROFILE_PATH` ögesini uygulamanızın temel hazırlık dosyası yoluyla değiştirin. Ön hazırlık profilleri hakkında daha fazla bilgi edinmek için bkz. “[Apple geliştirici dosyalarını edinme](#)” sayfa 4.

`SIGNING_OPTIONS` ögesini iPhone geliştirici sertifikası ve şifresini belirtecek şekilde değiştirin. Aşağıdaki sözdizimini kullanın:

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

`P12_FILE_PATH` ögesini P12 sertifika dosyanızın yoluyla değiştirin. `PASSWORD` ögesini sertifika şifresiyle değiştirin. (Aşağıdaki örneği inceleyin.) P12 sertifika dosyası hakkında daha fazla bilgi için bkz. “[Bir geliştirici sertifikasını P12 dosyasına dönüştürme](#)” sayfa 7.

`UYGULAMA_AÇIKLAYICI`'yı uygulama açıklayıcı dosyasını belirtecek şekilde değiştirin.

`KAYNAK_DOSYALAR`'ı projeye dahil edilecek diğer varlıklardan önce projenin başlıca SWF dosyasını belirtecek şekilde değiştirin. Flash CS5'teki uygulama ayarları iletişim kutusunda veya özel bir uygulama açıklayıcı dosyasında tanımladığımız tüm simgelerin yollarını dahil edin. Ayrıca, başlangıç ekranı resim dosyasını, `Default.png`'yi de ekleyin.

Aşağıdaki örneğe bakın:

```
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"
```

Aşağıdakileri kullanarak bir `HelloWorld.ipa` dosyası derler:

- `dfb7VKL19` sertifika şifresini kullanan özel bir PKCS#12 sertifikası
- `HelloWorld-uygulama.xml` uygulama açıklayıcı dosyası
- Bir kaynak `HelloWorld.swf` dosyası
- Özel `Default.png` ve simge dosyaları

Pfi uygulaması uygulama açıklayıcı dosyası, SWF dosyasına ve diğer varlıklara göre uygulamayı bir IPA dosyası içerisine derler.

Mac OS'de aşağıdaki seçenekleri pfi komutuna ekleyerek anahtarlıkta depolanan sertifikayı kullanabilirsiniz.

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

`ALIAS_NAME` ögesini kullanmak istediğiniz sertifikanın takma adıyla değiştirin. Mac anahtarlığında depolanan bir sertifikayı işaret ettiğinizde, sertifika dosyası konumuna işaret etmek yerine diğer adını belirtirsiniz.

Bir iPhone uygulaması yükleme

Geliştirme uygulamanızı bir iPhone'a yüklemek için, temel hazırlık dosyası iPhone'a yüklenir ve ardından uygulama iPhone'a yüklenir.

Temel hazırlık profilini iPhone'a ekleme

Temel hazırlık profilini iPhone'a eklemek için:

- 1 iTunes'ta File (Dosya) > Add To Library (Kütüphaneye Ekle) seçimini yapın. Ardından, temel hazırlık profili dosyasını seçin (dosya uzantısı olarak mobileprovision uzantısına sahiptir).

iPhone'unuzun temel hazırlık profiline ekli olduğundan emin olun. Apple iPhone Dev Center temel hazırlık profillerini yönetebilirsiniz (<http://developer.apple.com/iphone/>). Sitenin iPhone Developer Program Portal (iPhone Geliştirici Programı Portalı) bölümüne gidin. Geliştirme uygulamanızın yüklenebileceği aygıtların listesini yönetmek için Aygıtlar bağımlı tıklatın. Temel hazırlık profillerinizi yönetmek için Temel Hazırlık bağımlı tıklatın.

- 2 iPhone'unuzu bilgisayara bağlayın ve senkronize edin.

Temel hazırlık profili almayla ilgili bilgi için bkz. “[Apple geliştirici dosyalarını edinme](#)” sayfa 4.

Uygulamayı yükleme

Geliştirme uygulaması diğer IPA dosyalarıyla aynı şekilde yüklenir:

- 1 Daha önce uygulamanın bir sürümünü yüklediyseniz, uygulamayı aygıtınızdan ve iTunes'taki uygulama listesinden silin.
- 2 Uygulamayı aşağıdaki yollardan birini kullanarak iTunes'a ekleyin:
 - Dosya (File) menüsünde (iTunes'ta), Add to Library (Kütüphaneye Ekle) komutunu seçin. Daha sonra IPA dosyasını seçin ve Open (Aç) düğmesini tıklatın.
 - IPA dosyasını çift tıklatın.
 - IPA dosyasını iTunes kütüphanesine sürükleyin.
- 3 iPhone aygıtını bilgisayarın USB bağlantı noktasına bağlayın.
- 4 iTunes'ta aygıtın Uygulama sekmesini kontrol edin ve uygulamanın yüklenecek uygulamalar listesinde seçili olduğundan emin olun.
- 5 iPhone'unuzu senkronize edin.

Uygulama yükleme sorunlarını giderme

Uygulamayı yüklemeye çalışırken iTunes hata görüntülense, aşağıdakileri kontrol edin:

- Aygıt kimliğinin temel hazırlık profiline ekli olduğundan emin olun.
- Temel hazırlık profilinin yüklü olduğundan emin olmak için, profili iTunes'a sürükleyebilir veya File (Dosya) > Add to Library (Kitaplığa Ekle) komutunu kullanabilirsiniz.

Ayrıca, uygulamanızın uygulama kimliğinin Apple uygulama kimliği ile eşleşip eşleşmediğini kontrol edin:

- Apple uygulama kimliğinizin com.etkiAlanım.* ise, uygulama açıklayıcısı dosyasındaki uygulama kimliği veya iPhone Ayarları iletişim kutusundaki Uygulama Kimliği com.etkiAlanım (com.etkiAlanım.burayaherhangibirifadegelebilir gibi) ile başlamalıdır.
- Apple uygulama kimliğiniz com.etkiAlanım.uygulamam ise, uygulama açıklayıcı dosyasındaki veya Flash Professional CS5 kullanıcı arabirimindeki uygulama kimliği com.etkiAlanım.uygulamam şeklinde olmalıdır.
- Apple uygulama kimliğiniz * ise, uygulama açıklayıcı dosyasındaki veya Flash Professional CS5 kullanıcı arabirimindeki uygulama kimliği herhangi bir öge olabilir.

Uygulamanın uygulama kimliği Flash Professional CS5'te iPhone Ayarları iletişim kutusunda veya uygulama açıklayıcı dosyasında ayarlanır.

Bir iPhone uygulamasında hata ayıklama

Uygulamanın hata ayıklamasını, uygulama ADL'de çalışırken geliştirme bilgisayarında yapabilirsiniz. Uygulamanın hata ayıklamasını iPhone'da da yapabilirsiniz.

iPhone'da desteklenmeyen bazı AIR işlevleri, ADL kullanarak uygulamayı test ederken hala kullanılabilir (geliştirme yapılan bilgisayarda). Masaüstünde içerik test ederken bu farklara dikkat edin. Daha fazla bilgi için bkz. "[Mobil aygıtlarda desteklenmeyen ActionScript 3.0 API'leri](#)" sayfa 27.

Geliştirme bilgisayarında uygulamanın hata ayıklamasını yapma

Flash Professional CS5 kullanarak geliştirme bilgisayarında uygulamanın hata ayıklamasını yapmak için:

- ❖ Hata Ayıkla > Film Hatalarını Ayıkla > AIR Hata Ayıklama Başlatıcısı'nda (Mobil) ögesini seçin.

Uygulamanın hata ayıklamasını komut satırından ADL'yi çağırarak da yapabilirsiniz. Söz dizimi şu şekildedir:

```
adl -profile mobileDevice appDescriptorFile
```

appDescriptorFile ögesini uygulama açıklayıcı dosyasına giden yol ile değiştirin.

-profile mobileDevice seçeneğini dahil ettiğinizden emin olun.

Uygulamanın hata ayıklamasını iPhone'da yapma

iPhone'da uygulama hatalarını ayıklamak için:

1 Uygulamayı hata ayıklama desteğiyle derleyin:

- Flash Professional CS5'te, "Aygıt hata ayıklaması için hızlı yayınlama" ayarını kullanarak derleyin. (Bkz. "[Flash Professional CS5'te bulunan Packager for iPhone uygulamasını kullanarak bir iPhone uygulaması yükleyici dosyası oluşturma](#)" sayfa 19.)
- PFI komut satırı uygulamasını kullanarak uygulamayı `target ipa-debug` seçeneğiyle derleyin. (Bkz. "[Komut satırından bir iPhone uygulama yükleyici dosyası oluşturma](#)" sayfa 20.)

2 Uygulamayı iPhone aygıtına yükleyin.

3 iPhone'da Wi-Fi'ı açın ve geliştirme bilgisayarıyla aynı ağa bağlanın.

4 Geliştirme bilgisayarınızda bir hata ayıklama oturumu başlatın. Flash Professional CS5'te Hata Ayıkla > Uzaktan Hata Ayıklama Oturumuna Başla > ActionScript 3.0 ögesini seçin.

5 Uygulamayı iPhone aygıtında çalıştırın.

Uygulamanın hata ayıklama sürümü, geliştirici bilgisayarın IP adresini soracaktır. IP adresini girin ve Tamam düğmesine dokunun. Geliştirme bilgisayarının IP adresini almak için.

- Mac OS'ta Apple menüsünde System Preference (Sistem Tercih) seçimini yapın. System Preference (Sistem Tercih) penceresinde Network (Ağ) simgesini tıklayın. Network (Ağ) tercihleri penceresinde IP adresleri listelenir.
- Windows'ta bir komut satırı oturumu başlatın ve `ipconfig` komutunu çalıştırın.

Hata ayıklama oturumu uygulamanın her türlü `trace()` çıktısını gösterir.

iPhone'da yüklü bir uygulamanın hatalarını ayıklarken, Flash Professional CS5, kesme noktası kontrolü, kod geçişi ve çeşitli izlemeler gibi tüm hata ayıklama özelliklerini destekler.

GPU görüntü oluşturma tanılmasıyla hata ayıklama

GPU görüntü oluşturma tanılama özelliği uygulamanın donanım hızlandırmasını nasıl kullandığını görmenize izin verir (GPU görüntü oluşturma modunu kullanan uygulamalar için) Bu özelliği kullanmak için, komut satırında PFI aracını kullanarak uygulamayı derleyin ve `-renderingdiagnostics` seçeneğini dahil edin:

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

`-renderingdiagnostics` bayrağı doğrudan `-package` bayrağını takip etmelidir.

GPU görüntü oluşturma tanılama özelliği tüm görüntüleme nesneleri için renklendirilmiş dikdörtgenleri görüntüler:

- Mavi—Görüntüleme nesnesi bitmap değil veya bitmap olarak önbelleğe alınmamış ve oluşturuluyor.

Değişmeyen bir görüntüleme nesnesi için sürekli mavi görüntüleniyorsa, bunun sebebi hareket eden görüntü nesneleriyle kesişmesi olabilir. Örneğin, görüntüleme nesnesi hareket eden görüntüleme nesneleri için arka plan olabilir. Görüntüleme nesnesini bitmap olarak önbelleğe almayı göz önünde bulundurun.

Önbelleğe alınması gerektiğini düşündüğünüz bir nesne için mavi görüntülenirse, bunun sebebi nesnenin GPU'nun uygulayamayacağı bir efekt kullanıyor olması olabilir. Bu efektler belirli renk karışımlarını, renk dönüştürmelerini, `scrollRect` özelliğini ve maskeleri içerir.

Ayrıca, GPU'ya yüklenen görüntüleme nesneleri bellek sınırlarını aşarsa uygulama mavi görüntüler.

Uygulama her mavi dikdörtgen için günlüğe mesaj kaydeder. Uygulama bu mesajların çıktısını diğer `trace()` ve hata ayıklama çıktı mesajlarıyla birlikte alır.

- Yeşil—Görüntüleme nesnesi bitmap'tir veya bitmap olarak önbelleğe alınmıştır ve GPU'ya ilk kez yükleniyordur.

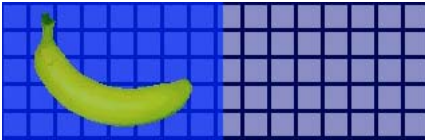
Görüntüleme nesnesi için sürekli yeşil görüntüleniyorsa, uygulama kod görüntüleme nesnesini yeniden oluşturuyordur. Örneğin, bu durum zaman çizelgesi görüntüleme nesnesini oluşturan bir kareye dönüyorsa oluşabilir. Aynı nesnelerin yeniden oluşmasını engellemek için içeriği değiştirmeyi düşünün.

- Kırmızı—Görüntüleme nesnesi bir bitmap'tir veya bitmap olarak önbelleğe alınmıştır ve GPU'ya yeniden yükleniyordur.

Bu tür bir görüntüleme nesnesi, uygulamanın bitmap temsilini yeniden oluşturmasını gerektirecek şekilde değiştiğinde kırmızı görüntülenir. Örneğin, bir 2B nesnenin `cacheAsBitmapMatrix` özelliği ayarlı değilse, ölçeklendiğinde veya döndürüldüğünde yeniden oluşturulur. Yeniden oluşturma ayrıca alt görüntüleme nesneleri taşındığında veya değiştiğinde de oluşur.

Boyamanın döngüler sırasında yeniden oluşmaması durumunda her renkli dikdörtgen, dört ekran yeniden çizme döngüsünden sonra solar. Ancak, ekranda hiçbir değişiklik yoksa, tanılama renklendirmesi değişmez.

Örneğin, bitmap olarak önbelleğe alınmamış bir vektör arka planının önündeki bir bitmap görüntüleme nesnesini (bir muz) düşünün. Muz ilk oluşturulduğunda yeşil renktedir. Arka plan ilk oluşturulduğunda mavi renktedir:

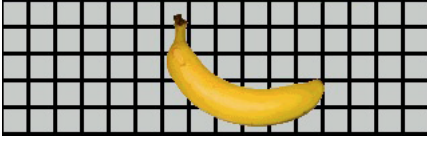


Muz hareket ettikçe, işlemci mavi gölgelemenin arka plan üzerinde görünmesine neden olarak arka planı yeniden oluşturmalıdır:



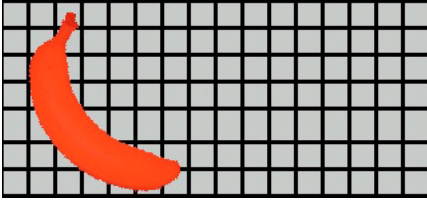
Arka plan üzerindeki mavi gölgeleme GPU'ya gönderilmesi gereken yeniden çizilmiş bölgeleri yansıtır.

Ancak, arka plan bitmap olarak önbelleğe alınmışsa, muz hareket ettiğinde oluşturma tanılama özelliği renkli renk tonlarını görüntüleyemez:



Tanılama özelliği GPU arka plan bitmap'ini koruduğundan, renkli renk tonu görüntüleyemez. GPU, işlemciye ihtiyacı olmadan muzun arka planla birleştirebilir.

Muzun `cacheAsBitmapMatrix` özelliği ayarlı olmayan bir 2B nesne olduğunu varsayın. Görüntüleme nesnesi döndürüldüğünde (veya ölçeklendiğinde), oluşturma tanılama özelliği kırmızı görüntüler. Bu, uygulamanın GPU'ya görüntüleme nesnesinin yeni bir sürümünü yüklemesi gerektiğini gösterir:



iPhone uygulamanızı App Store'a gönderme

Uygulamanızı App Store'a göndermek için:

- 1 iPhone Dev Center web sitesinden bir dağıtım sertifikası ve temel hazırlık profili edinin (<http://developer.apple.com/iphone/>).

Bir dağıtım sertifikası, "iPhone Developer: XXX.cer" olarak adlandırılır. Burada XXX sizin adınızdır.

Daha fazla bilgi edinmek için bkz. "[Apple geliştirici dosyalarını edinme](#)" sayfa 4.

- 2 Dağıtım sertifikasını bir P12 dosyasına dönüştürün.

Daha fazla bilgi için bkz. "[Bir geliştirici sertifikasını P12 dosyasına dönüştürme](#)" sayfa 7.

- 3 P12 ve temel hazırlık profili kullanarak uygulamanızı derleyin.

Dağıtım sertifikasına dayanan P12 dosyasını kullanın. Dağıtım temel hazırlık profiliyle ilişkilendirilmiş uygulama kimliğini kullanın.

Daha fazla bilgi için bkz. "[iPhone uygulaması yükleyici dosyasını \(IPA\) derleme](#)" sayfa 19.

- 4 Uygulamayı iPhone Dev Center web sitesine gönderin (<http://developer.apple.com/iphone/>).

Önemli: Apple, App Store'a uygulamaları yüklemek için Apple Application Loader (Uygulama Yükleyici) programını kullanmanızı gerektirir. Apple Application Loader (Uygulama Yükleyici) programını yalnızca Mac OS X için ayarlar. Windows bilgisayar kullanarak iPhone için bir AIR uygulaması geliştirebilirsiniz bile, uygulamayı App Store'a göndermek için OS X (sürüm 10.5.3 veya üzeri) çalıştıran bir bilgisayara erişiminizin olması gerekir. Application Loader (Uygulama Yükleyici) programını Apple iOS Dev Center'dan alabilirsiniz.

Bölüm 3: Mobil aygıtlar için ActionScript 3.0 API desteği

Mobil AIR uygulamaları oluştururken, Flash Player 10.1 ve AIR 2 masaüstü uygulamalarını geliştirmek için mevcut olan ActionScript 3.0 API'lerinin aynısını kullanabilirsiniz. Ancak istisnalar ve eklemeler mevcuttur.

Mobil aygıtlarda desteklenmeyen ActionScript 3.0 API'leri

Mobil aygıt profilinde çalışan uygulamalar (iPhone'da çalışan uygulamalar gibi) için desteklenmeyen bazı ActionScript 3.0 API'leri

Birden fazla profil (masaüstü ve mobil gibi) geliştirmek için aynı ActionScript kodunu kullanırken, kodu API'nin desteklenip desteklenmediğini test etmek için kullanın. Örneğin, iPhone uygulamalarında NativeWindow sınıfı desteklenmez. (iPhone uygulamaları yerel pencere kullanamaz veya oluşturamaz.) Bir uygulamanın yerel pencereleri destekleyen bir profile çalışıp çalışmadığını test etmek için (masaüstü profili gibi), `NativeWindow.isSupported` özelliğini kontrol edin.

Aşağıdaki tabloda mobil aygıt profilinde desteklenmeyen API'ler listelenmektedir. Tabloda aynı zamanda bir uygulamanın API desteği sunan bir platformda ne zaman çalışacağını belirlemek için kontrol edeceğiniz özellikler de listelenmektedir.

API	Destek testi
Erişilebilirlik	Capabilities.hasAccessibility
Camera	Camera.isSupported
DatagramSocket	DatagramSocket.isSupported
DNSResolver	DNSResolver.isSupported
DockIcon	NativeApplication.supportsDockIcon
DRMManager	DRMManager.isSupported
EncryptedLocalStore	EncryptedLocalStore.isSupported
HTMLLoader	HTMLLoader.isSupported
LocalConnection	LocalConnection.isSupported
Microphone	Microphone.isSupported
NativeApplication.exit()	—
NativeApplication.menu	NativeApplication.supportsMenu
NativeApplication.isSetAsDefaultApplication()	NativeApplication.supportsDefaultApplication
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported

API	Destek testi
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
PDF desteği	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

Mobil aygıt profili için HTML ve JavaScript temelli AIR uygulamaları yazamazsınız.

Bazı ActionScript 3.0 sınıfları sadece kısmen desteklenir:

File

iPhone uygulamalarının sadece uygulama dizinine ve uygulama depolama dizinine erişimi vardır. Aynı zamanda `File.createTempFile()` ve `File.createTempDirectory()` işlemlerini kullanabilirsiniz. Başka bir dizine erişmek için bir işlemi (bir `FileStream` okuma veya yazma yöntemi gibi) kullanmak `IOError` istisnasıyla sonuçlanır.

iPhone uygulamaları `File.browseForOpen()` yöntemiyle sağlananlar gibi yerel dosya tarayıcı iletişim kutularını desteklemez.

Yükleyici

Bir iPhone uygulamasında `Loader.load` yöntemini kullanamazsınız. Ancak, `Loaded.load` yöntemiyle yüklenmiş SWF içeriğinde hiçbir ActionScript kodunu çalıştıramazsınız. Ancak, SWF dosyasındaki varlıkları kullanabilirsiniz (örn. kütüphanedeki film klipleri, görüntüler, fontlar ve sesler). Ayrıca görüntü dosyalarını yüklemek için de `Loader.load` yöntemini kullanabilirsiniz.

Video

iPhone'daki bir AIR uygulamasında sadece Sorensen video ve ON2 VP6 video desteklenir.

Uygulamanın dışında bir H.264 video açmak için `navigateToURL()` yöntemini kullanabilirsiniz. `request` parametresi olarak videoya işaret eden bir URL içeren `URLRequest` nesnesi geçin. Video, iPhone aygıtının video oynatıcısında başlar.

Metin alanları

iPhone'daki fontlar ve diğer metin alanı ayarları için kısıtlamalar bulunmaktadır. Bkz. “[Fontlar ve metin girdisi](#)” sayfa 38.

Desteklenmeyen API'ler ve ADL kullanarak hata ayıklama

iPhone'da desteklenmeyen bazı AIR işlevleri, ADL kullanarak uygulamayı test ederken hala kullanılabilir (geliştirme yapılan bilgisayarda). ADL kullanarak içerik test ederken bu farklara dikkat edin.

Bu işlevler, aşağıdaki video ve ses kod çözücülerini içerir: Speex (ses), H.264/AVC (video) ve ACC (ses). Bu codec'ler iPhone'da çalışan AIR uygulamalarında kullanılamaz. Ancak, masaüstünde normal olarak çalışmaya devam ederler.

Ulaşılabilirlik ve ekran okuyucu desteği Windows'taki ADL'de çalışır. Ancak bu API'ler iPhone'da desteklenmez.

RTMPE protokolü, masaüstündeki ADL'den kullanıldığında normal olarak çalışır. Ancak, RTMPE protokolü kullanarak bağlanmaya çalışan bir NetConnection ögesi iPhone'da bunu başaramaz.

İçerik ADL ile çalıştırıldığında Loader sınıfı ek kısıtlamalar olmadan çalışır. Ancak iPhone'da çalıştırılırken ActionScript bit kodu içeren SWF içeriğinin yüklenmeye çalışılması hata mesajıyla sonuçlanır.

Gölgelendirici, örnekleri ADL'de çalıştırılır. Ancak iPhone Pixel Bender'da bit kodu yorumlanmaz ve gölgelerin grafik efekti olmaz.

Daha fazla bilgi için bkz. "[Bir iPhone uygulamasında hata ayıklama](#)" sayfa 23.

Mobil AIR uygulamalarına özgü ActionScript API'leri

Aşağıdaki API'ler yalnızca mobil aygıtlardaki AIR uygulamalarında mevcuttur. Şu anda AIR masaüstü sürümlerinde veya Flash Player'da kullanılamamaktadır.

Ekran yönlendirme API'si

Ekran yönlendirme API'si sahne ve iPhone yönlendirmesiyle çalışmanıza imkan verir:

- `Stage.autoOrients`—Uygulamanın, aygıt döndürüldüğünde sahne alanını otomatik olarak yeniden yönlendirecek şekilde ayarlanıp ayarlanmayacağı. Flash Professional CS5 iPhone Ayarlar iletişim kutusunda Otomatik Yönlendirme seçeneği seçili olduğunda özellik `true` olarak ayarlanır. (Uygulama açıklayıcı dosyasında `autoOrients` ögesini `true` olarak da ayarlayabilirsiniz.) Bkz. "[iPhone uygulama ayarları](#)" sayfa 14. Stage nesnesi için bir `orientationChanging` olay dinleyicisi ekleyerek otomatik yeniden yönlendirmeyi iptal edebilirsiniz. Bu olay nesnesinin `preventDefault()` yöntemi çağrıldığında, otomatik yeniden yönlendirme iptal edilir.

Otomatik yönlendirme kullanırken en iyi sonucu almak için Sahne Alanı'nın `align` özelliğini şu şekilde ayarlayın:

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

- `Stage.deviceOrientation`—Aygıtın fiziksel yönlendirmesi. `StageOrientation` sınıfı, bu özelliğin değerlerini tanımlar.
- `Stage.orientation`—Sahne alanının geçerli yönlendirmesi. `StageOrientation` sınıfı, bu özelliğin değerlerini tanımlar.
- `Stage.supportsOrientationChange`—iPhone'da `true`, AIR uygulamasında ise `false` olarak ayarlanır.
- `Stage.setOrientation()`—Sahne alanının yönlendirmesini ayarlar. Bu yöntemin, yeni sahne alanının yönünü tanımlayan bir dize olan tek bir parametresi vardır. `StageOrientation` sınıfındaki sabitler parametrenin muhtemel değerlerini tanımlar.
- `StageOrientation`—Sahne alanı yönlendirme değerlerini tanımlar. Örneğin, `StageOrientation.ROTATED_RIGHT` aygıtın varsayılan yönlendirmesine göre sağa döndürülen bir sahneyi ifade eder.

- StageOrientationEvent—Ekran yönlendirmesi değiştiğinde sahne alanının gönderdiği olayları tanımlar. Bu olay kullanıcı iPhone aygıtını döndürdüğünde meydana gelir. İki tür olay vardır. Aygıt döndürüldüğünde Sahne alanı orientationChanging olayını gönderir. Sahne alanının yeniden yönlendirilmesini önlemek için, orientationChanging olay nesnesinin preventDefault() yöntemini çağırın. Sahne alanının yeniden yönlendirmesi tamamlandığında Sahne alanı bir kez daha orientationChange olayını gönderir.

Ekran yönlendirme API'si şu anda sadece mobil aygıtlardaki AIR uygulamalarında kullanılabilir. Bir mobil AIR uygulaması ve masaüstü AIR uygulaması kaynak kodu paylaşıyorsa, API'nin desteklenip desteklenmediğini kontrol etmek için Stage.supportsOrientationChange özelliğini kullanın.

Aşağıdaki örnekte, aygıtın kullanıcı tarafından döndürülmesine nasıl yanıt verileceği gösterilmektedir:

```
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;
    }
}
```

Bu örnekte, farklı sahne yönlendirmeleri durumunda işlevsel kod yerine yorumlar mevcuttur.

Sahne alanının yönlendirmesini Stage nesnesinin setOrientation yöntemini çağırarak değiştirebilirsiniz. Yönlendirmenin ayarlanması eş zamanlı olmayan bir işlemdir. orientationChange olayını dinleyerek yönlendirmenin ne zaman tamamlanacağını kontrol edebilirsiniz. Aşağıdaki kod, sahne alanının sağ el yönlendirmesine göre nasıl ayarlanacağını gösterir:

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

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

Sahne döndükçe yeniden boyutlandırılır ve Stage nesnesi bir resize olayı gönderir. resize olayına yanıt olarak Sahnede display nesnelerini yeniden boyutlandırabilir ve konumlandırabilirsiniz.

NativeApplication.systemIdleMode ve SystemIdleMode

NativeApplication.systemIdleMode özelliği iPhone'un boşa moduna geçmesini engellemize izin verir. Varsayılan olarak, bir iPhone belli bir zaman boyunca dokunma ekranı etkileşimi olmazsa boşa moduna geçer. Boşa modu ekranın kararmasına neden olabilir. Ayrıca iPhone'un kilit moduna geçmesine de neden olabilir. Bu özellik iki değerden birine ayarlanabilir:

- SystemIdleMode.NORMAL—iPhone normal boşa modu davranışını sergiler.
- SystemIdleMode.KEEP_AWAKE—Uygulama iPhone'un boşa moduna girmesini önlemeye çalışır.

Bu işlev yalnızca mobil aygıtlarda desteklenir. Masaüstü işletim sistemi çalıştıran AIR uygulamalarında desteklenmez. Masaüstünde çalışan bir uygulamada, NativeApplication.systemIdleMode özelliğini ayarlamanın bir etkisi yoktur.

Aşağıdaki kod iPhone boşa modunun nasıl devre dışı bırakıldığını gösterir:

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

CameraRoll

CameraRoll sınıfı iPhone film rulosuna görüntü eklemenize izin verir. addBitmapData() yöntemi iPhone film rulosuna bir görüntü ekler. Yöntemin tek bir parametresi vardır, bitmapData. Bu parametre film rulosuna eklenecek görüntüyü içeren BitmapData nesnesidir.

CameraRoll işlevi yalnızca mobil aygıtlarda desteklenir. Masaüstü işletim sistemi çalıştıran AIR uygulamalarında desteklenmez. Uygulamanızın CamerRoll işlevini çalışma zamanında destekleyip desteklemediğini kontrol etmek için, statik CameraRoll.supportsAddBitmapData özelliğini kontrol edin.

addBitmapData() yöntemini çağırdıktan sonra, CameraRoll şu iki olaydan birini gönderir:

- complete—İşlem başarıyla tamamlandı.
- error—Bir hata oluştu. Örneğin, iPhone'da görüntüyü depolayacak kadar boş alan kalmamış olabilir.

Aşağıdaki kod film rulosuna sahne alanının (ekran yakalaması) bir görüntüsünü ekler:

```
if (CameraRoll.supportsAddBitmapData)
{
    var cameraRoll:CameraRoll = new CameraRoll();
    cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);
    cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);
    var bitmapData:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight);
    bitmapData.draw(stage);
    cameraRoll.addBitmapData(bitmapData);
}
else
{
    trace("not supported.");
}

function onCrError(event:ErrorEvent):void
{
    // Notify user.
}

function onCrComplete(event:Event):void
{
    // Notify user.
}
```

DisplayObject.cacheAsBitmapMatrix

`cacheAsBitmapMatrix` özelliği, `cacheAsBitmaptrue` olarak ayarlandığında bir görüntüleme nesnesinin nasıl oluşturulduğunu tanımlayan bir Matrix nesnesidir. Bu uygulama görüntüleme nesnesinin bitmap sürümünü oluştururken bu matrisi dönüştürme matrisi olarak kullanır.

`cacheAsBitmapMatrix` ayarlı olduğunda, uygulama görüntüleme matrisi yerine, bu matrisi kullanarak oluşturulmuş önbelleğe alınmış bitmap görüntüsünü korur. (Görüntüleme matrisi, görüntüleme nesnesinin `transform.concatenatedMatrix` ögesinin değeridir.) Matris görüntüleme matrisiyle eşleşmezse, bitmap gerektiği şekilde ölçeklenir ve döndürülür.

`cacheAsBitmapMatrix` olarak ayarlı bir görüntüleme nesnesi yalnızca `cacheAsBitmapMatrix` ögesinin değeri değiştiğinde oluşturulur. Bitmap, görüntüleme matrisine uymak için uygun olan şekilde ölçeklenir veya döndürülür.

Hem işlemci hem de GPU tabanlı görüntü oluşturma, GPU görüntü oluşturma genellikle daha fazla olsa da, `cacheAsBitmapMatrix` özelliğinin kullanımından faydalanır.

Not: Donanım hızlandırmayı kullanmak için, Flash Professional CS5'teki iPhone Ayarları iletişim kutusunun Genel sekmesinde Görüntü Oluşturma'yı GPU'ya ayarlayın. (Ya da `renderMode` özelliğini uygulama açıklayıcısı dosyasında `gpu` olarak ayarlayın.)

Örneğin, aşağıdaki kod görüntüleme nesnesinin dönüştürülmemiş bitmap temsili kullanır:

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

Aşağıdaki kod geçerli görüntü oluşturma ile eşleşen bir bitmap temsili kullanır:

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

Genellikle, kimlik matrisi (`new Matrix()`) veya `transform.concatenatedMatrix` yeterlidir. Ancak, GPU'ya farklı bir bitmap yüklemek için, küçültülmüş matris gibi farklı bir matris kullanabilirsiniz. Örneğin, aşağıdaki örnek x ve y eksenlerinde 0,5 küçültülmüş bir `cacheAsBitmapMatrix` matrisini uygular. GPU'nun kullandığı bitmap nesnesi daha küçüktür, ancak GPU onun boyutunu görüntüleme nesnesinin `transform.matrix` özelliğiyle eşleşmesi için ayarlar:

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

Genelde, görüntüleme nesnesini uygulamada görüntülenecek boyuta dönüştüren bir matris seçin. Örneğin, uygulamanız boyutu yarıya indirilmiş hareketli grafiğin bitmap sürümünü görüntülüyorsa, boyutu yarıya düşüren bir matris kullanın. Uygulamanız hareketli grafiği geçerli boyutlarından daha büyük görüntüleyecekse, o çarpanla ölçüyü arttıran bir matris kullanın.

`cacheAsBitmapMatrix` özelliğinin ayarlı olduğu görüntüleme nesnelere için boyut açısından gerçekçi bir sınır vardır. Sınır 1020 x 1020 pikseldir. `cacheAsBitmapMatrix` özelliğinin ayarlı olduğu tüm görüntüleme nesnelere toplam piksel sayısı için gerçekçi bir sınır vardır. Sınır yaklaşık dört milyon pikseldir.

`cacheAsBitmapMatrix` ögesini ve donanım hızlandırmayı kullanırken bir çok etken vardır. Hangi görüntüleme nesnelere o özellik kümesine sahip olması gerektiğini ve hangilerinin sahip olmaması gerektiğini bilmek önemlidir. Bu özelliği kullanmayla ilgili önemli bilgiler için bkz. “Donanım hızlandırma” sayfa 35.

Uygulamanızın hata ayıklama yapılarında GPU kullanımını tanılamak için GPU görüntü oluşturma tanılama özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. “Bir iPhone uygulamasında hata ayıklama” sayfa 23.

Ağ iletişimi notları

`navigateToURL()` işleviyle aşağıdaki URL şemalarının kullanılması belgenin harici bir uygulamada açılmasına neden olur:

URL şeması	<code>nativeToURL()</code> ögesine yapılan çağırının sonucu	Örnek
mailto:	Posta uygulamasında yeni bir mesaj açar.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	Metin mesajı uygulamasında yeni bir mesaj açar.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	Telefondaki bir telefon numarasını çevirir (kullanıcının onayı ile).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Bir iPhone uygulaması, https isteği gibi güvenli bir alışveriş sırasında sunucu kimliği doğrulama için yüklü kişisel imzalı kök sertifikaya dayanabilir. Bir sunucu yalnızca yaprak sertifikasını değil, kök sertifikaya zincirleme şekilde bağlı bulunan bütün orta düzey sertifikaları göndermelidir.

Mobil uygulama geliştiricilerini özellikle ilgilendiren ActionScript 3.0 API'leri

Aşağıdaki ActionScript 3.0 API'leri mobil aygıtlarda kullanılabilen işlevselliği tanımlar.

Hız Ölçer API'si

Aşağıdaki sınıflar uygulamanın aygıt hız ölçerinden olayları almasına izin verir:

- Accelerometer
- AccelerometerEvent

Daha fazla bilgi için bkz. [Akselerometre girdisi](#).

Coğrafi Konum API'si

Aşağıdaki sınıflar uygulamanın aygıt konum sensöründen olayları almasına izin verir:

- Geolocation
- GeolocationEvent

Daha fazla bilgi için bkz. [Coğrafi Konum](#).

Dokunma, çoklu dokunma ve hareket API'si

Aşağıdaki sınıflar uygulamanın dokunma ve mimik olaylarını almasına izin verir:

- GestureEvent
- GesturePhase
- MultiTouch

- MultitouchInputMode
- TouchEvent
- TransformGestureEvent

Daha fazla bilgi için bkz. [Dokunma, çoklu dokunma ve hareket girdisi](#).

Bölüm 4: iPhone uygulama tasarımı

iPhone işlem hızı ve ekran boyutu özel tasarım ve kodlamaya elverişlidir. Ancak, tasarım konularının birçoğu tüm uygulamaların veya mobil uygulamaların geliştirilmesinde yaygındır.

Uygulamaları en iyi duruma getirme ile ilgili daha fazla bilgi için bkz. [Flash Platform için Mobil İçeriği En İyileştirme](#). Bu belgede mobil içeriğinin, Flash Player içeriğinin, AIR içeriğinin ve genel olarak ActionScript tabanlı içeriğin performansının en iyileştirilmesine yönelik çok sayıda öneri bulunur. Bu önerilerin birçoğu iPhone için AIR uygulamalarına da yöneliktir.

Önemli: Tasarım düşüncelerinin ve en iyi duruma getirme tekniklerinin bir çoğu iPhone'da düzgün bir şekilde çalışacak uygulamalar geliştirmek için önemlidir.

Donanım hızlandırma

Bazı uygulamalarda grafik performansını iyileştirmek için OpenGL ES 1.1 donanım hızlandırmayı kullanabilirsiniz. Oyunlar ve görüntüleme nesnelerinin animasyonunun yapıldığı diğer uygulamalar donanım hızlandırma özelliğinden faydalanabilir. Donanım hızlandırmayı kullanan uygulamalar bazı grafik işlemlerini işlemciden iPhone GPU'ya yükleyerek performansı büyük ölçüde arttırabilir.

GPU'yu kullanmak için bir uygulama tasarlarken, içeriğin etkili bir şekilde GPU hızlandırmalı olduğundan emin olmak için kuralları takip etmek önemlidir.

Donanım hızlandırmayı kullanmak için, Flash Professional CS5'teki iPhone Ayarları iletişim kutusunun Genel sekmesinde Görüntü Oluşturma'yı GPU'ya ayarlayın. Uygulama açıklayıcı dosyasında `renderMode` ögesini `gpu` olarak da ayarlayabilirsiniz:

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

“[Flash Professional CS5'te iPhone uygulama özelliklerini ayarlama](#)” sayfa 14 ve “[Uygulama açıklayıcı dosyasında iPhone uygulama özelliklerini ayarlama](#)” sayfa 16 bölümlerine bakın.

İçerikleri seyrek olarak değişirse donanım hızlandırmada hızlı bir şekilde oluşturulabilecek dört görüntüleme nesnesi sınıfı vardır:

- Bitmap nesneleri
- `cacheAsBitmap` özelliğinin `true` olarak ayarlandığı ve isteğe bağlı olarak `cacheAsBitmapMatrix` özelliği ayarlı olan 2B Görüntüleme nesneleri (aşağıya bakın)
- 3B Görüntüleme nesneleri (örn. `z` özelliği ayarlı)
- Tek bir tek renkli dörtgen dolguya ve ekrandaki piksellerle hizalanan kenarlara sahip ekran nesneleri.

Vektöre dayalı nesnelere üstlerinde veya altlarında başka bir hareketli grafik hareket ettiğinde yeniden oluşur. Bu nedenle, bir animasyonun perde arkası veya ön planı olarak görev yapan her nesne aynı zamanda bu kategorilerden birine ait olmalıdır.

`cacheAsBitmap` öğeleri `true` olarak ayarlanmış ekran nesneleri için, `cacheAsBitmapMatrix` öğesini ayarlamak GPU'nun matris dönüşümü sonucu ortaya çıkan bitmap'i kullanmasına neden olur. Nesne döndürülse veya ölçeklense bile GPU bitmap temsilini kullanır. GPU bu bitmap'i, işlemcinin bir vektöre dayalı nesneyi yeniden çizmesinden çok daha hızlı bir şekilde oluşturur ve hareketlendirir.

`cacheAsBitmap` özelliğini tek başına `true` olarak ayarlamak görüntü nesnesinin (ve alt öğelerinin) önbelleğe alınmasına neden olur. Yeni bölgeler ortaya çıktığında veya tüm birleşik grafik çevrildiğinde görüntüleme nesnesi yeniden çizilmez.

Bir ekran nesnesinin `cacheAsBitmapMatrix` özelliği ayarlandığında uygulama, ekran nesnesi görünür olmadığında bile ekran nesnesinin bir temsilini oluşturabilir. Uygulama, bir sonraki karenin başlangıcında ekran nesnesinin önbelleğe alınmış temsilini oluşturur. Ardından ekran nesnesini sahne alanına eklediğinizde, uygulama bunu hızlı bir şekilde oluşturur. Ayrıca, uygulama hızlı bir şekilde nesneyi hareketlendirebilir, döndürebilir veya ölçekleyebilir. Döndürülen veya ölçeklenen nesnelere için, `cacheAsBitmap` özelliğini, `cacheAsBitmapMatrix` özelliğini de ayarlamadan ayarlamayın.

Ayrıca uygulama bitmap olarak önbelleğe alınan bir nesnede hızlı bir şekilde alfa dönüşümleri gerçekleştirebilir. Ancak, donanım hızlandırmalı alfa dönüşümleri için yalnızca 0 ve 1.0 arasındaki `alpha` değerleri desteklenir. Bu 0 ile 256 arasındaki bir `colorTransform.alphaMultiplier` ayarına karşılık gelir.

Metin alanları gibi sık bir şekilde güncellenen nesnelere `cacheAsBitmap` özelliğini `true` olarak ayarlamayın.

Sık bir şekilde değişen grafik içeriğine sahip görüntüleme nesnelere genellikle GPU oluşturma için zayıf adaylardır. Bu özellikle daha az güce sahip GPU'ları olan eski aygıtlar için geçerlidir. Grafikleri GPU'ya yüklemenin neden olduğu ek yük işlemci görüntü oluşturmalarını daha iyi bir seçenek haline getirebilir.

Üst nesneyle daha ilgili olan alt görüntüleme nesnelere içeren görüntüleme nesnelere yeniden yapılandırın. Alt görüntüleme nesnelere üst görüntüleme nesnelere aynı seviyeye gelmelerini sağlamak için değiştirin. Bu hepsinin kendi bitmap temsiline sahip olacağını garantiler. Ayrıca, her görüntüleme nesnesi, GPU'ya yeni grafiklerin yüklenmesine gerek kalmadan diğerlerine göre hareket edebilir.

Alt öğe ekran nesnelere hareketli olmadığı en yüksek ekran listesi seviyesinde `cacheAsBitmap` özelliğini `true` olarak ayarlayın. Diğer bir deyişle, özelliği hareketli parça içermeyen ekran nesnesi kaplarına ayarlayın. Alt öğe ekran nesnelere ayarlamayın. Diğer hareketli ekran nesnelere içeren hareketli grafikler için *ayarlamayın*.

Bir görüntüleme nesnesinin `z` özelliğini ayarladığınızda, uygulama her zaman önbelleğe alınmış bir bitmap temsili kullanır. Ayrıca bir görüntüleme nesnesinin `z` özelliğini ayarladıktan sonra uygulama, siz nesneyi döndürseniz veya ölçekleseniz bile önbelleğe alınmış bitmap temsilini kullanır. Uygulama ayarlanmış bir `z` özelliği olan görüntüleme nesnelere için `cacheAsBitmapMatrix` özelliğini kullanmaz. Üç boyutlu görüntüleme nesnesi özelliklerini, `rotationX`, `rotationY`, `rotationZ`, ve `transform.matrix3D` özellikleri de dahil olmak üzere ayarladığınızda aynı kurallar geçerlidir.

Görüntüleme nesnesi kabı içeriğini donanım hızlandırma için kullanacaksanız, `scrollRect` veya `mask` özelliklerini ayarlamayın. Bu özellikleri ayarlamak, görüntüleme nesnesi kabının ve alt nesnelere donanım hızlandırmasını devre dışı bırakır. `mask` özelliğini ayarlamaya bir alternatif olarak, maskelenen `display` nesnesinin üzerinde bir maske görüntüleme nesnesi katmanlayın.

Donanım hızlandırma için kullanılacak görüntüleme nesnelere boyutlarının sınırları vardır. Daha eski aygıtlarda, sınır hem genişlik hem yükseklikte 1024 piksel veya daha azdır. Daha yeni aygıtlarda, sınır 2048 piksel veya daha azdır. Bir aygıtın performansını ölçmek için GPU görüntü oluşturma teşhis aracını kullanabilirsiniz.

GPU, bitmap görüntülerini saklamak için iPhone RAM'ini de kullanır. En az bitmap görüntülerinin boyutu kadar bellek kullanır.

GPU, bitmap görüntüsünün her boyutu için 2'nin katlarında bellek ayırma kullanır. Örneğin, GPU 512 x 1024 veya 8 x 32 boyutlarında bellek ayırabilir. Böylelikle 9 x 15 piksel görüntü, bellekte, 16 x 16 piksel görüntü kadar yer kaplar. Önbellege alınmış görüntüleme nesneleri için, her yönde 2'nin katlarına yakın (ancak daha fazla değil) boyut kullanmak isteyebilirsiniz. Örneğin, 33 x 17 piksel yerine 32 x 16 piksel görüntüleme nesnesi kullanmak daha verimlidir.

Öteki platformlar için (masaüstü vb.) boyutlandırılmış varlıkları daha küçük olarak ölçeklendirmek için yeniden boyutlandırılmış bir Sahne Alanı'na bağlı kalmayın. Bunun yerine `cacheAsBitmapMatrix` özelliğini kullanın veya varlıkları iPhone için yayınlamadan önce yeniden boyutlandırın. 3B nesnelere yüzey görüntülerini ön belleğe alırken `cacheAsBitmapMatrix` özelliğini yok sayar. Bu nedenle görüntü nesneleri 3B yüzeyde oluşturulacaksa, bunları yayınlamadan önce yeniden boyutlandırmak daha iyidir.

Donanım hızlandırma ve RAM kullanımı arasında bir denge bulunmaktadır. Bellek doldukça iPhone işletim sistemi, çalışmakta olan diğer yerel iPhone uygulamalarını bellek boşaltmaları için uyarır. Bu uygulamalar uyarıyı işler ve belleği boşaltmaya başlarken, işlemci döngüleri için uygulamanızın önüne geçmeye çalışabilir. Bu durum uygulamanızın performansını bir an için düşürebilir. Uygulamanızı, çalışan işlemlerinizi için göreceli olarak daha az bellek bulunduracaklarından, eski aygıtlarda test etmeyi unutmayın.

iPhone'da uygulamada hata ayıklarken, GPU oluşturma tanılama özelliğini etkinleştirebilirsiniz. Bu özellik uygulamanızın GPU oluşturmaya nasıl kullandığını gözlemlemenize yardımcı olur. Daha fazla bilgi için "[Bir iPhone uygulamasında hata ayıklama](#)" sayfa 23 bölümünde "GPU görüntü oluşturma tanılama ile hata ayıklama" kısmına bakın.

`cacheAsBitmapMatrix` özelliğini kullanmayla ilgili daha fazla bilgi için bkz. "[Mobil AIR uygulamalarına özgü ActionScript API'leri](#)" sayfa 29 bölümünde "`DisplayObject.cacheAsBitmapMatrix`".

Ekran nesnesi performansını arttırmanın diğer yolları

Donanım hızlandırma bazı ekran nesnesi sınıflarında grafik performansını hızlandırabilir. Aşağıda grafik performansının nasıl en üst seviyeye çıkarılacağıyla ilgili birkaç ipucu verilmektedir:

- Sahnede görünen öğelerin sayısını sınırlandırmayı deneyin. Her öğenin görüntülenmesi ve etrafındaki diğer öğelerle birleştirilmesi biraz zaman alır.
Bir ekran nesnesini göstermenize gerek kalmazsa, `visible` özelliğini `false` olarak ayarlayın veya sahneden kaldırın (`removeChild()`). `alpha` özelliğini 0 olarak ayarlamayın.
- Genel olarak karışım modlarından, özellikle de katman karışımı modlarından sakının. Mümkün oldukça normal karışım modunu kullanın.
- Ekran nesnesi filtreleri sayısal olarak pahalıdır. Bunları tutumlu kullanın. Örneğin, bir giriş ekranında birkaç tane filtre kullanmak kabul edilebilir. Ancak, birçok nesnede veya oynatılmakta olan nesnelere ya da yüksek bir kare hızı kullanmak zorunda olduğunuzda filtre kullanmayın.
- Dönüşen şekillerden kaçının.
- Kırpma kullanmayın.
- Mümkünse, `Graphic.beginBitmapFill()` yöntemini kullanırken `repeat` parametresini `false` olarak ayarlayın.
- Fazladan çizim yapmayın. Arka plan olarak arka plan rengini kullanın. Büyük şekilleri birbirinin üstüne katmanlandırmayın. Çizilen her piksel yük oluşturur. Bu özellikle donanım hızlandırmalı görüntüleme nesneleri için geçerlidir.

- Uzun ve ince çivileri, birbirini kesen kenarları veya kenarlarında çok sayıda ince detayı bulunan şekillerden kaçının. Bu şekillerin görüntülenmesi düz kenarlı ekran nesnelere göre daha uzun sürer. Bu özellikle donanım hızlandırmalı görüntüleme nesnelere için geçerlidir.
- Bitmap'leri $2^n \times 2^m$ bite yakın fakat bundan az olmayacak boyutlarda yapın. Boyutlar 2'nin kuvvetleri olmak zorunda değilse de, daha büyük olmamak koşuluyla 2'nin kuvvetlerine yakın olmalıdır. Örneğin, 31x15 piksellik bir resim 33x17 piksellik bir resimden daha hızlı görüntülenir. (31 ve 15 2'nin katları olan 32 ve 16'dan daha küçüktür.) Bu tür görüntüler belleği de etkili bir biçimde kullanır.
- Görüntüleme nesnesinin boyutlarını 1024 x 1024 piksel olarak sınırlayın (veya daha yeni aygıtlarda 2048 x 2048)

Bilgi yoğunluğu

Mobil aygıtların fiziksel boyutu masaüstündekinden küçük olsa da, piksel yoğunluğu daha yüksektir. Keskin metne bakmak daha güzeldir, fakat gliflerin okunaklı olması için asgari bir fiziksel boyutu olmalıdır.

Mobil aygıtlar çoğunlukla hareket halinde ve zayıf aydınlatma koşulları altında kullanılır. Ekranda ne kadar bilgiyi gerçekçi bir şekilde okunaklı olarak görüntüleyebileceğinizi düşünün. Bir masaüstünde aynı piksel boyutlarına sahip bir ekranda görüntüleyebileceğinizden daha az olacaktır.

Önemli bilgileri vurgulamak için tipografik hiyerarşi kullanın. Kullanıcı arabirimi öğelerinin göreceli önemini göstermek için font boyutu, ağırlık, yerleşim ve aralık kullanın. Her hiyerarşi düzeyinde bir veya daha fazla ipucu kullanabilirsiniz. Bu ipuçlarını uygulamanızda tutarlı bir şekilde uygulayın. Bir işaret uzamsal (girinti, satır aralığı, yerleşim) veya grafiksel (boyut, stil, yazı karakteri rengi) olabilir. Gereksiz işaretler kullanmak, hiyerarşinin net bir şekilde yansıtıldığından emin olmanın etkili bir yoludur. Ancak, bir grup düzeyinde üçten fazla işaret kullanmamaya çalışın.

Etiketleri ve gereken açıklayıcı metinleri basitleştirmeye çalışın. Örneğin, metin alanında içeriği önerecek bir örnek girdi kullanın ve ayrı bir etiketten kaçının.

Fontlar ve metin girdisi

En iyi görünüm için aygıt fontlarını kullanın. Örneğin, aşağıdaki fontlar iPhone'daki aygıt fontlarıdır:

- Serif: Times New Roman, Georgia ve _serif
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma ve _sans
- Sabit genişlik: Courier New, Courier ve _typewriter

14 piksel veya daha büyük fontlar kullanın.

Düzenlenebilir metin alanları için aygıt fontlarını kullanın. Metin alanlarındaki aygıt fontları gömülü fontlardan da daha hızlı görüntülenir.

Girdi metin alanlarında altı çizili metin kullanmayın. Ayrıca metin alanının hizalamasını ayarlamayın. iPhone'daki metin alanları yalnızca sola hizalamayı (varsayılan) destekler.

Flash Professional CS5'teki bir metin alanında TLF Metni ayarı kullanıyorsanız, ActionScript 3.0 Ayarları'ndaki varsayılan bağlantıda çalışma zamanı paylaşılan kütüphaneyi kapatın. Aksi takdirde, uygulama çalışma zamanı paylaşılan kütüphanenin SWF dosyasını kullanmaya çalışacağından iPhone aygıtında çalışmaz:

1 Dosya > Yayınlama Ayarları'nı seçin.

- 2 Yayınlama Ayarları iletişim kutusunda Flash sekmesini tıklatın.
- 3 Komut Dosyası (ActionScript 3.0) açılır listesinin sağındaki Komut Dosyası düğmesini tıklatın.
- 4 Kütüphane Yolu sekmesini tıklatın.
- 5 Varsayılan Bağlantı açılır listesinde Kod İçine Birleştirilmiş seçimini yapın.

Giriş metni alanlarını kullanmanın alternatiflerini uygulamayı düşünün. Örneğin, kullanıcının sayısal bir değer girmesini sağlamak için bir metin alanı gerekmez. Değeri arttırmak veya azaltmak için iki düğme verebilirsiniz.

Sanal klavyenin kullanacağı alanı unutmayın. Sanal klavye etkinleştirildiğinde (örneğin, kullanıcı bir metin alanını tıklattığında), uygulama sahnenin konumunu ayarlar. Otomatik yeniden konumlandırma, seçili giriş metni alanının görünür olmasını sağlar:

- Sahnenin üstündeki metin alanı, görünür sahne alanının üstüne gider. (Görünür sahne alanı sanal klavyeyi barındırmak için küçüktür.)
- Sahnenin altındaki metin alanı ise yeni sahne alanının altında kalır.
- Sahnenin başka bir kısmındaki metin alanı ise sahnenin dikey merkezine taşınır.

Kullanıcı bir metin alanını düzenlemek için üzerini tıklattığında (ve sanal klavye görüntülediğinde), TextField nesnesi bir `focusIn` olayı gönderir. Metin alanını yeniden konumlandırmak üzere bu olay için bir olay dinleyicisi ekleyebilirsiniz.

Kullanıcı metni düzenlerken, tek satırlık metin alanına bir temizle düğmesi dahildir (metnin sağında). Ancak bu temizle düğmesi metin alanı çok darsa görüntülenmez.

Kullanıcı, tek satırlık metin alanında metin düzenledikten sonra, sanal klavyenin üzerindeki Bitti düğmesine basıp klavyeyi bırakır.

Kullanıcı, çok satırlı metin alanında metin düzenledikten sonra, sanal klavyeyi metin alanının dışına basarak bırakır. Bu, odağı metin alanından kaldırır. Sanal klavye görüntülediğinde, tasarımınıza metin alanının dışındaki alanın dahil olduğundan emin olun. Metin alanı fazla büyük olduğunda başka bir alan görünmeyebilir.

Bazı Flash Professional CS5 bileşenlerini kullanmak metin alanından odağı kaldırmanızı önleyebilir. Bu bileşenler, odak davranışının istendiği masaüstü makinelerde kullanılmak üzere tasarlanmıştır. Bu bileşenlerden biri TextArea bileşenidir. Odakta olduğunda (ve düzenlenirken), odağı başka bir görüntüleme nesnesini tıklatarak kaldıramazsınız. Sahne alanına bundan başka bazı Flash Professional CS5 bileşenlerini konumlandırmak da, odağı düzenlenen metin alanından başka bir yere değiştirmeyi önleyebilir.

Klavye olaylarına bağlı kalmayın. Örneğin, web için tasarlanmış bazı SWF içerikleri kullanıcının uygulamayı kontrol edebilmesi için klavyeyi kullanır. Ancak, iPhone aygıtında sanal klavye yalnızca kullanıcı bir metin alanını düzenlerken mevcuttur. Bir iPhone uygulaması klavye olaylarını ancak sanal klavye mevcutken gönderir.

Uygulama durumunu kaydetme

Uygulamadan dilediğiniz zaman çıkabilirsiniz (örneğin telefon çaldığında). Uygulamanızın durumunu her değiştirdiğinizde kaydetmeyi unutmayın. Örneğin, ayarları bir dosyaya veya bir veritabanını uygulama depolama dizinine kaydedebilirsiniz. Ayrıca, yerel olarak paylaşılan bir nesneye de veri kaydedebilirsiniz. Sonra yeniden başlatıldığında uygulamanızı eski durumuna döndürebilirsiniz. Bir telefon araması uygulamayı bölerse, uygulama arama bittiğinde yeniden başlar.

Uygulamadan çıkarken bir `çıkış` olayı gönderen `NativeApplication` özelliğine güvenmeyin; işe yaramayabilir.

Ekran yönlendirme değişiklikleri

iPhone içeriği dikey veya yatay yönde görüntülenebilir. Uygulamanızın ekran yönlendirme değişikliklerini nasıl işleme koyacağını düşünün. Daha fazla bilgi için bkz. [Aygıt yönlendirmeyi belirleme](#).

Vuruş hedefleri

Düğmeleri ve kullanıcının tıklattığı diğer kullanıcı arabirimi öğelerini tasarlarken vuruş hedeflerinin boyutunu düşünün. Bu öğeleri dokunmatik bir ekranın bir parmak ile rahatça etkinleştirilebileceği büyüklükte oluşturun. Ayrıca, hedefler arasında yeterince boşluk olduğundan emin olun. Vuruş hedefleri yaklaşık 44 piksel ile 57 piksel olmalıdır.

Bellek ayırma

Ferah bellek blokları ayırmak maliyetlidir. Uygulamayı yavaşlatabilir veya animasyon ya da etkileşim sırasında çöp toplama tetiklendiğinden performansın düşmesine neden olabilir.

Nesneleri atıp yenisini oluşturmak yerine her fırsatta geri dönüştürmeyi deneyin.

Vektör nesnelere dizilerden daha az bellek harcayabileceğini unutmayın. Bkz. [Vector sınıfı ve Array sınıfı karşılaştırması](#).

Bellek kullanımı hakkında daha fazla bilgi için bkz. [Belleği koruma](#).

Çizim API'si

Grafik oluşturmak için ActionScript çizim API'sini (Grafik sınıfı) kullanmamaya çalışın. Çizim API'si kullanmak nesnelere sahnedeki dinamik olarak oluşturur, ardından rasterleştiriciye dönüştürür. Mümkünse, bu nesnelere sahnedeki geliştirme zamanında durağan biçimde oluşturun.

Çizim API'si kullanılarak oluşturulan nesnelere tekrar tekrar kullanıldığında, ActionScript her çalıştırıldığında zarar görebilir ve yeniden oluşturulur. Ancak, statik nesnelere hafızada farklı zaman çizelgeleri içerisinde bulunur.

Olay köpürmesi

Derinlemesine yuvalanmış bir görüntüleme nesnesi kabı için olayların içten dışa tetiklenmesi fazla çaba gerektirebilir. Bu maliyeti olayı tamamen hedef nesnede işleyerek, ardından da olay nesnesinin `stopPropagation()` yöntemini kullanarak azaltın. Bu yöntemi kullanmak olayın köpürmesini engeller. Bu yöntemi kullanmak ana öğe nesnelere olayı almaması anlamına da gelir.

Uzun olay zincirlerini önlemek için ekran nesnesi yuvalanmasını düzleştirerek ilgili kazanımlar elde edebilirsiniz.

Mümkün olduğunda TouchEvent olayları yerine MouseEvent olaylarına başvurun. MouseEvent olayları, TouchEvent olaylarından daha az işlemci ek yükü kullanır.

Mümkün olduğunda `mouseEnabled` ve `mouseChildren` özelliklerini `false` olarak ayarlayın.

Video performansını en iyi hale getirme

Mobil video oynatımını en iyi hale getirmek için, video oynatılırken uygulamanızda daha az işlem olduğundan emin olun. Bu video çözümü ve oluşturma işlemlerinin mümkün oldukça çok işlemci kullanmasına izin verir.

Video oynatılırken çok az ActionScript kodu çalıştırın veya hiç çalıştırmayın. Sık aralıkla çalışan veya bir zaman çizelgesine göre çalışan kodlar çalıştırmaktan kaçının.

Video olmayan görüntüleme nesnelere yeniden çizimini en aza indirin. Özellikle video alanıyla kesişen görüntüleme nesnelere yeniden çizmekten kaçının. Videonun altına gizlenmiş olsalar bile bu durum geçerlidir. Yine de yeniden çizilirler ve işleme kaynaklarını kullanırlar. Örneğin, konum göstergesi için basit şekiller kullanın ve her karede güncellemek yerine, konum göstergesini saniyede bir kaç kere güncelleyin. Video kontrollerinin video alanıyla çakışmasına izin vermeyin; doğrudan video alanının altına koyun. Video arabelleğe alan bir animasyonunuz varsa, kullanılmadığı zamanlarda animasyonu videonun arkasına saklamayın; görünmez olarak ayarlayın.

Flex ve Flash bileşenleri

Birçok Flex ve Flash bileşeni masaüstü uygulamalarında kullanılmak için tasarlanmıştır. Bu bileşenler, özellikle görüntüleme bileşenleri, mobil aygıtlarda yavaş çalışabilir. Yavaş çalışmanın yanı sıra, masaüstü bileşenleri mobil aygıtlar için uygun olmayan etkileşim modellerine sahip olabilirler.

Adobe, Flex çerçevesinin mobil aygıtlar için en iyi duruma getirilmiş bir sürümünü geliştiriyor. Daha fazla bilgi için bkz. <http://labs.adobe.com/technologies/flex/mobile/>.

Uygulama dosyası boyutunu azaltma

Aşağıda IPA dosyanızın dosya boyutunu azaltabilmeniz için bir kaç ipucu bulunmaktadır:

- Boyutlarının doğru olup olmadığını (gerekenden daha geniş değil) görmek için arka plan bitmap'lerini kontrol edin.
- Herhangi bir ekstra fontun gömülüp gömülmediğini kontrol edin.
- Alfa kanalları için PNG varlıklarına bakın ve gereksizlerse kaldırın. PNG varlıklarının boyutunu azaltmak için PNG küçültücü gibi bir yardımcı program kullanın.
- Mümkünse, PNG varlıklarını JPG varlıklarına dönüştürün.
- Ses dosyalarını sıkıştırılmayı düşünün (daha düşük bit hızı kullanarak)
- Kullanılmayan varlıkları kaldırın.