

ACTIONSCRIPT® 3.0

Bileşenlerini Kullanma

Yasal bildirimler

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

İçindekiler

Bölüm 1: Giriş

Hedef kitle	1
Sistem gereksinimleri	1
Belgeler hakkında	1
Tipografik kurallar	2
Bu kılavuzda kullanılan terimler	2
Ek kaynaklar	2

Bölüm 2: ActionScript 3.0 Bileşenleri Hakkında

Bileşenleri kullanmanın avantajları	3
Bileşen türleri	4
Belgeye ekleme ve belgeden silme	6
Bileşenin sürümünü bulma	8
ActionScript 3.0 olay işleme modeli	8
Basit bir uygulama	9

Bölüm 3: Bileşenlerle Çalışma

Bileşen mimarisi	16
Bileşen dosyalarıyla çalışma	18
Bileşen uygulamalarının hatalarını ayıklama	20
Parametreleri ve özellikleri ayarlama	20
Kütüphane	21
Bileşenleri boyutlandırma	22
Canlı Önizleme	22
Olayları işleme	23
Görüntüleme listesiyle çalışma	24
FocusManager ile çalışma	26
List tabanlı bileşenler	28
DataProvider ile çalışma	28
CellRenderer ile çalışma	36
Bileşenleri erişilebilir duruma getirme	43

Bölüm 4: Kullanıcı Arabirimi Bileşenlerini Kullanma

Button bileşenini kullanma	44
CheckBox bileşenini kullanma	46
ColorPicker bileşenini kullanma	49
ComboBox bileşenini kullanma	52
DataGrid bileşenini kullanma	55
Label bileşenini kullanma	61
List bileşenini kullanma	63
NumericStepper bileşenini kullanma	67
ProgressBar bileşenini kullanma	70
RadioButton bileşenini kullanma	75

ScrollPane bileşenini kullanma	78
Slider bileşenini kullanma	81
TextArea bileşenini kullanma	84
TextInput bileşenini kullanma	87
TileList bileşenini kullanma	90
UILoader bileşenini kullanma	93
UIScrollBar bileşenini kullanma	95

Bölüm 5: Kullanıcı Arabirimi Bileşenlerini Özelleştirme

Kullanıcı arabirimi bileşeni özelleştirmesi hakkında	98
Stilleri ayarlama	98
Kaplamalar Hakkında	101
Button bileşenini özelleştirme	104
CheckBox bileşenini özelleştirme	106
ColorPicker bileşenini özelleştirme	107
ComboBox bileşenini özelleştirme	109
DataGrid bileşenini özelleştirme	111
Label bileşenini özelleştirme	116
List bileşenini özelleştirme	117
NumericStepper bileşenini özelleştirme	120
ProgressBar bileşenini özelleştirme	121
RadioButton bileşenini özelleştirme	123
ScrollPane bileşenini özelleştirme	125
Slider bileşenini özelleştirme	125
TextArea bileşenini özelleştirme	127
TextInput bileşenini özelleştirme	129
TileList bileşenini özelleştirme	130
UILoader bileşenini özelleştirme	132
UIScrollBar bileşenini özelleştirme	132

Bölüm 6: FLVPlayback Bileşenini kullanma

FLVPlayback bileşenini kullanma	135
FLVPlayback bileşenini özelleştirme	153
SMIL dosyasını kullanma	164

Bölüm 7: FLVPlayback Captioning Bileşenini Kullanma

FLVPlaybackCaptioning bileşenini kullanma	172
Timed Text resim yazılarını kullanma	174
Resim yazısı ile işaret noktalarını kullanma	180
Resim yazısı ile birden çok FLV dosyası oynatma	182
FLVPlaybackCaptioning bileşenini özelleştirme	182

Bölüm 1: Giriş

Adobe® Flash® CS5 Professional, çok etkili web deneyimleri üretilmesine yönelik standart bir geliştirme aracıdır. Bileşenler, bu deneyimleri sağlayan zengin Internet uygulamalarının yapıtaşlarıdır. *Bileşen* Adobe® ActionScript® yöntemleri, özellikleri ve olayları ile Flash'ta geliştirme sırasında veya çalışma zamanında bileşeni özelleştirmenizi sağlayan parametreleri içeren bir film klbidir. Bileşenler, ActionScript kullanmadan geliştiricilerin kodu yeniden kullanmasına ve paylaşmasına, tasarımcıların kullanıp özelleştirebileceği karmaşık işlevleri kapsüllemesine olanak sağlayacak şekilde tasarlanmıştır.

Bileşenler, hızlı ve kolayca tutarlı görünüm ve davranışa sahip güçlü uygulamalar oluşturmanızı sağlar. Bu kılavuz, Adobe ActionScript 3.0 bileşenleriyle uygulamaların nasıl oluşturulacağını açıklamaktadır. *Adobe® ActionScript® 3.0 Dil ve Bileşenler Başvurusu*'nda, her bileşenin uygulama programlama arabirimi (API) açıklanmaktadır.

Adobe® tarafından oluşturulan bileşenleri kullanabilir, diğer geliştiriciler tarafından oluşturulan bileşenleri indirebilir veya kendi bileşenlerinizi oluşturabilirsiniz.

Hedef kitle

Bu kılavuz, Flash uygulamaları oluşturan ve geliştirme işlemini hızlandıracak bileşenleri kullanmak isteyen geliştiriciler içindir. Flash'ta uygulamaları geliştirme ve ActionScript yazma konularında önceden bilgi sahibi olmanız gerekir.

ActionScript yazma konusunda fazla deneyiminiz yoksa, bir belgeye bileşenler ekleyebilir, Özellik denetçisinde veya Bileşen denetçisinde bunların parametrelerini ayarlayabilir ve bunların olaylarını işlemek için Davranışlar panelini kullanabilirsiniz. Örneğin, herhangi bir ActionScript kodu yazmaya gerek duymadan, düğme tıkladığında web tarayıcısında bir URL açan bir Düğme bileşenine Web Sayfasına Git davranışı ekleyebilirsiniz.

Daha sağlam uygulamalar oluşturmaya çalışan bir programcıysanız, bileşenleri dinamik olarak oluşturabilir, çalışma zamanında özellikleri ve çağrı yöntemlerini ayarlamak için ActionScript'i kullanabilir ve olayları işlemek için olay dinleyicisi modelini kullanabilirsiniz.

Daha fazla bilgi için, bkz. "[Bileşenlerle Çalışma](#)" sayfa 16.

Sistem gereksinimleri

Flash bileşenleri, Flash uygulamasının sistem gereksinimlerinden daha fazlasını gerektirmez.

Flash CS3 veya sonraki bileşenlerini kullanan herhangi bir SWF dosyasının Adobe® Flash® Player 9.0.28.0 veya sonrası ile görüntülenmesi ve ActionScript 3.0 için yayınlanması gerekir (Bunu Flash sekmesindeki Dosya > Yayınlama Ayarları ögesinden ayarlayabilirsiniz).

Belgeler hakkında

Bu belgede, Flash uygulamaları geliştirmek için bileşenlerin kullanımına ilişkin ayrıntılar açıklanmaktadır. Flash ve ActionScript 3.0 ile ilgili genel bilgiye sahip olduğunuz varsayılmaktadır. Flash ve ilgili ürünleri hakkında belirli belgeler ayrıca mevcuttur.

Bu belge, PDF dosyası ve çevrimiçi yardım olarak mevcuttur. Çevrimiçi yardımı görüntülemek için Flash uygulamasını başlatın ve Yardım > Flash Yardım > Adobe ActionScript 3.0 Bileşenlerini Kullanma seçeneklerini belirleyin.

Flash hakkında bilgi almak için şu belgelere bakın:

- *Flash Uygulamasını Kullanma*
- *ActionScript 3.0 Geliştirici Kılavuzu*
- *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*

Tipografik kurallar

Bu kılavuzda, aşağıdaki tipografi kuralları kullanılmıştır:

- *İtalik font*, değiştirilmesi gereken bir değeri belirtir (örneğin, bir klasör yolu).
- `Kod fontu`, yöntem ve özellik adları gibi ActionScript kodunu belirtir.
- *İtalik kod fontu*, değiştirilmesi gereken bir kod ögesini belirtir (örneğin, bir ActionScript parametresi).
- **Kalın font**, girdiğiniz bir değeri belirtir.

Bu kılavuzda kullanılan terimler

Bu kılavuzda aşağıdaki terimler kullanılmıştır:

çalışma zamanında Kod Flash Player'da çalışırken.

geliştirme sırasında Flash geliştirme ortamında çalışırken.

Ek kaynaklar

Bu kılavuzlara ek olarak Adobe, Adobe Geliştirici Merkezi ve Adobe Tasarım Merkezi'nde düzenli olarak güncellenen makaleler, tasarım fikirleri ve örnekler sunar.

www.adobe.com/go/learn_fl_samples_tr adreslerinde ek bileşen örnekleri bulabilirsiniz.

Adobe Geliştirici Merkezi

Adobe Geliştirici Merkezi'nde ActionScript ile ilgili en son bilgiler, gerçek uygulama geliştirme üzerine makaleler ve önemli yeni konular ile ilgili bilgiler yer almaktadır. Geliştirici Merkezi'ni www.adobe.com/go/flash_devcenter_tr adreslerinde görüntüleyin.

Adobe Tasarım Merkezi

Dijital tasarım ve hareket grafikleriyle ilgili en son yenilikleri öğrenin. Sanatçılara göre çalışmaya göz atın, yeni tasarım trendlerini keşfedin ve eğitimler, önemli iş akışları ve gelişmiş tekniklerle becerilerinizi geliştirin. Yeni dersler ve makaleler ile ilham verici galeri öğeleri için iki haftada bir kontrol edin. www.adobe.com/go/fl_designcenter_tr adresinde Tasarım Merkezi'ni görüntüleyin.

Bölüm 2: ActionScript 3.0 Bileşenleri Hakkında

Adobe® Flash® Professional CS5 bileşenleri, görünüm ve davranışlarını değiştirmenize olanak veren parametrelerin yer aldığı film klipleridir. Bileşen, RadioButton veya CheckBox gibi basit bir kullanıcı arabirimi denetimi olabileceği gibi, List ya da DataGrid benzeri içerikleri de barındırabilir.

Bileşenler, tutarlı davranış ve görünüme sahip güçlü Flash uygulamaları hızla ve kolayca oluşturmanızı sağlar. Özel düğmeler, birleşik kutular ve listeler oluşturmak yerine, bu denetimleri uygulayan Flash bileşenlerini kullanabilirsiniz. Tek yapmanız gereken bu bileşenleri Bileşenler panelinden uygulama belgenize sürüklemektir. Ayrıca bu bileşenlerin görünümünü uygulamanızın tasarımına uyacak şekilde kolayca özelleştirebilirsiniz.

ActionScript'i çok ayrıntılı bilmeden bunların tümünü yaparsanız da, bir bileşenin davranışını değiştirmek veya yeni davranış uygulamak için ActionScript 3.0'ı da kullanabilirsiniz. Her bileşenin benzersiz bir ActionScript yöntemleri, özellikleri ve olayları kümesi vardır ve bu küme, bileşenin *uygulama programlama arabirimi* (API) oluşturur. API, uygulama çalışırken bileşenler oluşturmanıza ve bu bileşenleri işlemenize olanak sağlar.

API, kendi yeni ve özel bileşenlerinizi oluşturmanıza da olanak sağlar. Flash topluluğu üyeleri tarafından oluşturulan bileşenleri www.adobe.com/go/flash_exchange_tr adreslerinde Adobe Exchange'den indirebilirsiniz. Bileşen oluşturma hakkında bilgi almak için bkz. www.adobe.com/go/learn_fl_creating_components_tr.

ActionScript 3.0 bileşen mimarisi, tüm bileşenlerin esas aldığı sınıfları, görünümü özelleştirmenize olanak sağlayan kaplama ve stilleri, bir olay işleme modelini, odak yönetimini, bir erişilebilirlik arabirimi ve daha fazlasını içerir.

Not: Adobe Flash CS5, ActionScript 3.0 bileşenlerinin yanı sıra ActionScript 2.0 bileşenlerini de içerir. Bu iki bileşen kümesini karıştıramazsınız. Belirli bir uygulama için kümelerden birini kullanmanız gerekir. Flash CS5, ActionScript 2.0 veya ActionScript 3.0 dosyası açmanıza bağlı olarak ActionScript 2.0 bileşenlerini ya da ActionScript 3.0 bileşenlerini sunar. Yeni bir Flash belgesi oluşturduğunuzda, Flash Dosyası (ActionScript 3.0) veya Flash Dosyası (ActionScript 2.0) belirtmeniz gerekir. Varolan bir belgeyi açtığımızda, Flash kullanılacak bileşenleri belirlemek için Yayınlama Ayarları'nı inceler. ActionScript 2.0 bileşenleri hakkında bilgi almak için, bkz. Adobe® ActionScript® 2.0 Bileşenlerini Kullanma.

Flash ActionScript 3.0 bileşenlerinin tam listesi için bkz. “Bileşen türleri” sayfa 4.

Bileşenleri kullanmanın avantajları

Bileşenler, uygulamanızı tasarlama işlemi kodlama işleminden ayırmanızı sağlar. Bunlar, tasarımcıların uygulamalarda kullanabileceği işlevleri geliştiricilerin oluşturmasına olanak sağlar. Geliştiriciler, sık kullanılan işlevleri bileşenlere kapsülleyebilir ve tasarımcılar da bileşenlerin parametrelerini değiştirerek bileşenlerin boyutunu, konumunu ve davranışını özelleştirebilir. Ayrıca bileşenin grafiksel öğelerini veya kaplamalarını düzenleyerek de bileşenin görünümünü değiştirebilirler.

Bileşenler, stil, kaplama ve odak yönetimi gibi temel işlevleri paylaşır. Bir uygulamaya birinci bileşeni eklediğinizde, bu temel işlev boyutun yaklaşık 20 kilobaytlık bölümünü alır. Başka bileşenler eklediğinizde, başlangıçta ayrılan bu bellek, eklenen bileşenler tarafından paylaşıldığından uygulamanızın boyutundaki artış daha az olur.

Bu bölümde, ActionScript 3.0 bileşenlerinin avantajlarından bazıları açıklanmaktadır.

ActionScript 3.0 güçlü Flash Player yeteneklerinin gelişiminde önemli bir adım olan güçlü ve nesne odaklı programlama dili sağlar. Bu dil, yeniden kullanılabilir kod tabanında zengin Internet uygulamaları oluşturmak üzere

tasarlanmıştır. ActionScript 3.0 komut dosyası oluşturmak için uluslararası standart olarak kabul edilmiş olan ECMAScript tabanlıdır ve ECMAScript (ECMA-262) sürüm 3 dil özellikleri ile uyumludur. ActionScript 3.0'a ayrıntılı bir giriş için bkz. *ActionScript 3.0 Geliştirici Kılavuzu*. Dil ile ilgili başvuru bilgileri için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

FLA tabanlı Kullanıcı Arabirimi bileşenleri geliştirme sırasında özelleştirmeyi kolaylaştırmak için kaplamalara kolay erişilmesini sağlar. Bu bileşenler ayrıca çalışma zamanında bileşenlerin görünümünü özelleştirmenizi ve kaplamalar yüklemenizi sağlayan kaplama stilleri gibi stilleri de sağlar. Daha fazla bilgi için bkz. “[Kullanıcı Arabirimi Bileşenlerini Özelleştirme](#)” sayfa 98 ve [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

Yeni FVLPlayback bileşeni, tam ekran desteği, gelişmiş canlı önizleme, renk ve alfa ayarları eklemenize olanak sağlayan kaplamalar ve gelişmiş FLV indirme ve mizanpaj özellikleriyle birlikte FLVPlaybackCaptioning bileşenini de ekler.

Özellik denetçisi ve Bileşen denetçisi Flash'ta geliştirme sırasında bileşen parametrelerini değiştirmenize olanak sağlar. Daha fazla bilgi için bkz. “[Bileşen dosyalarıyla çalışma](#)” sayfa 18 ve “[Parametreleri ve özellikleri ayarlama](#)” sayfa 20.

ComboBox, List ve TileList bileşenlerinin yeni koleksiyon iletişim kutusu, kullanıcı arabirimi üzerinden bunların dataProvider özelliğini doldurmanıza olanak sağlar. Daha fazla bilgi için bkz. “[DataProvider oluşturma](#)” sayfa 28.

ActionScript 3.0 olay modeli uygulamanızın olayları dinlemesine ve yanıt vermesi için olay işleyicilerini çağırmasına olanak sağlar. Daha fazla bilgi için bkz. “[ActionScript 3.0 olay işleme modeli](#)” sayfa 8 ve “[Olayları işleme](#)” sayfa 23.

Yönetici sınıfları bir uygulamada odağın işlenmesi ve stillerin yönetilmesi için kolay bir yol sağlar. Daha fazla bilgi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

UIComponent temel sınıfı kendisini genişleterek oluşturulan bileşenlere temel yöntemler, özellikler ve olaylar sağlar. Tüm ActionScript 3.0 kullanıcı arabirimi bileşenleri, UIComponent sınıfından miras alır. Daha fazla bilgi için [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki UIComponent sınıfına bakın.

Kullanıcı arabirimi FLA tabanlı bileşenlerde SWC kullanılması, derlemeyi hızlandırmak için bileşenin Zaman Çizelgesi içinde bir varlık olarak ActionScript tanımları sağlar.

ActionScript 3.0 kullanan, kolayca genişletilebilir bir sınıf hiyerarşisi, benzersiz ad alanları oluşturmanıza, gerektiğinde sınıfları içe aktarmanıza ve bileşenleri genişletmek için kolayca alt sınıflara ayırmanıza olanak sağlar.

Daha fazla bilgi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

Not: *Flash CS5, hem FLA tabanlı hem de SWC tabanlı bileşenleri destekler. Daha fazla bilgi için bkz. “[Bileşen mimarisi](#)” sayfa 16.*

Bileşen türleri

Flash CS5 yüklediğinizde, Flash bileşenlerini yüklersiniz.

ActionScript 3.0 bileşenleri şu kullanıcı arabirimi (UI) bileşenlerini içerir:

Düğme	List	TextArea
CheckBox	NumericStepper	TextInput
ColorPicker	RadioButton	TileList

ComboBox	ProgressBar	UILoader
DataGrid	ScrollPane	UIScrollBar
Etiket	Slider	

Flash ActionScript 3.0 bileşenleri, kullanıcı arabirimi bileşenlerinin yanı sıra, şu bileşenleri ve destekleyici sınıfları içerir:

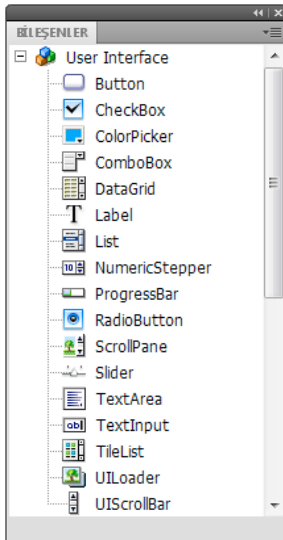
- SWC tabanlı bir bileşen olan FLVPlayback bileşeni (fl.video.FLVPlayback).
FLVPlayback bileşeni, HTTP üzerinden, Adobe® Flash® Video Streaming Service (FVSS) üzerinden veya Adobe Macromedia® Flash® Media Server (FMS) üzerinden aşamalı akış videosu oynatmak için Flash uygulamanıza önceden bir video oynatıcı dahil etmenizi sağlar. Daha fazla bilgi için, bkz. “[FLVPlayback Bileşenini kullanma](#)” sayfa 135.
- FLA tabanlı olan ve FLVPlayback bileşeninin ActionScript 2.0 ve ActionScript 3.0 sürümleriyle çalışan FLVPlayback Özel Kullanıcı Arabirimi bileşenleri. Daha fazla bilgi için, bkz. “[FLVPlayback Bileşenini kullanma](#)” sayfa 135.
- FLVPlayback için kapalı resim yazısı sağlayan FLVPlayback Captioning bileşeni. Bkz. “[FLVPlayback Captioning Bileşenini Kullanma](#)” sayfa 172.

ActionScript 3.0 bileşenlerinin ve destekleyen sınıfların bir listesi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

Flash bileşenlerini görüntüleyin:

Bu adımları izleyerek Bileşenler panelinde Flash ActionScript 3.0 bileşenlerini görüntüleyebilirsiniz.

- 1 Flash uygulamasını başlatın.
- 2 Yeni bir Flash dosyası (ActionScript 3.0) oluşturun veya Yayınlama Ayarları'nın ActionScript 3.0'ı belirttiği varolan bir Flash belgesini açın.
- 3 Açık değilse, Bileşenler panelini açmak için Pencere > Bileşenler seçeneğini belirleyin.



Kullanıcı Arabirimi bileşenleriyle Bileşenler paneli

www.adobe.com/go/flash_exchange_tr adreslerindeki Adobe Exchange'den ek bileşenler de indirebilirsiniz. Exchange'den indirilen bileşenleri yüklemek için, www.adobe.com/go/exchange_tr adreslerinden Adobe® Extension Manager'ı indirip yükleyin. Adobe Exchange Home bağına tıklattın ve Extension Manager bağına arayın. Tüm bileşenler, Flash uygulamasındaki Bileşenler panelinde görüntülenebilir. Bileşenleri Windows® veya Macintosh® bilgisayara yüklemek için bu adımları izleyin.

Bileşenleri Windows tabanlı veya Macintosh bilgisayara yükleyin:

- 1 Flash uygulamasından çıkın.
- 2 Bileşeni içeren SWC veya FLA dosyasını, sabit diskinizde şu klasöre yerleştirin:
 - Windows'da:
C:\Program Files\Adobe\Adobe Flash CS5\di\Configuration\Components
 - Macintosh'ta:
Macintosh Sabit Diski:Uygulamalar:Adobe Flash CS5:Configuration:Components
- 3 Flash uygulamasını başlatın.
- 4 Açık değilse, Bileşenler panelinde bileşeni görüntülemek için Pencere > Bileşenler seçeneğini belirleyin. Bileşen dosyaları hakkında daha fazla bilgi için bkz. “[Bileşen dosyalarıyla çalışma](#)” sayfa 18

Belgeye ekleme ve belgeden silme

FLA tabanlı bir bileşeni Bileşenler panelinden Sahne Alanı'na sürüklediğinizde, Flash, düzenlenebilir bir film klibini kütüphaneye içe aktarır. SWC tabanlı bir bileşeni Sahne Alanı'na sürüklediğinizde, Flash, derlenmiş bir klibi kütüphaneye içe aktarır. Bileşen kütüphaneye içe aktarıldıktan sonra, söz konusu bileşenin örneklerini Kütüphane panelinden veya Bileşenler panelinden Sahne Alanı'na sürükleyebilirsiniz.

Geliştirme sırasında bileşenler ekleme

Bir bileşeni, Bileşenler panelinden sürükleyerek belgeye ekleyebilirsiniz. Özellik denetçisinde veya Bileşen denetçisinin Parametreler sekmesinde bir bileşenin her örneği için özellikler ayarlayabilirsiniz.

- 1 Pencere > Bileşenler seçeneğini belirleyin.
- 2 Bileşenler panelinde bileşeni çift tıklattın veya bileşeni Sahne Alanı'na sürükleyin.
- 3 Sahne Alanı'nda bileşeni seçin.
- 4 Özellik denetçisi görünür değilse, Pencere > Özellikler > Özellikler seçeneğini belirleyin.
- 5 Özellik denetçisinde, bileşen örneği için bir örnek adı girin.
- 6 Örneğin parametrelerini belirtmek için, Pencere > Bileşen denetçisi seçeneklerini belirleyin ve Parametreler sekmesini seçin.
Daha fazla bilgi için, bkz. “[Parametreleri ve özellikleri ayarlama](#)” sayfa 20.
- 7 Genişlik (W:) ve yükseklik (H:) değerlerini düzenleyerek bileşenin boyutunu istediğiniz gibi değiştirin.
Belirli bileşen türlerini boyutlandırma hakkında daha fazla bilgi için bkz. “[Kullanıcı Arabirimi Bileşenlerini Özelleştirme](#)” sayfa 98.

- 8 Belgenizi derleyip ayarlarınızın sonuçlarını görmek için, Kontrol Et > Filmi Test Et seçeneklerini belirleyin veya Control+Enter tuşlarına basın.

Bileşenin stil özelliklerini ayarlayarak renk ve metin biçimlendirmesini değiştirebilir veya bileşenin kaplamalarını düzenleyerek görünümünü özelleştirebilirsiniz. Bu konular hakkında daha fazla bilgi için bkz. “[Kullanıcı Arabirimi Bileşenlerini Özelleştirme](#)” sayfa 98.

Geliştirme sırasında bir bileşeni Sahne Alanı'na sürüklerseniz, bileşenin örnek adını (örneğin myButton) kullanarak bileşeni ifade edebilirsiniz.

ActionScript ile çalışma zamanında bileşenler ekleme

ActionScript ile çalışma zamanında belgeye bir bileşen eklemek için öncelikle SWF derlendiğinde bileşenin uygulamanın kütüphanesinde (Pencere > Kütüphane) olması gerekir. Kütüphaneye bileşen eklemek için, bileşeni Bileşenler panelinden Kütüphane paneline sürükleyin. Kütüphane hakkında daha fazla bilgi için bkz. “[Kütüphane](#)” sayfa 21.

Ayrıca bileşenin API'sini uygulamanız için kullanılabilir hale getirmek üzere bileşenin sınıf dosyasını da içe aktarmanız gerekir. Bileşen sınıf dosyaları, bir veya daha fazla sınıf içeren *paketlere* yüklenir. Bileşen sınıfını içe aktarmak için, `import` ifadesini kullanın ve paket adını ve sınıf adını belirtin. Örneğin, Button sınıfı şu `import` ifadesiyle içe aktarılır:

```
import fl.controls.Button;
```

Bir bileşenin hangi pakette olduğuyla ilgili bilgiler için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#). Bileşen kaynak dosyalarının konumu hakkında bilgi almak için bkz. “[Bileşen dosyalarıyla çalışma](#)” sayfa 18.

Bileşenin bir örneğini oluşturmak için, bileşenin ActionScript yapıcı yöntemini çağırmanız gerekir. Örneğin, şu ifade `aButton` adında bir Button örneği oluşturur:

```
var aButton:Button = new Button();
```

Son adım, bileşen örneğini Sahne Alanı'na veya uygulama kabına eklemek için statik `addChild()` yönteminin çağırılmasıdır. Örneğin, şu ifade `aButton` örneğini ekler:

```
addChild(aButton);
```

Bu noktada, bileşenin boyutunu ve Sahne Alanı'ndaki konumunu dinamik olarak belirtmek için bileşenin API'sini kullanabilir, olayları dinleyebilir ve davranışını değiştirmek için özelliklerini ayarlayabilirsiniz. Belirli bir bileşen için API ile ilgili daha fazla bilgi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

`addChild()` yöntemi hakkında daha fazla bilgi için bkz. “[Görüntüleme listesiyle çalışma](#)” sayfa 24.

Bileşen silme

Geliştirme sırasında Sahne Alanı'ndan bileşen örneğini silmek için, bileşen örneğini seçin ve Delete tuşuna basın. Böylece örnek Sahne Alanı'ndan kaldırılır ancak bileşen uygulamanızdan kaldırılmaz.

Sahne Alanı'na veya kütüphaneye yerleştirdikten sonra Flash belgenizden bir bileşeni silmek için, bileşeni ve ilişkilendirilmiş varlıklarını kütüphaneden silmeniz gerekir. Yalnızca bileşenin Sahne Alanı'ndan silinmesi yeterli değildir. Kütüphaneden kaldırmazsanız, bileşeni derlediğinizde bileşen uygulamanıza dahil edilir.

- 1 Kütüphane panelinde, bileşenin sembolünü seçin.
- 2 Kütüphane panelinin alt kısmında Sil düğmesini tıklatın veya Kütüphane paneli menüsünden Sil seçeneğini belirleyin.

Bileşenle ilişkilendirilmiş herhangi bir varlığı silmek için bu adımları yineleyin.

Uygulamanız çalışırken bileşenin kabından nasıl kaldırıldığı hakkında bilgi almak için bkz. “[Görüntüleme listesinden bileşeni kaldırma](#)” sayfa 26.

Bileşenin sürümünü bulma

Flash ActionScript 3.0 bileşenlerinin, Adobe Teknik Desteğine sağlamanız gerektiğinde veya kullandığınız bileşenin sürümünü öğrenmeniz gerektiğinde görüntüleyebileceğiniz bir sürüm özelliği vardır.

Kullanıcı arabirimi bileşeninin sürüm numarasını görüntüleyin:

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 Bileşeni Sahne Alanı'na sürükleyin ve bu bileşene bir örnek adı verin. Örneğin, ComboBox öğesini Sahne Alanı'na sürükleyin ve **aCb** adını verin.
- 3 Eylemler panelini açmak için **F9** tuşuna basın veya Pencere > Eylemler seçeneklerini belirleyin.
- 4 Ana Zaman Çizelgesi'nde Kare 1'i tıklattın ve Eylemler paneline şu kodu ekleyin:

```
trace(aCb.version);
```

Aşağıdakine benzeyen bir sürüm numarasının Çıktı panelinde görüntülenmesi gerekir.

Sürüm numarası sınıf sabitinde saklandığından, FLVPlayback ve FLVPlaybackCaptioning bileşenleri için, örnek adı yerine sınıf adına başvurmanız gerekir.

FLVPlayback ve FLVPlaybackCaptioning bileşenleri için sürüm numarasını görüntüleyin:

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 FLVPlayback ve FLVPlaybackCaptioning bileşenlerini Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açmak için **F9** tuşuna basın veya Pencere > Eylemler seçeneklerini belirleyin.
- 4 Ana Zaman Çizelgesi'nde Kare 1'i tıklattın ve Eylemler paneline şu kodu ekleyin:

```
import fl.video.*;
trace("FLVPlayback.VERSION: " + FLVPlayback.VERSION);
trace("FLVPlaybackCaptioning.VERSION: " + FLVPlaybackCaptioning.VERSION);
```

Sürüm numaraları Çıktı panelinde görüntülenir.

ActionScript 3.0 olay işleme modeli

ActionScript 3.0, önceki ActionScript sürümlerinde varolan farklı olay işleme mekanizmalarının yerini alan tek bir olay işleme modelini sunmaktadır. Yeni olay modeli, Belge Nesnesi Modeli (DOM) Düzey 3 Olaylar Belirtimi'ni esas alır.

ActionScript 2.0 `addListener()` yöntemini kullanma konusunda deneyimli geliştiriciler için, ActionScript 2.0 olay dinleyicisi modeli ile ActionScript 3.0 olay modeli arasındaki farklılıkların belirtilmesi yararlı olabilir. Aşağıdaki listede, iki etkinlik modeli arasındaki önemli farklılıklardan birkaçı açıklanmaktadır:

- ActionScript 2.0'da olay dinleyicileri eklemek için, bazı durumlarda `addListener()` öğesini ve bazı durumlarda da `addEventListener()` öğesini kullanırsınız, ancak ActionScript 3.0'da ise tüm durumlarda `addEventListener()` öğesini kullanırsınız.

- ActionScript 2.0'da herhangi bir olay akışı yoktur, başka bir deyişle, `addListener()` yöntemi yalnızca olayı yayınlayan nesnede çağrılabilirken, ActionScript 3.0'da `addEventListener()` yöntemi, olay akışının parçası olan herhangi bir nesnede çağrılabilir.
- ActionScript 2.0'da, olay dinleyicileri işlev, yöntem veya nesne olabilirken, ActionScript 3.0'da yalnızca işlevler ya da yöntemler olay dinleyicisi olabilir.
- `on(event)` sözdizimi artık ActionScript 3.0'da desteklenmez, bu nedenle ActionScript olay kodunu bir film klibine ekleyemezsiniz. Olay dinleyicisi eklemek için yalnızca `addEventListener()` kullanabilirsiniz.

`aButton` adındaki bir `Button` bileşeninde `MouseEvent.CLICK` olayını dinleyen aşağıdaki örnekte temel ActionScript 3.0 olay işleme modeli gösterilmektedir:

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler);
function clickHandler(event:MouseEvent):void {
    trace("clickHandler detected an event of type: " + event.type);
    trace("the event occurred on: " + event.target.name);
}
```

ActionScript 3.0 olay işleme hakkında daha fazla bilgi için bkz. *ActionScript 3.0'ı Programlama*. Bileşenler için ActionScript 3.0 olay işleme hakkında daha fazla bilgi almak için bkz. "[Olayları işleme](#)" sayfa 23.

Basit bir uygulama

Bu bölüm, Flash bileşenlerini ve Flash geliştirme aracını kullanarak basit bir ActionScript 3.0 uygulaması oluşturma adımlarına sizi götürür. Bu örnek hem Zaman Çizelgesi'ne dahil edilen ActionScript koduna sahip bir FLA dosyası olarak hem de yalnızca kütüphanedeki bileşenleri içeren bir FLA dosyasına sahip harici bir ActionScript sınıf dosyası olarak sağlanır. Genelde, sınıflar ve uygulamalar arasında kodu paylaşabilmeniz ve uygulamalarınızın bakımını kolaylaştırmanız için harici sınıf dosyalarını kullanarak daha büyük uygulama geliştirmek istersiniz. ActionScript 3.0 ile programlama hakkında daha fazla bilgi için bkz. *ActionScript 3.0'ı Programlama*.

Uygulamanın tasarımı

ActionScript bileşen uygulamasına yönelik ilk örneğimiz, standart "Hello World" uygulamasının bir çeşitlemesi olup bunun tasarımı oldukça basittir:

- Uygulama, Greetings olarak adlandırılır.
- Bu, başlangıçta Hello World olan bir selamlamayı görüntülemek için `TextArea` öğesini kullanır.
- Metnin rengini değiştirmenize olanak sağlayan `ColorPicker` öğesini kullanır.
- Metnin boyutunu küçük, büyük veya daha büyük olarak ayarlamaya olanak sağlayan üç `RadioButton` öğesini kullanır.
- Açılır listeden farklı bir selamlama seçmenize olanak sağlayan `ComboBox` öğesini kullanır.
- Uygulama, Bileşenler panelindeki bileşenleri kullanır ve ActionScript kodu üzerinden uygulama öğeleri oluşturur.

Bu tanım yerindeyken uygulama oluşturma işlemine başlayabilirsiniz.

Greetings uygulaması oluşturma

Aşağıdaki adımlar sayesinde, FLA dosyası oluşturmak için Flash geliştirme aracı kullanılarak Greetings uygulaması oluşturulur, Sahne Alanı'na bileşenler yerleştirilir ve Zaman Çizelgesi'ne ActionScript kodu eklenir.

FLA dosyasında Greetings uygulaması oluşturma:

- 1 Dosya > Yeni'yi seçin.
- 2 Yeni Belge iletişim kutusunda Flash Dosyası (ActionScript 3.0) öğesini seçip Tamam'ı tıklatın.
Yeni bir Flash penceresi açılır.
- 3 Dosya > Kaydet seçeneklerini belirleyin, Flash dosyasını **Greetings fla** olarak adlandırın ve Kaydet düğmesini tıklatın.
- 4 Flash Bileşenleri panelinde bir TextArea bileşenini seçin ve Sahne Alanı'na sürükleyin.
- 5 Özellikler penceresinde, Sahne Alanı'nda TextArea seçili durumdayken örnek adı için **aTa** yazın ve şu bilgileri girin:
 - W (genişlik) değeri için **230** girin.
 - H (yükseklik) değeri için **44** girin.
 - X değeri (yatay konum) için **165** girin.
 - Y değeri (dikey konum) için **57** girin.
 - Parametreler sekmesi üzerinde, metin parametresi olarak **Hello World!** metnini girin.
- 6 Bir ColorPicker bileşenini Sahne Alanı'na sürükleyin, TextArea öğesinin soluna yerleştirin ve bu bileşene **txtCp örnek adını verin**. Özellik denetçisinde şu bilgileri girin:
 - X değeri için **96** girin.
 - Y değeri için **72** girin.
- 7 Üç RadioButton bileşenini Sahne Alanı'na birer birer sürükleyin ve bunlara **smallRb**, **largerRb** ve **largestRb** örnek adlarını verin. Bunlar için Özellik denetçisinde şu bilgileri girin:
 - Her birine W değeri için **100** ve H değeri için **22** girin.
 - X değeri için **155** girin.
 - smallRb'nin Y değeri için **120**, largerRb'nin Y değeri için **148** ve largestRb'nin Y değeri için **175** girin.
 - Her birinin groupName parametresi için **fontRbGrp** girin.
 - **Small**, **Larger**, **Largest** öğesinin Parametreler sekmesinde bunların etiketlerini girin.
- 8 Bir ComboBox bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **msgCb** örnek adını verin. Bileşen için Özellik denetçisinde şu bilgileri girin:
 - W değeri için **130** girin.
 - X değeri için **265** girin.
 - Y değeri için **120** girin.
 - Parametreler sekmesinde, istem parametresi için **Greetings** girin.
 - Değerler iletişim kutusunu açmak için, dataProvider parametresinin metin alanını çift tıklatın.
 - Artı işaretini tıklatın ve etiket değerinin yerine **Hello World!** öğesini getirin.
 - Önceki adımı **Have a nice day!** ve **Top of the Morning!** etiket değerlerini eklemek için tekrarlayın.
 - Değerler iletişim kutusunu kapatmak için Tamam'ı tıklatın.
- 9 Dosyayı kaydedin.
- 10 Açık değilse, **F9** tuşuna basarak veya Pencere menüsünden Eylemler seçeneğini belirleyerek Eylemler panelini açın. Ana Zaman Çizelgesi'nde Kare 1'i tıklatın ve Eylemler paneline şu kodu ekleyin:

```
import flash.events.Event;
import fl.events.ComponentEvent;
import fl.events.ColorPickerEvent;
import fl.controls.RadioButtonGroup;

var rbGrp:RadioButtonGroup = RadioButtonGroup.getGroup("fontRbGrp");
rbGrp.addEventListener(MouseEvent.CLICK, rbHandler);
txtCp.addEventListener(ColorPickerEvent.CHANGE, cpHandler);
msgCb.addEventListener(Event.CHANGE, cbHandler);
```

İlk üç satır, uygulamanın kullandığı olay sınıflarını içe aktarır. Kullanıcı bileşenlerden biriyle etkileşim kurduğunda bir olay gerçekleşir. Sonraki beş satır, uygulamanın dinlemek istediği olaylar için olay işleyicilerini kaydeder. Kullanıcı üzerine tıklattığında RadioButton için bir `click` olayı gerçekleşir. Kullanıcı ColorPicker'da farklı bir renk seçtiğinde `change` olayı gerçekleşir. Kullanıcı açılır listeden farklı bir selamlama seçtiğinde, ComboBox'ta `change` olayı gerçekleşir.

Dördüncü satır RadioButtonGroup sınıfını içe aktarır, böylece uygulama her düğmeye ayrı ayrı olay dinleyicisi atamak yerine bir olay dinleyicisini RadioButtons grubuna atayabilir.

- 11** Uygulamanın TextArea ögesindeki metnin `size` ve `color` stil özelliklerini değiştirmek için kullandığı `tf` TextFormat nesnesini oluşturmak için Eylemler paneline şu kod satırını ekleyin.

```
var tf:TextFormat = new TextFormat();
```

- 12** `rbHandler` olay işleme işlevini oluşturmak için şu kodu ekleyin. Bu işlev, kullanıcı RadioButton bileşenlerinden birini tıklattığında `click` olayını işler.

```
function rbHandler(event:MouseEvent):void {
    switch(event.target.selection.name) {
        case "smallRb":
            tf.size = 14;
            break;
        case "largerRb":
            tf.size = 18;
            break;
        case "largestRb":
            tf.size = 24;
            break;
    }
    aTa.setStyle("textFormat", tf);
}
```

Bu işlev, olayı tetikleyen RadioButton ögesini belirlemek üzere `event` nesnesinin `target` özelliğini incelemek için bir `switch` ifadesini kullanır. `currentTarget` özelliği, olayı tetikleyen nesnenin adını içerir. Kullanıcının tıklattığı RadioButton ögesine bağlı olarak, uygulama TextArea ögesindeki metnin boyutunu 14, 18 veya 24 nokta olarak değiştirir.

- 13** ColorPicker'daki değere değişiklik işleyen `cpHandler()` işlevi uygulamak için şu kodu ekleyin:

```
function cpHandler(event:ColorPickerEvent):void {
    tf.color = event.target.selectedColor;
    aTa.setStyle("textFormat", tf);
}
```

Bu işlev, `tf` TextFormat nesnesinin `color` özelliğini, ColorPicker'da seçilen renge ayarlar ve sonra bunu `aTa` TextArea örneğindeki metne uygulamak için `setStyle()` ögesini çağırır.

- 14** ComboBox'taki seçime değişiklik işleyen `cbHandler()` işlevi uygulamak için şu kodu ekleyin:

```
function cbHandler(event:Event):void {  
    aTa.text = event.target.selectedItem.label;  
}
```

Bu işlev, TextArea ögesindeki metnin yerine ComboBox'ta seçili metni (`event.target.selectedItem.label`) getirir.

- 15 Kodu derleyip Greetings uygulamasını test etmek için Kontrol Et > Filmi Test Et seçeneklerini belirleyin ve Control+Enter tuşlarına basın.

Aşağıdaki bölümde, harici bir ActionScript sınıfı ve yalnızca kütüphanedeki zorunlu bileşenlere sahip bir FLA dosyası ile aynı uygulamanın nasıl oluşturulacağı gösterilmektedir.

Greetings2 uygulamasını bir harici sınıf dosyası ile oluşturma:

- 1 Dosya > Yeni'yi seçin.
- 2 Yeni Belge iletişim kutusunda Flash Dosyası (ActionScript 3.0) ögesini seçip Tamam'ı tıklatın.
Yeni bir Flash penceresi açılır.
- 3 Dosya > Kaydet seçeneklerini belirleyin, Flash dosyasını **Greetings2 fla** olarak adlandırın ve Kaydet düğmesini tıklatın.
- 4 Şu bileşenlerin her birini Bileşenler panelinden kütüphaneye sürükleyin:
 - ColorPicker
 - ComboBox
 - RadioButton
 - TextArea

Derlenen SWF dosyası bu varlıkların her birini kullanır, bu nedenle bunları kütüphaneye eklemeniz gerekir. Bileşenleri Kütüphane panelinin alt kısmına sürükleyin. Bu bileşenleri kütüphaneye eklediğinizde, diğer varlıklar da (List, TextInput ve UIScrollBox gibi) otomatik olarak eklenir.
- 5 Özellikler penceresinde, Belge Sınıfı için **Greetings2** yazın.
Flash, “belge sınıfı için tanım bulunamıyor,” şeklinde bir uyarı görüntülense, bu uyarıyı yoksayın. Greetings2 sınıfını sonraki adımlarda tanımlarsınız. Bu sınıf, uygulamanın ana işlevlerini tanımlar.
- 6 Greetings2 fla dosyasını kaydedin.
- 7 Dosya > Yeni'yi seçin.
- 8 Yeni Belge iletişim kutusunda ActionScript Dosyası'nı seçin ve Tamam'ı tıklatın.
Yeni bir komut dosyası penceresi açılır.
- 9 Şu kodu komut dosyası penceresine ekleyin:


```
package {
    import flash.display.Sprite;
    import flash.events.Event;
    import flash.events.MouseEvent;
    import flash.text.TextFormat;
    import fl.events.ComponentEvent;
    import fl.events.ColorPickerEvent;
    import fl.controls.ColorPicker;
    import fl.controls.ComboBox;
    import fl.controls.RadioButtonGroup;
    import fl.controls.RadioButton;
    import fl.controls.TextArea;
    public class Greetings2 extends Sprite {
        private var aTa:TextArea;
        private var msgCb:ComboBox;
        private var smallRb:RadioButton;
        private var largerRb:RadioButton;
        private var largestRb:RadioButton;
        private var rbGrp:RadioButtonGroup;
        private var txtCp:ColorPicker;
        private var tf:TextFormat = new TextFormat();
        public function Greetings2() {
```

Komut dosyası, Greetings2 adında ActionScript 3.0 sınıfını tanımlar. Komut dosyası şunları yapar:

- Dosyada kullanılacak sınıfları içe aktarır. Normalde, kodda farklı sınıflara başvurduğunuzda bu içe aktarma ifadelerini eklersiniz ancak kısa sürmesi için bu örnek tümünü tek bir adımda içe aktarmaktadır.
- Bu, koda eklenecek farklı bileşen nesnesi türlerini temsil eden değişkenleri bildirir. Başka bir değişken, `tf` `TextFormat` nesnesini oluşturur.
- Sınıf için `Greetings2()` yapıcı işlevini tanımlar. Aşağıdaki adımlarda, bu işleve satırlar ekleyeceğiz ve sınıfa da başka yöntemler ekleyeceğiz.

10 Dosya > Kaydet seçeneklerini belirleyin, dosyayı **Greetings2.as** olarak adlandırın ve Kaydet düğmesini tıklayın.

11 Şu kod satırlarını `Greeting2()` işlevine ekleyin:

```
        createUI();
        setUpHandlers();
    }
```

Şimdi işlevin şöyle görünmesi gerekir:

```
public function Greetings2() {
    createUI();
    setUpHandlers();
}
```

12 `Greeting2()` yönteminin kapanış ayarından sonra şu kod satırlarını ekleyin:

```
private function createUI() {
    bldTxtArea();
    bldColorPicker();
    bldComboBox();
    bldRadioButtons();
}
private function bldTxtArea() {
    aTa = new TextArea();
    aTa.setSize(230, 44);
    aTa.text = "Hello World!";
    aTa.move(165, 57);
    addChild(aTa);
}
private function bldColorPicker() {
    txtCp = new ColorPicker();
    txtCp.move(96, 72);
    addChild(txtCp);
}
private function bldComboBox() {
    msgCb = new ComboBox();
    msgCb.width = 130;
    msgCb.move(265, 120);
    msgCb.prompt = "Greetings";
    msgCb.addItem({data:"Hello.", label:"English"});
    msgCb.addItem({data:"Bonjour.", label:"Français"});
    msgCb.addItem({data:"¡Hola!", label:"Español"});
    addChild(msgCb);
}
private function bldRadioButtons() {
    rbGrp = new RadioButtonGroup("fontRbGrp");
    smallRb = new RadioButton();
    smallRb.setSize(100, 22);
    smallRb.move(155, 120);
    smallRb.group = rbGrp; //"fontRbGrp";
    smallRb.label = "Small";
    smallRb.name = "smallRb";
    addChild(smallRb);
    largerRb = new RadioButton();
    largerRb.setSize(100, 22);
    largerRb.move(155, 148);
    largerRb.group = rbGrp;
    largerRb.label = "Larger";
    largerRb.name = "largerRb";
    addChild(largerRb);
    largestRb = new RadioButton();
    largestRb.setSize(100, 22);
    largestRb.move(155, 175);
    largestRb.group = rbGrp;
    largestRb.label = "Largest";
    largestRb.name = "largestRb";
    addChild(largestRb);
}
```

Bu satırlar şunları yapar:

- Uygulamadan kullanılan bileşenleri başlatır.
- Her bileşenin boyutunu, konumunu ve özelliklerini ayarlar.

- addChild() yöntemini kullanarak her bileşeni Sahne Alanı'na ekler.

13 bldRadioButtons() yönteminin kapanış ayracından sonra, setUpHandlers() yöntemi için şu kodu ekleyin:

```
private function setUpHandlers():void {
    rbGrp.addEventListener(MouseEvent.CLICK, rbHandler);
    txtCp.addEventListener(ColorPickerEvent.CHANGE, cpHandler);
    msgCb.addEventListener(Event.CHANGE, cbHandler);
}
private function rbHandler(event:MouseEvent):void {
    switch(event.target.selection.name) {
        case "smallRb":
            tf.size = 14;
            break;
        case "largerRb":
            tf.size = 18;
            break;
        case "largestRb":
            tf.size = 24;
            break;
    }
    aTa.setStyle("textFormat", tf);
}
private function cpHandler(event:ColorPickerEvent):void {
    tf.color = event.target.selectedColor;
    aTa.setStyle("textFormat", tf);
}
private function cbHandler(event:Event):void {
    aTa.text = event.target.selectedItem.data;
}
}
```

Bu işlevler, bileşenler için olay dinleyicilerini tanımlar.

14 Dosyayı kaydetmek için Dosya > Kaydet seçeneklerini belirleyin.

15 Kodu derleyip Greetings2 uygulamasını test etmek için Kontrol Et > Filmi Test Et seçeneklerini belirleyin ve Control+Enter tuşlarına basın.

Sonraki örnekleri geliştirip çalıştırın.

Greetings uygulamasını geliştirip çalıştırdıktan sonra, bu kitapta sunulan diğer kod örneklerini çalıştırmak için gereken temel bilgilere sahip olmanız gerekir. Her örnekte ilgili ActionScript 3.0 kodu vurgulanıp ele alınacaktır ve bu kitaptaki örneklerin her birini kesip bir FLA dosyasına yapıştırabilmeniz ve derleyip çalıştırabilmeniz gerekir.

Bölüm 3: Bileşenlerle Çalışma

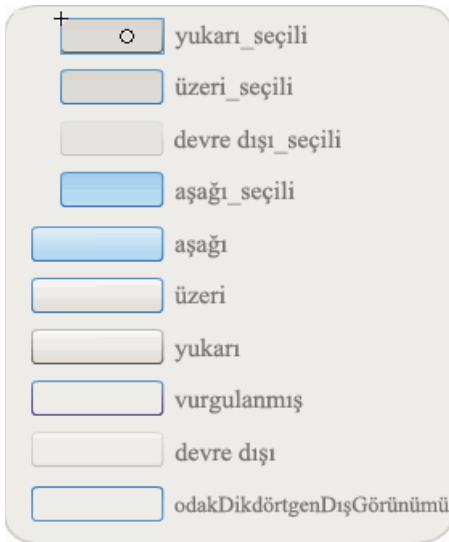
Bileşen mimarisi

Adobe® Flash Player sürüm 9.0.28.0 ve sonrası tarafından Adobe® ActionScript® 3.0 bileşenleri desteklenir. Bu bileşenler, Flash CS4 öncesinde oluşturulan bileşenlerle uyumlu değildir. Adobe® ActionScript® 2.0 bileşenlerini kullanma hakkında bilgi almak için, bkz. *Using Adobe® ActionScript® 2.0 Bileşenlerini Kullanma* ve *Adobe® ActionScript® 2.0 Bileşenleri Dil Başvurusu*.

Adobe ActionScript 3.0 Kullanıcı Arabirimi (UI) bileşenleri, FLA tabanlı bileşenler olarak uygulanır ancak Flash CS5 hem SWC hem de FLA tabanlı bileşenleri destekler. Örneğin, FLVPlayback ve FLVPlaybackCaptioning bileşenleri SWC tabanlı bileşenlerdir. Bileşenler panelinde görüntülenmesi için, Bileşenler klasörüne herhangi türde bir bileşen yerleştirebilirsiniz. Bu iki tür bileşen farklı şekilde oluşturulmuş olduğundan burada da ayrı ayrı açıklanmıştır.

ActionScript 3.0 FLA tabanlı bileşenler

ActionScript 3.0 Kullanıcı Arabirimi bileşenleri, düzenlemek üzere Sahne Alanı'nda bileşeni çift tıklatarak eriştiğiniz yerleşik kaplamalar içeren FLA tabanlı (.fla) dosyalardır. Bileşenin kaplamaları ve diğer varlıkları Zaman Çizelgesi'nin Kare 2'sine yerleştirilir. Bileşeni çift tıklattığınızda, Flash uygulaması otomatik olarak Kare 2'ye geçer ve bileşen kaplamalarının bir paletini açar. Aşağıdaki şekilde, Button bileşeni için görüntülenen kaplamaların olduğu palet gösterilmektedir.



Button bileşenin kaplamaları

Bileşen kaplamaları ve bileşenleri özelleştirme hakkında daha fazla bilgi için bkz. “[Kullanıcı Arabirimi Bileşenlerini Özelleştirme](#)” sayfa 98 ve “[FLVPlayback bileşenini özelleştirme](#)” sayfa 153.

Uygulamalarınızın derlemesini hızlandırmak ve ActionScript 3.0 ayarlarınızla olan çakışmaları önlemek için, Flash CS5 FLA tabanlı kullanıcı arabirimi bileşenleri, bileşenin önceden derlenmiş ActionScript kodunu içeren bir SWC dosyası da içerir. ComponentShim SWC dosyası, önceden derlenmiş tanımları kullanılabilir duruma getirmek için her Kullanıcı Arabirimi bileşeninde Kare 2'deki Sahne Alanı'na yerleştirilir. Bir bileşenin ActionScript ile kullanılabilir olması için, Bağlantı özelliklerinde, İlk Karede Dışa Aktar seçeneği belirlenmiş şekilde ya Sahne Alanı'nda olması ya da kütüphanede olması gerekir. ActionScript kullanarak bir bileşen oluşturmak için, bu bileşene erişmek üzere ayrıca `import` ifadesiyle bir sınıfı içe aktarmanız gerekir. `import` ifadesiyle ilgili daha fazla bilgi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

SWC Tabanlı Bileşenler

SWC tabanlı bileşenler de bir FLA dosyası ve bir ActionScript sınıf dosyası içerir ancak SWC olarak dışa aktarılır. SWC dosyası, önceden derlenmiş Flash sembollerinden ve değişmeyecek sembolleri ve kodu yeniden derlemenizi önleyen ActionScript kodundan oluşan bir pakettir.

FLVPlayback ve FLVPlaybackCaptioning bileşenleri, SWC tabanlı bileşenlerdir. Bu bileşenler yerleşik değil, harici kaplamalar içerir. FLVPlayback bileşeni, önceden tasarlanmış kaplamalar koleksiyonundan bir tanesini seçerek, Bileşenler panelindeki kullanıcı arabirimi denetimlerinde yer alan denetimleri (BackButton, BufferingBar vb.) özelleştirerek veya özel bir kaplama oluşturarak değiştirebileceğiniz varsayılan bir kaplamaya sahiptir. Daha fazla bilgi için, bkz. "[FLVPlayback bileşenini özelleştirme](#)" sayfa 153.

Flash uygulamasında bir film klibini şu şekilde derlenmiş bir klibe dönüştürebilirsiniz:

Film klibi derleme

- Kütüphane panelinde film klibini sağ tıklatın (Windows) veya Control tuşunu basılı tutarak tıklatın (Macintosh) ve ardından Derlenmiş Klibe Dönüştür seçeneğini belirleyin.

Derlenmiş klip, derlenmiş olduğu film klibi gibi davranır ancak derlenmiş klipler normal film kliplerinden daha hızlı şekilde görüntülenir ve yayınlanır. Derlenmiş klipler düzenlenemez ancak Özellik denetçisinde ve Bileşen denetçisinde özellikleri görüntülenebilir.

SWC bileşenlerinde, derlenmiş bir klip, bileşenin önceden derlenmiş ActionScript tanımları ve bileşeni açıklayan diğer dosyalar yer alır. Kendi bileşeninizi oluşturursanız, bu bileşeni dağıtmak için SWC dosyası olarak dışa aktarabilirsiniz.

SWC dosyası dışa aktarma

- Kütüphane panelinde film klibini seçin ve sağ tıklatın (Windows) veya Control tuşunu basılı tutarak tıklatın (Macintosh), ardından SWC Dosyasını Dışa Aktar seçeneğini belirleyin.

Not: Flash CS4 biçimi veya sonraki bir SWC dosyası biçimi, Flex SWC biçimiyle uyumludur, bu nedenle SWC dosyaları iki ürün arasında değiştirilebilse de bu her zaman değişiklik olmayacağı anlamına gelmez.

SWC tabanlı bileşenler oluşturma hakkında bilgi almak için bkz.

www.adobe.com/go/learn_fl_creating_components_tr.

ActionScript 3.0 Bileşenleri API'si

Her ActionScript 3.0 bileşeni, paket klasöründe bulunan ve fl biçiminde ada sahip olan yerleşik bir ActionScript 3.0 sınıfıdır. `packagename.classname`. Örneğin, Button bileşeni Button sınıfının bir örneği olup `fl.controls.Button` paket adına sahiptir. Uygulamanızda bir bileşen sınıfını içe aktarırken paket adına başvurmanız gerekir. Button sınıfını şu ifadeyle içe aktarırsınız:

```
import fl.controls.Button;
```

Bileşen sınıfı dosyalarının konumu hakkında bilgi almak için bkz. “[Bileşen dosyalarıyla çalışma](#)” sayfa 18.

Bileşenin sınıfı, uygulamanızda o bileşenle etkileşim kurmanıza olanak sağlayan yöntemleri, özellikleri, olayları ve stilleri tanımlar. ActionScript 3.0 kullanıcı arabirimi bileşenleri, Sprite ve UIComponent sınıflarının alt sınıfları olup bu sınıfların özelliklerini, yöntemlerini ve olaylarını miras alır. Sprite sınıfı, temel görüntüleme listesi yapıtaşı olup MovieClip ögesine benzer ancak Zaman Çizelgesi'ne sahip değildir. UIComponent sınıfı hem etkileşimli hem de etkileşimli olmayan, tüm görsel bileşenler için taban sınıftır. Her bileşenin miras yolu, özellikleri, yöntemleri, olayları ve stilleri de dahil olmak üzere, *Adobe Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* bölümünde açıklanır.

Tüm ActionScript 3.0 bileşenleri, ActionScript 3.0 olay işleme modelini kullanır. Olay işleme hakkında daha fazla bilgi için, bkz. “[Olayları işleme](#)” sayfa 23 ve *ActionScript 3.0'i Programlama*.

Bileşen dosyalarıyla çalışma

Bu bölümde, bileşen dosyalarının nerede saklandığı, ActionScript kaynak dosyalarının nerede bulunduğu ve Bileşenler panelinden nasıl bileşen eklenip kaldırıldığı açıklanmaktadır.

Bileşen dosyalarının saklandığı yer

Flash bileşenleri, uygulama düzeyinde Konfigürasyon klasöründe saklanır.

Not: Bu klasörler hakkında bilgi almak için, bkz. “[Flash ile yüklenen konfigürasyon klasörleri](#)”, *Flash Uygulamasını Kullanma*.

Bileşenler şu konumlara yüklenir:

- Windows 2000 veya Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\dil\Configuration\Components
- Mac OS X: Macintosh HD:Applications:Adobe Flash CS5:Configuration:Components

Bileşenler klasöründe, Kullanıcı Arabirimi (UI) bileşenleri, User Interface fla dosyasında yer alırken FLVPlayback (FLVPlaybackAS3.swc) ve FLVPlaybackCaptioning bileşenleri de Video klasöründe yer alır.

Bileşenleri aşağıdaki kullanıcı tabanlı konumlarda da saklayabilirsiniz:

- Windows 2000 veya Windows XP: C:\Documents and Settings\kullanıcı adı\Local Settings\Application Data\Adobe\Adobe Flash CS5\tr\Configuration\Components
- Windows Vista: C:\Users\kullanıcı adı\Local Settings\Application Data\Adobe\Adobe Flash CS5\tr\Configuration\Components

Not: Windows'ta, *Application Data* klasörü varsayılan olarak gizlenmiştir. Gizli klasörleri ve dosyaları göstermek için, *Bilgisayarım*'i seçip *Windows Gezgini*'ni açın ve sonra *Araçlar>Klasör Seçenekleri* ve *Görünüm* sekmesini seçin. *Görünüm* sekmesinin altında, *Gizli dosya ve klasörleri göster* radyo düğmesini seçin.

- Mac OS X: Macintosh HD:Users:<kullanıcı adı>:Library:Application Support:Adobe Flash CS5:Configuration:Components

Bileşen kaynak dosyalarının saklandığı yer

Bileşenlerin ActionScript (.as) sınıf dosyaları (veya *kaynak dosyaları*), Windows 2000 veya Windows XP için şu uygulama klasörlerine yüklenir:

Kullanıcı Arabirimi bileşenleri C:\Program Files\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\User Interface\fl

FLVPlayback C:\Program Files\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlayback\fl\video

FLVPlaybackCaptioning C:\Program Files\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

Mac OS X için, bileşen kaynak dosyaları burada bulunur:

Kullanıcı Arabirimi bileşenleri Macintosh HD:Applications:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

FLVPlayback Macintosh HD:Applications:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0:FLVPlayback:fl:video

FLVPlaybackCaptioning Macintosh HD:Applications:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0:FLVPlaybackCaptioning:fl:video

Bileşen kaynak dosyaları ve Classpath

ActionScript 3.0 bileşenlerinin kodları derlenmiş olduğundan, Classpath değişkeninizde ActionScript sınıf dosyalarının konumunu belirtmemiz gerekir. Classpath'e ActionScript sınıf dosyalarının konumlarını dahil ederseniz, uygulamalarınızın derlenmesi için gerekli süreyi artırmış olursunuz. Ancak Flash uygulaması, Classpath ayarınızda bileşen sınıf dosyalarını bulursa, sınıf dosyası her zaman bileşenin derlenmiş kodu üzerinde öncelik kazanır.

Classpath ayarınıza bileşen kaynak dosyalarının konumunu eklemek isteyebileceğiniz tek durum, bileşenlerle bir uygulamanın hatalarını ayıkladığınız zamandır. Daha fazla bilgi için bkz. "[Bileşen uygulamalarının hatalarını ayıklama](#)" sayfa 20.

Bileşen dosyalarını değiştirme

SWC tabanlı bileşenleri günceller, ekler veya kaldırabilirsiniz ya da Flash uygulamasına yeni FLA tabanlı bileşenler eklerseniz, bunları kullanılabilir duruma getirmek için yeniden Bileşenler paneline yüklemeniz gerekir. Flash uygulamasını yeniden başlatarak veya Bileşenler paneli menüsünden Yeniden Yükle seçeneğini belirleyerek bileşenleri yeniden yükleyebilirsiniz. Böylece Flash uygulamasının Bileşenler klasörüne eklediğiniz tüm bileşenleri alması sağlanır.

Flash uygulaması çalışırken bileşenleri Bileşenler paneline yeniden yükleme:

- Bileşenler paneli menüsünde Yeniden Yükle seçeneğini belirleyin.

Bir bileşeni Bileşenler panelinden kaldırma:

- Bileşenler klasöründen FLA, SWC veya MXP dosyasını kaldırın ve Flash uygulamasını yeniden başlatın ya da Bileşenler paneli menüsünden Yeniden Yükle seçeneğini belirleyin. MXP dosyası, Adobe Exchange'den indirilmiş bir bileşen dosyasıdır.

Flash uygulaması çalışırken SWC tabanlı bileşenleri kaldırabilir ve değiştirebilirsiniz, yeniden yükleme yapıldığında değişiklikler yansıtılır ancak FLA tabanlı bileşenleri değiştirir ya da silerseniz, siz Flash uygulamasını sonlandırınca ya da yeniden başlatınca kadar değişiklikler yansıtılmaz. Ancak FLA tabanlı bileşenler ekleyebilir ve Yeniden Yükle komutuyla bu bileşenleri yükleyebilirsiniz.



Adobe, öncelikle değiştireceğiniz Flash bileşen dosyasının (.fla veya .as) bir kopyasını oluşturmanızı önerir. Daha sonra gerekirse bu dosyayı geri yükleyebilirsiniz.

Bileşen uygulamalarının hatalarını ayıklama

ActionScript 3.0 bileşenleri, uygulamanızı derlerken derleme süresini azaltmak için tüm kaynak kodunu içerir. Ancak Flash hata ayıklayıcısı, derlenen kliplerin içindeki kodu inceleyemez. Bu nedenle, bileşenin kaynak kodunu da içerecek şekilde uygulamanızın hatalarını ayıklamak istiyorsanız, bileşen kaynak dosyalarını Classpath ayarınıza eklemeniz gerekir.

Bileşen paketi klasörlerinin konumu, bileşen türüne yönelik kaynak dosyalarının konumuna göre değişir. Tüm kullanıcı arabirimi bileşenlerinin ActionScript 3.0 kaynak dosyalarının tamamına başvurmak için, Kullanıcı Arabirimi paketlerinin Classpath öğesine şu konumu ekleyin:

- \$(AppConfig)/Component Source/ActionScript 3.0/Kullanıcı Arabirimi

Not: Böylece, tüm kullanıcı arabirimi bileşenlerinin derlenmiş dahili kodu geçersiz kılınır ve uygulamanızın derleme süresi artar. Bir bileşenin kaynak kodunu herhangi bir nedenle değiştirdiyse, bunun sonucunda söz konusu bileşen farklı davranışlar sergileyebilir.

Classpath öğesini ayarlamak için, Düzen menüsünden Tercihler'i seçin ve sonra Kategori listesinden ActionScript seçeneğini belirleyip ActionScript 3.0 Ayarları düğmesini tıklatın Yeni bir giriş eklemek için, geçerli ayarları görüntüleyen pencerenin yukarıdaki artı işaretini tıklatın.

\$(AppConfig) değişkeni, Flash CS5 uygulamasını yüklediğiniz konumdaki Flash CS5 Konfigürasyon klasörünü ifade eder. Yol, genel olarak şöyle görünür:

- Windows 2000 veya Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\dil\Configuration\
- Mac OS X: Macintosh HD:Applications:Adobe Flash CS5:Configuration

Not: Bir bileşen kaynak dosyasını değiştirmeniz gerekiyorsa, Adobe orijinal kaynak dosyasını farklı bir konuma kopyalamanızı ve sonra bu konumu Classpath öğenize eklemenizi önemle önerir.

Bileşen kaynak dosyaları hakkında daha fazla bilgi almak için bkz. "[Bileşen kaynak dosyalarının saklandığı yer](#)" sayfa 18.

Parametreleri ve özellikleri ayarlama

Her bileşen, bileşenin görünüm ve davranışını değiştirmek için ayarlayabileceğiniz parametrelere sahiptir. Parametre, bileşen sınıfının bir özelliği olup Özellik denetçisinde ve Bileşen denetçisinde görüntülenir. En yaygın kullanılan özellikler, geliştirme parametreleri olarak görüntülenir; diğerlerini ActionScript ile ayarlamanız gerekir. Geliştirme sırasında ayarlanabilen tüm parametreler ActionScript ile de ayarlanabilir. Bir parametre ActionScript ile ayarlandığında, geliştirme sırasında ayarlanan değerler geçersiz kılınır.

ActionScript 3.0 Kullanıcı Arabirimi bileşenlerinin çoğu, UICOMPONENT sınıfının yanı sıra bir taban sınıfının da özellik ve yöntemlerini miras alır. Örneğin, Button ve CheckBox sınıfları, hem UICOMPONENT sınıfının hem de BaseButton sınıfının özelliklerini miras alır. Bileşenin miras alınan özellikleri ve kendi sınıf özellikleri erişiminize açıktır. Örneğin, ProgressBar bileşeni, UICOMPONENT öğesinden ProgressBar.enabled özelliğini miras alır ancak kendi ProgressBar.percentComplete özelliğine de sahiptir. ProgressBar bileşeninin bir örneğiyle etkileşim kurmak için, bu özelliklerin ikisine de erişebilirsiniz. Bir bileşenin özellikleriyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde sınıf girişlerine bakın.

Özellik denetçisini veya Bileşen parametresini kullanarak bileşen örneği için parametreler ayarlayabilirsiniz.

Özellik denetçisinde bir bileşen için örnek adı girme:

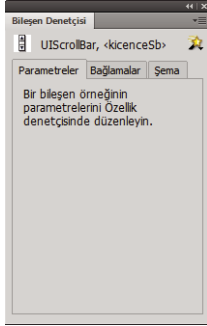
- 1 Pencere > Özellikler > Özellikler'i seçin.

- 2 Sahne Alanı'nda bir bileşen örneği seçin.
- 3 Film Klipi yazan açılır listenin aşağısındaki <Örnek Adı> yazan kutuya bileşen örneği için bir ad girin. Veya Parametreler sekmesini tıklattın ve *Component* kelimesinin aşağısındaki kutuya adı girin. Ayarlamak istediğiniz tüm parametreler için değer girin.

Bileşenin türünü belirtmek için örnek adına bir sonek eklemek iyi fikirdir; böylece ActionScript kodunuzun okunması kolaylaşır. Örneğin, **licenseSb** örnek adı, bileşenin **licenseTa** metin alanındaki bir lisans sözleşmesinde kaydırılan kaydırma çubuğu olduğunu belirtir.

Bileşen denetçisinde bir bileşen örneği için parametre girme:

- 1 Pencere > Bileşen Denetçisi'ni seçin.
- 2 Sahne Alanı'nda bir bileşen örneği seçin.
- 3 Parametreler sekmesini tıklattın ve listelenen parametrelerden herhangi biri için değer girin.



Bileşen denetçisinde bileşen parametreleri

ActionScript'te bileşen özelliklerini ayarlama

ActionScript'te, Sahne Alanı'ndaki bir nesneye veya örneğe ait özelliklere ya da yöntemlere erişmek için bir nokta (.) operatörünü kullanırsınız. Bir nokta sözdizimi ifadesi, bir nokta tarafından takip edilen örnek adıyla başlar ve belirtmek istediğiniz öğeyle sonlanır. Örneğin, şu ActionScript kodu, `aCh` CheckBox örneğini 50 piksel genişlikte yapmak için bu örneğin `width` özelliğini ayarlar:

```
aCh.width = 50;
```

Şu `if` ifadesi, kullanıcının onay kutusunu seçip seçmediğini kontrol eder:

```
if (aCh.selected == true) {  
    displayImg(redCar);  
}
```

Kütüphane

Bir belgeye ilk kez bileşen eklediğinizde, Flash uygulaması bunu Kütüphane paneline film klipi olarak içe aktarır. Ayrıca bir bileşeni, doğrudan Bileşenler panelinden Kütüphane paneline sürükleyebilir ve ardından bu bileşenin bir örneğini Sahne Alanı'na ekleyebilirsiniz. Her durumda, sınıf öğelerine erişmeden önce kütüphaneye bir bileşen eklemeniz gerekir.

Kütüphaneye bir bileşen ekler ve ActionScript kullanarak bu bileşenin bir örneğini oluşturursanız öncelikle `import` ifadesiyle bunun sınıfını içe aktarmanız gerekir. `import` ifadesinde hem bileşenin paket adını hem de sınıf adını belirtmeniz gerekir. Örneğin, aşağıdaki ifade `Button` sınıfını içe aktarır:

```
import fl.controls.Button;
```

Kütüphaneye bir bileşen eklediğinizde, Flash uygulaması farklı durumlara yönelik kaplamaları da içeren, bu bileşenin varlıklarının bir klasörünü de içe aktarır. Bir bileşenin *kaplamaları*, uygulamada bileşenin grafiksel görünümünü oluşturan semboller koleksiyonunu içerir. Tek bir kaplama, bileşenin belirli bir durumunu belirten grafiksel temsil veya film klibidir.

Component Assets klasörünün içerikleri, istediğiniz zaman bileşenin kaplamalarını değiştirmenize olanak sağlar. Daha fazla bilgi için bkz. “[Kullanıcı Arabirimi Bileşenlerini Özelleştirme](#)” sayfa 98.

Bileşen kütüphanede olduktan sonra, bileşenin simgesini Bileşenler panelinden veya Kütüphane panelinden Sahne Alanı'na sürükleyerek belgenize daha fazla bileşen örneği ekleyebilirsiniz.

Bileşenleri boyutlandırma

Bileşen örneklerini yeniden boyutlandırmak için, Serbest Dönüştürme aracını veya `setSize()` yöntemini kullanın. Herhangi bir bileşen örneğini yeniden boyutlandırmak için o bileşen örneğinden `setSize()` yöntemini çağırabilirsiniz (bkz. `UIComponent.setSize()`). Şu kod, List bileşeninin örneğini 200 piksel genişlik ve 300 piksel yüksekliğe yeniden boyutlandırır:

```
aList.setSize(200, 300);
```

Bileşen, etiketine sığmak için otomatik olarak yeniden boyutlandırılmaz. Bir belgeye eklenen bileşen örneği, etiketini görüntüleyebilecek kadar büyük değilse, etiket metni kırpılır. Bileşeni etikete sığacak şekilde yeniden boyutlandırmanız gerekir.

Bileşenleri boyutlandırmayla ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde tek tek girişlerine bakın.

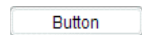
Canlı Önizleme

Varsayılan olarak etkinleştirilen Canlı Önizleme özelliği, yayınlanan Flash içeriğinde görüntülediğinizde bileşenleri Sahne Alanı'nda görüntülemenize olanak sağlar; bileşenler yaklaşık boyutlarında görüntülenir.

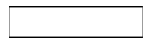
Canlı Önizleme özelliğini etkinleştirmek veya devre dışı bırakmak için:

- Kontrol Et > Canlı Önizlemeyi Etkinleştir seçeneklerini belirleyin. Seçeneğin yanındaki onay işareti, o seçeneğin etkinleştirildiğini belirtir.

Canlı önizleme, farklı bileşenler için farklı parametreler yansıtır. Hangi bileşen parametrelerinin canlı görünümde yansıtıldığıyla ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde her bileşenin girişine bakın.



Canlı Önizleme özelliği etkinleştirilmiş bir Button bileşeni



Canlı Önizleme özelliği devre dışı bırakılmış bir Button bileşeni

Canlı Önizlemede bileşenler çalışmaz. İşlevselliği test etmek için, Kontrol Et > Filmi Test Et komutunu kullanmanız gerekir.

Olayları İşleme

Her bileşen, kullanıcı kendisiyle etkileşim kurduğunda olaylar yayımlar. Örneğin, kullanıcı bir Button bileşenini tıklattığında, bu bileşen bir `MouseEvent.CLICK` olayı gönderir ve kullanıcı List bileşeninde bir öğeyi seçtiğinde List bileşeni bir `Event` ögesi gönderir.`CHANGE` olayı. Ayrıca `UILoader` örneği için içerik yüklemesi bittiğinde, `Event.COMPLETE` olayı oluştuğunda olduğu gibi, bir bileşende önemli bir şey olduğunda da olay gerçekleşebilir. Bir olayı işlemek için, olay gerçekleştiğinde çalıştırılan `ActionScript` kodu yazarsınız.

Bileşenin olayları arasında, bileşenin miras aldığı herhangi bir sınıfın olayları da yer alır. Başka bir deyişle, `UIComponent` sınıfı `ActionScript 3.0` Kullanıcı Arabirimi bileşenlerinin taban sınıfı olduğundan, tüm `ActionScript 3.0` Kullanıcı Arabirimi bileşenleri, `UIComponent` sınıfının olaylarını miras alır. Bir bileşenin yayınladığı olayların listesini görmek için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde bileşenin sınıf girişinin Olaylar bölümüne bakın.

`ActionScript 3.0`'da olay işlemenin tam açıklaması için, bkz. *ActionScript 3.0'da Programlama*.

Olay dinleyicileri hakkında

Aşağıdaki temel noktalar, `ActionScript 3.0` bileşenleri için olay işleminde geçerlidir:

- Tüm olaylar, bir bileşen sınıfının örneği tarafından yayımlanır. Bileşen örneği *yayımlayıcıdır*.
- Bileşen örneği için `addEventListener()` yöntemini çağırarak bir olay *dinleyicisi* kaydedersiniz. Örneğin, şu kod satırı, `aButton` Button örneğine `MouseEvent.CLICK` olayı için bir dinleyici ekler:

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler);
```

`addEventListener()` yönteminin ikinci parametresi, olay gerçekleştiğinde çağrılacak işlevin adını (`clickHandler`) kaydeder. Bu işlev, *geri çağrı işlevi* olarak da ifade edilir.

- Tek bir bileşen örneğine birden çok dinleyici kaydedebilirsiniz.

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler1);  
aButton.addEventListener(MouseEvent.CLICK, clickHandler2);
```

- Birden çok bileşen örneğine tek bir dinleyici kaydedebilirsiniz.

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler1);  
bButton.addEventListener(MouseEvent.CLICK, clickHandler1);
```

- Olay işleyici işlevi, olay türü ve olayı yayımlayan örnek hakkında bilgi içeren bir olay nesnesine iletilir. Daha fazla bilgi için, bkz. "[Olay nesnesi hakkında](#)" sayfa 23.
- Uygulama sona erinceye kadar veya siz `removeEventListener()` yöntemini kullanıp açıkça uygulamayı kaldırıncaya kadar dinleyici etkin olarak kalır. Örneğin, şu kod satırı, `aButton` ögesinde `MouseEvent.CLICK` olayı için dinleyiciyi kaldırır:

```
aButton.removeEventListener(MouseEvent.CLICK, clickHandler);
```

Olay nesnesi hakkında

Olay nesnesi, `Event` nesnesi sınıfından miras alır ve olay hakkında önemli bilgiler sağlayan `target` ve `type` özellikleri dahil olmak üzere, gerçekleşen olay hakkında bilgi içeren özelliklere sahiptir:

Özellik	Açıklama
<code>type</code>	Olayın türünü belirten bir dize.
<code>target</code>	Olayı yayımlayan bileşen örneğinin başvurusu.

Bir olayın ek özellikleri olduğunda, *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* bölümünde olayın sınıf açıklamasında listelenirler.

Olay gerçekleştiğinde, olay nesnesi otomatik olarak oluşturulur ve olay işleyici işlevine iletilir.

Yayınlanan olayın adına veya olayı yayınlayan bileşenin örnek adına erişmek için, işlevin içinde olay nesnesini kullanabilirsiniz. Örnek adından diğer bileşen özelliklerine erişebilirsiniz. Örneğin, şu kod, `aButton` ögesinin `label` özelliğine erişmek ve bu özelliği Çıktı panelinde görüntülemek için, `evtObj` olay nesnesinin `target` özelliğini kullanır:

```
import fl.controls.Button;
import flash.events.MouseEvent;

var aButton:Button = new Button();
aButton.label = "Submit";
addChild(aButton);
aButton.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(evtObj:MouseEvent) {
    trace("The " + evtObj.target.label + " button was clicked");
}
```

Görüntüleme listesiyle çalışma

Tüm ActionScript 3.0 bileşenleri, `DisplayObject` sınıfından miras alır ve böylece görüntüleme listesiyle etkileşim kurmak için bu sınıfın yöntem ve özelliklerine erişir. *Görüntüleme listesi*, bir uygulamada görüntülenen nesnelere ve görsel öğelerin hiyerarşisidir. Bu hiyerarşi şu öğeleri içerir:

- Üst düzey kap olan Sahne Alanı
- Arasında şekiller, `MovieClip` öğeleri, metin alanları, vb'nin yer aldığı görüntüleme nesnelere
- Alt görüntüleme nesnelere içerebilen özel görüntüleme nesnelere niteliğindeki görüntüleme nesnesi kapları.

Görüntüleme listesindeki nesnelere sırası, bu öğelerin üst öğe kabındaki derinliklerini belirler. Bir nesnenin derinliği, o nesnenin Sahne Alanı'nda ya da görüntüleme kabında yukarıdan aşağıya veya önden arkaya konumunu ifade eder. Derinlik sırası, nesne örtüştüğünde belirgin olur ancak örtüşmediğinde de vardır. Görüntüleme listesindeki her nesne, Sahne Alanı'nda karşılık gelen bir derinliğe sahiptir. Nesneyi diğer nesnelere önüne veya arkasına yerleştirerek bir nesnenin derinliğini değiştirmek istiyorsanız, nesnenin konumunu görüntüleme listesinde değiştirmeniz gerekir. Görüntüleme listesindeki nesnelere varsayılan sıralaması, Sahne Alanı'na yerleştirildikleri sıradır. Görüntüleme listesindeki 0 konumu, derinlik sıralamasında en altta yer alan nesnedir.

Görüntüleme listesine bileşen ekleme

Kabın `addChild()` veya `addChildAt()` yöntemini çağırarak `DisplayObjectContainer` nesnesine bir nesne ekleyebilirsiniz. Sahne Alanı olması durumunda, geliştirme sırasında nesne oluşturarak veya bileşenler olması durumunda, nesneyi Bileşenler panelinden Sahne Alanı'na sürükleyerek görüntüleme listesine bir nesne ekleyebilirsiniz. ActionScript ile kaba bir nesne eklemek için öncelikle `new` operatörüyle örneğin yapısını çağırarak bir örnek oluşturun ve sonra bu örneği Sahne Alanı'na ve görüntüleme listesine yerleştirmek için `addChild()` ya da `addChildAt()` yöntemini çağırın. `addChild()` yöntemi, nesneyi görüntüleme listesindeki bir sonraki konuma

yerleştirirken, `addChildAt()` yöntemi de nesnenin ekleneceği konumu belirtir. Önceden doldurulmuş bir konum belirtirseniz, o konumdaki nesne ve daha yüksek konumlardaki nesnelere 1 düzey yukarı taşınır. `DisplayObjectContainer` nesnesinin `numChildren` özelliği, nesnenin içerdiği görüntüleme nesnelerinin sayısını içerir. `getChildAt()` yöntemini çağırıp konumu belirterek veya nesnenin adını biliyorsanız, `getChildByName()` yöntemini çağırarak görüntüleme listesinden bir nesne alabilirsiniz.

Not: Görüntüleme listesinde bileşene adıyla erişmek istiyorsanız, *ActionScript* ile bileşeni eklediğinizde, bileşenin ad özelliğine bir ad atmanız gerekir.

Aşağıdaki örnek, görüntüleme listesinde üç bileşenin adını ve konumunu görüntüler. İlk olarak, bir `NumericStepper`, bir `Button` ve bir `ComboBox` öğesini birbiriyle örtüşecek şekilde Sahne Alanı'na sürükleyin ve bunlara **aNs**, **aButton** ve **aCb** örnek adlarını verin. Daha sonra Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
var i:int = 0;
while(i < numChildren) {
    trace(getChildAt(i).name + " is at position: " + i++);
}
```

Çıktı panelinde şu satırları görmemiz gerekir:

```
aNs is at position: 0
aButton is at position: 1
aCb is at position: 2
```

Görüntüleme listesinde bileşeni taşıma

`addChildAt()` yöntemini çağırıp bir nesne adı ve nesneyi yerleştirmek istediğiniz konumu yöntemin parametreleri olarak sağlayarak o nesnenin görüntüleme listesindeki konumunu ve görüntüleme derinliğini değiştirebilirsiniz. Örneğin, `NumericStepper` öğesini en üste yerleştirmek için bir önceki örneğe şu kodu ekleyin ve bileşenin görüntü listesindeki yeni konumlarını görüntülemek için döngüyü yineleyin:

```
this.addChildAt(aNs, numChildren - 1);
i = 0;
while(i < numChildren) {
    trace(getChildAt(i).name + " is at position: " + i++);
}
```

Çıktı panelinde şunları görmemiz gerekir:

```
aNs is at position: 0
aButton is at position: 1
aCb is at position: 2
aButton is at position: 0
aCb is at position: 1
aNs is at position: 2
```

`NumericStepper` öğesinin de ekrandaki diğer bileşenlerin önünde görüntülenmesi gerekir.

Listedeki birinci konum 0 olsa da, `numChildren` öğesinin görüntüleme listesindeki nesnelerin sayısı (1 ile n arasında) olduğunu unutmayın. Bu nedenle, listede üç öğe varsa, üçüncü nesnenin dizin konumu 2 olur. Başka bir deyişle, görüntüleme derinliği için görüntüleme listesindeki son konuma veya en üst nesneye `numChildren - 1` olarak başvurabilirsiniz.

Görüntüleme listesinden bileşeni kaldırma

`removeChild()` ve `removeChildAt()` yöntemleriyle görüntüleme nesnesi kabından ve görüntüleme listesinden bir bileşeni kaldıracabilirsiniz. Aşağıdaki örnek, Sahne Alanı'nda birbirinin önüne üç Button bileşeni yerleştirir ve bunların her biri için bir olay dinleyicisi ekler. Her bir Button öğesini tıklattığınızda, olay işleyici o Button öğesini görüntüleme listesinden ve Sahne Alanı'ndan kaldırır.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 Button öğesini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu ekleyin:

```
import fl.controls.Button;

var i:int = 0;
while(i++ < 3) {
    makeButton(i);
}
function removeButton(event:MouseEvent):void {
    removeChildAt(numChildren - 1);
}
function makeButton(num) {
    var aButton:Button = new Button();
    aButton.name = "Button" + num;
    aButton.label = aButton.name;
    aButton.move(200, 200);
    addChild(aButton);
    aButton.addEventListener(MouseEvent.CLICK, removeButton);
}
```

Görüntüleme listesinin tam açıklaması için, bkz. "Görüntü programlama", *ActionScript 3.0'i Programlama*.

FocusManager ile çalışma

Kullanıcı Flash uygulamasında Sekme tuşuna bastığında veya uygulamada herhangi bir yeri tıklattığında, FocusManager sınıfı hangi bileşenin girdi odağını alacağını belirler. Bir bileşen oluşturmuyorsanız FocusManager öğesini etkinleştirmek için uygulamaya FocusManager örneğini eklemenize veya herhangi bir kod yazmanıza gerek yoktur.

Bir RadioButton nesnesi odak alırsa, FocusManager öğesi, o nesneyi veya aynı `groupName` değerine sahip tüm nesnelere bakar ve `selected` özelliği `true` değerine ayarlanmış şekilde nesnede odağı ayarlar.

Her kalıcı Window bileşeni bir FocusManager örneği içerir, böylece o penceredeki denetimler de bu örneğin kendi sekme kümesi olur. Böylece kullanıcının Sekme tuşuna basarak yanlışlıkla diğer pencerelerdeki bileşenlere gitmesi önlenir.

FocusManager öğesi, varsayılan gezinme şeması veya *sekme döngüsü* olarak kaptaki öğelerin derinlik düzeyini (veya z sıralamasını) kullanır. Kullanıcılar tipik olarak Sekme tuşunu kullanarak sekme döngüsünde gezinir; bu durumda odak bulunduğu birinci bileşenden sonuncuya gider ve sonra tekrar birinciye geri gelir. Derinlik düzeyleri öncelikli olarak bileşenlerin Sahne Alanı'na sürüklendiği sıralamaya göre ayarlanır; ancak son z sıralamasını belirlemek için Değiştir > Yerleştir > En Öne Getir/En Alta Gönder komutlarını da kullanabilirsiniz. Derinlik düzeyleri hakkında daha fazla bilgi almak için bkz. "Görüntüleme listesiyle çalışma" sayfa 24.

Bir uygulamada bileşen örneğine odak vermek için `setFocus()` yöntemini çağırabilirsiniz. Örneğin, aşağıdaki örnek, geçerli kap (`this`) için bir FocusManager örneği oluşturur ve `aButton` Button örneğine odağı verir.

```
var fm:FocusManager = new FocusManager(this);  
fm.setFocus(aButton);
```

`getFocus()` yöntemini çağırarak hangi bileşenin odağa sahip olduğunu belirleyebilir ve `getNextFocusManagerComponent()` yöntemini çağırarak da bir sonraki defa sekme döngüsünde hangi bileşenin odağı alacağını belirleyebilirsiniz. Aşağıdaki örnekte, bir `CheckBox`, bir `RadioButton` ve bir `Button` Sahne Alanı'ndadır ve her bileşen `MouseEvent.CLICK` ve `FocusEvent.MOUSE_FOCUS_CHANGE` olayları için dinleyicilere sahiptir. `MouseEvent.CLICK` olayı gerçekleştiğinde, kullanıcı bileşeni tıklattığından, `showFocus()` işlevi bir sonraki defa sekme döngüsünde hangi bileşenin odağı alacağını belirlemek için `getNextFocusManagerComponent()` yöntemini çağırır. Daha sonra da o bileşene odağı vermek için `setFocus()` yöntemini çağırır. `FocusEvent.MOUSE_FOCUS_CHANGE` olayı gerçekleştiğinde, `fc()` işlevi bu olayın gerçekleştiği bileşenin adını görüntüler. Kullanıcı sekme döngüsünde sonraki bileşenden başka bir bileşeni tıklattığında bu olay tetiklenir.

```
// This example assumes a CheckBox (aCh), a RadioButton (aRb) and a Button  
// (aButton) have been placed on the Stage.
```

```
import fl.managers.FocusManager;  
import flash.display.InteractiveObject;  
  
var fm:FocusManager = new FocusManager(this);  
  
aCh.addEventListener(MouseEvent.CLICK, showFocus);  
aRb.addEventListener(MouseEvent.CLICK, showFocus);  
aButton.addEventListener(MouseEvent.CLICK, showFocus);  
aCh.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);  
aRb.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);  
aButton.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);  
  
function showFocus(event:MouseEvent):void {  
    var nextComponent:InteractiveObject = fm.getNextFocusManagerComponent();  
    trace("Next component in tab loop is: " + nextComponent.name);  
    fm.setFocus(nextComponent);  
}  
  
function fc(fe:FocusEvent):void {  
    trace("Focus Change: " + fe.target.name);  
}
```

Kullanıcı Enter (Windows) veya Return (Macintosh) tuşuna bastığında odağı alan bir `Button` öğesi oluşturmak için, aşağıdaki kodda olduğu gibi, `FocusManager.defaultButton` özelliğini varsayılan `Button` olmasını istediğiniz `Button` örneğine ayarlayın:

```
import fl.managers.FocusManager;  
  
var fm:FocusManager = new FocusManager(this);  
fm.defaultButton = okButton;
```

`FocusManager` sınıfı varsayılan Flash Player odak dikdörtgenini geçersiz kılar ve yuvarlak köşeli bir özel odak dikdörtgeni çizer.

Flash uygulamasında bir odak şeması oluşturmayla ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FocusManager` class öğesine bakın. Özel bir odak yöneticisi oluşturmak için, `IFocusManager` arabirimini uygulayan bir sınıf oluşturmanız gerekir. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `IFocusManager` öğesine bakın.

List tabanlı bileşenler

List, DataGrid ve TileList bileşenlerinin tümü SelectableList taban sınıfından miras alır. Bu nedenle, bu bileşenler List tabanlı olarak değerlendirilir. ComboBox bir metin kutusu ve List ögesi içerir, bu nedenle o da List tabanlı bir bileşendir.

List ögesi satırlardan oluşur. DataGrid ve TileList öğeleri, birden çok sütuna bölünebilen satırlardan oluşur. Bir satır ile sütunun kesişimi hücredir. Tek bir satırlar sütunu olan List ögesinde, her satır bir hücredir. Hücrelerle ilgili şu iki nokta önemlidir:

- Hücrelerin barındırdığı veri değerlerine öğeler denir. Öğeler, bir List ögesindeki bilgi birimlerini saklamak için kullanılan ActionScript nesnesidir. List ögesi, tüm dizinli boşluklarının öğe olduğu bir dizi olarak düşünülebilir. Bir List bileşeninde öğe, tipik olarak görüntülenen bir `label` özelliğine ve veri saklamak için kullanılan bir `data` özelliğine sahip bir nesnedir. *Veri sağlayıcı*, List bileşenindeki öğelerin bir veri modelidir. Veri sağlayıcı, List tabanlı bir bileşeni bileşenin `dataProvider` özelliğine atayarak doldurmanıza olanak sağlar.
- Hücre, metinden görüntülere, MovieClip öğelerine veya oluşturabileceğiniz tüm sınıflara kadar birçok farklı veri türlerini barındırabilir. Bu nedenle, hücrenin içeriğine uygun şekilde çizilmesi veya oluşturulması gerekir. Bunun sonucunda, List tabanlı bileşenler, hücrelerini oluşturmak için bir *hücre oluşturucuya* sahiptir. DataGrid olması durumunda, her sütun ayrıca bir `cellRenderer` özelliğine sahip olan `DataGridColumn` nesnesidir, böylece her sütun içeriğine uygun şekilde oluşturulabilir.

Tüm List tabanlı bileşenler, bu bileşenlerin hücrelerini yüklemek ve oluşturmak için ayarlayabileceğiniz `cellRenderer` ve `dataProvider` özelliklerine sahiptir. Bu özellikleri kullanma ve List tabanlı bileşenlerle çalışma hakkında bilgi almak için, bkz. "[DataProvider ile çalışma](#)" sayfa 28 ve "[CellRenderer ile çalışma](#)" sayfa 36.

DataProvider ile çalışma

DataProvider, ComboBox, DataGrid, List ve TileList bileşenlerine veri sağlamak için kullanabileceği bir veri kaynağıdır. Bu bileşen sınıflarının her biri, bileşenin hücrelerini verilerle doldurmak için bir DataProvider nesnesi atayabileceğiniz `dataProvider` özelliğine sahiptir. Tipik olarak, veri sağlayıcı, Array veya XML nesnesi gibi bir veri koleksiyonudur.

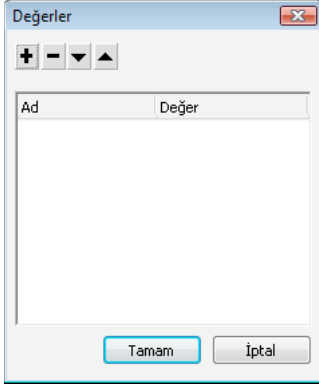
DataProvider oluşturma

ComboBox, List ve TileList bileşenleri için, geliştirme ortamında `dataProvider` parametresini kullanarak bir DataProvider ögesi oluşturabilirsiniz. DataGrid bileşeninin birçok sütunu olabileceğinden ve dolayısıyla veri sağlayıcısı daha karmaşık olabileceğinden bu bileşenin Özellik denetçisinde bir `dataProvider` parametresi yoktur. DataGrid ögesinin yanı sıra bu bileşenler için de DataProvider oluşturmak üzere ActionScript kullanabilirsiniz.

dataProvider parametresini kullanma

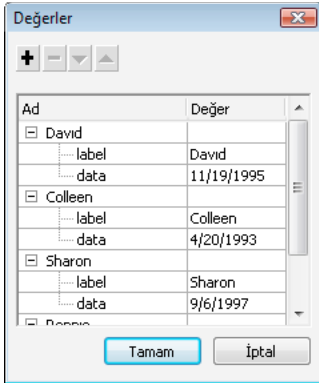
Özellik denetçisinin veya Bileşen denetçisinin Parametreler sekmesinde `dataProvider` parametresini tıklayarak ComboBox, List ve TileList bileşenleri için basit bir veri sağlayıcı oluşturabilirsiniz.

Başlangıçta boş bir Array ögesi görüntüleyen değer hücrelerini çift tıklayarsanız, veri sağlayıcısı oluşturmak için birden çok etiket ve veri değeri girmenize olanak sağlayan Değerler iletişim kutusunu açarsınız.



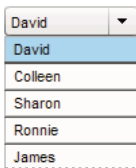
dataProvider için Değerler iletişim kutusu

dataProvider bileşenine bir öge eklemek için artı işaretini tıklatın. Bir ögeyi silmek için eksi işaretini tıklatın. Seçili ögeyi listede bir öge yukarı taşımak için yukarı oku veya seçili ögeyi listede bir öge aşağı taşımak için aşağı oku tıklatın. Aşağıdaki resimde, çocukların adlarının ve doğum tarihlerinin bir listesini oluşturan Değerler iletişim kutusu gösterilmektedir.



Verilerin yer aldığı Değerler iletişim kutusu

Oluşturduğunuz Array ögesi, etiket ve değer alanları çiftlerinden oluşur. Etiket alanları, label ve data iken, değer alanları da çocukların adları ve doğum tarihleridir. Etiket alanı, List ögesinde görüntülenen içeriği tanımlar, bu örnekte içerik çocukların adlarıdır. Sonuçta elde edilen ComboBox şöyle görünür:



DataProvider tarafından doldurulan ComboBox

Veri ekleme işlemini bitirdiğinizde, iletişim kutusunu kapatmak için Tamam'ı tıklatın. dataProvider parametresindeki Array ögesi şimdi oluşturduğunuz öğelerle doldurulur.

allowMultipleSelection	false
dataProvider	{label:David,data:11/19/1995},{label:Colleen,data:4/20/1993},{label:Sharon,data:9/6/1997}
horizontalLineScrollSize	1
horizontalPageScrollSize	0
horizontalScrollPolicy	auto
verticalLineScrollSize	1

verilerin yer aldığı `dataProvider` parametresi

Bileşenin `dataProvider` özelliğine erişmek için ActionScript kullanarak oluşturduğunuz etiket ve veri değerlerine erişebilirsiniz.

ActionScript kullanarak DataProvider oluşturma

Bir Array veya XML nesnesinde veri oluşturup DataProvider yapıcısına `value` parametresi olarak nesneyi sağlayarak bir DataProvider oluşturabilirsiniz.

Not: ActionScript 3.0'da, özellik bir DataProvider nesnesi olarak tanımlandığından ve yalnızca DataProvider türünde bir nesne alabildiğinden, bir Array veya XML nesnesini doğrudan bir dataProvider özelliğine atayamazsınız.

Aşağıdaki örnek, tek bir satırlar sütunu olan List bileşenini birçok çocuk adı ve doğum tarihiyle doldurur. Bu örnek, `items` Array ögesinde listeyi tanımlar ve DataProvider örneğini (yeni DataProvider ö eler) oluşturduğunda bunu parametre olarak sunup List bileşeninin `dataProvider` özelliğine atar.

```
import fl.controls.List;
import fl.data.DataProvider;

var aList:List = new List();
var items:Array = [
    {label:"David", data:"11/19/1995"},
    {label:"Colleen", data:"4/20/1993"},
    {label:"Sharon", data:"9/06/1997"},
    {label:"Ronnie", data:"7/6/1993"},
    {label:"James", data:"2/15/1994"},
];
aList.dataProvider = new DataProvider(items);
addChild(aList);
aList.move(150,150);
```

Array ögesi, etiket ve değer alanları çiftlerinden oluşur. Etiket alanları, `label` ve `data` iken, değer alanları da çocukların adları ve doğum tarihleridir. Etiket alanı, List ögesinde görüntülenen içeriği tanımlar, bu örnekte içerik çocukların adlarıdır. Sonuçta elde edilen List şöyle görünür:

David
Colleen
Sharon
Ronnie
James

DataProvider tarafından doldurulmuş bir List ögesi

Kullanıcı üzerine tıklatıp listedeki bir ögeyi seçtiğinde, veri alanının değeri kullanılabilir duruma geçer ve bir `change` olayı başlatılır. Aşağıdaki örnek, kullanıcı List ögesinde bir ad seçtiğinde çocuğun doğum gününü görüntülemek için önceki örneğe bir TextArea (`aTa`) ve bir olay işleyici (`changeHandler`) ekler.

```
import fl.controls.List;
import fl.controls.TextArea;
import flash.events.Event;
import fl.data.DataProvider;

var aList:List = new List();
var aTa:TextArea = new TextArea();
var items:Array = [
    {label:"David", data:"1/19/1995"},
    {label:"Colleen", data:"4/20/1993"},
    {label:"Sharon", data:"9/06/1994"},
    {label:"Ronnie", data:"7/6/1993"},
    {label:"James", data:"2/15/1994"},
];
aList.dataProvider = new DataProvider(items);

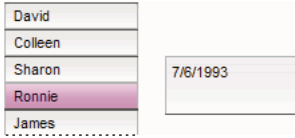
addChild(aList);
addChild(aTa);

aList.move(150,150);
aTa.move(150, 260);

aList.addEventListener(Event.CHANGE, changeHandler);

function changeHandler(event:Event):void {
    aTa.text = event.target.selectedItem.data;
};
```

Şimdi kullanıcı List öğesinde bir çocuğun adını seçtiğinde, aşağıdaki resimde gösterildiği gibi çocuğun doğum günü TextArea bileşeninde görüntülenir. Bu, `changeHandler()` işlevi tarafından TextArea (`aTa.text`) öğesinin `text` özelliğinin seçili öğedeki veri alanının değerine (`event.target.selectedItem.data`) ayarlanmasıyla gerçekleştirilir. `event.target` özelliği, olayı tetikleyen nesnedir (bu durumda List).



List bileşeninin DataProvider öğesinden veri alanını görüntüleme

DataProvider öğesine metin dışında da veriler dahil edebilirsiniz. Aşağıdaki örnek, TileList öğesine veri sağlayan DataProvider öğesine MovieClip öğelerini dahil eder. Bu, MovieClip (renkli kutu) oluşturduktan sonra her bir öğeyi eklemek için `addItem()` öğesini çağırarak DataProvider bileşenini oluşturur.

```
import fl.data.DataProvider;
import flash.display.DisplayObject;

var aBox:MovieClip = new MovieClip();
var i:uint = 0;
var colors:Array = new Array(0x000000, 0xFF0000, 0x0000CC, 0x00CC00, 0xFFFF00);
var colorNames:Array = new Array("Midnight", "Cranberry", "Sky", "Forest", "July");
var dp:DataProvider = new DataProvider();
for(i=0; i < colors.length; i++) {
    drawBox(aBox, colors[i]); // draw box w next color in array
    dp.addItem( {label:colorNames[i], source:aBox} );
}
aTl.dataProvider = dp;
aTl.columnWidth = 110;
aTl.rowHeight = 130;
aTl.setSize(280,150);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(0, 0, 100, 100);
    box.graphics.endFill();
}
```

Bir DataProvider nesnesini doldurmak için de XML verilerini (dizi yerine) kullanabilirsiniz. Örneğin, şu kod, verileri employeesXML adındaki bir XML nesnesine saklar ve sonra DataProvider() yapıcısı işlevinin değer parametresi olarak bu nesneyi iletir:

```
import fl.controls.DataGrid;
import fl.data.DataProvider;

var aDg:DataGrid = new DataGrid();
addChild(aDg);

var employeesXML:XML =
    <employees>
        <employee Name="Edna" ID="22" />
        <employee Name="Stu" ID="23" />
    </employees>;

var myDP:DataProvider = new DataProvider(employeesXML);

aDg.columns = ["Name", "ID"];
aDg.dataProvider = myDP;
```

Verileri önceki kodda olduğu gibi XML verilerinin nitelikleri olarak veya aşağıdaki kodda olduğu gibi XML verilerinin özellikleri olarak sağlayabilirsiniz:

```
var employeesXML:XML =
    <employees>
        <employee>
            <Name>Edna</Name>
            <ID>22</ID>
        </employee>
        <employee>
            <Name>Stu</Name>
            <ID>23</ID>
        </employee>
    </employees>;
```

DataProvider ögesi ayrıca kendisine erişip bu ögeyi işlemenizi sağlayan yöntemler ve özellikler kümesine de sahiptir. DataProvider içinde öğeler eklemek, kaldırmak, değiştirmek, sıralamak ve birleştirmek için DataProvider API'sini kullanabilirsiniz.

DataProvider ögesini işleme

addItem() ve addItemAt() yöntemleriyle DataProvider bileşenine öğeler ekleyebilirsiniz. Aşağıdaki örnek, kullanıcının düzenlenebilir bir ComboBox bileşeninin metin alanına girdiği öğeleri ekler. Bu, ComboBox bileşeninin Sahne Alanı dışına sürüklendiğini ve bu bileşene aCb örnek adının verildiğini varsayar.

```
import fl.data.DataProvider;
import fl.events.ComponentEvent;

var items:Array = [
    {label:"Roger"},
    {label:"Carolyn"},
    {label:"Darrell"},
    {label:"Rebecca"},
    {label:"Natalie"},
    {label:"Mitchell"},
];
aCb.dataProvider = new DataProvider(items);

aCb.addEventListener(ComponentEvent.ENTER, newItemHandler);

function newItemHandler(event:ComponentEvent):void {
    var newRow:int = event.target.length + 1;
    event.target.addItemAt({label:event.target.selectedLabel},
        event.target.length);
}
```

DataProvider yoluyla da bir bileşendeki öğeleri değiştirebilir ve kaldırabilirsiniz. Aşağıdaki örnek, iki ayrı List bileşenini (listA ve listB) uygular ve Sync etiketli bir Button ögesi sağlar. Kullanıcı Button ögesini tıklattığında, örnek listB içindeki öğeleri listA içindeki öğelerle değiştirmek için replaceItemAt() yöntemini kullanır. listA ögesi listB ögesinden daha uzun olursa, örnek, listB ögesine fazladan öğe eklemek için addItem() yöntemini çağırır. listB ögesi listA ögesinden daha uzun olursa, örnek, listB ögesindeki fazladan öğeleri kaldırmak için removeItemAt() yöntemini çağırır.

```
// Requires the List and Button components to be in the library

import fl.controls.List;
import fl.controls.Button;
import flash.events.Event;
import fl.data.DataProvider;

var listA:List = new List();
var listB:List = new List();
var syncButton:Button = new Button();
syncButton.label = "Sync";

var itemsA:Array = [
    {label:"David"},
    {label:"Colleen"},
    {label:"Sharon"},
    {label:"Ronnie"},
    {label:"James"},
];
var itemsB:Array = [
    {label:"Roger"},
    {label:"Carolyn"},
    {label:"Darrell"},
    {label:"Rebecca"},
    {label:"Natalie"},
    {label:"Mitchell"},
];
listA.dataProvider = new DataProvider(itemsA);
listB.dataProvider = new DataProvider(itemsB);

addChild(listA);
addChild(listB);
addChild(syncButton);

listA.move(100, 100);
listB.move(250, 100);
syncButton.move(175, 220);

syncButton.addEventListener(MouseEvent.CLICK, syncHandler);

function syncHandler(event:MouseEvent):void {
    var i:uint = 0;
    if(listA.length > listB.length) { //if listA is longer, add items to B
        while(i < listB.length) {
            listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        }
    }
}
```

```
        ++i;
    }
    while(i < listA.length) {
        listB.dataProvider.addItem(listA.dataProvider.getItemAt(i++));
    }
} else if(listA.length == listB.length) { //if listA and listB are equal length
    while(i < listB.length) {
        listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        ++i;
    }
} else { //if listB is longer, remove extra items from B
    while(i < listA.length) {
        listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        ++i;
    }
    while(i < listB.length) {
        listB.dataProvider.removeItemAt(i++);
    }
}
}
```

Ayrıca `merge()`, `sort()` ve `sortOn()` yöntemlerini kullanarak da `DataProvider` ögesini birleştirip sıralayabilirsiniz. Aşağıdaki örnek, iki `DataGrid` örneğini (`aDg` ve `bDg`) iki beysbol takımının kısmi görev listesiyle doldurur. `Merge` etiketine sahip bir `Button` ögesi ekler ve kullanıcı bunun üzerine tıklattığında, olay işleyici (`mrgHandler`) `bDg` görev listesini `aDg` görev listesiyle birleştirir ve sonuçta elde edilen `DataGrid` ögesini `Ad` sütununda sıralar.

```
import fl.data.DataProvider;
import fl.controls.DataGrid;
import fl.controls.Button;

var aDg:DataGrid = new DataGrid();
var bDg:DataGrid = new DataGrid();
var mrgButton:Button = new Button();
addChild(aDg);
addChild(bDg);
addChild(mrgButton);
bldRosterGrid(aDg);
bldRosterGrid(bDg);
var aRoster:Array = new Array();
var bRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter", Bats:"R", Throws:"R", Year:"So", Home: "Redlands, CA"},
    {Name:"Sue Pennypacker", Bats:"L", Throws:"R", Year:"Fr", Home: "Athens, GA"},
    {Name:"Jill Smithfield", Bats:"R", Throws:"L", Year:"Sr", Home: "Spokane, WA"},
    {Name:"Shirley Goth", Bats:"R", Throws:"R", Year:"Sr", Home: "Carson, NV"}
];
bRoster = [
    {Name:"Angelina Davis", Bats:"R", Throws:"R", Year:"So", Home: "Odessa, TX"},
    {Name:"Maria Santiago", Bats:"L", Throws:"L", Year:"Sr", Home: "Tacoma, WA"},
    {Name:"Debbie Ferguson", Bats:"R", Throws:"R", Year: "Jr", Home: "Bend, OR"}
];
aDg.dataProvider = new DataProvider(aRoster);
bDg.dataProvider = new DataProvider(bRoster);
aDg.move(50,50);
aDg.rowCount = aDg.length;
```

```
bDg.move(50,200);
bDg.rowCount = bDg.length;
mrgButton.label = "Merge";
mrgButton.move(200, 315);
mrgButton.addEventListener(MouseEvent.CLICK, mrgHandler);

function bldRosterGrid(dg:DataGrid) {
    dg.setSize(400, 300);
    dg.columns = ["Name", "Bats", "Throws", "Year", "Home"];
    dg.columns[0].width = 120;
    dg.columns[1].width = 50;
    dg.columns[2].width = 50;
    dg.columns[3].width = 40;
    dg.columns[4].width = 120;
};

function mrgHandler(event:MouseEvent):void {
    aDg.dataProvider.merge(bDg.dataProvider);
    aDg.dataProvider.sortOn("Name");
}
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki DataProvider sınıfına bakın.

CellRenderler ile çalışma

CellRenderler, List, DataGrid, TileList ve ComboBox gibi List tabanlı bileşenlerin kendi satırları için özel hücre içeriğini işlemek ve görüntülemek için kullandığı bir sınıftır. Özelleştirilmiş bir hücre, metin, CheckBox gibi önceden oluşturulmuş bir bileşen veya oluşturabileceğiniz herhangi bir görüntüleme nesnesini içerebilir. Özel CellRenderler öğesini kullanarak veri oluşturmak için, CellRenderler sınıfını genişletebilir veya kendi özel CellRenderler sınıfınızı oluşturmak için ICellRenderler arabirimini uygulayabilirsiniz.

List, DataGrid, TileList ve ComboBox sınıfları, SelectableList sınıfının alt sınıflarıdır. SelectableList sınıfı bir cellRenderler stili içerir. Bu stil, bileşenin hücre oluşturmak için kullandığı görüntüleme nesnesini tanımlar.

List nesnesinin setRenderlerStyle() yöntemini çağırarak CellRenderler tarafından kullanılan stillerin biçimlendirmesini ayarlayabilirsiniz (bkz. “[Hücreleri biçimlendirme](#)” sayfa 36). Veya CellRenderler olarak kullanmak için özel bir sınıf tanımlayabilirsiniz (bkz. “[Özel bir CellRenderler sınıfı tanımlama](#)” sayfa 37).

Hücreleri biçimlendirme

CellRenderler sınıfı, hücrenin biçimini denetlemenize olanak sağlayan bir grup stil içerir.

Şu stiller, farklı hücre durumları (devre dışı, basılmış, üzerinde ve basılmamış) için kullanılan kaplamaları tanımlamanıza olanak sağlar:

- disabledSkin ve selectedDisabledSkin
- downSkin ve selectedDownSkin
- overSkin ve selectedOverSkin
- upSkin ve selectedUpSkin

Aşağıdaki stiller metin biçimlendirmesi için geçerlidir:

- disabledTextFormat
- textFormat
- textPadding

List nesnesinin `setRendererStyle()` yöntemini çağırarak veya `CellRenderer` nesnesinin `setStyle()` yöntemini çağırarak bu stilleri ayarlayabilirsiniz. List nesnesinin `getRendererStyle()` yöntemini çağırarak veya `CellRenderer` nesnesinin `getStyle()` yöntemini çağırarak bu stilleri ayarlayabilirsiniz. List nesnesinin `rendererStyles` özelliği veya `CellRenderer` nesnesinin `getStyleDefinition()` yöntemi yoluyla tüm oluşturucu stillerini tanımlayan bir nesneye de (nesnenin özellikleri denir) erişebilirsiniz.

Bir stili varsayılan değerine sıfırlamak için `clearRendererStyle()` yöntemini çağırabilirsiniz.

Listedeki satırların uzunluğunu almak veya ayarlamak için List nesnesinin `rowHeight` özelliğini kullanın.

Özel bir CellRenderer sınıfı tanımlama

Özel bir CellRenderer tanımlamak için CellRenderer sınıfını genişleten bir sınıf oluşturma

Örneğin, şu kod iki sınıf içerir. `ListSample` sınıfı bir List bileşenini başlatır ve List bileşeni için kullanılacak hücre oluşturucuyu tanımlamak üzere diğer sınıfı (`CustomRenderer`) kullanır. `CustomRenderer` sınıfı, `CellRenderer` sınıfını genişletir.

- 1 Dosya > Yeni'yi seçin.
- 2 Görüntülenen Yeni Belge iletişim kutusunda Flash Dosyası (ActionScript 3.0) öğesini seçip Tamam'ı tıklayın.
- 3 Bileşenler panelini görüntülemek için Pencere > Bileşenler seçeneklerini belirleyin.
- 4 Bileşenler panelinde, bir List bileşenini Sahne Alanı'na sürükleyin.
- 5 Flash uygulaması Özellik denetçisini görüntülemiyorsa, Pencere > Özellikler > Özellikler seçeneklerini belirleyin.
- 6 List bileşeni seçili durumdayken Özellik denetçisinde özellikleri ayarlayın:
 - Örnek Adı: `myList`
 - W (genişlik): 200
 - H (yükseklik): 300
 - X: 20
 - Y: 20
- 7 Zaman Çizelgesi'nde Kare 1'i seçin ve Pencere > Eylemler seçeneklerini belirleyin.
- 8 Eylemler paneline şu komut dosyasını yazın:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Dosya > Kaydet'i seçin. Dosyaya bir ad verin ve Tamam düğmesini tıklayın.
- 10 Dosya > Yeni'yi seçin.
- 11 Yeni Belge iletişim kutusunda ActionScript Dosyası'nı seçip Tamam düğmesini tıklayın.
- 12 Komut dosyası penceresinde, `CustomCellRenderer` sınıfını tanımlamak için şu kodu girin:

```
package {
    import fl.controls.listClasses.CellRenderer;
    import flash.text.TextFormat;
    import flash.filters.BevelFilter;
    public class CustomCellRenderer extends CellRenderer {
        public function CustomCellRenderer() {
            var format:TextFormat = new TextFormat("Verdana", 12);
            setStyle("textFormat", format);
            this.filters = [new BevelFilter()];
        }
    }
}
```

13 Dosya > Kaydet'i seçin. Dosyayı CustomCellRenderer.as olarak adlandırın ve FLA dosyasıyla aynı dizine koyup Tamam düğmesini tıklatın.

14 Kontrol Et > Filmi Test Et'i seçin.

Özel bir CellRenderer ögesini tanımlamak için ICellRenderer arabirimini uygulayan bir sınıfı kullanma

DisplayObject sınıfını miras alıp ICellRenderer arabirimini uygulayan herhangi bir sınıfı kullanarak da CellRenderer ögesini tanımlayabilirsiniz. Örneğin, şu kod iki sınıfı tanımlar. ListSample2 sınıfı, List nesnesini görüntüleme listesine ekler ve CustomRenderer sınıfını kullanmak için CellRenderer ögesini tanımlar. CustomRenderer sınıfı, CheckBox sınıfını (DisplayObject sınıfını genişleten) genişletir ve ICellRenderer arabirimini uygular. CustomRenderer sınıfının ICellRenderer arabiriminde tanımlanan data ve listData özellikleri için getter ve setter yöntemleri tanımladığını unutmayın. ICellRenderer arabiriminde tanımlanan diğer özellik ve yöntemler (selected özelliği ve setSize() yöntemi) önceden CheckBox sınıfında tanımlanmıştır:

- 1 Dosya > Yeni'yi seçin.
- 2 Görüntülenen Yeni Belge iletişim kutusunda Flash Dosyası (ActionScript 3.0) ögesini seçip Tamam'ı tıklatın.
- 3 Bileşenler panelini görüntülemek için Pencere > Bileşenler seçeneklerini belirleyin.
- 4 Bileşenler panelinde, bir List bileşenini Sahne Alanı'na sürükleyin.
- 5 Flash uygulaması Özellik denetçisini görüntülemiyorsa, Pencere > Özellikler > Özellikler seçeneklerini belirleyin.
- 6 List bileşeni seçili durumdayken Özellik denetçisinde özellikleri ayarlayın:

- Örnek Adı: myList
- W (genişlik): 100
- H (yükseklik): 300
- X: 20
- Y: 20

7 Zaman Çizelgesi'nde Kare 1'i seçin ve Pencere > Eylemler seçeneklerini belirleyin.

8 Eylemler paneline şu komut dosyasını yazın:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({name:"Burger", price:"$5.95"});
myList.addItem({name:"Fries", price:"$1.95"});
```

9 Dosya > Kaydet'i seçin. Dosyaya bir ad verin ve Tamam düğmesini tıklatın.

10 Dosya > Yeni'yi seçin.

11 Yeni Belge iletişim kutusunda ActionScript Dosyası'nı seçip Tamam düğmesini tıklatın.

12 Komut dosyası penceresinde, CustomCellRenderere sınıfını tanımlamak için şu kodu girin:

```
package
{
    import fl.controls.CheckBox;
    import fl.controls.listClasses.ICellRenderer;
    import fl.controls.listClasses.ListData;
    public class CustomCellRenderere extends CheckBox implements ICellRenderer {
        private var _listData:ListData;
        private var _data:Object;
        public function CustomCellRenderere() {
        }
        public function set data(d:Object):void {
            _data = d;
            label = d.label;
        }
        public function get data():Object {
            return _data;
        }
        public function set listData(ld:ListData):void {
            _listData = ld;
        }
        public function get listData():ListData {
            return _listData;
        }
    }
}
```

13 Dosya > Kaydet'i seçin. Dosyayı CustomCellRenderere.as olarak adlandırın ve FLA dosyasıyla aynı dizine koyup Tamam düğmesini tıkkatın.

14 Kontrol Et > Filmi Test Et'i seçin.

CellRenderere tanımlamak için sembol kullanma

CellRenderere tanımlamak için kütüphanede bir sembol de kullanabilirsiniz. Sembolün ActionScript için dışa aktarılması ve kütüphane sembolünün sınıf adının ICellRenderere arabirimini uygulayan veya CellRenderere sınıfını (ya da alt sınıflarından birini) genişleten ilişkilendirilmiş bir sınıf dosyasına sahip olması gerekir.

Aşağıdaki örnek, kütüphane sembolünü kullanarak özel bir CellRenderere öğesini tanımlar.

- 1 Dosya > Yeni'yi seçin.
- 2 Görüntülenen Yeni Belge iletişim kutusunda Flash Dosyası (ActionScript 3.0) öğesini seçip Tamam'ı tıkkatın.
- 3 Bileşenler panelini görüntülemek için Pencere > Bileşenler seçeneklerini belirleyin.
- 4 Bileşenler panelinde, bir List bileşenini Sahne Alanı'na sürükleyin.
- 5 Flash uygulaması Özellik denetçisini görüntülemiyorsa, Pencere > Özellikler > Özellikler seçeneklerini belirleyin.
- 6 List bileşeni seçili durumdayken Özellik denetçisinde özellikleri ayarlayın:
 - Örnek Adı: myList
 - W (genişlik): 100
 - H (yükseklik): 400
 - X: 20
 - Y: 20

- 7 Parametreler panelini tıklatın ve dataProvider satırında ikinci sütunu çift tıklatın.
- 8 Değerler iletişim kutusunda, iki veri ögesini (etiketleri label0 ve label1 olarak ayarlanmış şekilde) eklemek için artı işaretini iki defa tıklatın ve sonra da Tamam düğmesini tıklatın.
- 9 Metin aracıyla Sahne Alanı'nda bir metin alanı çizin.
- 10 Metin alanı seçili durumdayken Özellik denetçisinde özellikleri ayarlayın:
 - Metin türü: Dinamik Metin
 - Örnek Adı: textField
 - W (genişlik): 100
 - Font boyutu: 24
 - X: 0
 - Y: 0
- 11 Metin alanı seçili durumdayken, Değiştir > Sembole Dönüştür seçeneklerini belirleyin.
- 12 Sembole Dönüştür iletişim kutusunda şu ayarları yapın ve Tamam'ı tıklatın.
 - Ad: MyCellRenderere
 - Tür: MovieClip
 - ActionScript için Dışa Aktar: Seçili
 - İlk Karede Dışa Aktar: Seçili
 - Sınıf: MyCellRenderere
 - Temel Sınıf: flash.display.MovieClipFlash uygulaması bir ActionScript Sınıf Uyarısı görüntülense, uyarı kutusunda Tamam düğmesini tıklatın.
- 13 Yeni film klibi örneğinin sembolünü Sahne Alanı'ndan silin.
- 14 Zaman Çizelgesi'nde Kare 1'i seçin ve Pencere > Eylemler seçeneklerini belirleyin.
- 15 Eylemler paneline şu komut dosyasını yazın:

```
myList.setStyle("cellRenderere", MyCellRenderere);
```
- 16 Dosya > Kaydet'i seçin. Dosyaya bir ad verin ve Tamam düğmesini tıklatın.
- 17 Dosya > Yeni'yi seçin.
- 18 Yeni Belge iletişim kutusunda ActionScript Dosyası'nı seçip Tamam düğmesini tıklatın.
- 19 Komut dosyası penceresinde, MyCellRenderere sınıfını tanımlamak için şu kodu girin:

```
package {
    import flash.display.MovieClip;
    import flash.filters.GlowFilter;
    import flash.text.TextField;
    import fl.controls.listClasses.ICellRenderer;
    import fl.controls.listClasses.ListData;
    import flash.utils.setInterval;
    public class MyCellRenderer extends MovieClip implements ICellRenderer {
        private var _listData:ListData;
        private var _data:Object;
        private var _selected:Boolean;
        private var glowFilter:GlowFilter;
        public function MyCellRenderer() {
            glowFilter = new GlowFilter(0xFFFF00);
            setInterval(toggleFilter, 200);
        }
        public function set data(d:Object):void {
            _data = d;
            textField.text = d.label;
        }
        public function get data():Object {
            return _data;
        }
        public function set listData(ld:ListData):void {
            _listData = ld;
        }
        public function get listData():ListData {
            return _listData;
        }
        public function set selected(s:Boolean):void {
            _selected = s;
        }
        public function get selected():Boolean {
            return _selected;
        }
        public function setSize(width:Number, height:Number):void {
        }
        public function setStyle(style:String, value:Object):void {
        }
        public function setMouseState(state:String):void{
        }
        private function toggleFilter():void {
            if (textField.filters.length == 0) {
                textField.filters = [glowFilter];
            } else {
                textField.filters = [];
            }
        }
    }
}
```

20 Dosya > Kaydet'i seçin. Dosyayı MyCellRender.as olarak adlandırın ve FLA dosyasıyla aynı dizine koyup Tamam düğmesini tıklayın.

21 Kontrol Et > Filmi Test Et'i seçin.

CellRenderer özellikleri

data özelliği, CellRenderer için ayarlanmış tüm özellikleri içeren bir nesnedir. Örneğin, CheckBox sınıfını genişleten özel bir CellRenderer ögesini tanımlayan şu sınıfta, data özelliğinin setter işlevinin data.label değerini CheckBox sınıfından miras alınan label özelliğine ilettiğini unutmayın:

```
public class CustomRenderer extends CheckBox implements ICellRenderer {
    private var _listData:ListData;
    private var _data:Object;
    public function CustomRender() {
    }
    public function set data(d:Object):void {
        _data = d;
        label = d.label;
    }
    public function get data():Object {
        return _data;
    }
    public function set listData(ld:ListData):void {
        _listData = ld;
    }
    public function get listData():ListData {
        return _listData;
    }
}
}
```

selected özelliği, bir hücrenin listede seçili olup olmadığını tanımlar.

DataGrid nesnesinin sütunu için CellRenderer uygulama

DataGrid nesnesinde birden çok sütun bulunabilir ve siz de her sütun için farklı hücre oluşturucuları belirtebilirsiniz. DataGrid ögesinin her sütunu bir DataGridColumn nesnesiyle temsil edilir ve DataGridColumn sınıfı, sütun için CellRenderer ögesini tanımlayabileceğiniz bir cellRenderer özelliğini içerir.

Düzenlenebilir bir hücre için CellRenderer tanımlama

DataGridCellEditor sınıfı, bir DataGrid nesnesindeki düzenlenebilir hücreler için kullanılan bir oluşturucuyu tanımlar. DataGrid nesnesinin editable özelliği true değerine ayarlandığında ve kullanıcı düzenlenecek hücreyi tıklattığında bu, hücrenin oluşturucusu olur. Düzenlenebilir bir hücre için CellRenderer ögesi tanımlamak üzere, DataGrid nesnesinin columns dizisinin her bir ögesi için itemEditor özelliğini ayarlayın.

CellRenderer olarak bir görüntü, SWF dosyası veya film klibi kullanma

CellRenderer ögesinin alt sınıfı olan ImageCell sınıfı, hücrenin ana içeriğinin bir görüntü, SWF dosyası veya film klibi olduğu hücreler oluşturmak için kullanılan bir nesneyi tanımlar. ImageCell sınıfı, hücrenin görünümünün tanımlanması için şu stilleri içerir:

- imagePadding—Piksel olarak, hücrenin kenarını görüntünün kenarından ayıran dolgu
- selectedSkin—Seçili durumunu göstermek için kullanılan kaplama
- textOverlayAlpha—Hücre etiketinin arkasındaki kaplamanın opaklığı
- textPadding—Piksel olarak, hücrenin kenarını metnin kenarından ayıran dolgu

ImageCell sınıfı, TileList sınıfının varsayılan CellRenderer ögesidir.

Bileşenleri erişilebilir duruma getirme

Flash uygulamalarındaki görsel içeriği, ekran okuyucu yoluyla görme engelli kullanıcılar için erişilebilir duruma getirebilirsiniz, böylece ekran içeriğinin ses açıklaması sağlanır. Flash uygulamanızı ekran okuyucusu için erişilebilir duruma getirme hakkında bilgi almak için, bkz. Bölüm 18, “Erişilebilir İçerik Oluşturma,” *Flash Uygulamasını Kullanma*.

ActionScript 3.0 bileşenini ekran koruyucusu için erişilebilir duruma getirmek üzere bileşenin erişilebilirlik sınıfını ve `enableAccessibility()` yöntemine yapılan bir çağrıyı da içe aktarmanız gerekir. Şu ActionScript 3.0 bileşenlerini ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

Bileşen	Erişilebilirlik Sınıfı:
Düğme	ButtonAccImpl
CheckBox	CheckBoxAccImpl
ComboBox	ComboBoxAccImpl
List	ListAccImpl
RadioButton	RadioButtonAccImpl
TileList	TileListAccImpl

Bileşen erişilebilirliği sınıfları, `fl.accessibility` paketindedir. Bir CheckBox ögesini ekran okuyucusu için erişilebilir duruma getirmek için, örneğin uygulamanıza şu ifadeleri eklersiniz:

```
import fl.accessibility.CheckBoxAccImpl;  
  
CheckBoxAccImpl.enableAccessibility();
```

Kaç tane örnek oluşturduğunuza bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirebilirsiniz.

Not: *Erişilebilirlik etkinleştirildiğinde, derleme sırasında zorunlu sınıflar dahil edilerek dosya boyutu artırılır.*

Çoğu bileşen, klavye ile de gezinilebilir durumdadır. Erişilebilir bileşenleri etkinleştirme ve klavye ile gezinme hakkında daha fazla bilgi için “[Kullanıcı Arabirimi Bileşenlerini Kullanma](#)” sayfa 44 ögesindeki Kullanıcı Etkileşimi bölümlerine ve *Adobe® Flash® Professional CS5 için ActionScript® 3.3 Başvurusu* ögesindeki erişilebilirlik sınıflarına bakın.

Bölüm 4: Kullanıcı Arabirimi Bileşenlerini Kullanma

Bu bölümde, Flash uygulamasına dahil edilen ActionScript 3.0 kullanıcı arabirimi (UI) bileşenlerinin nasıl kullanılacağı açıklanmaktadır.

Button bileşenini kullanma

Button bileşeni, kullanıcının uygulamada bir eylem başlatmak için fare veya boşluk çubuğu ile basabileceği yeniden boyutlandırılabilir, dikdörtgen bir düğmedir. Button öğesine özel bir simge ekleyebilirsiniz. Button öğesinin davranışını basma yerine aç/kapa olarak değiştirebilirsiniz. Aç/kapa Button öğesi, tıklatıldığında basılı kalır ve tekrar tıklatıldığında basılmamış durumuna geri döner.

Button, birçok form ve web uygulamasının temel parçasını oluşturur. Kullanıcının bir olay başlatmasını istediğiniz her yerde düğmeleri kullanabilirsiniz. Örneğin, çoğu formda bir Gönder düğmesi bulunur. Ayrıca bir sunuma Geri ve İleri düğmeleri de ekleyebilirsiniz.

Button bileşeniyle kullanıcı etkileşimi

Bir uygulamada herhangi bir düğmeyi etkinleştirebilir veya devre dışı bırakabilirsiniz. Bir düğme devre dışı durumunda fare veya klavye girdisi almaz. Etkinleştirilmiş bir düğme üzerini tıklattığınızda veya bastığınızda odağı alır. Bir Button örneği odağa sahip olduğunda, odağı denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Üst Karakter+Sekme	Odağı bir önceki nesneye taşır.
Boşluk çubuğu	Düğmeye basar veya düğmeyi serbest bırakır ve click olayını tetikler.
Sekme	Odağı bir sonraki nesneye taşır.
Enter/Return	Bir düğme FocusManager'ın varsayılan Button öğesi olarak ayarlanmışsa odağı bir sonraki nesneye taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki IFocusManager arabirimine ve FocusManager sınıfına bakın.

Her bir Button örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır.

Not: Bir simge düğmeden büyükse, düğmenin kenarlıklarını aşar.

Bir düğmeyi bir uygulamada varsayılan basma düğmesi olarak atamak için (kullanıcı Enter düğmesine bastığında click olayını alan düğme), FocusManager.defaultButton öğesini ayarlayın. Örneğin, aşağıdaki kod varsayılan düğmeyi submitButton adında bir Button örneği olacak şekilde ayarlar.

```
FocusManager.defaultButton = submitButton;
```

Bir uygulamaya Button bileşenini eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:


```
import fl.accessibility.ButtonAccImpl;  
  
ButtonAccImpl.enableAccessibility();
```

Kaç tane örnek oluşturduğunuza bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirebilirsiniz.

Button bileşeni parametreleri

Her Button örneği için Özellik denetçisinde (Pencere > Özellikler > Özellikler) ya da Bileşen denetçisinde (Pencere > Bileşen Denetçisi) şu geliştirme parametrelerini ayarlayabilirsiniz: `emphasized`, `label`, `labelPlacement`, `selected` ve `toggle`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelere bir değer atadığınızda, uygulamada özelliğin başlangıç durumunu ayarlarsınız. Özellik ActionScript'te ayarlandığında, parametrede ayarladığınız değer geçersiz kalır. Bu parametrelerin olası değerleriyle ilgili bilgi için [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki Button sınıfına bakın.

Button bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl Button bileşeni ekleneceğini açıklamaktadır. Bu örnekte, Button ögesi tıklatıldığında, ColorPicker bileşeninin durumunu değiştirir.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir Button bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Button bileşeni için Özellik denetçisinde şu değerleri girin:
 - **aButton** örnek adını girin.
 - label parametresi için **Show** girin.
- 3 ColorPicker bileşenini Sahne Alanı'na ekleyin ve bu bileşene **aCp** örnek adını verin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
aCp.visible = false;  
  
aButton.addEventListener(MouseEvent.CLICK, clickHandler);  
  
function clickHandler(event:MouseEvent):void {  
  
    switch(event.currentTarget.label) {  
        case "Show":  
            aCp.visible = true;  
            aButton.label = "Disable";  
            break;  
        case "Disable":  
            aCp.enabled = false;  
            aButton.label = "Enable";  
            break;  
        case "Enable":  
            aCp.enabled = true;  
            aButton.label = "Hide";  
            break;  
        case "Hide":  
            aCp.visible = false;  
            aButton.label = "Show";  
            break;  
    }  
}
```

Kodun ikinci satırı `clickHandler()` işlevini `MouseEvent.CLICK` olayı için olay işleme işlevi olarak kaydeder. Kullanıcı Button öğesini tıklattığında olay gerçekleşir ve Button öğesinin değerine bağlı olarak `clickHandler()` işlevinin şu eylemlerden birini gerçekleştirmesine neden olur:

- Göster seçeneği, `ColorPicker` öğesini görünür hale getirir ve Button öğesinin etiketini Devre Dışı Bırak olarak değiştirir.
- Devre Dışı Bırak seçeneği, `ColorPicker` öğesini devre dışı bırakır ve Button öğesinin etiketini Etkinleştir olarak değiştirir.
- Etkinleştir seçeneği, `ColorPicker` öğesini etkinleştirir ve Button öğesinin etiketini Gizle olarak değiştirir.
- Gizle seçeneği, `ColorPicker` öğesini görünmez hale getirir ve Button öğesinin etiketini Göster olarak değiştirir.

5 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

Button bileşeniyle uygulama oluşturma

Aşağıdaki yordam, `ActionScript` kullanarak bir aç/kapa Button öğesi oluşturur ve Button öğesini tıklattığınızda Çıktı panelinde olay türünü görüntüler. Örnek, sınıfın yapıcısına başvurarak Button örneğini oluşturur ve onu `addChild()` yöntemini çağırarak Sahne Alanı'na ekler.

1 Yeni bir Flash (`ActionScript 3.0`) belgesi oluşturun.

2 Button bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin.

Bu işlem sonucunda bileşen kütüphaneye eklenir ancak uygulamada görünür duruma getirilmez.

3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve bir Button örneği oluşturmak için şu kodu girin:

```
import fl.controls.Button;

var aButton:Button = new Button();
addChild(aButton);
aButton.label = "Click me";
aButton.toggle = true;
aButton.move(50, 50);
```

`move()` yöntemi, düğmeyi Sahne Alanı'nda 50 (x koordinatı), 50 (y koordinatı) konumuna yerleştirir.

4 Şimdi bir olay dinleyicisi ve olay işleyici işlevi oluşturmak için şu `ActionScript`'i ekleyin:

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {
    trace("Event type: " + event.type);
}
```

5 Kontrol Et > Filmi Test Et'i seçin.

Düğmeyi tıklattığınızda, Flash uygulaması Çıktı panelinde "Olay türü: click" mesajını görüntüler.

CheckBox bileşenini kullanma

`CheckBox` seçilebilir veya seçimi kaldırılabilir kare bir kutudur. Seçildiğinde, kutuda bir onay işareti görüntülenir. `CheckBox` öğesine bir metin etiketi ekleyebilir ve bu metin etiketini `CheckBox` öğesinin soluna, sağına, yukarısına veya aşağısına yerleştirebilirsiniz.

Birbirini dışlamayan `true` veya `false` değerlerinin bir kümesini toplamak için `CheckBoxes` ögesini kullanabilirsiniz. Örneğin, ne tür bir araba satın almak istediğiniz hakkında bilgi toplayan bir uygulama, özellikleri seçmenize olanak sağlamak için `CheckBox` öğelerini kullanabilir.

CheckBox ile kullanıcı etkileşimi

Bir uygulamada `CheckBox` ögesini etkinleştirebilir veya devre dışı bırakabilirsiniz. `CheckBox` ögesi etkinleştirilirse ve kullanıcı bu ögeyi veya bu ögenin etiketini tıklarsa, `CheckBox` ögesi girdi odağını alır ve basılı durumdaki görünümünü görüntüler. Kullanıcı fare düğmesine basarken işaretçiyi `CheckBox` ögesinin veya etiketinin sınırlama alanının dışına getirirse, bileşenin görünümü orijinal durumuna geri döner ve girdi odağını korur. Fare bileşenin üzerinde serbest bırakılıncaya kadar `CheckBox` ögesinin durumu değişmez. Ayrıca, `CheckBox` ögesi iki devre dışı durumuna sahiptir: seçili ve seçimi kaldırılmış; bunlar sırayla `selectedDisabledSkin` ve `disabledSkin` öğelerini kullanır ve fare ya da klavye etkileşimine izin vermez.

`CheckBox` devre dışı bırakılırsa, kullanıcı müdahalesine bakılmaksızın devre dışı durumda görüntülenir. `CheckBox` ögesi, düğme devre dışı durumunda fare veya klavye girdisi almaz.

Kullanıcı üzerini tıklattığında veya sekme ile seçtiğinde `CheckBox` örneği, odağı alır. Bir `CheckBox` örneği odağa sahip olduğunda, odağı denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Üst Karakter+Sekme	Odağı bir önceki ögeye taşır.
Boşluk çubuğu	Bileşeni seçer veya bileşenin seçimini kaldırır ve <code>change</code> olayını tetikler.
Sekme	Odağı bir sonraki ögeye taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki "[FocusManager ile çalışma](#)" sayfa 26 ve `FocusManager` sınıfına bakın.

Her bir `CheckBox` örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır.

Bir uygulamaya `CheckBox` bileşenini eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.CheckBoxAccImpl;  
  
CheckBoxAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğine sahip olduğunuza bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirirsiniz.

CheckBox bileşeni parametreleri

Her `CheckBox` bileşeni örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: `label`, `labelPlacement` ve `selected`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `CheckBox` sınıfına bakın.

CheckBox kullanarak uygulama oluşturma

Aşağıdaki yordamda, borç uygulama formundan bir alıntıyı kullanarak geliştirme sırasında bir uygulamaya CheckBox bileşeninin nasıl ekleneceği açıklanmaktadır. Formda başvuran kişinin ev sahibi olup olmadığı sorulur ve kullanıcının “evet” yanıtı vermesi için bir CheckBox sağlanır. Bu durumda, formda kullanıcının evin göreceli değerini belirtmesi için iki radyo düğmesi sunulur.

Checkbox bileşenini kullanarak uygulama oluşturma

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 CheckBox bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde şunları yapın:
 - Örnek adı için **homeCh** girin.
 - Genişlik (W) değeri için **140** girin.
 - Etiket parametresi için “**Eviniz kendinize mi ait?**” cümlesini girin.
- 4 İki RadioButton bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve CheckBox öğesinin aşağısına ve sağına yerleştirin. Bunlar için Özellik denetçisinde şu değerleri girin:
 - Örnek adları için **underRb** ve **overRb** girin.
 - Her iki RadioButton öğesinin W (genişlik) parametresi için **120** girin.
 - **500.000\$'dan az mı?** öğesini **underRb** etiket parametresi için girin.
 - **500.000\$'dan çok mu?** öğesini **overRb** etiket parametresi için girin.
 - Her iki RadioButton öğesinin **groupName** parametresi için **valueGrp** öğesini girin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
homeCh.addEventListener(MouseEvent.CLICK, clickHandler);
underRb.enabled = false;
overRb.enabled = false;

function clickHandler(event:MouseEvent):void {
    underRb.enabled = event.target.selected;
    overRb.enabled = event.target.selected;
}
```

Bu kod, **homeCh** CheckBox öğesi seçilirse **underRb** ve **overRb** RadioButton öğelerini etkinleştiren **CLICK** olayı için bir olay işleyicisi oluşturur ve **homeCh** öğesi seçilmezse bu öğeleri devre dışı bırakır. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki **MouseEvent** sınıfına bakın.

- 6 Kontrol Et > Filmi Test Et'i seçin.

Aşağıdaki örnek, önceki uygulamayı çoğaltır ancak ActionScript ile CheckBox ve RadioButtons öğelerini oluşturur.

ActionScript kullanarak Checkbox oluşturma

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 CheckBox bileşenini ve RadioButton bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin. Kütüphane paneli açılmazsa, Kütüphane panelini açmak için **Ctrl+L** tuşlarına basın veya **Pencere > Kütüphane** seçeneklerini belirleyin.

Bu işlem sonucunda bileşenler uygulamanız için kullanılabilir duruma getirilir ancak Sahne Alanı'na koyulmaz.

- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve bileşen örnekleri oluşturup konumlandırmak için şu kodu girin:

```
import fl.controls.CheckBox;
import fl.controls.RadioButton;

var homeCh:CheckBox = new CheckBox();
var underRb:RadioButton = new RadioButton();
var overRb:RadioButton = new RadioButton();
addChild(homeCh);
addChild(underRb);
addChild(overRb);
underRb.groupName = "valueGrp";
overRb.groupName = "valueGrp";
homeCh.move(200, 100);
homeCh.width = 120;
homeCh.label = "Own your home?";
underRb.move(220, 130);
underRb.enabled = false;
underRb.width = 120;
underRb.label = "Under $500,000?";
overRb.move(220, 150);
overRb.enabled = false;
overRb.width = 120;
overRb.label = "Over $500,000?";
```

Bu kod, bileşenler oluşturmak için `CheckBox()` ve `RadioButton()` yapıcılarını ve bu bileşenleri Sahne Alanı'na yerleştirmek için `addChild()` yöntemini kullanır. Bileşenleri Sahne Alanı'nda konumlandırmak için `move()` yöntemini kullanır.

- 4 Şimdi bir olay dinleyicisi ve olay işleyici işlevi oluşturmak için şu ActionScript'i ekleyin:

```
homeCh.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {
    underRb.enabled = event.target.selected;
    overRb.enabled = event.target.selected;
}
```

Bu kod, `homeCh` `CheckBox` öğesi seçilirse `underRb` ve `overRb` radyo düğmelerini etkinleştiren `CLICK` olayı için bir olay işleyicisi oluşturur ve `homeCh` öğesi seçilmezse bu öğeleri devre dışı bırakır. Daha fazla bilgi için, [Adobe® Flash® Professional CSS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `MouseEvent` sınıfına bakın.

- 5 Kontrol Et > Filmi Test Et'i seçin.

ColorPicker bileşenini kullanma

`ColorPicker` bileşeni, kullanıcının renk örneği listesinden bir renk seçmesine olanak tanır. `ColorPicker` öğesinin varsayılan modu, kare düğmede tek bir renk gösterir. Kullanıcı düğmeyi tıklattığında, geçerli renk seçiminin onaltılık değerini görüntüleyen bir metin alanıyla birlikte renk örneği panelinde kullanılabilir renklerin listesi görüntülenir.

`colors` özelliğini, görüntülemek istediğiniz renk değerleriyle ayarlayarak `ColorPicker` öğesinde görüntülenen renkleri ayarlayabilirsiniz.

ColorPicker bileşeniyle kullanıcı etkileşimi

ColorPicker, kullanıcının bir renk seçip bu rengi uygulamadaki başka bir nesneye uygulamasına olanak sağlar. Örneğin, arka plan rengi veya metin rengi gibi uygulama öğelerini kullanıcının kişiselleştirmesine olanak sağlamak istiyorsanız, ColorPicker öğesi dahil edebilir ve kullanıcının seçtiği rengi uygulayabilirsiniz.

Kullanıcı paneldeki renk örneğini tıklatarak veya metin alanına onaltılık değerini girerek bir renk seçer. Kullanıcı bir renk seçtikten sonra, metne veya uygulamadaki başka bir nesneye bu rengi uygulamak için ColorPicker öğesinin `selectedColor` özelliğini kullanabilirsiniz.

Kullanıcı ColorPicker örneğinin üzerine işaretçiyi getirirse veya bu örneğe basarsa, ColorPicker örneği odağı alır. ColorPicker öğesinin renk örneği paneli açık olduğunda bu paneli denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Home	Renk örneği panelinde seçimi ilk renge taşır.
Yukarı Ok	Renk örneği panelinde seçimi bir satır yukarı taşır.
Aşağı Ok	Renk örneği panelinde seçimi bir satır aşağı taşır.
Sağ Ok	Renk örneği panelinde seçimi bir renk sağa taşır.
Sol Ok	Renk örneği panelinde seçimi bir renk sola taşır.
End	Renk örneği panelinde seçimi son renge taşır.

ColorPicker bileşeni parametreleri

Her ColorPicker örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: `selectedColor` ve `showTextField`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki ColorPicker sınıfına bakın.

ColorPicker bileşeniyle uygulama oluşturma

Şu örnek, geliştirme sırasında bir uygulamaya ColorPicker bileşeni ekler. Bu örnekte, ColorPicker öğesindeki rengi her değiştirdiğinizde `changeHandler()` işlevi ColorPicker öğesinde seçtiğiniz rengin yer aldığı yeni bir kutu çizmek için `drawBox()` işlevini çağırır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir ColorPicker öğesini Bileşenler panelinden Sahne Alanı'nın ortasına sürükleyin ve bu öğeye `aCp` örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.events.ColorPickerEvent;

var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xFF0000); //draw a red box
addChild(aBox);

aCp.addEventListener(ColorPickerEvent.CHANGE, changeHandler);

function changeHandler(event:ColorPickerEvent):void {
    drawBox(aBox, event.target.selectedColor);
}

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1);
    box.graphics.drawRect(100, 150, 100, 100);
    box.graphics.endFill();
}
```

- 4 Kontrol Et > Filmi Test Et'i seçin.
- 5 ColorPicker öğesini tıkkatın ve kutuyu renklendirmek için bir renk seçin.

ActionScript kullanarak ColorPicker oluşturma

Bu örnek, Sahne Alanı'nda bir ColorPicker oluşturmak için ColorPicker() yapıcısını ve addChild() öğesini kullanır. ColorPicker öğesinin görüntüleyeceği renkleri belirtmek üzere kırmızı (0xFF0000), yeşil (0x00FF00) ve mavi (0x0000FF) renk değerlerine colors özelliğine ayarlar. Ayrıca bir TextArea oluşturur ve ColorPicker öğesinde her farklı bir renk seçtiğinizde örnek, eşleşmek için TextArea öğesindeki metnin rengine dönüşür.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ColorPicker bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 TextArea bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.ColorPicker;
import fl.controls.TextArea;
import fl.events.ColorPickerEvent;

var aCp:ColorPicker = new ColorPicker();
var aTa:TextArea = new TextArea();
var aTf:TextFormat = new TextFormat();

aCp.move(100, 100);
aCp.colors = [0xff0000, 0x00ff00, 0x0000ff];
aCp.addEventListener(ColorPickerEvent.CHANGE, changeHandler);

aTa.text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus quis nisl vel
tortor nonummy vulputate. Quisque sit amet eros sed purus euismod tempor. Morbi tempor. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Curabitur
diam. Suspendisse at purus in ipsum volutpat viverra. Nulla pellentesque libero id libero.";
aTa.setSize(200, 200);
aTa.move(200,100);

addChild(aCp);
addChild(aTa);

function changeHandler(event:ColorPickerEvent):void {
    if(TextFormat(aTa.getStyle("textFormat"))){
        aTf = TextFormat(aTa.getStyle("textFormat"));
    }
    aTf.color = event.target.selectedColor;
    aTa.setStyle("textFormat", aTf);
}
```

5 Kontrol Et > Filmi Test Et'i seçin.

ComboBox bileşenini kullanma

ComboBox bileşeni, kullanıcının açılır bir listeden tek bir seçim yapmasına olanak tanır. ComboBox statik veya düzenlenebilir olabilir. Düzenlenebilir bir ComboBox, kullanıcının listenin en üst kısmındaki metin alanına doğrudan metin girmesine olanak sağlar. Açılır liste belgenin alt kısmına değerse, aşağı yerine yukarı doğru açılır. ComboBox üç alt bileşenden oluşur: BaseButton, TextInput ve List bileşenleri.

Düzenlenebilir bir ComboBox öğesinde metin kutusu değil yalnızca düğme vuruş alanıdır. Statik bir ComboBox öğesinde düğme ve metin kutusu vuruş alanını oluşturur. Vuruş alanı, açılır listeyi açarak veya kapatarak yanıt verir.

Kullanıcı fare veya klavye ile listede bir seçim yaptığında, seçimin etiketi ComboBox öğesinin üst kısmındaki metin alanına kopyalanır.

ComboBox bileşeniyle kullanıcı etkileşimi

ComboBox bileşenini, herhangi bir formda veya listeden tek bir seçim yapılmasını gerektiren uygulamada kullanabilirsiniz. Örneğin, bir müşteri adresi formunda şehirlerin açılır bir listesini sağlayabilirsiniz. Daha karmaşık senaryolar için düzenlenebilir bir ComboBox kullanabilirsiniz. Örneğin, sürüş talimatları sağlayan bir uygulamada, kullanıcının başlangıç ve hedef adreslerini girmesine olanak sağlamak için düzenlenebilir bir ComboBox kullanabilirsiniz. Açılır listede önceden girilmiş adresler bulunur.

ComboBox düzenlenebilir durumdaysa, başka bir deyişle `editable` özelliği `true` değerindeyse, şu tuşlar odağı metin girdi kutusundan kaldırır ve önceki değeri bırakır. Tek istisna, kullanıcı metni girdiğinde ilk olarak yeni değeri uygulayan Enter tuşudur.

Anahtar	Açıklama
Üst Karakter + Sekme	Odağı bir önceki öğeye taşır. Yeni bir öge seçilirse, <code>change</code> olayı gönderilir.
Sekme	Odağı bir sonraki öğeye taşır. Yeni bir öge seçilirse, <code>change</code> olayı gönderilir.
Aşağı Ok	Seçimi bir öge aşağı taşır.
End	Seçimi listenin en alt kısmına taşır.
Escape	Açılır listeyi kapatır ve odağı ComboBox ögesine geri getirir.
Enter	Açılır listeyi kapatır ve odağı ComboBox ögesine geri getirir. ComboBox düzenlenebilir durumda olduğunda ve kullanıcı metin girdiğinde, Enter ögesi, değeri girilen metne ayarlar.
Home	Seçimi listenin en üst kısmına taşır.
Sayfa Yukarı	Seçimi bir sayfa yukarı taşır.
Sayfa Aşağı	Seçimi bir sayfa aşağı taşır.

Bir uygulamaya ComboBox bileşenini eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.ComboBoxAccImpl;  
  
ComboBoxAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğine sahip olduğunuza bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirirsiniz.

ComboBox bileşeni parametreleri

Her ComboBox örneği için Özellik denetçisinde veya Bileşen denetçisinde şu parametreleri ayarlayabilirsiniz: `dataProvider`, `editable`, `prompt` ve `rowCount`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki ComboBox sınıfına bakın. `dataProvider` parametresini kullanma hakkında bilgi almak için, bkz. “[dataProvider parametresini kullanma](#)” sayfa 28.

ComboBox bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl ComboBox bileşeni ekleneceğini açıklamaktadır. ComboBox ögesi düzenlenebilir durumdadır ve metin alanına **Add** yazarsanız, örnek açılır listeye bir öge ekler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir ComboBox bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aCb** örnek adını verin. Parametreler sekmesinde `editable` parametresini `true` değerine ayarlayın.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
import fl.data.DataProvider;
import fl.events.ComponentEvent;

var items:Array = [
    {label:"screen1", data:"screenData1"},
    {label:"screen2", data:"screenData2"},
    {label:"screen3", data:"screenData3"},
    {label:"screen4", data:"screenData4"},
    {label:"screen5", data:"screenData5"},
];
aCb.dataProvider = new DataProvider(items);

aCb.addEventListener(ComponentEvent.ENTER, onAddItem);

function onAddItem(event:ComponentEvent):void {
    var newRow:int = 0;
    if (event.target.text == "Add") {
        newRow = event.target.length + 1;
        event.target.addItemAt({label:"screen" + newRow, data:"screenData" + newRow},
            event.target.length);
    }
}
```

4 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak ComboBox oluşturma

Aşağıdaki örnek, ActionScript ile bir ComboBox öğesi oluşturur ve bu öğeyi San Francisco, California bölgesindeki üniversitelerin listesiyle doldurur. ComboBox öğesinin `width` özelliğini istem metninin genişliğini alacak şekilde ayarlar ve `dropdownWidth` özelliğini en uzun üniversite adını alacak şekilde daha geniş bir değere ayarlar.

Bu örnek, okul adlarını saklamak için `label` özelliğini ve her bir okulun web sitesini saklamak için `data` özelliğini kullanarak Array örneğinde üniversitelerin listesini oluşturur. `dataProvider` özelliğini ayarlayarak Array öğesini ComboBox öğesine atar.

Kullanıcı listeden bir üniversite seçtiğinde, bir Event öğesini tetikler.`CHANGE` olayı ve bir URL'ye `data` özelliğini yükleyen `changeHandler()` işlevi çağırısı, okulun web sitesine erişme isteğinde bulunur.

Liste kapandığında istemi yeniden görüntülemek için son satırın, ComboBox örneğinin `selectedIndex` özelliğini -1 değerine ayarladığına dikkat edin. Aksi takdirde, seçilmiş olan okulun adı istemin yerine geçer.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ComboBox bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.ComboBox;
import fl.data.DataProvider;
import flash.net.navigateToURL;

var sfUniversities:Array = new Array(
    {label:"University of California, Berkeley",
      data:"http://www.berkeley.edu/"},
    {label:"University of San Francisco",
      data:"http://www.usfca.edu/"},
    {label:"San Francisco State University",
      data:"http://www.sfsu.edu/"},
    {label:"California State University, East Bay",
      data:"http://www.csu Hayward.edu/"},
    {label:"Stanford University", data:"http://www.stanford.edu/"},
    {label:"University of Santa Clara", data:"http://www.scu.edu/"},
    {label:"San Jose State University", data:"http://www.sjsu.edu/"},
);

var aCb:ComboBox = new ComboBox();
aCb.dropdownWidth = 210;
aCb.width = 200;
aCb.move(150, 50);
aCb.prompt = "San Francisco Area Universities";
aCb.dataProvider = new DataProvider(sfUniversities);
aCb.addEventListener(Event.CHANGE, changeHandler);

addChild(aCb);

function changeHandler(event:Event):void {
    var request:URLRequest = new URLRequest();
    request.url = ComboBox(event.target).selectedItem.data;
    navigateToURL(request);
    aCb.selectedIndex = -1;
}
```

4 Kontrol Et > Filmi Test Et'i seçin.

Bu örneği Flash geliştirme ortamında uygulayabilir ve çalıştırabilirsiniz ancak ComboBox öğesindeki öğeleri tıklatarak üniversite web sitelerine erişmeyi denerseniz, uyarı mesajları alırsınız. İnternet'te tam işlevsel ComboBox öğesine erişmek için şu konuma erişin:

<http://www.helpexamples.com/peter/bayAreaColleges/bayAreaColleges.html>

DataGrid bileşenini kullanma

DataGrid bileşeni, bir diziden veya DataProvider için bir diziye ayrıştırabileceğiniz harici bir XML dosyasından verileri çizerek satır ve sütun ızgarasında görüntülemenize olanak sağlar. DataGrid bileşeni, dikey ve yatay kaydırma, olay desteği (düzenlenebilir hücreler desteği dahil) ve sıralama yeteneklerini içerir.

Font, renk ve ızgaradaki sütunların kenarlıkları gibi özellikleri yeniden boyutlandırabilir ve özelleştirebilirsiniz. ızgaradaki herhangi bir sütun için hücre oluşturucu olarak özel bir film klibi kullanabilirsiniz. (Hücre oluşturucu, hücrenin içeriklerini görüntüler) Sayfa görünüm stili görüntüsü oluşturmak için kaydırma çubuklarını devre dışı bırakabilir ve DataGrid yöntemlerini kullanabilirsiniz. Özelleştirmeyle ilgili daha fazla bilgi için, *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* bölümündeki DataColumn sınıfına bakın.

Daha fazla Yardım konusu

[DataGrid bileşeni oluşturma, doldurma ve yeniden boyutlandırma](#)

[DataGrid bileşeni özelleştirme ve sıralama](#)

[DataGrid bileşeninde veri filtreleme ve biçimlendirme](#)

DataGrid bileşeniyle kullanıcı etkileşimi

DataGrid bileşeniyle etkileşim kurmak için fare ve klavyeyi kullanabilirsiniz.

Hem `sortableColumns` özelliği hem de sütunun `sortable` özelliği, `true` olursa, sütun üstbilgisi tıklatıldığında sütunun değerleri esas alınarak veriler sıralanır. `sortable` özelliğini `false` değerine ayarlayarak tek bir sütun için sıralamayı devre dışı bırakabilirsiniz.

`resizableColumns` özelliği `true` olursa, üstbilgi satırında sütun bölücülerini sürükleyerek sütunları yeniden boyutlandırabilirsiniz.

Düzenlenebilir bir hücre tıklatıldığında odak o hücreye verilir; düzenlenemez bir hücre tıklatıldığında odak üzerine herhangi bir etki olmaz. Hücrenin hem `DataGrid.editable` hem de `DataGridColumn.editable` özellikleri `true` olduğunda tek bir hücre düzenlenebilir olur.

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki DataGrid ve DataColumn sınıflarına bakın.

DataGrid örneği, tıklatma veya basma işlemiyle odağı aldığı anda, örneği denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Aşağı Ok	Bir hücre düzenlenirken, ekleme noktası hücre metninin sonuna kayar. Hücre düzenlenebilir değilse, Aşağı Ok tuşu, List bileşeninin işlediği şekilde seçimi işler.
Yukarı Ok	Bir hücre düzenlenirken, ekleme noktası hücre metninin başına kayar. Hücre düzenlenebilir değilse, Yukarı Ok tuşu, List bileşeninin işlediği şekilde seçimi işler.
Üst Karakter+Yukarı/Aşağı Ok	DataGrid düzenlenebilir değilse ve <code>allowMultipleSelection</code> ögesi <code>true</code> değerindeyse, bitişik satırları seçer. Karşı yön okuyla yön tersine çevrildiğinde, başlangıç satırını geçinceye kadar seçili satırların seçimi kaldırılır ve bu noktada o yöndeki satırlar seçilir.
Üst Karakter+Tıklatma	<code>allowMultipleSelection</code> ögesi <code>true</code> olursa, seçili satır ile geçerli düzeltme konumu (vurgulu hücre) arasındaki tüm satırları seçer.
Ctrl+Tıklatma	<code>allowMultipleSelection</code> ögesi <code>true</code> olursa, bitişik olmak zorunda olmayan ek satırları seçer.
Sağ Ok	Bir hücre düzenlenirken, ekleme noktası bir karakter sağa kayar. Hücre düzenlenebilir değilse, Sağ Ok herhangi bir şey yapmaz.
Sol Ok	Bir hücre düzenlenirken, ekleme noktası bir karakter sola kayar. Hücre düzenlenebilir değilse, Sol Ok herhangi bir şey yapmaz.
Home	DataGrid ögesinde ilk satırı seçer.
End	DataGrid ögesinde son satırı seçer.
PageUp	DataGrid sayfasında ilk satırı seçer. Sayfa, DataGrid ögesinin kaydırma olmadan görüntüleyebildiği satırlardan oluşur.

Anahtar	Açıklama
PageDown	DataGrid sayfasında son satırı seçer. Sayfa, DataGrid ögesinin kaydırma olmadan görüntüleyebildiği satırlardan oluşur.
Return/Enter/Üst Karakter+Enter	Hücre düzenlenebilir olduğunda, değişiklik yapılır ve ekleme noktası aynı sütunda, sonraki satıra (kaydırma geçişine bağlı olarak yukarı veya aşağı) taşınır.
Üst Karakter+Sekme/Sekme	DataGrid düzenlenebilir durumdaysa, sütunun sonuna ulaşıncaya kadar odağı önceki/sonraki öğeye ve sonra ilk veya son hücreye ulaşıncaya kadar önceki/sonraki satıra taşır. İlk hücre seçildiğinde, Üst Karakter+Sekme tuşları odağı bir önceki denetime taşır. Son hücre seçildiğinde, Sekme tuşu odağı bir sonraki denetime taşır. DataGrid ögesi düzenlenebilir değilse, odağı önceki/sonraki denetime taşır.

Çok sayıda veri tabanlı uygulama türleri için temel olarak DataGrid bileşenini kullanabilirsiniz. Verilerin biçimlendirilmiş çizelgeli görünümünü kolayca görüntüleyebilirsiniz ancak daha karmaşık ve düzenlenebilir kullanıcı arabirimleri oluşturmak için hücre oluşturucusu yeteneklerini de kullanabilirsiniz. DataGrid bileşeninin pratik kullanımları şunlardır:

- Web postası istemcisi
- Arama sonuçları sayfaları
- Borç hesaplayıcıları ve vergi formu uygulamaları gibi elektronik tablo uygulamaları

DataGrid bileşeniyle bir uygulama tasarladığınızda, DataGrid sınıfı SelectableList sınıfını uzattığından, List bileşeninin tasarımının anlaşılması yararlı olacaktır. SelectableList sınıfı ve List bileşeniyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki SelectableList ve List sınıflarına bakın.

Uygulamanıza bir DataGrid bileşeni eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.DataGridAccImpl;  
DataGridAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğe sahip olduğuna bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirebilirsiniz. Daha fazla bilgi için bkz. Bölüm 18, “Erişilebilir İçerik Oluşturma,” *Flash Uygulamasını Kullanma*.

DataGrid bileşeni parametreleri

Her DataGrid bileşen örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: allowMultipleSelection, editable, headerHeight, horizontalLineScrollSize, horizontalPageScrollSize, horizontalScrollPolicy, resizableColumns, rowHeight, showHeaders, verticalLineScrollSize, verticalPageScrollSize ve verticalScrollPolicy. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki DataGrid sınıfına bakın.

DataGrid bileşeniyle uygulama oluşturma

DataGrid bileşeniyle bir uygulama oluşturmak için öncelikle verilerinizin geldiği yeri belirlemeniz gerekir. Tipik olarak veriler dataProvider özelliğini ayarlayarak ızgaraya çekebileceğiniz bir Array ögesinden gelir. ızgaraya veri eklemek için DataGrid ve DataGridColumn sınıflarının yöntemlerini de kullanabilirsiniz.

DataGrid bileşeniyle yerel bir veri sağlayıcısı kullanın

Bu örnekte, beysbol takımının görev listesini görüntülemek için bir DataGrid oluşturulmaktadır. Bir Array ögesinde (aRoster) görev listesini tanımlar ve bunu DataGrid'in dataProvider özelliğine atar.

- 1 Flash uygulamasında, Dosya > Yeni seçeneklerini belirleyin ve sonra Flash Dosyası (ActionScript 3.0) ögesini seçin.
- 2 DataGrid bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde, aDg örnek adını girin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.data.DataProvider;

bldRosterGrid(aDg);
var aRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter", Bats:"R", Throws:"R", Year:"So", Home: "Redlands, CA"},
    {Name:"Sue Pennypacker", Bats:"L", Throws:"R", Year:"Fr", Home: "Athens, GA"},
    {Name:"Jill Smithfield", Bats:"R", Throws:"L", Year:"Sr", Home: "Spokane, WA"},
    {Name:"Shirley Goth", Bats:"R", Throws:"R", Year:"Sr", Home: "Carson, NV"},
    {Name:"Jennifer Dunbar", Bats:"R", Throws:"R", Year:"Fr", Home: "Seaside, CA"},
    {Name:"Patty Crawford", Bats:"L", Throws:"L", Year:"Jr", Home: "Whittier, CA"},
    {Name:"Angelina Davis", Bats:"R", Throws:"R", Year:"So", Home: "Odessa, TX"},
    {Name:"Maria Santiago", Bats:"L", Throws:"L", Year:"Sr", Home: "Tacoma, WA"},
    {Name:"Debbie Ferguson", Bats:"R", Throws:"R", Year: "Jr", Home: "Bend, OR"},
    {Name:"Karen Bronson", Bats:"R", Throws:"R", Year: "Sr", Home: "Billings, MO"},
    {Name:"Sylvia Munson", Bats:"R", Throws:"R", Year: "Jr", Home: "Pasadena, CA"},
    {Name:"Carla Gomez", Bats:"R", Throws:"L", Year: "Sr", Home: "Corona, CA"},
    {Name:"Betty Kay", Bats:"R", Throws:"R", Year: "Fr", Home: "Palo Alto, CA"},
];
aDg.dataProvider = new DataProvider(aRoster);
aDg.rowCount = aDg.length;

function bldRosterGrid(dg:DataGrid){
    dg.setSize(400, 300);
    dg.columns = ["Name", "Bats", "Throws", "Year", "Home"];
    dg.columns[0].width = 120;
    dg.columns[1].width = 50;
    dg.columns[2].width = 50;
    dg.columns[3].width = 40;
    dg.columns[4].width = 120;
    dg.move(50,50);
};
```

bldRosterGrid() işlevi, DataGrid ögesinin boyutunu ayarlar ve sütunların ve boyutlarının sırasını belirler.

- 5 Kontrol Et > Filmi Test Et'i seçin.

Bir uygulamada DataGrid bileşeni için sütunları belirtme ve sıralama ekleme

Herhangi bir sütun başlığını tıklararak, DataGrid ögesinin içeriklerini o sütunun değerlerine göre azalan sırada düzenleyebileceğinizi unutmayın.

Aşağıdaki örnek, bir DataGrid ögesine DataColumn örnekleri eklemek için addColumn() yöntemini kullanır. Sütunlar, oynatıcı adlarını ve puanlarını temsil eder. Ayrıca bu örnek her sütun için sıralama seçeneklerini belirtmek üzere sortOptions özelliğini kullanır: Ad sütunu için Array.CASEINSENSITIVE ve Puan sütunu için Array.NUMERIC. Uzunluğu satır sayısına ve genişliği de 200 değerine ayarlayarak DataGrid ögesini uygun şekilde boyutlandırır.

- 1 Flash uygulamasında, Dosya > Yeni seçeneklerini belirleyin ve sonra Flash Dosyası (ActionScript 3.0) ögesini seçin.
- 2 DataGrid bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde, aDg örnek adını girin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.dataGridClasses.DataGridColumn;
import fl.events.DataGridEvent;
import fl.data.DataProvider;
// Create columns to enable sorting of data.
var nameDGC:DataGridColumn = new DataColumn("name");
nameDGC.sortOptions = Array.CASEINSENSITIVE;
var scoreDGC:DataGridColumn = new DataColumn("score");
scoreDGC.sortOptions = Array.NUMERIC;
aDg.addColumn(nameDGC);
aDg.addColumn(scoreDGC);
var aDP_array:Array = new Array({name:"clark", score:3135}, {name:"Bruce", score:403},
{name:"Peter", score:25})
aDg.dataProvider = new DataProvider(aDP_array);
aDg.rowCount = aDg.length;
aDg.width = 200;
```

- 5 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak DataGrid bileşen örneği oluşturma

Bu örnek, ActionScript kullanarak bir DataGrid oluşturur ve bunu oynatıcı adları ve puanlarını içeren bir Array ögesi ile doldurur.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 DataGrid bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin.
Bu işlem sonucunda bileşen kütüphaneye eklenir ancak uygulamada görünür duruma getirilmez.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.DataGrid;
import fl.data.DataProvider;

var aDg:DataGrid = new DataGrid();
addChild(aDg);
aDg.columns = [ "Name", "Score" ];
aDg.setSize(140, 100);
aDg.move(10, 40);
```

Bu kod DataGrid örneğini oluşturur ve sonra ızgarayı boyutlandırıp konumlandırır.

- 4 Bir dizi oluşturun, diziye veriler ekleyin ve DataGrid için veri sağlayıcısı olarak diziyi tanımlayın:

```
var aDP_array:Array = new Array();
aDP_array.push({Name:"Clark", Score:3135});
aDP_array.push({Name:"Bruce", Score:403});
aDP_array.push({Name:"Peter", Score:25});
aDg.dataProvider = new DataProvider(aDP_array);
aDg.rowCount = aDg.length;
```

- 5 Kontrol Et > Filmi Test Et'i seçin.

XML dosyasıyla DataGrid yükleme

Aşağıdaki örnek, DataGrid öğesinin sütunlarını oluşturmak için DataGridColumn sınıfını kullanır. DataProvider() yapıcısının value parametresi olarak XML nesnesini ileterek DataGrid öğesini doldurur.

- 1 Metin düzenleyici kullanarak şu veriler ile bir XML dosyası yaratın ve team.xml olarak FLA dosyasını kaydedeceğiniz klasöre kaydedin.

```
<team>
  <player name="Player A" avg="0.293" />
  <player name="Player B" avg="0.214" />
  <player name="Player C" avg="0.317" />
</team>
```

- 2 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 3 Bileşenler panelinde DataGrid bileşenini çift tıklatarak Sahne Alanı'na ekleyin.
- 4 Özellik denetçisinde, aDg örnek adını girin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:


```
import fl.controls.dataGridClasses.DataGridColumn;
import fl.data.DataProvider;
import flash.net.*;
import flash.events.*;

var request:URLRequest = new URLRequest("team.xml");
var loader:URLLoader = new URLLoader;

loader.load(request);
loader.addEventListener(Event.COMPLETE, loaderCompleteHandler);

function loaderCompleteHandler(event:Event):void {

    var teamXML:XML = new XML(loader.data);

    var nameCol:DataGridColumn = new DataGridColumn("name");
    nameCol.headerText = "Name";
    nameCol.width = 120;
    var avgCol:DataGridColumn = new DataGridColumn("avg");
    avgCol.headerText = "Average";
    avgCol.width = 60;

    var myDP:DataProvider = new DataProvider(teamXML);

    aDg.columns = [nameCol, avgCol];
    aDg.width = 200;
    aDg.dataProvider = myDP;
    aDg.rowCount = aDg.length;
}
```

6 Kontrol Et > Filmi Test Et'i seçin.

Label bileşenini kullanma

Etiket bileşeni genellikle bir web sayfasında başka bir öğeyi ya da etkinliği tanımlamak için tek bir metin satırı görüntüler. Metin biçimlendirme etiketlerinden yararlanmak için bir etiketin HTML ile biçimlendirileceğini belirtebilirsiniz. Ayrıca etiketin hizalamasını ve boyutunu da denetleyebilirsiniz. Etiket bileşenleri kenarlıklara sahip değildir, odaklanamazlar ve olayları yayınlamazlar.

Her bir Label örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır. Etiket kenarlığı yoktur, bu nedenle canlı önizlemesini görmeyen tek yolu, etiketin metin parametresini ayarlamaktır.

Label bileşeniyle kullanıcı etkileşimi

Bir formda, kullanıcının adını kabul eden TextInput alanının solunda "Ad:" etiketi gibi başka bir metin etiketi oluşturmak için Label bileşenini kullanın. Tutarlı bir görünüm elde etmek için stilleri kullanabildiğinizden, düz metin alanı yerine Label bileşeninin kullanılması iyi bir fikirdir.

Bir Label bileşenini döndürmek istiyorsanız, fontları gömmeniz gerekir; aksi takdirde siz filmi test ederken gösterilmezler.

Label bileşeni parametreleri

Her Label bileşeni örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: `autoSize`, `CondenseWhite`, `seçilebilir`, `text` ve `wordWrap`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CSS için ActionScript® 3.0 Başvurusu](#) bölümündeki Label sınıfına bakın.

Label bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl Label bileşeni ekleneceğini açıklamaktadır. Bu örnekte, etiket yalnızca “Expiration Date” metnini görüntüler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir Label bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene şu değerleri verin:
 - Örnek adı için **aLabel** girin.
 - W değeri için **80** girin.
 - X değeri için **100** girin.
 - Y değeri için **100** girin.
 - `text` parametresi için **Expiration Date** girin.
- 3 Bir TextArea bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene şu değerleri verin:
 - Örnek adı için **aTag** girin.
 - H değeri için **22** girin.
 - X değeri için **200** girin.
 - Y değeri için **100** girin.

- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
var today:Date = new Date();
var expDate:Date = addDays(today, 14);
aTa.text = expDate.toString();

function addDays(date:Date, days:Number):Date {
    return addHours(date, days*24);
}

function addHours(date:Date, hrs:Number):Date {
    return addMinutes(date, hrs*60);
}

function addMinutes(date:Date, mins:Number):Date {
    return addSeconds(date, mins*60);
}

function addSeconds(date:Date, secs:Number):Date {
    var mSecs:Number = secs * 1000;
    var sum:Number = mSecs + date.getTime();
    return new Date(sum);
}
```

- 5 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak Label bileşen örneği oluşturma

Aşağıdaki örnek, ActionScript kullanarak bir Label parametresi oluşturur. ColorPicker bileşeninin işlevini tanımlamak için bir Label ögesini kullanır ve Label metnine biçimlendirme uygulamak için `htmlText` özelliğini kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Label bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin.
- 3 ColorPicker bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.Label;
import fl.controls.ColorPicker;

var aLabel:Label = new Label();
var aCp:ColorPicker = new ColorPicker();

addChild(aLabel);
addChild(aCp);

aLabel.htmlText = '<font face="Arial" color="#FF0000" size="14">Fill:</font>';
aLabel.x = 200;
aLabel.y = 150;
aLabel.width = 25;
aLabel.height = 22;

aCp.x = 230;
aCp.y = 150;
```

- 5 Kontrol Et > Filmi Test Et'i seçin.

List bileşenini kullanma

List bileşeni kaydırılabilir bir tekli veya çoklu seçim listesi kutusudur. Liste, diğer bileşenler dahil olmak üzere görüntüleme grafiklerini de görüntüleyebilir. Etiketler veya veri parametresi alanlarını tıklattığınızda görüntülenen Değerler iletişim kutusunu kullanarak listede görüntülenen öğeleri eklersiniz. Listeye öge eklemek için `List.addItem()` ve `List.addItemAt()` yöntemlerini de kullanabilirsiniz.

List bileşeni, 0 dizinine sahip ögenin en üstte görüntülenen öge olduğu sıfır temelli dizini kullanır. List sınıfı yöntemlerini ve özelliklerini kullanarak liste öğeleri eklendiğinde, kaldırıldığında veya değiştirildiğinde liste ögesinin dizinini belirtmeniz gerekebilir.

List bileşeniyle kullanıcı etkileşimi

Kullanıcıların tek veya çoklu seçim yapabileceği şekilde bir listeyi ayarlayabilirsiniz. Örneğin, e-ticaret web sitesini ziyaret eden bir kullanıcının hangi öğeyi satın alacağını seçmesi gerekir. Kullanıcının kaydırarak gezindiği ve öğeleri tıklayarak seçtiği listede 30 öge vardır.

Ayrıca kullanıcıya daha fazla bilgi görüntüleyebilmeniz için özel film kliplerini satırlar olarak kullanan bir List ögesi de tasarlayabilirsiniz. Örneğin, bir e-posta uygulamasında her posta kutusu bir List bileşeni olabilir ve her satır öncelik ve durumunu belirten simgeler içerebilir.

Üzerini tıklattığınızda veya üzerine bastığınızda List öğesi odağı alır ve bunu denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Alfayısal tuşlar	Etiketinde ilk karakteri <code>Key.getAsci()</code> olan bir sonraki öğeye atlar.
Kontrol Et	Birden çok bitişik seçime veya birden çok bitişik seçimin kaldırılmasına olanak sağlayan aç/kapa tuşu.
Aşağı Ok	Seçimi bir öğe aşağı taşır.
Home	Seçimi listenin en üstüne taşır.
Sayfa Aşağı	Seçimi bir sayfa aşağı taşır.
Sayfa Yukarı	Seçimi bir sayfa yukarı taşır.
Üst Karakter	Bitişik seçime olanak sağlar.
Yukarı Ok	Seçimi bir öğe yukarı taşır.

Not: Kaydırma boyutlarının satır değil piksel cinsinden olduğunu unutmayın.

Not: Sayfa Yukarı ve Sayfa Aşağı tuşları tarafından kullanılan sayfa boyutu, görüntüye sığan öğe sayısından bir tane azdır. Örneğin, on satırlık açılır listede sayfa aşağı inildiğinde, her sayfada bir öğe örtüşerek 0-9, 9-18, 18-27, vb. öğeleri gösterilir.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki IFocusManager arabirimine ve FocusManager sınıfına bakın.

Her bir List örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır.

Bir uygulamaya List bileşenini eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.ListAccImpl;  
  
ListAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğe sahip olduğuna bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirebilirsiniz. Daha fazla bilgi için bkz. Bölüm 18, “Erişilebilir İçerik Oluşturma,” [Flash Uygulamasını Kullanma](#).

List bileşeni parametreler

Her List bileşeni örneği için Özellik denetçisinde ve Bileşen denetçisinde şu parametreleri ayarlayabilirsiniz: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` ve `verticalScrollPolicy`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki List sınıfına bakın. `dataProvider` parametresini kullanma hakkında bilgi almak için, bkz. “[dataProvider parametresini kullanma](#)” sayfa 28.

List bileşeniyle uygulama oluşturma

Aşağıdaki örnekler, geliştirme sırasında bir uygulamaya nasıl List bileşeni ekleneceğini açıklamaktadır.

Uygulamaya basit bir List bileşeni ekleme

Bu örnekte List ögesi, araba modellerini ve fiyatların yer aldığı veri alanlarını tanımlayan etiketlerden oluşur.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde şunları yapın:
 - **aList** örnek adını girin.
 - W (genişlik) ögesine **200** değerini atayın.
- 4 **aList** ögesinin aşağısında bir metin alanı oluşturmak için Metin aracını kullanın ve bu metin alanına **aTf** örnek adını verin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.List;
import flash.text.TextField;

aTf.type = TextFieldType.DYNAMIC;
aTf.border = false;

// Create these items in the Property inspector when data and label
// parameters are available.
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});
aList.addItem({label:"1966 Mustang (Classic)", data:27000});
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});
aList.allowMultipleSelection = true;

aList.addEventListener(Event.CHANGE, showData);

function showData(event:Event) {
    aTf.text = "This car is priced at: $" + event.target.selectedItem.data;
}
```

Bu kod, **aList** ögesini üç öğeyle doldurmak için `addItem()` yöntemini kullanır ve öğelerin her birine listede görüntülenen bir `label` değeri ve bir `data` değeri atar. List ögesinde bir öğeyi seçtiğinizde, olay dinleyicisi, seçili öğe için `data` değerini görüntüleyen `showData()` işlevini çağırır.

- 6 Bu uygulamayı derlemek ve çalıştırmak için Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

Liste örneğini veri sağlayıcısıyla doldurma

Bu örnek, araba modellerinin ve fiyatlarının bir List ögesini oluşturur. Ancak List ögesini doldurmak için `addItem()` yöntemi yerine bir veri sağlayıcısını kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde şunları yapın:
 - **aList** örnek adını girin.
 - W (genişlik) ögesine **200** değerini atayın.
- 4 **aList** ögesinin aşağısında bir metin alanı oluşturmak için Metin aracını kullanın ve bu metin alanına **aTf** örnek adını verin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.List;
import fl.data.DataProvider;
import flash.text.TextField;

aTf.type = TextFieldType.DYNAMIC;
aTf.border = false;

var cars:Array = [
    {label:"1956 Chevy (Cherry Red)", data:35000},
    {label:"1966 Mustang (Classic)", data:27000},
    {label:"1976 Volvo (Xcllnt Cond)", data:17000},
];
aList.dataProvider = new DataProvider(cars);
aList.allowMultipleSelection = true;

aList.addEventListener(Event.CHANGE, showData);

function showData(event:Event) {
    aTf.text = "This car is priced at: $" + event.target.selectedItem.data;
}
```

6 Öğelerin yer aldığı List öğesini görmek için Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

MovieClip örneğini denetlemek için List bileşenini kullanma

Aşağıdaki örnek, renk adlarının bir List öğesini oluşturur ve bir renk seçtiğinizde o rengi MovieClip öğesine uygular.

- 1 Flash (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene şu değerleri verin:
 - Örnek adı için **aList** girin.
 - H değeri için **60** girin.
 - X değeri için **100** girin.
 - Y değeri için **150** girin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
aList.addItem({label:"Blue", data:0x0000CC});
aList.addItem({label:"Green", data:0x00CC00});
aList.addItem({label:"Yellow", data:0xFFFF00});
aList.addItem({label:"Orange", data:0xFF6600});
aList.addItem({label:"Black", data:0x000000});

var aBox:MovieClip = new MovieClip();
addChild(aBox);

aList.addEventListener(Event.CHANGE, changeHandler);
function changeHandler(event:Event) {
    drawBox(aBox, event.target.selectedItem.data);
};

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(225, 150, 100, 100);
    box.graphics.endFill();
}
```

- 4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.
- 5 List öğesindeki renkleri MovieClip öğesinde görüntülenmesi için tıklatın.

ActionScript kullanarak List bileşen örneği oluşturma

Bu örnek ActionScript kullanarak basit bir liste oluşturur ve addItem() yöntemi ile doldurur.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.List;

var aList:List = new List();
aList.addItem({label:"One", data:1});
aList.addItem({label:"Two", data:2});
aList.addItem({label:"Three", data:3});
aList.addItem({label:"Four", data:4});
aList.addItem({label:"Five", data:5});
aList.setSize(60, 40);
aList.move(200,200);
addChild(aList);
aList.addEventListener(Event.CHANGE, changeHandler);
function changeHandler(event:Event):void {
    trace(event.target.selectedItem.data);
}
```

- 4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

NumericStepper bileşenini kullanma

NumericStepper bileşeni, kullanıcının sıralanmış bir sayı kümesinde ilerlemesine olanak tanır. Bu bileşen, küçük yukarı ve aşağı ok düğmelerinin yanında görüntülenen metin kutusundaki bir sayıyı içerir. Kullanıcı düğmelere bastığında, düğmeleri serbest bırakıncaya veya maksimum ya da minimum değere ulaşıncaya kadar stepSize parametresinde belirtilen birime göre sayı basamaklı olarak artar veya azalır. NumericStepper bileşeninin metin kutusundaki metin de düzenlenebilir.

Her bir NumericStepper örneğinin canlı önizlemesi, Özellik denetçisinde veya Bileşen denetçisinde value parametresinin ayarını yansıtır. Ancak canlı önizlemede NumericStepper öğesinin ok düğmeleriyle fare veya klavye etkileşimi yoktur.

NumericStepper bileşeniyle kullanıcı etkileşimi

Kullanıcının sayısal bir değer seçmesini istediğiniz her yerde NumericStepper bileşenini kullanabilirsiniz. Örneğin, bir formda kredi kartı son kullanma tarihinin ay, gün ve yılını ayarlamak için NumericStepper bileşenini kullanabilirsiniz. Kullanıcının font boyutunu artırmasına ve azaltmasına olanak sağlamak için de NumericStepper bileşenini kullanabilirsiniz.

NumericStepper bileşeni yalnızca sayısal verileri işler. Ayrıca ikiden çok sayısal basamağı görüntülemek için (örneğin, 5246 veya 1,34 sayıları) geliştirme sırasında adımlayıcıyı yeniden boyutlandırmanız gerekir.

Bir uygulamada NumericStepper öğesini etkinleştirebilir veya devre dışı bırakabilirsiniz. NumericStepper öğesi, devre dışı durumunda fare veya klavye girdisi almaz. NumericStepper öğesi etkinleştirilmiş durumundayken, üzerine tıklattığınızda veya üzerine bastığınızda odağı alır ve dahili odak, metin kutusuna ayarlanır. Bir NumericStepper örneği odağa sahip olduğunda, odağı denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Aşağı Ok	Değer, birer birimler olarak değişir.
Sol Ok	Ekleme noktasını metin kutusu içinde sola taşır.
Sağ Ok	Ekleme noktasını metin kutusu içinde sağa taşır.
Üst Karakter+Sekme	Odağı bir önceki nesneye taşır.
Sekme	Odağı bir sonraki nesneye taşır.
Yukarı Ok	Değer, birer birimler olarak değişir.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki FocusManager sınıfına bakın.

NumericStepper bileşeni parametreleri

Her NumericStepper örneği için Özellik denetçisinde veya Bileşen denetçisinde şu parametreleri ayarlayabilirsiniz: `maximum`, `minimum`, `stepSize` ve `value`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki NumericStepper sınıfına bakın.

NumericStepper ile uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl NumericStepper bileşeni ekleneceğini açıklamaktadır. Bu örnek, Sahne Alanı'na bir NumericStepper bileşeni ve Label bileşeni ekler ve Event öğesi için bir dinleyici oluşturur. NumericStepper örneğinde CHANGE olayı. Bu örnek, sayısal adımlayıcıdaki değer değiştiğinde, yeni değeri Etiket örneğinin `text` özelliğinde görüntüler.

- 1 NumericStepper bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 2 Özellik denetçisinde, **aNs** örnek adını girin.
- 3 Label bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 4 Özellik denetçisinde, **aLabel** örnek adını girin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import flash.events.Event;

aLabel.text = "value = " + aNs.value;

aNs.addEventListener(Event.CHANGE, changeHandler);
function changeHandler(event:Event) :void {
    aLabel.text = "value = " + event.target.value;
};
```

Bu örnek, etiketin `text` özelliğini NumericStepper değerine ayarlar. NumericStepper örneğindeki değer her değiştiğinde, `changeHandler()` işlevi, etiketin `text` özelliğini günceller.

- 6 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak NumericStepper oluşturun:

Bu örnek, ActionScript koduyla üç NumericStepper ögesi oluşturur; bunların her biri kullanıcının doğum tarihinin ay, gün ve yılını girmek içindir. Ayrıca bu, NumericStepper öğelerinin her birinin istemi ve tanımlayıcıları için Label örnekleri ekler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Label ögesini Kütüphane paneline sürükleyin.
- 3 NumericStepper bileşenini Kütüphane paneline sürükleyin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.Label;
import fl.controls.NumericStepper;

var dobPrompt:Label = new Label();
var moPrompt:Label = new Label();
var dayPrompt:Label = new Label();
var yrPrompt:Label = new Label();

var moNs:NumericStepper = new NumericStepper();
var dayNs:NumericStepper = new NumericStepper();
var yrNs:NumericStepper = new NumericStepper();

addChild(dobPrompt);
addChild(moPrompt);
addChild(dayPrompt);
addChild(yrPrompt);
addChild(moNs);
addChild(dayNs);
addChild(yrNs);

dobPrompt.setSize(65, 22);
dobPrompt.text = "Date of birth:";
dobPrompt.move(80, 150);

moNs.move(150, 150);
moNs.setSize(40, 22);
moNs.minimum = 1;
moNs.maximum = 12;
moNs.stepSize = 1;
moNs.value = 1;

moPrompt.setSize(25, 22);
moPrompt.text = "Mo.";
moPrompt.move(195, 150);

dayNs.move(225, 150);
```

```
dayNs.setSize(40, 22);
dayNs.minimum = 1;
dayNs.maximum = 31;
dayNs.stepSize = 1;
dayNs.value = 1;

dayPrompt.setSize(25, 22);
dayPrompt.text = "Day";
dayPrompt.move(270, 150);

yrNs.move(300, 150);
yrNs.setSize(55, 22);
yrNs.minimum = 1900;
yrNs.maximum = 2006;
yrNs.stepSize = 1;
yrNs.value = 1980;

yrPrompt.setSize(30, 22);
yrPrompt.text = "Year";
yrPrompt.move(360, 150);
```

5 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

ProgressBar bileşenini kullanma

ProgressBar bileşeni, içeriğin büyük olduğu ve uygulamanın çalıştırılmasını geciktirebileceği durumlarda kullanıcının emin olmasını sağlayan yükleme içeriği ilerlemesini görüntüler. Görüntü ve uygulama parçaları yüklemesinin ilerlemesini görüntülemek için ProgressBar ögesi yararlıdır. Yükleme işlemi, belirli veya belirsiz olabilir. *determinate* ilerleme çubuğu, bir görevin zaman içindeki ilerlemesinin doğrusal temsilidir ve yüklenecek içeriğin miktarı bilindiğinde kullanılır. *indeterminate* ilerleme çubuğu, yüklenecek içeriğin miktarı bilinmediğinde kullanılır. Yükleme ilerlemesini yüzde olarak görüntülemek için de Label etiketi ekleyebilirsiniz.

ProgressBar bileşeni 9 dilimli ölçekleme kullanır ve bir çubuk kaplaması, bir yol kaplaması ve bir belirsiz kaplaması içerir.

ProgressBar bileşeniyle kullanıcı etkileşimi

ProgressBar bileşeninin kullanılacağı üç mod vardır. En yaygın kullanılan modlar, olay modu ve sorgulu moddur. Bu modlar, *progress* ve *complete* olayları yayan (olay modu ve sorgulu mod) veya *bytesLoaded* ve *bytesTotal* özelliklerini kullanıma sunan (sorgulu mod) bir yükleme işlemini belirtir. *ProgressBar.setProgress()* yöntemine yapılan çağrılarla birlikte *maximum*, *minimum* ve *value* özelliklerini ayarlayarak ProgressBar bileşenini manuel modda da kullanabilirsiniz. ProgressBar ögesinin çizgili dolgu ve bilinmeyen boyutta bir kaynak mı (*true*) yoksa düz dolgu ve bilinmeyen boyutta bir kaynak (*false*) mı içerdiğini belirtmek için belirsiz özelliğini ayarlayabilirsiniz.

Özellik denetçisinde veya Bileşen denetçisinde *mode* parametresi üzerinden *mode* özelliğini ayarlayarak ya da ActionScript kullanarak ProgressBar ögesinin modunu ayarlayabilirsiniz.

100.000 öge ayrıştırma gibi, ilerleme durumunu göstermek için ProgressBar ögesini kullanırsanız, tek bir kare döngüsünde olursa, ekranın yeniden çizimi olmadığından ProgressBar ögesindeki güncellemeler görünür olmaz.

ProgressBar bileşeni parametreleri

Her ProgressBar örneği için Özellik denetçisinde veya Bileşen denetçisinde şu parametreleri ayarlayabilirsiniz: `direction`, `mode` ve `source`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir.

Bunları ve ProgressBar bileşeninin ek seçeneklerini denetlemek için ProgressBar bileşeninin özelliklerini, yöntemlerini ve olaylarını kullanarak ActionScript yazabilirsiniz. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki ProgressBar sınıfına bakın.

ProgressBar ile uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl ProgressBar bileşeni ekleneceğini göstermektedir. Bu örnekte, ProgressBar ögesi olay modunu kullanır. Olay modunda, yükleme içeriği ProgressBar ögesinin ilerlemeyi belirtmek için gönderdiği `progress` ve `complete` olaylarını gösterir. `progress` olayı gerçekleştiğinde, içeriğin yüklenen yüzdesini göstermek için örnek bir etiketi günceller. `complete` olayı gerçekleştiğinde, örnek, “Yükleme tamamlandı” mesajını ve dosyanın boyutu olan `bytesTotal` özelliğinin değerini görüntüler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ProgressBar bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
 - Özellik denetçisinde, **aPb** örnek adını girin.
 - Parametreler bölümünde X değeri için **200** girin.
 - Y değeri için **260** girin.
 - `mode` parametresi için `event` ögesini seçin.
- 3 Button bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
 - Özellik denetçisinde örnek adı olarak **loadButton** girin.
 - X parametresi için **220** girin.
 - Y parametresi için **290** girin.
 - `label` parametresi için **Load Sound** girin.
- 4 Label bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **progLabel** örnek adını verin.
 - W değeri için **150** girin.
 - X parametresi için **200** girin.
 - Y parametresi için **230** girin.
 - Parametreler bölümünde `text` parametresinin değerini temizleyin.
- 5 Eylemler panelinde, ana Zaman Çizelgesi'nde Kare 1 ögesini seçin ve bir mp3 ses dosyası yükleyen şu ActionScript kodunu girin:

```
import fl.controls.ProgressBar;
import flash.events.ProgressEvent;
import flash.events.IOErrorEvent;

var aSound:Sound = new Sound();
aPb.source = aSound;
var url:String = "http://www.helpexamples.com/flash/sound/song1.mp3";
var request:URLRequest = new URLRequest(url);

aPb.addEventListener(ProgressEvent.PROGRESS, progressHandler);
aPb.addEventListener(Event.COMPLETE, completeHandler);
aSound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
loadButton.addEventListener(MouseEvent.CLICK, clickHandler);

function progressHandler(event:ProgressEvent):void {
    progLabel.text = ("Sound loading ... " + aPb.percentComplete);
}

function completeHandler(event:Event):void {
    trace("Loading complete");
    trace("Size of file: " + aSound.bytesTotal);
    aSound.close();
    loadButton.enabled = false;
}

function clickHandler(event:MouseEvent) {
    aSound.load(request);
}

function ioErrorHandler(event:IOErrorEvent):void {
    trace("Load failed due to: " + event.text);
}
```

6 Kontrol Et > Filmi Test Et'i seçin.

Sorgulu modda ProgressBar bileşeniyle bir uygulama oluşturma

Aşağıdaki örnek, ProgressBar öğesini sorgulu moda ayarlar. Sorgulu modda, ilerleme, yüklenen içerikteki progress olayları dinlenerek ve ilerlemenin hesaplanması için bytesLoaded ve bytesTotal özellikleri kullanılarak belirlenir. Bu örnek bir Sound nesnesi yükler, bu nesnenin progress olaylarını dinler ve bytesLoaded ve bytesTotal özelliklerini kullanarak yüklenen yüzdeyi hesaplar. Hem bir etikette hem de Çıktı panelinde yüklenen yüzdesini görüntüler.

1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.

2 Bir ProgressBar bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Özellik denetçisinde şu değerleri girin:

- Örnek adı için **aPb** girin.
- X değeri için **185** girin.
- Y değeri için **225** girin.

3 Bir Label bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde şu değerleri girin:

- Örnek adı için **progLabel** girin.
- X değeri için **180** girin.

- Y değeri için **180** girin.
- Parametreler bölümünde text parametresinin değerini temizleyin.

4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve bir Sound nesnesi (aSound) oluşturup Sound nesnesine ses yüklemek için loadSound() ögesini çağıran şu ActionScript kodunu girin:

```
import fl.controls.ProgressBarMode;
import flash.events.ProgressEvent;
import flash.media.Sound;

var aSound:Sound = new Sound();
var url:String = "http://www.helpexamples.com/flash/sound/song1.mp3";
var request:URLRequest = new URLRequest(url);

aPb.mode = ProgressBarMode.POLLED;
aPb.source = aSound;
aSound.addEventListener(ProgressEvent.PROGRESS, loadListener);

aSound.load(request);

function loadListener(event:ProgressEvent) {
    var percentLoaded:int = event.target.bytesLoaded / event.target.bytesTotal * 100;
    progLabel.text = "Percent loaded: " + percentLoaded + "%";
    trace("Percent loaded: " + percentLoaded + "%");
}
```

5 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

Manuel modunda ProgressBar bileşeniyle bir uygulama oluşturma

Aşağıdaki örnek, ProgressBar ögesini manuel moduna ayarlar. Manuel modunda, setProgress() yöntemini çağırıp ilerlemeyi manuel ayarlamamız ve ilerlemenin boyutunu belirlemek için ilerlemenin geçerli ve maksimum değerlerini sağlamanız gerekir. Manuel modunda source özelliği ayarlanmaz. Bu örnek, ProgressBar ögesini artırmak için maksimum 250 değeriyle bir NumericStepper bileşenini kullanır. NumericStepper ögesindeki değer değişip bir CHANGE olayını tetiklediğinde, olay işleyicisi (nsChangeHandler) ProgressBar ögesini yönlendirmek için setProgress() yöntemini çağırır. Ayrıca maksimum değeri esas alarak, ilerlemenin tamamlanan yüzdesini de görüntüler.

1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.

2 ProgressBar bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene şu değerleri verin:

- Örnek adı için **aPb** girin.
- X değeri için **180** girin.
- Y değeri için **175** girin.

3 Bir NumericStepper bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde şu değerleri girin:

- Örnek adı için **aNs** girin.
- X değeri için **220** girin.
- Y değeri için **215** girin.
- Parametreler bölümünde, maximum parametresi için **250**, minimum parametresi için **0**, stepSize parametresi için **1** ve value parametresi için **0** değerini girin.

4 Bir Label bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde şu değerleri girin:

- Örnek adı için **progLabel** girin.
- W değeri için **150** girin.
- X değeri için **180** girin.
- Y değeri için **120** girin.
- Parametreler sekmesinde text parametresinin Label değerini temizleyin.

5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
import fl.controls.ProgressBarDirection;
import fl.controls.ProgressBarMode;
import flash.events.Event;

aPb.direction = ProgressBarDirection.RIGHT;
aPb.mode = ProgressBarMode.MANUAL;
aPb.minimum = aNs.minimum;
aPb.maximum = aNs.maximum;
aPb.indeterminate = false;

aNs.addEventListener(Event.CHANGE, nsChangeHandler);

function nsChangeHandler(event:Event):void {
    aPb.value = aNs.value;
    aPb.setProgress(aPb.value, aPb.maximum);
    progLabel.text = "Percent of progress = " + int(aPb.percentComplete) + "%";
}
```

6 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

7 ProgressBar ögesini yönlendirmek için NumericStepper ögesinde Yukarı Ok'u tıklatın.

ActionScript kullanarak ProgressBar oluşturma

Bu örnek, ActionScript kullanarak bir ProgressBar oluşturur. Ayrıca, manuel modunda ProgressBar oluşturan önceki örneğin işlevselliğini çoğaltır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ProgressBar bileşenini Kütüphane paneline sürükleyin.
- 3 NumericStepper bileşenini Kütüphane paneline sürükleyin.
- 4 Label bileşenini Kütüphane paneline sürükleyin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
import fl.controls.ProgressBar;  
import fl.controls.NumericStepper;  
import fl.controls.Label;  
import fl.controls.ProgressBarDirection;  
import fl.controls.ProgressBarMode;  
import flash.events.Event;  
  
var aPb:ProgressBar = new ProgressBar();  
var aNs:NumericStepper = new NumericStepper();  
var progLabel:Label = new Label();  
  
addChild(aPb);  
addChild(aNs);  
addChild(progLabel);  
  
aPb.move(180,175);  
aPb.direction = ProgressBarDirection.RIGHT;  
aPb.mode = ProgressBarMode.MANUAL;  
  
progLabel.setSize(150, 22);  
progLabel.move(180, 150);  
progLabel.text = "";  
  
aNs.move(220, 215);  
aNs.maximum = 250;  
aNs.minimum = 0;  
aNs.stepSize = 1;  
aNs.value = 0;  
  
aNs.addEventListener(Event.CHANGE, nsChangeHandler);  
  
function nsChangeHandler(event:Event):void {  
    aPb.setProgress(aNs.value, aNs.maximum);  
    progLabel.text = "Percent of progress = " + int(aPb.percentComplete) + "%";  
}
```

- 6 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.
- 7 ProgressBar öğesini yönlendirmek için NumericStepper öğesinde Yukarı Ok'u tıklattın.

RadioButton bileşenini kullanma

RadioButton bileşeni kullanıcıyı seçenekler kümesinde tek bir tercih yapmaya zorlamanıza imkan tanır. Bu bileşen, en az iki RadioButton örneğinin bulunduğu bir grupta kullanılmalıdır. Herhangi bir belirli zamanda yalnızca bir grup üyesi seçilebilir. Grupta bir radyo düğmesi seçildiğinde, grupta geçerli olarak seçilen radyo düğmesinin seçimini kaldırılır. `groupName` parametresini, bir radyo düğmesinin ait olduğu grubu belirtecek şekilde ayarlırsınız.

Radyo düğmesi, web'deki çoğu form uygulamasının temel bölümünü oluşturur. Kullanıcının seçenekler grubunda tek bir tercih yapmasını istediğiniz her yerde radyo düğmelerini kullanabilirsiniz. Örneğin, müşterinin hangi kredi kartını kullanmak istediğini sormak için formda radyo düğmelerini kullanabilirsiniz.

RadioButton bileşeniyle kullanıcı etkileşimi

Radyo düğmesi etkinleştirilebilir veya devre dışı bırakılabilir. Devre dışı bırakılmış bir radyo düğmesi, fare veya klavye girdisini kabul etmez. Kullanıcı, bir RadioButton bileşen grubunu tıklattığında veya sekme ile geçtiğinde, yalnızca seçili radyo düğmesi odağı alır. Daha sonra kullanıcı odağı denetlemek için şu tuşları kullanabilir:

Anahtar	Açıklama
Yukarı Ok/Sol Ok	Seçim radyo düğmesi grubu içinde önceki radyo düğmesine taşınır.
Aşağı Ok/Sağ Ok	Seçim radyo düğmesi grubu içinde sonraki radyo düğmesine taşınır.
Sekme	Odağı radyo düğmesi grubundan sonraki bileşene taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki IFocusManager arabirimine ve FocusManager sınıfına bakın.

Sahne Alanı'nda her bir RadioButton örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır. Ancak seçimin birbirini dışlaması canlı önizlemede görüntülenmez. Aynı gruptaki iki radyo düğmesi için seçili parametreyi `true` değerine ayarlarsanız, çalışma zamanında yalnızca oluşturulan son örnek seçilmiş olarak görüntülense de, her iki örnek birden seçilmiş olarak görüntülenir. Daha fazla bilgi için bkz. “[RadioButton bileşeni parametreleri](#)” sayfa 76.

Bir uygulamaya RadioButton bileşenini eklediğinizde, aşağıdaki kod satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.RadioButtonAccImpl;  
RadioButtonAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğine sahip olduğunuza bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirirsiniz. Daha fazla bilgi için bkz. Bölüm 18, “Erişilebilir İçerik Oluşturma,” Flash Uygulamasını Kullanma.

RadioButton bileşeni parametreleri

Her RadioButton bileşeni örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: `groupName`, `label`, `LabelPlacement`, `selected` ve `value`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki RadioButton sınıfına bakın.

RadioButton sınıfının yöntemlerini, özelliklerini ve olaylarını kullanarak RadioButton ögesi için ek seçenekler ayarlamak üzere ActionScript yazabilirsiniz.

RadioButton bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl RadioButton bileşenleri ekleneceğini açıklamaktadır. Bu örnekte, evet veya hayır yanıtı vermek için RadioButton ögeleri kullanılır. RadioButton ögesindeki veriler TextArea üzerinde görüntülenir.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 İki RadioButton bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Birinci radyo düğmesini seçin. Özellik denetçisinde, bu radyo düğmesine **yesRb** örnek adını ve **rbGroup** grup adını verin.
- 4 İkinci radyo düğmesini seçin. Özellik denetçisinde, bu radyo düğmesine **noRb** örnek adını ve **rbGroup** grup adını verin.

5 TextArea bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene aTa örnek adını verin.

6 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
yesRb.label = "Yes";  
yesRb.value = "For";  
noRb.label = "No";  
noRb.value = "Against";  
  
yesRb.move(50, 100);  
noRb.move(100, 100);  
aTa.move(50, 30);  
noRb.addEventListener(MouseEvent.CLICK, clickHandler);  
yesRb.addEventListener(MouseEvent.CLICK, clickHandler);  
  
function clickHandler(event:MouseEvent):void {  
    aTa.text = event.target.value;  
}
```

7 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak RadioButton oluşturma

Bu örnek, kırmızı, mavi ve yeşil renkleri için üç RadioButtons öğesi oluşturmak üzere ActionScript kullanır ve gri bir kutu çizer. Her RadioButton öğesinin value özelliği, o düğmeyle ilişkilendirilmiş rengin onaltılık değerini belirtir. Kullanıcı RadioButton öğelerinden birini tıklattığında, clickHandler() işlevi drawBox() öğesini çağırarak kutuyu renklendirmek için RadioButton öğesinin value özelliğindeki rengi iletir.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 RadioButton bileşenini Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.RadioButton;
import fl.controls.RadioButtonGroup;

var redRb:RadioButton = new RadioButton();
var blueRb:RadioButton = new RadioButton();
var greenRb:RadioButton = new RadioButton();
var rbGrp:RadioButtonGroup = new RadioButtonGroup("colorGrp");

var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xCCCCCC);

addChild(redRb);
addChild(blueRb);
addChild(greenRb);
addChild(aBox);

redRb.label = "Red";
redRb.value = 0xFF0000;
blueRb.label = "Blue";
blueRb.value = 0x0000FF;
greenRb.label = "Green";
greenRb.value = 0x00FF00;
redRb.group = blueRb.group = greenRb.group = rbGrp;
redRb.move(100, 260);
blueRb.move(150, 260);
greenRb.move(200, 260);

rbGrp.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {
    drawBox(aBox, event.target.selection.value);
}

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(125, 150, 100, 100);
    box.graphics.endFill();
}
```

4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki RadioButton sınıfına bakın.

ScrollPane bileşenini kullanma

ScrollPane bileşenini, yükleneceği alana göre çok büyük olan içerikleri görüntülemeye kullanabilirsiniz. Örneğin, büyük bir görüntünüz varsa ve uygulamanızda bu görüntü için çok küçük bir alan bulunuyorsa, görüntüyü ScrollPane ögesine yükleyebilirsiniz. ScrollPane film kliplerini, JPEG, PNG, GIF ve SWF dosyalarını kabul edebilir.

ScrollPane ve UILoader gibi bileşenler, içeriğin ne zaman yüklenmeyi bitirdiğini belirlemenize olanak sağlayan complete olayları vardır. ScrollPane veya UILoader bileşeninin özelliklerini ayarlamak istiyorsanız, complete olayını dinleyin ve olay işleyicisinde özelliği ayarlayın. Örneğin, şu kod Event ögesi için bir dinleyici oluşturur.COMPLETE olayı ve ScrollPane içeriğinin alpha özelliğini 0,5 değerine ayarlayan olay işleyicisi:

```
function spComplete(event:Event):void{  
    aSp.content.alpha = .5;  
}  
aSp.addEventListener(Event.COMPLETE, spComplete);
```

ScrollPane ögesine içerik yüklerken bir konum belirtirseniz, konumu (X ve Y koordinatları) 0, 0 olarak belirtmeniz gerekir. Örneğin, kutu 0, 0 konumunda çizildiğinden, şu kod ScrollPane ögesini düzgün şekilde yükler:

```
var box:MovieClip = new MovieClip();  
box.graphics.beginFill(0xFF0000, 1);  
box.graphics.drawRect(0, 0, 150, 300);  
box.graphics.endFill();  
aSp.source = box;//load ScrollPane
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki ScrollPane sınıfına bakın.

ScrollPane bileşeniyle kullanıcı etkileşimi

ScrollPane etkinleştirilebilir veya devre dışı bırakılabilir. Devre dışı bırakılmış ScrollPane ögesi, fare veya klavye girdisini kabul etmez. ScrollPane odağı aldığında ScrollPane ögesini denetlemek için kullanıcı şu tuşları kullanabilir:

Anahtar	Açıklama
Aşağı Ok	İçerik bir dikey satır yukarı taşınır.
Yukarı Ok	İçerik bir dikey satır aşağı taşınır.
End	İçerik ScrollPane ögesinin en alt kısmına taşınır.
Sol Ok	İçerik bir yatay satır sağa taşınır.
Sağ Ok	İçerik bir yatay satır sola taşınır.
Home	İçerik ScrollPane ögesinin en üst kısmına taşınır.
End	İçerik ScrollPane ögesinin en alt kısmına taşınır.
PageDown	İçerik bir dikey sayfa yukarı taşınır.
PageUp	İçerik bir dikey sayfa aşağı taşınır.

Kullanıcı ScrollPane ögesinin içeriğinde ve dikey ve yatay kaydırma çubuklarında ScrollPane ögesiyle etkileşim kurma için fareyi kullanabilir. `scrollDrag` özelliği `true` değerine ayarlandığında kullanıcı fareyi kullanarak içeriği sürükleyebilir. İçerikteki el işaretçisi görünümü, kullanıcının içeriği sürükleyebildiğini belirtir. Diğer denetimlerden farklı olarak, fare düğmesine basıldığında eylemler gerçekleşir ve fare düğmesi serbest bırakılıncaya kadar devam eder. İçeriğin geçerli sekme durakları yoksa, `scrollDrag` ögesini `false` değerine ayarlamanız gerekir. Aksi takdirde, içeriklerdeki tüm fare vuruşları kaydırarak sürüklemeyi çağırır.

ScrollPane bileşeni parametreleri

Özellik denetçisinde ya da Bileşen denetçisinde her ScrollPane örneği için şu parametreleri ayarlayabilirsiniz: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` ve `verticalScrollPolicy`. Bu parametrelerin her biri, aynı anda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki ScrollPane sınıfına bakın.

Bunları ve ScrollPane bileşeninin ek seçeneklerini denetlemek için ScrollPane bileşeninin özelliklerini, yöntemlerini ve olaylarını kullanarak ActionScript yazabilirsiniz.

ScrollPane bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl ScrollPane bileşeni ekleneceğini açıklamaktadır. Bu örnekte, ScrollPane ögesi `source` özelliği tarafından belirtilen bir yoldan resim yükler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ScrollPane bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **aSp** örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.events.ScrollEvent;

aSp.setSize(300, 200);

function scrollListener(event:ScrollEvent):void {
    trace("horizontalScPosition: " + aSp.horizontalScrollPosition +
        ", verticalScrollPosition = " + aSp.verticalScrollPosition);
};
aSp.addEventListener(ScrollEvent.SCROLL, scrollListener);

function completeListener(event:Event):void {
    trace(event.target.source + " has completed loading.");
};
// Add listener.
aSp.addEventListener(Event.COMPLETE, completeListener);

aSp.source = "http://www.helpexamples.com/flash/images/image1.jpg";
```

- 4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak ScrollPane örneği oluşturma

Bu örnek bir ScrollPane ögesi oluşturur, bu ögenin boyutunu ayarlar ve `source` özelliğini kullanarak bu ögeye bir görüntü yükler. Ayrıca iki dinleyici oluşturur. Birinci dinleyici, `scroll` olayını dinler ve kullanıcı dikey veya yatay olarak kaydırma yaptıkça görüntünün konumunu görüntüler. İkinci dinleyici, `complete` olayını dinler ve görüntünün yüklemeyi tamamladığını bildiren bir mesajı Çıktı panelinde görüntüler.

Bu örnek, ActionScript kullanarak bir ScrollPane ögesi oluşturur ve bu ögeye 150 piksel genişliğinde ve 300 piksel uzunluğunda bir MovieClip (kırmızı bir kutu) yerleştirir.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ScrollPane bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 DataGrid bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.containers.ScrollPane;
import fl.controls.ScrollPolicy;
import fl.controls.DataGrid;
import fl.data.DataProvider;

var aSp:ScrollPane = new ScrollPane();
var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xFF0000);//draw a red box

aSp.source = aBox;
aSp.setSize(150, 200);
aSp.move(100, 100);

addChild(aSp);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1);
    box.graphics.drawRect(0, 0, 150, 300);
    box.graphics.endFill();
}
```

5 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

Slider bileşenini kullanma

Slider bileşeni, grafiksel *thumb* öğesini değerler aralığına karşılık gelen bir yolun uç noktaları arasında kaydırarak kullanıcının bir değer seçmesine olanak tanır. Örneğin kullanıcının bir sayı veya yüzde gibi bir değer seçmesine olanak sağlamak için kaydırıcı kullanabilirsiniz. Kaydırıcının değerinin ikinci nesnenin davranışını etkilemesini sağlamak için de ActionScript kullanabilirsiniz. Örneğin, kaydırıcıyı bir resimle ilişkilendirebilir ve kaydırıcının kaydırma kutusunun göreceli konumunu veya değerini esas alarak kaydırıcıyı daraltabilir ya da genişletebilirsiniz.

Slider öğesinin geçerli değeri, kaydırma kutusunun yolun uç noktaları arasındaki göreceli konumuyla veya Slider öğesinin minimum ve maksimum değerleriyle belirlenir.

Slider öğesi, minimum ve maksimum değerleri arasında sürekli bir değer aralığına olanak sağlar ancak minimum ve maksimum değer arasında aralıkları belirtmek için `snapInterval` parametresini de ayarlayabilirsiniz. Slider öğesi, kaydırıcının atanmış değerlerinden bağımsız olan onay işaretlerini yol boyunca belirtilen aralıklarda gösterebilir.

Kaydırıcının varsayılan olarak yönlendirmesi yataydır ancak `direction` parametresinin değerini dikey olarak ayarlayarak kaydırıcının yönlendirmesini dikey durumuna getirebilirsiniz. Kaydırıcı yolu bir uçtan diğer uca uzanır ve onay işaretleri yolun hemen üzerinden soldan sağa doğru yerleştirilir.

Slider bileşeniyle kullanıcı etkileşimi

Bir Slider örneği odağa sahip olduğunda, odağı denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Sağ Ok	Yatay kaydırıcı için ilişkilendirilmiş değeri artırır.
Yukarı Ok	Dikey kaydırıcı için ilişkilendirilmiş değeri artırır.
Sol Ok	Yatay kaydırıcı için ilişkilendirilmiş değeri azaltır.

Anahtar	Açıklama
Aşağı Ok	Dikey kaydırıcı için ilişkilendirilmiş değeri azaltır.
Üst Karakter+Sekme	Odağı bir önceki nesneye taşır.
Sekme	Odağı bir sonraki nesneye taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki IFocusManager arabirimine ve FocusManager sınıfına bakın.

Her bir Slider örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır.

Slider bileşeni parametreleri

Her Slider bileşeni örneği için Özellik denetçisinde ya da Bileşen denetçisinde aşağıdaki yönetim parametrelerini ayarlayabilirsiniz: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` ve `value`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki Slider sınıfına bakın.

Slider ile uygulama oluşturma

Şu örnek, kullanıcının varsayım olayıyla memnuniyet düzeyini ifade etmesine olanak sağlamak için bir Slider örneği oluşturur. Kullanıcı, daha yüksek veya daha alçak bir memnuniyet düzeyi belirtmek için Slider öğesini sağa ya da sola taşır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Label bileşenini Bileşenler panelinden Sahne Alanı'nın ortasına sürükleyin.
 - Bileşene **valueLabel** örnek adını verin.
 - `text` parametresine **0percent** değerini atayın.
- 3 Slider bileşenini Bileşenler panelinden sürükleyip `value_1bl` öğesinin aşağısına ortalayın.
 - Bileşene **aSlider** örnek adını verin.
 - **200** genişlik (W:) değeri atayın.
 - **10** yükseklik (H:) değeri atayın.
 - `maximum` parametresine **100** değeri atayın.
 - Hem `snapInterval` hem de `tickInterval` parametresine **10** değeri atayın.
- 4 Kütüphane panelinden başka bir Label örneğini sürükleyip `aSlider` öğesinin aşağısına ortalayın.
 - Bileşene **promptLabel** örnek adını verin.
 - **250** genişlik (W:) değeri atayın.
 - **22** yükseklik (H:) değeri atayın.
 - `text` parametresi için **Lütfen memnuniyet düzeyinizi belirtin** metnini girin.
- 5 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.Slider;
import fl.events.SliderEvent;
import fl.controls.Label;

aSlider.addEventListener(SliderEvent.CHANGE, changeHandler);

function changeHandler(event:SliderEvent):void {
    valueLabel.text = event.value + "percent";
}
```

6 Kontrol Et > Filmi Test Et'i seçin.

Bu örnekte, kaydırıcının kaydırma kutusunu bir aralıktan diğerine taşıdığımızda, kaydırma kutusunun konumuna karşılık gelen yüzdeyi görüntülemek için, `SliderEvent.CHANGE` olayının bir dinleyicisi, `valueLabel` öğesinin `text` özelliğini günceller.

ActionScript kullanarak Slider bileşeniyle uygulama oluşturma

Aşağıdaki örnek, ActionScript kullanarak bir Slider öğesi oluşturur. Bu örnek, bir çiçek görüntüsünü indirir ve Slider öğesinin değerine karşılık gelmesi için görüntünün `alpha` özelliğini değiştirerek kullanıcının görüntüyü karartmasına veya aydınlatmasına olanak sağlamak için Slider öğesini kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Label bileşenini ve Slider bileşenini Bileşenler panelinden geçerli belgenin Kütüphane paneline sürükleyin.
Bu işlem sonucunda bileşenler kütüphaneye eklenir ancak uygulamada görünür duruma getirilmez.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve bileşen örnekleri oluşturup konumlandırmak için şu kodu girin:

```
import fl.controls.Slider;
import fl.events.SliderEvent;
import fl.controls.Label;
import fl.containers.UILoader;

var sliderLabel:Label = new Label();
sliderLabel.width = 120;
sliderLabel.text = "< Fade - Brighten >";
sliderLabel.move(170, 350);

var aSlider:Slider = new Slider();
aSlider.width = 200;
aSlider.snapInterval = 10;
aSlider.tickInterval = 10;
aSlider.maximum = 100;
aSlider.value = 100;
aSlider.move(120, 330);

var aLoader:UILoader = new UILoader();
aLoader.source = "http://www.flash-mx.com/images/image1.jpg";
aLoader.scaleContent = false;

addChild(sliderLabel);
addChild(aSlider);
addChild(aLoader);

aLoader.addEventListener(Event.COMPLETE, completeHandler);

function completeHandler(event:Event) {
    trace("Number of bytes loaded: " + aLoader.bytesLoaded);
}

aSlider.addEventListener(SliderEvent.CHANGE, changeHandler);

function changeHandler(event:SliderEvent):void {
    aLoader.alpha = event.value * .01;
}
```

4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

5 Slider öğesinin kaydırma kutusunu görüntüyü karartmak için sola, görüntüyü aydınlatmak için sağa taşıyın.

TextArea bileşenini kullanma

TextArea bileşeni, yerel ActionScript TextField nesnesi için bir sözcük kaydırıcısıdır. Metni görüntülemek ve editable özelliği true olduğunda metin girdisi almak için de TextArea bileşenini kullanabilirsiniz. wordWrap özelliği true değerine ayarlanırsa, bileşen birden çok metin satırını görüntüleyebilir veya alabilir ve uzun metin satırlarını sarar. restrict özelliği, kullanıcının girebileceği karakterleri kısıtlamanıza olanak sağlar, maxChars ögesi de kullanıcının girebileceği maksimum karakter sayısı belirtmenize olanak sağlar. Metin, metin alanının yatay ve dikey sınırlarını aşarsa, kaydırma çubuklarının ilişkilendirilmiş özellikleri (horizontalScrollPolicy ve verticalScrollPolicy) off olarak ayarlanmadığı sürece, otomatik olarak yatay ve dikey kaydırma çubukları görüntülenir.

Birden çok metin alanına ihtiyaç duyduğunuz her yerde bir TextArea bileşeni kullanabilirsiniz. Örneğin, bir formda yorum alanı olarak TextArea bileşenini kullanabilirsiniz. Kullanıcı sekmenin seçimini kaldırarak alandan çıktığında alanın boş olup olmadığını kontrol eden bir dinleyici ayarlayabilirsiniz. Dinleyici, alana bir yorumun girilmesi gerektiğini belirten bir hata mesajı görüntüleyebilir.

Tek satırlık bir metin alanına ihtiyacınız varsa TextInput bileşenini kullanın.

TextArea örneğinde görüntülenen metnin stilini değiştirmek için, `setStyle()` yöntemini kullanarak `textFormat` stilini ayarlayabilirsiniz. Ayrıca ActionScript'te `htmlText` özelliğini kullanarak TextArea bileşenini HTML ile biçimlendirebilir ve metni yıldız işaretleriyle maskeleyerek için `displayAsPassword` özelliğini `true` değerine ayarlayabilirsiniz. `condenseWhite` özelliğini `true` değerine ayarlarsanız, Flash boşluklardan, satır kesmelerinden, vb'den kaynaklanan fazladan beyaz alanı kaldırır. Denetimde önceden yer alan metin üzerinde herhangi bir etkisi yoktur.

TextArea bileşeniyle kullanıcı etkileşimi

TextArea bileşeni, bir uygulamada etkinleştirilebilir veya devre dışı bırakılabilir. Devre dışı durumunda fare veya klavye girdisi alamaz. Etkinleştirildiğinde, ActionScript TextField nesnesiyle aynı odak, seçim ve gezinme kurallarını izler. TextArea örneği odağı aldığı anda, şu tuşları kullanarak odağı denetleyebilirsiniz:

Anahtar	Açıklama
Ok tuşları	Metin düzenlenebilir durumdaysa, metin içinde ekleme noktasını yukarı, aşağı, sola veya sağa taşır.
Sayfa Aşağı	Metin düzenlenebilir durumdaysa, ekleme noktasını metnin sonuna taşır.
Sayfa Yukarı	Metin düzenlenebilir durumdaysa, ekleme noktasını metnin başına taşır.
Üst Karakter+Sekme	Sekme döngüsünde odağı bir önceki nesneye taşır.
Sekme	Sekme döngüsünde odağı bir sonraki nesneye taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki FocusManager sınıfına bakın.

TextArea bileşeni parametreleri

Özellik denetçisinde ya da Bileşen denetçisinde her TextArea bileşeni örneği için şu yönetim parametrelerini ayarlayabilirsiniz: `condenseWhite`, `düzenlenebilir`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` ve `wordwrap`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki TextArea sınıfına bakın.

Her bir TextArea örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır. Kaydırma çubuğu gerekiyorsa, canlı önizlemede görüntülenir ancak çalışmaz. Metin canlı önizlemede seçilemez ve Sahne Alanı'nda bileşen örneğine metin giremezsiniz.

Bunları ve TextArea bileşeninin ek seçeneklerini denetlemek için TextArea bileşeninin özelliklerini, yöntemlerini ve olaylarını kullanarak ActionScript yazabilirsiniz. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki TextArea sınıfına bakın.

TextArea bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl TextArea bileşeni ekleneceğini açıklamaktadır. Bu örnek, TextArea ögesinde, arabirimin farklı bir bölümüne odağı vermeden önce kullanıcının metin alanına bir şeyler yazdığını doğrulayan `focusOut` olay işleyicisini ayarlar.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 TextArea bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **aTa** örnek adını verin. Parametrelerini varsayılan ayarlarında bırakın.
- 3 İkinci bir TextArea bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyip birinci TextArea ögesinin aşağısına yerleştirin ve bu bileşene **bTa** örnek adını verin. Parametrelerini varsayılan ayarlarında bırakın.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import flash.events.FocusEvent;

aTa.restrict = "a-z, \\\" \";
aTa.addEventListener(Event.CHANGE, changeHandler);
aTa.addEventListener(FocusEvent.KEY_FOCUS_CHANGE, k_m_fHandler);
aTa.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, k_m_fHandler);

function changeHandler(ch_evt:Event):void {
    bTa.text = aTa.text;
}
function k_m_fHandler(kmf_event:FocusEvent):void {
    kmf_event.preventDefault();
}
```

Bu örnek, aTa metin alanına girebileceğiniz karakterleri küçük harfli karakterler, virgül, kesme işareti ve boşluklarla sınırlar. Ayrıca aTa metin alanında da `change`, `KEY_FOCUS_CHANGE` ve `MOUSE_FOCUS_CHANGE` olaylarını ayarlar. `changeHandler()` işlevi, her bir `change` olayında `aTa.text` ögesini `bTa.text` olarak ayarlayarak metin alanına girebileceğiniz metnin otomatik olarak bTa metin alanında görüntülenmesini sağlar. `KEY_FOCUS_CHANGE` ve `MOUSE_FOCUS_CHANGE` olaylarının `k_m_fHandler()` işlevi, herhangi bir metin girmeden bir sonraki alana geçmeniz için Sekme tuşuna basmanızı önler. Varsayılan davranışı önleyerek bunu yapar.

- 5 Kontrol Et > Filmi Test Et'i seçin.

Herhangi bir metin girmeden odağı ikinci metin alanına taşımak için Sekme tuşuna basarsanız, bir hata mesajı görmeniz ve odağın birinci metin alanına geri dönmesi gerekir. Birinci metin alanına metin girdiğinizde, bu metnin ikinci metin alanında yinelendiğini görürsünüz.

ActionScript kullanarak TextArea örneği oluşturma

Aşağıdaki örnek, ActionScript ile TextArea bileşeni oluşturur. Bu örnek, beyaz alanı sıkıştırmak için `condenseWhite` özelliğini `true` değerine ayarlar ve HTML metin biçimlendirme niteliklerinden yararlanmak için `htmlText` özelliğine metin atar.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 TextArea bileşenini Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

aTa.move(100,100);
aTa.setSize(200, 200);
aTa.condenseWhite = true;
aTa.htmlText = '<b>Lorem ipsum dolor</b> sit amet, consectetur adipiscing elit. <u>Vivamus quis nisl vel tortor nonummy vulputate.</u> Quisque sit amet eros sed purus euismod tempor. Morbi tempor. <font color="#FF0000">Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos.</font> Curabitur diam. Suspendisse at purus in ipsum volutpat viverra. Nulla pellentesque libero id libero.';
addChild(aTa);
```

Bu örnek, bir metin bloğuna HTML kalın ve alt çizgi niteliklerini uygulamak ve bu metin bloğunu a_ta metin alanında görüntülemek için htmlText özelliğini kullanır. Bu örnek aynı zamanda metin bloğu içinde beyaz alanı sıkıştırmak için condenseWhite özelliğini true değerine ayarlar. setSize() yöntemi, metin alanının yükseklik ve genişliğini ayarlarken, move() yöntemi de konumunu ayarlar. addChild() yöntemi, TextArea örneğini Sahne Alanı'na ekler.

4 Kontrol Et > Filmi Test Et'i seçin.

TextInput bileşenini kullanma

TextInput bileşeni, yerel ActionScript TextField nesnesi için bir sözcük kaydırıcı olan tek satırlı bir metin bileşenidir. Çok satırlı bir metin bileşeni gerekiyorsa, TextArea bileşenini kullanın. Örneğin, bir formda şifre alanı olarak TextInput bileşenini kullanabilirsiniz. Kullanıcı sekmenin seçimini kaldırarak alandan çıktığında alanda yeterince karakter olup olmadığını kontrol eden bir dinleyici ayarlayabilirsiniz. Bu dinleyici, uygun sayıda karakter girilmesi gerektiğini belirten bir hata mesajı görüntüleyebilir.

TextInput örneğinde görüntülenen metnin stilini değiştirmek için, setStyle() yöntemini kullanarak textFormat özelliğini ayarlayabilirsiniz. TextInput bileşeni, HTML ile veya metni gizleyen bir şifre alanı olarak da biçimlendirilebilir.

TextInput bileşeniyle kullanıcı etkileşimi

TextInput bileşeni, bir uygulamada etkinleştirilebilir veya devre dışı bırakılabilir. Devre dışı durumunda fare veya klavye girdisi alamaz. Etkinleştirildiğinde, ActionScript TextField nesnesiyle aynı odak, seçim ve gezinme kurallarını izler. Bir TextInput örneği odağa sahip olduğunda, odağı denetlemek için şu tuşları kullanabilirsiniz:

Anahtar	Açıklama
Ok tuşları	Ekleme noktasını bir karakter sola veya sağa taşır.
Üst Karakter+Sekme	Odağı bir önceki nesneye taşır.
Sekme	Odağı bir sonraki nesneye taşır.

Odağı kontrol etmeyle ilgili daha fazla bilgi için, *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* ve “FocusManager ile çalışma” sayfa 26 bölümündeki FocusManager sınıfına bakın.

Her bir TextInput örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır. Metin canlı önizlemede seçilemez ve Sahne Alanı'nda bileşen örneğine metin giremezsiniz.

Bir uygulamaya TextInput bileşenini eklediğinizde, bileşeni ekran okuyucusu için erişilebilir duruma getirmek için Erişilebilirlik panelini kullanabilirsiniz.

TextInput bileşeni parametreleri

Özellik denetçisinde veya Bileşen denetçisinde her TextInput bileşeni örneği için şu yönetim parametrelerini ayarlayabilirsiniz: `editable`, `displayAsPassword`, `maxChars`, `restrict` ve `text`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelerin olası değerleriyle ilgili bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki TextInput sınıfına bakın.

Bunları ve TextInput bileşeninin ek seçeneklerini denetlemek için TextInput bileşeninin özelliklerini, yöntemlerini ve olaylarını kullanarak ActionScript yazabilirsiniz. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki TextInput sınıfına bakın.

TextInput bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl TextInput bileşeni ekleneceğini açıklamaktadır. Bu örnek, şifre almak ve şifreyi onaylamak için iki TextInput alanı kullanır. Bu örnek, minimum sekiz karakter girildiğini ve iki alandaki metnin eşleştiğini görmek için bir olay dinleyicisi kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir Label bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene şu değerleri verin:
 - **pwdLabel** örnek adını girin.
 - W için **100** değerini girin.
 - X için **50** değerini girin.
 - Y için **150** değerini girin.
 - Parametreler bölümünde, metin parametresi için bir **Password:** değeri girin.
- 3 İkinci bir Label bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bileşene şu değerleri verin:
 - **confirmLabel** örnek adını girin.
 - W için **100** değerini girin.
 - X için **50** değerini girin.
 - Y için **200** değerini girin.
 - Parametreler bölümünde, metin parametresi için bir **ConfirmPassword:** değeri girin.
- 4 Bir TextInput bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bileşene şu değerleri verin:
 - **pwdTi** örnek adını girin.
 - W için **150** değerini girin.
 - X için **190** değerini girin.
 - Y için **150** değerini girin.
 - Parametreler bölümünde, `displayAsPassword` parametresinin değerini çift tıklatın ve **true** seçeneğini belirleyin. Böylece metin alanına girilen değer yıldız işaretleriyle maskelenmesi sağlanır.
- 5 İkinci bir TextInput bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bileşene şu değerleri verin:
 - **confirmTi** örnek adını girin.

- W için **150** değerini girin.
- X için **190** değerini girin.
- Y için **200** değerini girin.
- Parametreler bölümünde, `displayAsPassword` parametresinin değerini çift tıklatın ve **true** seçeneğini belirleyin. Böylece metin alanına girilen değer yıldız işaretleriyle maskelenmesi sağlanır.

6 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
function tiListener(evt_obj:Event){
    if(confirmTi.text != pwdTi.text || confirmTi.length < 8)
    {
        trace("Password is incorrect. Please reenter it.");
    }
    else {
        trace("Your password is: " + confirmTi.text);
    }
}
confirmTi.addEventListener("enter", tiListener);
```

Bu kod, `confirmTi` adındaki `TextInput` örneğinde bir `enter` olay işleyicisi ayarlar. İki şifre eşleşmezse veya kullanıcı sekizden az karakter girerse, örnek, “Şifre yanlış. Lütfen yeniden girin.” mesajını görüntüler. Şifreler en az sekiz karakterse ve birbirleriyle eşleşiyorsa, örnek, Çıktı panelinde girilen değeri görüntüler.

7 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak TextInput örneği oluşturma

Aşağıdaki örnek, ActionScript ile `TextInput` bileşeni oluşturur. Bu örnek ayrıca kullanıcının adını girmesini istemek için kullanılan bir `Label` ögesi de oluşturur. Örnek, bileşenin `restrict` özelliğini yalnızca büyük harf ve küçük harf, nokta ve boşluğa izin verecek şekilde ayarlar. Ayrıca, hem `Label` hem de `TextInput` bileşeninde metni biçimlendirmek için kullanılan bir `TextFormat` nesnesi de oluşturur.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 `TextInput` bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 `Label` bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.controls.Label;
import fl.controls.TextInput;

var nameLabel:Label = new Label();
var nameTi:TextInput = new TextInput();
var tf:TextFormat = new TextFormat();

addChild(nameLabel);
addChild(nameTi);

nameTi.restrict = "A-Z .a-z";

tf.font = "Georgia";
tf.color = 0x0000CC;
tf.size = 16;

nameLabel.text = "Name: " ;
nameLabel.setSize(50, 25);
nameLabel.move(100,100);
nameLabel.setStyle("textFormat", tf);
nameTi.move(160, 100);
nameTi.setSize(200, 25);
nameTi.setStyle("textFormat", tf);
```

5 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

TileList bileşenini kullanma

TileList bileşeni, bir veri sağlayıcısının sunduğu verilerle birlikte gelen satır ve sütunlardan oluşan bir listeden oluşur. Bir *item*, TileList içindeki bir hücrede saklanan veri birimini ifade eder. Veri sağlayıcısının sunduğu bir öge genellikle bir *label* özelliğine ve bir *source* özelliğine sahiptir. *label* özelliği, bir hücrede görüntülenecek içeriği tanımlarken *source* ögesi de bunun için bir değer sağlar.

Bir Array örneği oluşturabilir veya sunucudan bir Array örneği alabilirsiniz. TileList bileşeni, veri sağlayıcısı için proxy olarak çalışan *addItem()* ve *removeItem()* gibi yöntemlere sahiptir. Listeye herhangi bir harici veri sağlayıcısı sağlanmazsa, bu yöntemler otomatik olarak *List.dataProvider* üzerinden kullanıma sunulan bir veri sağlayıcısı oluşturur.

TileList bileşeniyle kullanıcı etkileşimi

TileList ögesi, her bir hücreyi *ICellRenderer* arabirimini uygulayan bir *Sprite* ögesini kullanarak oluşturur. *TileList cellRenderer* özelliğiyle bu oluşturucuyu belirtebilirsiniz. *TileList* bileşeninin varsayılan *CellRenderer* ögesi, görüntü (sınıf, bitmap, örnek veya URL) gösteren *ImageCell* ve isteğe bağlı bir etikettir. Etiket, her zaman hücrenin alt kısmında hizalanan tek bir satırdır. *TileList* ögesini yalnızca bir yönde kaydırabilirsiniz.

TileList örneği odağı aldığı anda, örneğin içindeki öğelere erişmek için şu tuşları da kullanabilirsiniz:

Anahtar	Açıklama
Yukarı Ok ve Aşağı Ok	Bir sütunda yukarı be aşağı hareket etmenize olanak sağlar. <code>allowMultipleSelection</code> özelliği <code>true</code> olursa, birden çok hücre seçmek için Üst Karakter tuşuyla birlikte bu tuşları kullanabilirsiniz.
Sol Ok ve Sağ Ok	Bir satırda sola veya sağa hareket etmenize olanak sağlar. <code>allowMultipleSelection</code> özelliği <code>true</code> olursa, birden çok hücre seçmek için Üst Karakter tuşuyla birlikte bu tuşları kullanabilirsiniz.
Home	TileList ögesinde birinci hücreyi seçer. <code>allowMultipleSelection</code> özelliği <code>true</code> olursa, Üst Karakter tuşu basılı tutulduğunda Home tuşu geçerli seçiminizden birinci hücreye kadar tüm hücreleri seçer.
End	TileList ögesinde son hücreyi seçer. <code>allowMultipleSelection</code> özelliği <code>true</code> olursa, Üst Karakter tuşu basılı tutulduğunda End tuşu geçerli seçiminizden son hücreye kadar tüm hücreleri seçer.
Ctrl	<code>allowMultipleSelection</code> özelliği <code>true</code> değerine ayarlanırsa, belirli bir sıralama olmaksızın birden çok hücre seçmenize olanak sağlar.

Bir uygulamaya TileList bileşenini eklediğinizde, aşağıdaki ActionScript kodu satırlarını ekleyerek bu bileşeni bir ekran okuyucusu için erişilebilir duruma getirebilirsiniz:

```
import fl.accessibility.TileListAccImpl;  
  
TileListAccImpl.enableAccessibility();
```

Bileşenin kaç tane örneğe sahip olduğuna bakılmaksızın, bir bileşenin erişilebilirliğini yalnızca bir defa etkinleştirebilirsiniz. Daha fazla bilgi için bkz. Bölüm 18, “Erişilebilir İçerik Oluşturma,” *Flash Uygulamasını Kullanma*.

TileList bileşeni parametreleri

Özellik denetçisinde veya Bileşen denetçisinde her TileList bileşeni örneğinde şu yönetim parametrelerini ayarlayabilirsiniz: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `ScrollPolicy`, `verticalScrollLineSize` ve `verticalScrollPageSize`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. `dataProvider` parametresini kullanma hakkında bilgi almak için, bkz. “[dataProvider parametresini kullanma](#)” sayfa 28.

TileList örneklerinin yöntemlerini, özelliklerini ve olaylarını kullanarak TileList örnekleri için ek seçenekler ayarlamak üzere ActionScript yazabilirsiniz. Daha fazla bilgi için, *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* bölümündeki TileList sınıfına bakın.

TileList bileşeniyle uygulama oluşturma

Bu örnek, bir TileList ögesini boya renkleri dizisiyle doldurmak için MovieClip öğelerini kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 TileList bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aTI** örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.data.DataProvider;
import flash.display.DisplayObject;

var aBoxes:Array = new Array();
var i:uint = 0;
var colors:Array = new Array(0x000000, 0xFF0000, 0x0000CC, 0x00CC00, 0xFFFF00);
var colorNames:Array = new Array("Midnight", "Cranberry", "Sky", "Forest", "July");
var dp:DataProvider = new DataProvider();
for(i=0; i < colors.length; i++) {
    aBoxes[i] = new MovieClip();
    drawBox(aBoxes[i], colors[i]); // draw box w next color in array
    dp.addItem( {label:colorNames[i], source:aBoxes[i]} );
}
aTl.dataProvider = dp;
aTl.columnWidth = 110;
aTl.rowHeight = 130;
aTl.setSize(280,150);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(0, 0, 100, 100);
    box.graphics.endFill();
}
```

4 Uygulamayı test etmek için, Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

ActionScript kullanarak TileList bileşeni oluşturma

Bu örnek, dinamik olarak bir TileList örneği oluşturur ve ColorPicker, ComboBox, NumericStepper ve CheckBox bileşenlerinin örneklerini ekler. Görüntülenecek bileşenin etiketlerini ve adlarını içeren bir Array oluşturur ve TileList ögesinin dataProvider özelliğine Array (dp) atar. TileList ögesini düzenlemek için columnWidth ve rowHeight özelliklerini ve setSize() yöntemini, TileList ögesini Sahne Alanı'nda konumlandırmak için move() yöntemini, TileList örneğinin kenarlıklarını ile içerikleri arasına boşluk koymak için contentPadding stilini ve içeriği etiketlerine göre sıralamak için sortItemsOn() yöntemini kullanır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Şu bileşenleri Bileşenler panelinden Kütüphane paneline sürükleyin: ColorPicker, ComboBox, NumericStepper, CheckBox ve TileList.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:


```
import fl.controls.CheckBox;
import fl.controls.ColorPicker;
import fl.controls.ComboBox;
import fl.controls.NumericStepper;
import fl.controls.TileList;
import fl.data.DataProvider;

var aCp:ColorPicker = new ColorPicker();
var aCb:ComboBox = new ComboBox();
var aNs:NumericStepper = new NumericStepper();
var aCh:CheckBox = new CheckBox();
var aTl:TileList = new TileList();

var dp:Array = [
    {label:"ColorPicker", source:aCp},
    {label:"ComboBox", source:aCb},
    {label:"NumericStepper", source:aNs},
    {label:"CheckBox", source:aCh},
];
aTl.dataProvider = new DataProvider(dp);
aTl.columnWidth = 110;
aTl.rowHeight = 100;
aTl.setSize(280,130);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);
aTl.sortItemsOn("label");
addChild(aTl);
```

4 Uygulamayı test etmek için, Test Et > Filmi Kontrol Et seçeneklerini belirleyin.

UILoader bileşenini kullanma

UILoader bileşeni; SWF, JPEG, aşamalı JPEG, PNG ve GIF dosyalarını görüntüleyebilen bir kaptır. Uzak bir konumdan içerik alıp Flash uygulamasına çekmeniz gerektiğinde bir UILoader ögesini kullanabilirsiniz. Örneğin, bir forma şirket logosu (JPEG dosyası) eklemek için UILoader ögesini kullanabilirsiniz. Ayrıca fotoğrafları görüntüleyen bir uygulamada da UILoader bileşenini kullanabilirsiniz. İçerik yüklemek için `load()` yöntemini, ne kadar içerik yüklendiğini belirlemek için `percentLoaded` özelliğini ve yüklemenin bittiği zamanı belirlemek için `complete` olayını kullanın.

UILoader ögesinin içeriklerini ölçekleyebilir veya içeriklerin boyutu sığacak şekilde UILoader ögesini yeniden boyutlandırabilirsiniz. Varsayılan olarak içerikler UILoader ögesine sığacak şekilde ölçeklenir. Ayrıca çalışma zamanında içerik yükleyebilir ve yükleme ilerlemesini izleyebilirsiniz (içerik önbelleğe alındıktan sonra yüklense de, ilerleme çabucak %100 değerine atlar). UILoader ögesine içerik yüklerken bir konum belirtirseniz, konumu (X ve Y koordinatları) 0, 0 olarak belirtmeniz gerekir.

UILoader bileşeniyle kullanıcı etkileşimi

UILoader bileşeni odak alamaz. Ancak UILoader bileşenine yüklenen içerik odağı kabul edebilir ve kendi odak etkileşimlerine sahip olabilir. Odağı kontrol etmeyle ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) ve “[FocusManager ile çalışma](#)” sayfa 26 bölümündeki FocusManager sınıfına bakın.

UILoader bileşeni parametreleri

Özellik denetçisinde veya Bileşen denetçisinde her UILoader bileşeni örneği için şu geliştirme parametrelerini ayarlayabilirsiniz: `autoLoad`, `maintainAspectRatio`, `source` ve `scaleContent`. Bu parametrelerin her biri, aynı anda, karşılık gelen ActionScript özelliğine sahiptir.

Her bir UILoader örneğinin canlı önizlemesi, geliştirme sırasında Özellik denetçisinde veya Bileşen denetçisinde parametrelere yapılan değişiklikleri yansıtır.

UILoader örneklerinin yöntemlerini, özelliklerini ve olaylarını kullanarak UILoader örnekleri için ek seçenekler ayarlamak üzere ActionScript yazabilirsiniz. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki UILoader sınıfına bakın.

UILoader bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl UILoader bileşeni ekleneceğini açıklamaktadır. Bu örnekte, yükleyici logonun GIF görüntüsünü yükler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 UILoader bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Özellik denetçisinde, **aUI** örnek adını girin.
- 4 Sahne Alanı'nda ve Bileşen denetçisinde yükleyiciyi seçin ve `source` parametresi için <http://www.helpexamples.com/images/logo.gif> girin.

ActionScript kullanarak UILoader bileşen örneği oluşturma

Bu örnek, ActionScript kullanarak bir UILoader bileşeni oluşturur ve JPEG biçiminde bir çiçek görüntüsünü yükler. `complete` olayı gerçekleştiğinde, Çıktı panelinde yüklenen bayt sayısını görüntüler.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 UILoader bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import fl.containers.UILoader;

var aLoader:UILoader = new UILoader();
aLoader.source = "http://www.flash-mx.com/images/image1.jpg";
aLoader.scaleContent = false;
addChild(aLoader);

aLoader.addEventListener(Event.COMPLETE, completeHandler);
function completeHandler(event:Event) {
    trace("Number of bytes loaded: " + aLoader.bytesLoaded);
}
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

UIScrollBar bileşenini kullanma

UIScrollBar, bir metin alanına kaydırma çubuğu eklemenize olanak tanır. Geliştirme sırasında veya çalışma zamanında ActionScript ile metin alanına bir kaydırma çubuğu ekleyebilirsiniz. UIScrollBar bileşenini kullanmak için, Sahne Alanı'nda bir metin alanı oluşturun, UIScrollBar bileşenini Bileşenler panelinden metin alanının sınırlama kutusunun herhangi bir çeyreğine sürükleyin.

Kaydırma çubuğunun uzunluğu, kaydırma oklarının birleştirilmiş boyutundan küçük olursa, kaydırma çubuğu düzgün görüntülenmez. Ok düğmelerinden biri diğerinin arkasına gizlenir. Flash uygulaması bununla ilgili hata vermez. Bu durumda kaydırma çubuğunu ActionScript ile gizlemek iyi bir fikirdir. Kaydırma çubuğu, kaydırma kutusu için yeterli boşluk olmayacak şekilde boyutlandırılırsa, Flash kaydırma kutusunu görünmez hale getirir.

UIScrollBar bileşeni, diğer tüm kaydırma çubukları gibi çalışır. Uçta ve kaydırma yolunda ve bunların arasındaki kaydırma kutusunda ok düğmeleri içerir. Metin alanının herhangi bir kenarına hem dikey hem de yatay olarak eklenebilir.

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki UILoader sınıfına bakın.

UIScrollBar bileşeniyle kullanıcı etkileşimi

Diğer pek çok bileşenden farklı olarak UIScrollBar bileşeni, art arda tıklamalara gerek kalmadan örneğin kullanıcının fare düğmesini basılı tutmasıyla sürekli fare girdisi alabilir.

UIScrollBar bileşeniyle klavye etkileşimi olmaz.

UIScrollBar bileşeni parametreleri

Her UIScrollBar bileşeni örneği için Özellik denetçisinde veya Bileşen denetçisinde şu geliştirme parametrelerini ayarlayabilirsiniz: `direction` ve `scrollTargetName`. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir.

UIScrollBar örneklerinin yöntemlerini, özelliklerini ve olaylarını kullanarak UIScrollBar örnekleri için ek seçenekler ayarlamak üzere ActionScript yazabilirsiniz. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki UIScrollBar sınıfına bakın.

UIScrollBar bileşeniyle uygulama oluşturma

Aşağıdaki prosedür, geliştirme sırasında bir uygulamaya nasıl UIScrollBar bileşeni ekleneceğini açıklamaktadır.

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 Bir veya iki metin satırını alacak kadar dinamik bir metin alanı oluşturun ve bu metin alanına Özellik denetçisinde `myText` örnek adını verin.
- 3 Özellik denetçisinde, metin girdi alanının Satır Türü'nü Çok Satırlı olarak ayarlayın veya kaydırma çubuğunu yatay olarak kullanmayı düşünüyorsanız, Çok Satırlı Sarma Yok olarak ayarlayın.
- 4 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve kullanıcının tümünü görüntülemek için kaydırma yapmasına gerek kalmayacak şekilde `text` alanını doldurmak için şu ActionScript kodunu girin:

```
myText.text="When the moon is in the seventh house and Jupiter aligns with Mars, then peace will guide the planet and love will rule the stars."
```

***Not:** Sahne Alanı'nda metin alanının tüm metnin görüntülenmesi için kaydırma yapılmasına gerek kalmayacak kadar küçük olduğundan emin olun. Yeterince küçük değilse, kaydırma çubuğu görüntülenmez veya kaydırma kutusu tutamacı (içeriği kaydırmak için sürüklediğiniz bölüm) olmadan yalnızca iki satır olarak görüntülenebilir.*

5 Nesne yapışmasının etkinleştirilmiş olduğunu (Görünüm > Yapışma > Nesnelere Yapış) doğrulayın.

6 Bir UIScrollBar örneğini Bileşenler panelinden örneği yapıştırmak istediğiniz tarafın yakınındaki metin girdi alanına sürükleyin. Alana düzgün şekilde yapışması için fareyi serbest bıraktığınızda, bileşenin metin alanıyla örtüşmesi gerekir. Bileşene **mySb** örnek adını verin.

Bileşenin `scrollTargetName` özelliği, Özellik ve Bileşen denetçilerinde otomatik olarak metin alanı örnek adıyla doldurulur. Bu, Parametreler sekmesinde görüntülenmezse, UIScrollBar örneğini yeterince örtüşürmemiş olabilirsiniz.

7 Kontrol Et > Filmi Test Et'i seçin.

ActionScript kullanarak UIScrollBar bileşen örneği oluşturma

ActionScript ile bir UIScrollBar oluşturabilir ve bunu çalışma zamanında bir metin alanıyla ilişkilendirebilirsiniz. Şu örnek, yatay olarak yönlendirilmiş bir UIScrollBar örneği oluşturur ve bu örneği, bir URL'deki metinle yüklenen **myTxt** adındaki metin alanı örneğinin en alt kısmına ekler. Bu örnek ayrıca kaydırma çubuğunun boyutunu metin alanının boyutuyla eşleştirecek şekilde ayarlar:

- 1 Yeni bir Flash (ActionScript 3.0) belgesi oluşturun.
- 2 ScrollBar bileşenini Kütüphane paneline sürükleyin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu ActionScript kodunu girin:

```
import flash.net.URLLoader;
import fl.controls.UIScrollBar;
import flash.events.Event;

var myTxt:TextField = new TextField();
myTxt.border = true;
myTxt.width = 200;
myTxt.height = 16;
myTxt.x = 200;
myTxt.y = 150;

var mySb:UIScrollBar = new UIScrollBar();
mySb.direction = "horizontal";
// Size it to match the text field.
mySb.setSize(myTxt.width, myTxt.height);

// Move it immediately below the text field.
mySb.move(myTxt.x, myTxt.height + myTxt.y);

// put them on the Stage
addChild(myTxt);
addChild(mySb);
// load text
var loader:URLLoader = new URLLoader();
var request:URLRequest = new URLRequest("http://www.helpexamples.com/flash/lorem.txt");
loader.load(request);
loader.addEventListener(Event.COMPLETE, loadcomplete);

function loadcomplete(event:Event) {
    // move loaded text to text field
    myTxt.text = loader.data;
    // Set myTxt as target for scroll bar.
    mySb.scrollTarget = myTxt;
}
```

4 Kontrol Et > Filmi Test Et'i seçin.

Bölüm 5: Kullanıcı Arabirimi Bileşenlerini Özelleştirme

Kullanıcı arabirimi bileşeni özelleştirme hakkında

Aşağıdaki öğelerden birini veya her ikisini değiştirerek, uygulamalarınızdaki bileşenlerin görünümünü özelleştirebilirsiniz:

Stiller Her bileşen, bileşenin görünümünü oluşturmak için Flash uygulamasının hangi değerleri kullanacağını belirtmek üzere ayarlayabildiğiniz stiller kümesine sahiptir. Stiller genellikle bir bileşen için farklı durumlarda kullanılacak kaplamalar ve simgelerin yanı sıra metin biçimlendirmesi ve dolgusu için hangi değerlerin kullanılacağını belirtir.

Kaplamalar *Kaplama*, belirli bir durumda bileşenin grafiksel görünümünü oluşturan semboller koleksiyonunu içerir. Stil hangi kaplamanın kullanılacağını belirtirken, kaplama ise Flash uygulamasının bileşeni çizmek için kullandığı grafiksel bir öğedir. *Kaplama oluşturma*, bileşenin grafiklerini değiştirerek bileşenin görünümünü değiştirme işlemidir.

Not: *ActionScript 3.0 bileşenlerinin varsayılan görünümü bir tema (Aeon Halo) olarak değerlendirilebilir ancak bu kaplamalar bileşenlerde yerleşik olarak bulunur. ActionScript 3.0 bileşenleri, ActionScript 2.0 bileşenlerinin desteklediği harici tema dosyalarını desteklemez.*

Stilleri ayarlama

Bileşenin stilleri genellikle, Flash söz konusu bileşeni çeşitli durumlarda çizdiğinde o bileşenin kaplamalarının, simgelerinin, metin biçimlendirmesinin ve dolgusunun değerlerini belirtir. Örneğin, bir düğmenin üzerinde fare düğmesini tıklattığınızda Flash basılmış durumunu göstermek için Button öğesini basılmamış ya da normal durumunda gösterilenden farklı bir kaplamayla çizer. Ayrıca `enabled` özelliğinin `false` olarak ayarlanmış olması sonucunda devre dışı durumundayken de farklı bir kaplama kullanır.

Belge, sınıf ve örnek düzeylerindeki bileşenler için stiller ayarlayabilirsiniz. Ayrıca üst bileşenden de bazı stil özellikleri miras alınabilir. Örneğin, List bileşeni, BaseScrollPane öğesini miras alarak ScrollBar stillerini miras alır.

Aşağıdaki yolları kullanarak bir bileşeni özelleştirmek için stilleri ayarlayabilirsiniz:

- Bir bileşen örneğinde stilleri ayarlayın. Tek bir bileşen örneği için rengi ve metin özelliklerini değiştirebilirsiniz. Bu yöntem bazı durumlar için etkili olmasına rağmen bir belgedeki tüm bileşenlerde ayrı ayrı özellikler ayarlamanız gerektiğinde zaman alıcı olabilir.
- Bir belgede belirli bir türün tüm bileşenleri için stiller ayarlayın. Bir belgedeki belirli bir türün tüm bileşenlerine (örneğin, tüm CheckBox veya tüm Button öğelerine) tutarlı bir görünüm uygulamak istiyorsanız, bileşen düzeyinde stiller ayarlayabilirsiniz.

Kaplarda ayarlanan stil özelliklerinin değerleri, içerilen bileşenler tarafından miras alınır.

Siz Canlı Önizleme özelliğini kullanarak Sahne Alanındaki bileşenleri görüntülerken, stil özelliklerine yapılan değişiklikleri Flash görüntülemez.

Stil ayarlarını anlama

Aşağıda, stil kullanımıyla ilgili birkaç önemli noktaya yer verilmiştir:

Miras Bileşen alt ögesi, tasarım sırasında üst bileşenden bir stili miras alacak şekilde ayarlanır. ActionScript içindeki stiller için miras almayı ayarlayamazsınız

Öncelik Bir bileşen stili birden çok şekilde ayarlanırsa, Flash öncelik sırasına göre karşılaştığı ilk stili kullanır. Flash, bir değer bulununcaya kadar aşağıdaki sıraya göre stilleri arar:

- 1 Flash, bileşen örneğinde stil özelliğini arar.
- 2 Stil, miras alan stillerden biriye, Flash miras alınan bir değer üst öge hiyerarşisine bakar.
- 3 Flash, bileşende stili arar.
- 4 Flash, StyleManager'da genel bir ayar arar.
- 5 Özellik halen tanımlanmamışsa, özelliğin değeri undefined olur.

Bileşenin varsayılan stillerini erişme

Bileşen sınıfı için statik `getStyleDefinition()` yöntemini kullanarak bir bileşenin varsayılan stillerine erişebilirsiniz. Örneğin şu kod, ComboBox bileşeninin varsayılan stillerini alır ve `buttonWidth` ve `downArrowDownSkin` özelliklerinin varsayılan değerlerini görüntüler:

```
import fl.controls.ComboBox;
var styleObj:Object = ComboBox.getStyleDefinition();
trace(styleObj.buttonWidth); // 24
trace(styleObj.downArrowDownSkin); // ScrollArrowDown_downSkin
```

Bileşen örneğinde stilleri ayarlama ve alma

Herhangi bir kullanıcı arabirimi bileşeni örneği, bir stili ayarlamak veya almak için doğrudan `setStyle()` ve `getStyle()` yöntemlerini çağırabilir. Şu sözdizimi, bileşen örneği için bir stil ve değer ayarlar:

```
instanceName.setStyle("styleName", value);
```

Bu sözdizimi, bileşen örneği için bir stil alır:

```
var a_style:Object = new Object();
a_style = instanceName.getStyle("styleName");
```

Farklı veri türlerine sahip birden çok stil döndürebildiğinden `getStyle()` yönteminin `Object` türünü döndürdüğüne dikkat edin. Örneğin, şu kod, TextArea örneği (`aTa`) için font stilini ayarlar ve sonra `getStyle()` yöntemini kullanarak bu font stilini alır. Bu örnek, döndürülen değeri bir `TextFormat` değişkenine atamak için `TextFormat` nesnesine çevirir. Çevrim olmadan, derleyici `Object` değişkenini `TextFormat` değişkenine işleme girişimine yönelik bir hata yayınlar.

```
import flash.text.TextFormat;

var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

Metin özelliklerini ayarlamak için TextFormat ögesini kullanma

Bir bileşen örneğinin metnini biçimlendirmek için TextFormat nesnesini kullanın. TextFormat nesnesi, bold, bullet, color, font, italic, size, vb. gibi metin özelliklerini belirtmenizi sağlayan özellikler içerir. TextFormat nesnesinde bu özellikleri ayarlayabilir ve sonra bunları bir bileşen örneğine uygulamak üzere setStyle() yöntemini çağırabilirsiniz. Örneğin, şu kod bir TextFormat nesnesinin font, size ve bold özelliklerini ayarlar ve bu özellikleri bir Button örneğine uygular:

```
/* Create a new TextFormat object to set text formatting properties. */  
var tf:TextFormat = new TextFormat();  
tf.font = "Arial";  
tf.size = 16;  
tf.bold = true;  
a_button.setStyle("textFormat", tf);
```

Aşağıda, bu ayarların, Submit etiketine sahip bir düğme üzerindeki etkisi gösterilmektedir:

Submit

Bir bileşen örneğinde setStyle() aracılığıyla ayarlanan stil özellikleri en yüksek önceliğe sahip olup diğer tüm stil ayarlarını geçersiz kılar. Ancak, tek bir bileşen örneğinde setStyle() kullanarak ne kadar çok özellik ayarlarsanız, söz konusu bileşen çalışma zamanında oluşturma işlemini o kadar yavaş yapar.

Bir bileşenin tüm örnekleri için stil ayarlama

StyleManager sınıfının statik setComponentStyle() yöntemini kullanarak bir bileşen sınıfının tüm örnekleri için stil ayarlayabilirsiniz. Örneğin, önce bir Button ögesini Sahne Alanı'na sürükleyip sonra Zaman Çizelgesi'nin Kare 1'indeki Eylemler paneline şu ActionScript kodunu ekleyerek tüm Button öğeleri için metnin rengini kırmızı olarak ayarlayabilirsiniz:

```
import fl.managers.StyleManager;  
import fl.controls.Button;  
  
var tf:TextFormat = new TextFormat();  
tf.color = 0xFF0000;  
StyleManager.setComponentStyle(Button, "textFormat", tf);
```

Sonradan Sahne Alanı'na eklediğiniz tüm Button öğelerinde kırmızı etiketler olur.

Tüm bileşenler için bir stil ayarlama

StyleManager sınıfının statik setStyle() yöntemini kullanarak tüm bileşenler için bir stil ayarlayabilirsiniz.

- 1 Bir List bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aList** örnek adını verin.
- 2 Bir Button bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aButton** örnek adını verin.
- 3 Önceden açılmadıysa, Eylemler panelini açmak için **F9** tuşuna basın veya Pencere menüsünden Eylemler seçeneğini belirleyin ve tüm bileşenler için metnin rengini kırmızı olarak ayarlamak üzere Zaman Çizelgesi'nin Kare 1'ine şu kodu girin:

```
import fl.managers.StyleManager;  
  
var tf:TextFormat = new TextFormat();  
tf.color = 0xFF0000;  
StyleManager.setStyle("textFormat", tf);
```

- 4 Listeyi metinle doldurmak için Eylemler paneline şu kodu ekleyin.


```
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});  
aList.addItem({label:"1966 Mustang (Classic)", data:27000});  
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});  
aList.allowMultipleSelection = true;
```

- 5 Kodu derleyip içeriğinizi test etmek için Kontrol Et > Filmi Test Et seçeneklerini belirleyin veya Ctrl+Enter tuşlarına basın. Hem düğme etiketindeki hem de listedeki metnin kırmızı olması gerekir.

Kaplamalar Hakkında

Bir bileşenin görünümü, anahat, dolgu rengi, simgeler ve diğer bileşenler gibi grafiksel öğelerden oluşur. Örneğin, ComboBox bir List bileşeni içerir ve List bileşeni de bir ScrollBar bileşeni içerir. Grafiksel öğeler birlikte ComboBox'ın görünümünü oluşturur. Ancak bileşenin görünümü, geçerli durumuna göre değişir. Örneğin, etiketi olmadan bir CheckBox uygulamanızda görüntülenirse, şöyle görünür:



CheckBox normal basılmamış durumunda

CheckBox'in üzerinde fare düğmesini tıklatıp basılı tutarsanız, CheckBox'in görünümü şöyle değişir:



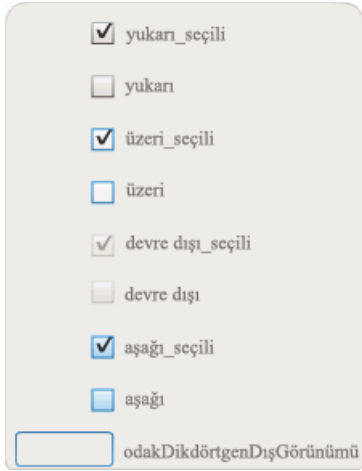
CheckBox basılmış durumunda

Fare düğmesini serbest bıraktığınızda, CheckBox orijinal görünümüne geri döner, ancak bu defa seçilmiş olduğunu gösteren bir onay işaretiyle birlikte görüntülenir.



CheckBox seçili durumunda

Bileşeni çeşitli durumlarında temsil eden simgelerin hepsine birden topluca bileşenin *kaplamaları* denir. Tıpkı diğer Flash sembollerinde olduğu gibi, Flash uygulamasında kaplamalarını düzenleyerek bileşenlerin herhangi bir durumunda veya tüm durumlarında görünümünü değiştirebilirsiniz. Bileşenin kaplamalarına iki şekilde erişebilirsiniz. Bunun en kolay yolu, bileşeni Sahne Alanı'na sürükleyip çift tıklatmaktır. Bu işlem sonucunda CheckBox için şöyle görünen bir bileşen kaplamalarının paleti açılır.



CheckBox'ın kaplamaları

Bir bileşenin kaplamalarına ayrı ayrı Kütüphane panelinden de erişebilirsiniz. Bir bileşeni Sahne Alanı'na sürüklediğinizde, bunu aynı zamanda, içindeki varlıkların bulunduğu bir klasör ve içerdiği diğer tüm bileşenlerle birlikte kütüphaneye kopyalamış olursunuz. Örneğin, Sahne Alanı'na bir ComboBox sürüklerseniz, Kütüphane paneli ComboBox'ta yerleşik olan List, ScrollBar ve TextInput bileşenlerini, bu bileşenlerin her birinin kaplamasını içeren bir klasörü ve bu bileşenlerin paylaştığı öğeleri içeren bir Paylaşılan varlıklar klasörünü de içerir. Kaplamalar klasörünü (ComboBoxSkins, ListSkins, ScrollBarSkins veya TextInputSkins) açıp düzenlemek istediğiniz kaplamanın simgesini çift tıklatarak bu bileşenlerden herhangi birinin kaplamalarını düzenleyebilirsiniz. Örneğin, ComboBox_downSkin öğesi çift tıklatıldığında, aşağıda gösterildiği gibi kaplama sembol düzenleme modunda açılır:



ComboBox_downSkin

Yeni bir kaplama oluşturma

Belgenizdeki bir bileşen için yeni bir görünüm oluşturmak istiyorsanız, bileşenin kaplamalarını düzenleyerek görünümü değiştirin. Bileşenin kaplamalarına erişmek üzere kaplamaların paletini açmak için Sahne Alanı üzerinde bileşeni çift tıklatın. Ardından düzenlemek istediğiniz kaplamayı sembol düzenleme modunda açmak için çift tıklatın. Örneğin, TextArea bileşeninin varlıklarını sembol düzenleme modunda açmak için Sahne Alanı üzerinde TextArea bileşenini çift tıklatın. Yakınlaştırma denetimini %400 olarak ayarlayın ve görünümünü değiştirmek için sembolü düzenleyin. İşleminiz bitince, değişiklik belgedeki bileşenlerin tüm örneklerini etkiler. Alternatif olarak, Kütüphane panelindeki belirli bir kaplamayı Sahne Alanı'nda sembol düzenleme modunda açmak için çift tıklatabilirsiniz.

Bileşen kaplamalarını şu yöntemlerle değiştirebilirsiniz:

- Tüm örnekler için yeni bir kaplama oluşturma
- Bazı örnekler için yeni kaplamalar oluşturma

Tüm örnekler için bir kaplama oluşturma

Bir bileşenin kaplamasını düzenlediğinizde, varsayılan olarak bileşenin belgedeki tüm örnekleri için de görünümü değiştirirsiniz. Aynı bileşen için farklı görünümler oluşturmak isterseniz, değiştirmek istediğiniz kaplamaları çoğaltıp bunlara farklı adlar vermeniz, düzenlemeniz ve sonra uygun stilleri ayarlayıp uygulamanız gerekir. Daha fazla bilgi için bkz. “Bazı örnekler için kaplamalar oluşturma” sayfa 103.

Bu bölümde, kullanıcı arabirimi bileşenlerinin her biri için bir veya birkaç kaplamanın nasıl değiştirileceği açıklanmaktadır. Kullanıcı arabirimi bileşeninin kaplamalarından bir veya birkaçını değiştirmek için bu prosedürlerden birini izlerseniz, belgedeki tüm örnekler için kaplamayı değiştirirsiniz.

Bazı örnekler için kaplamalar oluşturma

Aşağıdaki genel prosedürü kullanarak bir bileşenin bazı örnekleri için kaplama oluşturabilirsiniz:

- Kütüphane panelinde bileşenin Varlıklar klasöründe kaplamayı seçin.
- Kaplamayı çoğaltın ve kaplamaya benzersiz bir sınıf adı atayın.
- Kaplamaya istediğiniz görünümü vermek için düzenleyin.
- Kaplama stiline yeni kaplama atamak üzere bileşen örneği için `setStyle()` yöntemini çağırın.

Aşağıdaki prosedür, iki Button örneğinden biri için yeni bir `selectedDownSkin` oluşturur.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 Bileşenler panelinden iki Button öğesini Sahne Alanı'na sürükleyin ve bunlara **aButton** ve **bButton** örnek adlarını verin.
- 3 Kütüphane panelini ve sonra panelin içindeki Bileşen Varlıkları ve ButtonSkins klasörlerini açın.
- 4 `selectedDownSkin` kaplamasını seçmek için tıklatın.
- 5 Bağlam menüsünü açmak için sağ tıklatın ve Çoğalt öğesini seçin.
- 6 Sembolü Çoğalt iletişim kutusunda, yeni kaplamaya benzersiz bir ad (örneğin, **Button_mySelectedDownSkin**) verin. Sonra Tamam'ı tıklatın.
- 7 Kütüphane > Bileşen Varlıkları > ButtonSkins klasöründe, `Button_mySelectedDownSkin` öğesini seçin ve bağlam menüsünü açmak için sağ tıklatın. Bağlantı Özellikleri iletişim kutusunu açmak için Bağlantı öğesini seçin.
- 8 ActionScript için Dışa Aktar onay kutusunu tıklatın. İlk Karede Dışa Aktar onay kutusunu seçili bırakın ve sınıf adının benzersiz olduğundan emin olun. Tamam'ı tıklatın ve sonra sınıf tanımının bulunamadığını ve oluşturulacağını belirten uyarıya yanıt olarak tekrar Tamam'ı tıklatın.
- 9 Kütüphane panelindeki `Button_mySelectedDownSkin` kaplamasını sembol düzenleme modundan açmak için çift tıklatın.
- 10 Özellik denetçisindeki Dolgu rengi seçicisinde renk görüntüleninceye kadar kaplamanın ortasında mavi dolguyu tıklatın. Renk seçiciyi tıklatın ve kaplama dolgusu için #00CC00 rengini seçin.
- 11 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 12 Özellik denetçisinde, her düğmenin Parametreler sekmesini tıklatın ve `toggle` parametresini `true` değerine ayarlayın.
- 13 Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);  
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```
- 14 Kontrol Et > Filmi Test Et'i seçin.

15 Her düğmeyi tıklatın. bButton nesnesinin aşağı kaplamasının (seçili ve seçimi kaldırılmış) yeni kaplama sembolünü kullandığını unutmayın.

Button bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında Button bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında setSize() yöntemini veya height ve width ile scaleX ve scaleY gibi Button sınıfının ilgili özelliklerini kullanın.

Düğme yeniden boyutlandırıldığında, simgenin veya etiketin boyutu değişmez. Button öğesinin sınırlama kutusu, Button öğesinin sınırına karşılık gelir ve örneğin vuruş alanını atar. Örneğin boyutunu artırırsanız, vuruş alanının boyutunu da artırırsınız. Sınırlama kutusu etiket için çok küçükse, etiket sınırlama kutusuna sığacak şekilde kırılır.

Button öğesinde bir simge varsa ve bu simge Button öğesinden büyükse, simge Button öğesinin sınırlarının dışına uzanır.

Button bileşeniyle stilleri kullanma

Button öğesinin stilleri genellikle bileşen çeşitli durumlarda çizildiğinde Button kaplamalarının, simgelerinin, metin biçimlendirmesinin ve dolgusunun değerlerini belirtir.

Aşağıdaki prosedür Sahne Alanı'na iki Button öğesini koyar ve kullanıcı bu Button öğelerinden birini tıklattığında her iki Button öğesi için emphasized özelliğini true değerine ayarlar. Ayrıca kullanıcı ikinci Button öğesini tıklattığında, ikinci Button öğesinin emphasizedSkin stilini de selectedOverSkin stiline ayarlar, böylece iki Button öğesi aynı durum için farklı kaplamalar gösterir.

- 1 Bir Flash dosyası (ActionScript 3.0) oluşturun.
- 2 İki Button öğesini teker teker Sahne Alanı'na sürükleyin ve bunlara aBtn ve bBtn örnek adlarını verin. Özellik denetçisinin Parametreler sekmesinde, bunlara Button A ve Button B etiketlerini verin.
- 3 Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
bBtn.emphasized = true;
aBtn.emphasized = true;
bBtn.addEventListener(MouseEvent.CLICK, Btn_handler);
function Btn_handler(evt:MouseEvent):void {
    bBtn.setStyle("emphasizedSkin", "Button_selectedOverSkin");
}
```

- 4 Kontrol Et > Filmi Test Et'i seçin.
- 5 Düğmelerin her birinde emphasizedSkin stilinin etkisini görmek için düğmelerden birini tıklatın.

Button bileşeniyle kaplamaları kullanma

Button bileşeni, farklı durumlarına karşılık gelen şu kaplamaları kullanır. Button öğesinin görünümünü değiştirmek üzere bir veya birkaç kaplama düzenlemek için, Sahne Alanı'nda Button örneğini çift tıklatarak aşağıda gösterildiği gibi bu örneğin kaplamalarının bir paletini açın:

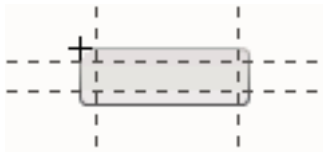


Button kaplamaları

Bir düğme etkinleştirilirse, düğmenin üzerine işaretçi geldiğinde düğme üzerinde durumunda görüntülenir. Düğme girdi odağını alır ve düğmenin üzerine basıldığında düğme basılmış durumda görüntülenir. Fare serbest bırakıldığında, düğme basılmamış durumuna geri döner. Fare basılı durumdayken işaretçi düğmeden ayrılırsa, düğme orijinal durumuna geri döner. Toggle parametresi `true` olarak ayarlanırsa, `selectedDownSkin` ile basılı durumu gösterilir, `selectedUpSkin` ile basılmamış durumu gösterilir ve `selectedOverSkin` ile üzerinde durumu gösterilir.

Button öğesi devre dışı bırakılırsa, kullanıcı müdahalesine bakılmaksızın Button öğesi devre dışı durumunda görüntülenir.

Kaplamalardan birini düzenlemek için, aşağıda gösterildiği gibi kaplamayı çift tıklararak sembol düzenleme modunda açın:



Sembol düzenleme modunda Button

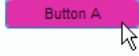
Bu noktada, kaplamayı istediğiniz şekilde düzenlemek için Flash geliştirme araçlarını kullanabilirsiniz.

Aşağıdaki prosedür, Button öğesinin `selected_over` kaplamasının rengini değiştirir.

- 1 Yeni Flash dosyası (ActionScript 3.0) oluşturun.
- 2 Bileşenler panelinden Sahne Alanı'na bir Button öğesini sürükleyin. Parametreler sekmesinde toggle parametresini `true` değerine ayarlayın.
- 3 Kaplamalarının paletini açmak için Button öğesini çift tıklayın.
- 4 `selected_over` kaplamasını sembol düzenleme modunda açmak için çift tıklayın.
- 5 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %400 değerine ayarlayın.
- 6 Özellik denetçisindeki Dolgu rengi seçicisinde rengi görüntüleninceye kadar arka planı çift tıklayın.
- 7 `selected_over` kaplamasının arka planına uygulamak için Dolgu rengi seçicisinde `#CC0099` rengini seçin.

- 8 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 9 Kontrol Et > Filmi Test Et'i seçin.
- 10 Seçili duruma koymak için düğmeyi tıklatın.

Fare işaretçisini Button öğesinin üzerine getirdiğinizde, selected_over durumunun aşağıda gösterildiği gibi görüntülenmesi gerekir.



Değiştirilmiş renkli selected_over kaplamasını gösteren Button öğesi

CheckBox bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında CheckBox bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında setSize() yöntemini veya CheckBox sınıfının ilgili özelliklerini kullanın. Örneğin, height ve width ve scaleX ve scaleY özelliklerini değiştirerek CheckBox öğesinin boyutunu değiştirebilirsiniz. CheckBox öğesi yeniden boyutlandırıldığında, etiketin boyutu veya onay kutusu simgesi değişmez; yalnızca sınırlama kutusunun boyutu değişir.

CheckBox örneğinin sınırlama kutusu görünür hale gelir ve örneğin vuruş alanını atar. Örneğin boyutunu artırırsanız, vuruş alanının boyutunu da artırırsınız. Sınırlama kutusu etiket için çok küçükse, etiket sınırlama kutusuna sığacak şekilde kırpılır.

CheckBox ile stilleri kullanma

CheckBox örneğinin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. Örneğin, aşağıdaki prosedür, CheckBox etiketinin boyutunu ve rengini değiştirir.

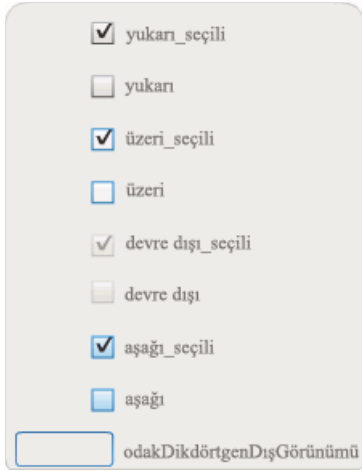
- 1 CheckBox bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene myCb örnek adını verin.
- 2 Özellik denetçisinde Parametreler sekmesini tıklatın ve etiket parametresi için şu değeri girin: **500\$'dan az?**
- 3 Ana Zaman Çizelgesi'nin Kare 1'inde, Eylemler Paneli'ne şu kodu girin:

```
var myTf:TextFormat = new TextFormat();  
myCb.setSize(150, 22);  
myTf.size = 16;  
myTf.color = 0xFF0000;  
myCb.setStyle("textFormat", myTf);
```

Daha fazla bilgi için bkz. “[Stilleri ayarlama](#)” sayfa 98. Bileşenin simgelerini ve kaplamalarını değiştirmek üzere stil özelliklerini ayarlama hakkında bilgi almak için bkz. “[Yeni bir kaplama oluşturma](#)” sayfa 102 ve “[CheckBox ile kaplamaları kullanma](#)” sayfa 106.

CheckBox ile kaplamaları kullanma

CheckBox bileşeni, görünümünü değiştirmek için düzenleyebileceğiniz şu kaplamaları içerir.



CheckBox kaplamaları

Bu örnek, `up` ve `selectedUp` durumlarında bileşenin anahat rengini ve arka plan rengini değiştirir. Diğer durumlara yönelik kaplamaları değiştirmek için de benzer adımlar izlenir.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 CheckBox bileşenini Sahne Alanı'na sürükleyin, böylece bileşen varlıklarının yer aldığı bir klasörle birlikte kütüphaneye yerleştirilir.
- 3 Kaplama simgelerinin panelini açmak için Sahne Alanı'nda CheckBox bileşenini çift tıklayın.
- 4 `selected_up` simgesini sembol düzenleme modunda açmak için çift tıklayın.
- 5 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %800 değerine ayarlayın.
- 6 Seçmek için CheckBox öğesinin kenarlığını tıklayın. #0033FF rengini seçip kenarlığa uygulamak için Özellik denetçisinde Dolgu rengi seçicisini kullanın.
- 7 CheckBox öğesinin arka planını çift tıklayıp seçin ve arka plan rengini #00CCFF olarak ayarlamak için tekrar Dolgu rengi seçicisini kullanın.
- 8 CheckBox up kaplaması için 4 ile 8 arasındaki adımları yineleyin.
- 9 Kontrol Et > Filmi Test Et'i seçin.

ColorPicker bileşenini özelleştirme

ColorPicker öğesini yalnızca stilleri üzerinden yeniden boyutlandırabilirsiniz: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` ve `textFieldHeight`. `setSize()` yöntemini veya `width`, `height`, `scaleX` ya da `scaleY` özelliklerini kullanarak Dönüştürme aracı veya ActionScript ile ColorPicker öğesinin boyutunu değiştirmeyi denerseniz, siz SWF dosyası oluştururken bu değerler yoksayılr ve ColorPicker varsayılan boyutunda görüntülenir. Palet arka planı, `columnCount` stili için `setStyle()` kullanılarak ayarlanan sütunların sayısı ile eşleşmesi için yeniden boyutlandırılır. Varsayılan sütun sayısı 18'dir. Özel renkleri 1024 olarak ayarlayabilirsiniz, böylece palet, renk örneklerinin sayısı ile eşleşmek için dikey olarak yeniden boyutlandırılır.

ColorPicker bileşeniyle Stilleri kullanma

ColorPicker bileşeninin görünümünü değiştirmek için birçok stili ayarlayabilirsiniz. Örneğin, aşağıdaki prosedür, ColorPicker'daki sütun sayısını (columnCount) 12 olarak değiştirir, renk örneklerinin yüksekliğini (swatchHeight) ve genişliğini (swatchWidth) değiştirir ve hem metin alanının textPadding) hem de arka planın (backgroundPadding) dolgusunu değiştirir.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 ColorPicker bileşenini Sahne Alanı'na sürükleyin ve bu bileşene aCp örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
aCp.setStyle("columnCount", 12);  
aCp.setStyle("swatchWidth", 8);  
aCp.setStyle("swatchHeight", 12);  
aCp.setStyle("swatchPadding", 2);  
aCp.setStyle("backgroundPadding", 3);  
aCp.setStyle("textPadding", 7);
```

- 4 Kontrol Et > Filmi Test Et'i seçin.
- 5 ColorPicker öğesini tıklatarak açın ve bu ayarların ColorPicker öğesinin görünümünü nasıl değiştirdiğini görün.

ColorPicker bileşeniyle Kaplamaları kullanma

ColorPicker bileşeni, görsel durumlarını temsil etmesi için şu kaplamaları kullanır.



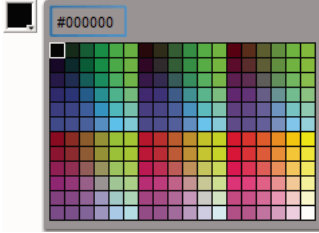
ColorPicker kaplamaları

Palet arka planının rengini değiştirmek için, Arka Plan kaplamasının rengini değiştirebilirsiniz.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 ColorPicker bileşenini Sahne Alanı'na sürükleyin.
- 3 Kaplamalar paletini açmak için üzerini çift tıklatın.
- 4 Arka Plan kaplamasını seçilinceye kadar çift tıklatın, böylece Özellik denetçisinde Dolgu rengi seçici görüntülenir.
- 5 Arka Plan kaplamasına uygulamak için Dolgu rengi seçicisini kullanarak #999999 rengini seçin.
- 6 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.

7 Kontrol Et > Filmi Test Et'i seçin.

ColorPicker ögesini tıklattığınızda, paletin arka planının aşağıda gösterildiği gibi gri olması gerekir.



Koyu gri Arka Plan kaplaması ile ColorPicker

ComboBox bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında ComboBox bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini veya `height` ve `width` ve `scaleX` ve `scaleY` gibi ComboBox sınıfının ilgili özelliklerini kullanın.

ComboBox ögesi, belirtilen genişlik ve yüksekliğe sığacak şekilde yeniden boyutlandırılır. `dropdownWidth` özelliği ayarlanmadığı sürece, bileşenin genişliğine sığacak şekilde liste yeniden boyutlandırılır.

Metin, ComboBox ögesine sığamayacak kadar uzunsa, buraya sığacak şekilde kırpılır. Metni sığdırmak için, ComboBox ögesini yeniden boyutlandırmanız ve `dropdownWidth` özelliğini ayarlamanız gerekir.

ComboBox bileşeniyle Stilleri kullanma

ComboBox bileşeninin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. Stilleri, bileşenin kaplamaları, hücre oluşturucusu, dolgusu ve düğme genişliğinin değerlerini belirtir. Şu örnek, `buttonWidth` ve `textPadding` stillerini ayarlar. `buttonWidth` stili, düğmenin vuruş alanının genişliğini ayarlar ve ComboBox düzenlenebilir durumda olduğunda etkin olur, böylece açılır listeyi açmak için yalnızca düğmeye basabilirsiniz. `textPadding` stili ise, metin alanının dış kenarlığı ile metin arasındaki boşluk miktarını belirtir. ComboBox ögesini daha uzun hale getirirseniz, metin alanında metni dikey olarak ortalamak için bu kullanışlıdır. Aksi takdirde metin, metin alanının en üst kısmında görüntülenebilir.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 ComboBox bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aCb** örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
import fl.data.DataProvider;

aCb.setSize(150, 35);
aCb.setStyle("textPadding", 10);
aCb.setStyle("buttonWidth", 10);
aCb.editable = true;

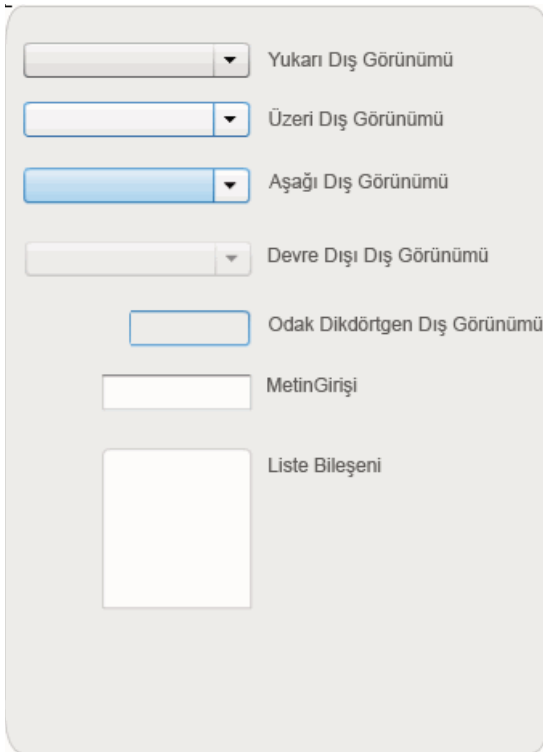
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera Street, Suite 311"}
];
aCb.dataProvider = new DataProvider(items);
```

4 Kontrol Et > Filmi Test Et'i seçin.

Açılır listeyi açmak için tıklatabileceğiniz düğme alanının yalnızca sağ taraftaki dar alan olduğuna dikkat edin. Ayrıca metin alanında metnin dikey şekilde ortalandığına da dikkat edin. Etkilerini görmek için iki `setStyle()` ifadesi olmadan örneği çalıştırmayı deneyebilirsiniz.

ComboBox ile Kaplamaları kullanma

ComboBox öğesi, görsel durumlarını temsil etmesi için şu kaplamaları kullanır:



ComboBox kaplamaları

Sahne Alanı üzerinde devre dışı durumundayken bileşenin rengini değiştirmek için Up kaplamasının rengini değiştirebilirsiniz.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 ComboBox bileşenini Sahne Alanı'na sürükleyin.
- 3 Kaplamalar paletini açmak için üzerini çift tıklatın.
- 4 Up kaplamasını seçilinceye kadar çift tıklatın ve düzenlemek için açın.
- 5 Yakınlaştırma denetimini %400 değerine ayarlayın.
- 6 Özellik denetçisindeki Dolgu rengi seçicisinde kaplamanın rengi görüntüleninceye kadar kaplamanın orta kısmını tıklatın.
- 7 Up kaplamasına uygulamak için Dolgu rengi seçicisini kullanarak #33FF99 rengini seçin.
- 8 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 9 Kontrol Et > Filmi Test Et'i seçin.

Aşağıda gösterildiği gibi, ComboBox öğesinin Sahne Alanı üzerinde görüntülenmesi gerekir.



Arka Plan kaplaması için özelleştirilmiş renkli ComboBox

DataGrid bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında DataGrid bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında, `setSize()` yöntemini veya `width`, `height`, `scaleX` ve `scaleY` gibi ilgili özellikleri kullanın. Herhangi bir yatay kaydırma çubuğu yoksa, sütun genişlikleri orantılı olarak ayarlanır. Sütun (ve dolayısıyla hücre) boyutu ayarlanırsa, hücrelerdeki metinler kırılabilir.

DataGrid bileşeniyle stilleri kullanma

DataGrid bileşenin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. DataGrid bileşeni, List bileşeninden stilleri miras alır. (Bkz. "[List bileşeniyle stilleri kullanma](#)" sayfa 117.)

Tek bir sütun için stiller ayarlama

DataGrid nesnesinde birden çok sütun bulunabilir ve siz de her sütun için farklı hücre oluşturucuları belirtebilirsiniz. DataGrid öğesinin her sütunu bir `DataGridColumn` nesnesiyle temsil edilir ve `DataGridColumn` sınıfı, sütun için `CellRenderer` öğesini tanımlayabileceğiniz bir `cellRenderer` özelliğini içerir.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 DataGrid bileşenini Kütüphane paneline sürükleyin.
- 3 Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin. Bu kod, üçüncü sütunda uzun bir metin dizesiyle birlikte bir DataGrid öğesi oluşturur. Sonunda sütunun `cellRenderer` özelliğini, birden çok satır hücresi oluşturan bir hücre oluşturucusunun adına ayarlar.

```
/* This is a simple cell renderer example. It invokes
the MultiLineCell cell renderer to display a multiple
line text field in one of a DataGrid's columns. */

import fl.controls.DataGrid;
import fl.controls.dataGridClasses.DataGridColumn;
import fl.data.DataProvider;
import fl.controls.ScrollPolicy;

// Create a new DataGrid component instance.
var aDg:DataGrid = new DataGrid();

var aLongString:String = "An example of a cell renderer class that displays a multiple line
TextField"
var myDP:Array = new Array();
myDP = [{firstName:"Winston", lastName:"Elstad", note:aLongString, item:100},
        {firstName:"Ric", lastName:"Dietrich", note:aLongString, item:101},
        {firstName:"Ewing", lastName:"Canepa", note:aLongString, item:102},
        {firstName:"Kevin", lastName:"Wade", note:aLongString, item:103},
        {firstName:"Kimberly", lastName:"Dietrich", note:aLongString, item:104},
        {firstName:"AJ", lastName:"Bilow", note:aLongString, item:105},
        {firstName:"Chuck", lastName:"Yushan", note:aLongString, item:106},
        {firstName:"John", lastName:"Roo", note:aLongString, item:107},
];

// Assign the data provider to the DataGrid to populate it.
// Note: This has to be done before applying the cellRenderers.
aDg.dataProvider = new DataProvider(myDP);

/* Set some basic grid properties.
Note: The data grid's row height should reflect
the number of lines you expect to show in the multiline cell.
The cell renderer will size to the row height.
About 40 for 2 lines or 60 for 3 lines.*/

aDg.columns = ["firstName", "lastName", "note", "item"];
aDg.setSize(430,190);
aDg.move(40,40);
aDg.rowHeight = 40;// Allows for 2 lines of text at default text size.
aDg.columns[0].width = 70;
aDg.columns[1].width = 70;
aDg.columns[2].width = 230;
aDg.columns[3].width = 60;
aDg.resizableColumns = true;
aDg.verticalScrollPolicy = ScrollPolicy.AUTO;
addChild(aDg);
// Assign cellRenderers.
var col3:DataGridColumn = new DataGridColumn();
col3 = aDg.getColumnAt(2);
col3.cellRenderer = MultiLineCell;
```

- 4 FLA dosyasını MultiLineGrid.fla olarak kaydedin.
- 5 Yeni bir ActionScript dosyası oluşturun.
- 6 Şu ActionScript kodunu Komut Dosyası penceresine kopyalayın:

```
package {  
  
    import fl.controls.listClasses.CellRenderer;  
  
    public class MultiLineCell extends CellRenderer  
    {  
  
        public function MultiLineCell()  
        {  
            textField.wordWrap = true;  
            textField.autoSize = "left";  
        }  
        override protected function drawLayout():void {  
            textField.width = this.width;  
            super.drawLayout();  
        }  
    }  
}
```

7 ActionScript dosyasını, MultiLineGrid.fla dosyasını kaydettiğiniz aynı klasöre MultiLineCell.as olarak kaydedin.

8 MultiLineGrid.fla uygulamasına geri dönün ve Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

DataGrid öğesinin şöyle görünmesi gerekir:

firstName	lastName	note	item
Winston	Elstad	An example of a cell renderer class that displays a multiple line TextField	100
Ric	Dietrich	An example of a cell renderer class that displays a multiple line TextField	101
Ewing	Canepa	An example of a cell renderer class that displays a multiple line TextField	102
Kevin	Wade	An example of a cell renderer class that displays a multiple line TextField	103

MultiLineGrid.fla uygulaması için DataGrid

Üstbilgi stillerini ayarlama

headerTextFormat stilini kullanarak üstbilgi satırının metin stilini ayarlayabilirsiniz. Şu örnek, Arial fontunu, kırmızı rengini, 14 font boyutunu ve italik özelliklerini kullanmak üzere headerTextFormat stilini ayarlamak için TextFormat nesnesini kullanır.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 DataGrid bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **aDg** örnek adını verin.
- 3 Eylemler panelini açın, ana Zaman Çizelgesi'nde Kare 1'i seçin ve şu kodu girin:

```
import fl.data.DataProvider;
import fl.controls.dataGridClasses.DataGridColumn;

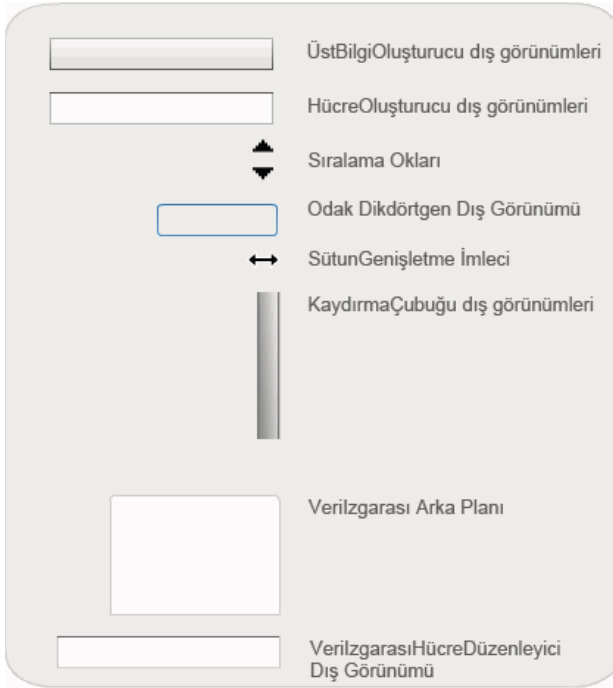
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
        {FirstName:"Ric", LastName:"Dietrich"},
        {FirstName:"Ewing", LastName:"Canepa"},
        {FirstName:"Kevin", LastName:"Wade"},
        {FirstName:"Kimberly", LastName:"Dietrich"},
        {FirstName:"AJ", LastName:"Bilow"},
        {FirstName:"Chuck", LastName:"Yushan"},
        {FirstName:"John", LastName:"Roo"}
];

// Assign the data provider to the DataGrid to populate it.
// Note: This has to be done before applying the cellRenderers.
aDg.dataProvider = new DataProvider(myDP);
aDg.setSize(160,190);
aDg.move(40,40);
aDg.columns[0].width = 80;
aDg.columns[1].width = 80;
var tf:TextFormat = new TextFormat();
tf.size = 14;
tf.color = 0xff0000;
tf.italic = true;
tf.font = "Arial"
aDg.setStyle("headerTextFormat", tf);
```

4 Uygulamayı çalıştırmak için Kontrol Et > Filmi Test Et'i seçin.

DataGrid bileşeniyle kaplamaları kullanma

DataGrid bileşeni, görsel durumlarını temsil etmesi için şu kaplamaları kullanır:



DataGrid kaplamaları

CellRenderer kaplaması, DataGrid öğesinin gövde hücreleri için kullanılan kaplamayken, HeaderRenderer kaplaması üstbilgi satırı için kullanılır. Aşağıdaki prosedür, üstbilgi satırının arka plan rengini değiştirir ancak CellRenderer kaplamasını düzenleyerek DataGrid öğesinin gövde hücrelerinin arka plan rengini değiştirmek için de aynı işlemi kullanabilirsiniz.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 DataGrid bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **adg** örnek adını verin.
- 3 Kaplamalar paletini açmak için bileşeni çift tıklayın.
- 4 Düzenlemek üzere simgeleri büyütme için yakınlaştırma denetimini %400 değerine ayarlayın.
- 5 HeaderRenderer kaplamalarının paletini açmak için HeaderRenderer kaplamasını çift tıklayın.
- 6 Up_Skin öğesini sembol düzenleme modunda açmak için çift tıklayın ve seçilinceye kadar arka planını tıklayın, böylece Özellik denetçisinde Dolgu rengi seçici görüntülenir.
- 7 Up_Skin HeaderRenderer kaplamasının arka planına uygulamak için Dolgu rengi seçicisini kullanarak #00CC00 rengini seçin.
- 8 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 9 DataGrid öğesine veri eklemek için Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
import fl.data.DataProvider;

bldRosterGrid(aDg);
var aRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter",Home: "Redlands, CA"},
    {Name:"Sue Pennypacker",Home: "Athens, GA"},
    {Name:"Jill Smithfield",Home: "Spokane, WA"},
    {Name:"Shirley Goth", Home: "Carson, NV"},
    {Name:"Jennifer Dunbar",Home: "Seaside, CA"}
];
aDg.dataProvider = new DataProvider(aRoster);
function bldRosterGrid(dg:DataGrid) {
    dg.setSize(400, 130);
    dg.columns = ["Name", "Home"];
    dg.move(50,50);
    dg.columns[0].width = 120;
    dg.columns[1].width = 120;
};
```

10 Uygulamayı test etmek için, Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

DataGrid öğesinin, yeşil üstbilgi satırı arka planıyla aşağıdaki gibi görüntülenmesi gerekir.

Name	Home
Wilma Carter	Redlands, CA
Sue Pennypacker	Athens, GA
Jill Smithfield	Spokane, WA
Shirley Goth	Carson, NV
Jennifer Dunbar	Seaside, CA

Özelleştirilmiş üstbilgi satırı arka planıyla DataGrid

Label bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında Label bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. `autoSize` geliştirme parametresini de ayarlayabilirsiniz; bu parametre ayarlandığında, canlı önizlemedeki sınırlama kutusu değişmez ancak Label öğesi yeniden boyutlandırılır. Label öğesi, `wordwrap` parametresine bağlı olarak yeniden boyutlandırılır. Parametre `true` olursa, Label öğesi metne sığacak şekilde dikey olarak yeniden boyutlandırılır. Parametre `false` olursa, Label öğesi yatay olarak yeniden boyutlandırılır. Çalışma zamanında `setSize()` yöntemini kullanın. Daha fazla bilgi için *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*'ndaki `Label.setSize()` yöntemi ve `Label.autoSize` özelliğine bakın. Ayrıca bkz. "[Label bileşeniyle uygulama oluşturma](#)" sayfa 62.

Label bileşeniyle stilleri kullanma

Etiket örneğinin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. Label bileşeni örneğindeki metnin tamamı aynı stili paylaşmalıdır. Label bileşeni bir `textFormat` stilini içerir, bu stil `TextFormat` nesnesiyle aynı niteliklere sahip olup `Label.text` içeriği için normal Flash `TextField` ile aynı özellikleri ayarlamayı sağlar. Şu örnek, bir etiketteki metnin rengini kırmızı olarak ayarlar.

- 1 Label bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene `a_label` örnek adını verin.
- 2 Parametreler sekmesini tıklatın ve metin özelliği değerinin yerine metni getirin.

Beni kırmızı yap

- 3 Ana Zaman Çizelgesi'nde Kare 1 ögesini seçin, Eylemler panelini açın ve şu kodu girin:

```
/* Create a new TextFormat object, which allows you to set multiple text properties at a
time. */

var tf:TextFormat = new TextFormat();
tf.color = 0xFF0000;
/* Apply this specific text format (red text) to the Label instance. */
a_label.setStyle("textFormat", tf);
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

Etiket stilleri ile ilgili daha fazla bilgi için şu adresteki Label sınıfına bakın:[Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

Kaplamalar ve Etiket

Label bileşeni, kaplama için herhangi bir görsel öge içermez.

List bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında List bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini veya `height`, `width`, `scaleX` ve `scaleY` gibi List sınıfının ilgili özelliklerini kullanın.

Bir liste yeniden boyutlandırıldığında, listenin satırları yatay olarak daraltılır ve içindeki metinler kırpılır. Dikey olarak, liste gerektiğinde satır ekler veya kaldırır. Kaydırma çubukları, otomatik olarak gerektiği şekilde konumlandırılır.

List bileşeniyle stilleri kullanma

List bileşeninin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. Stiller, bileşen çizildiğinde bileşenin kaplamaları ve dolgusu için değerleri belirtir.

Çeşitli kaplama stilleri, kaplama için kullanılmak üzere farklı sınıflar belirtmenizi sağlar. Kaplama stillerini kullanma hakkında daha fazla bilgi için bkz. "[Kaplamalar Hakkında](#)" sayfa 101.

Aşağıdaki prosedür, List bileşeni için `contentPadding` stilinin değerini ayarlar. İçerik etrafında dolgu elde etmek için, bu ayarın değerinin, List ögesinin boyutundan çıkarıldığına dikkat edin, böylece List ögesindeki metnin kırpılmasını önlemek için List ögesinin boyutunu artırmanız gerekebilir.

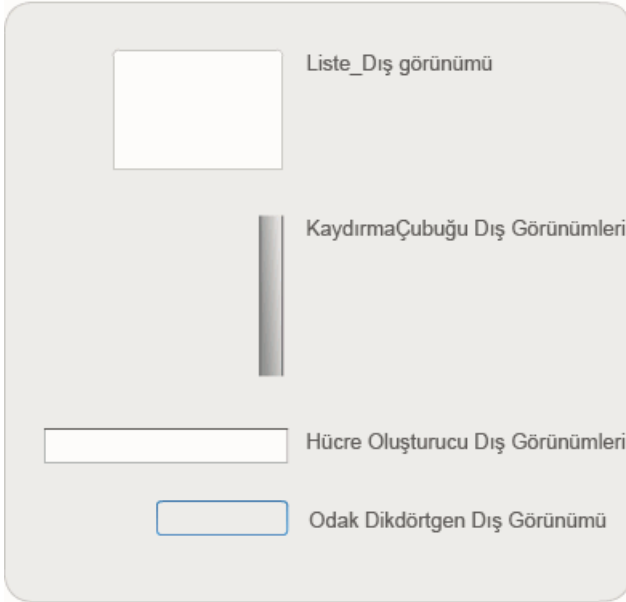
- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **aList** örnek adını verin.
- 3 Ana Zaman Çizelgesi'nde Kare 1 ögesini seçin, Eylemler panelini açın ve `contentPadding` stilini ayarlayıp List ögesine verileri ekleyen şu kodu girin:

```
aList.setStyle("contentPadding", 5);  
aList.setSize(145, 200);  
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});  
aList.addItem({label:"1966 Mustang (Classic)", data:27000});  
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});  
aList.rowCount = aList.length;
```

4 Kontrol Et > Filmi Test Et'i seçin.

List bileşeniyle kaplamaları kullanma

List bileşeni, görsel durumlarını temsil etmesi için şu kaplamaları kullanır:

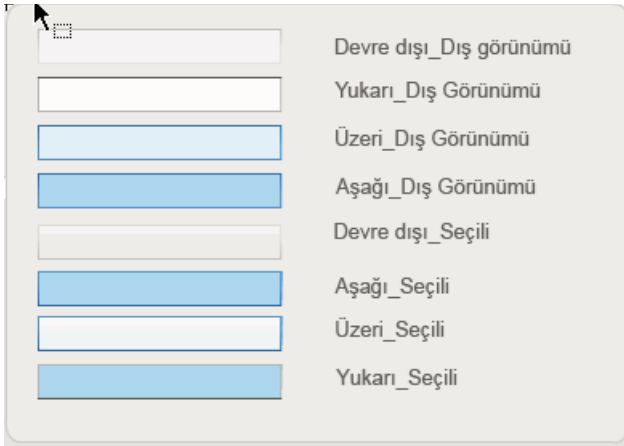


List kaplamaları

ScrollBar ögesi için kaplama oluşturma hakkında daha fazla bilgi için bkz. "[UIScrollBar bileşenini özelleştirme](#)" sayfa 132. Focus Rect kaplaması için kaplama oluşturma hakkında bilgi için bkz. "[TextArea bileşenini özelleştirme](#)" sayfa 127

Not: Bir bileşendeki ScrollBar kaplaması değiştirildiğinde, ScrollBar ögesini kullanan diğer tüm bileşenler için de ScrollBar kaplaması değiştirilir.

List hücresinin farklı durumları için ikinci bir kaplamalar paletini açmak için Cell Renderer kaplamasını çift tıklayın.



List Cell Renderer kaplamaları

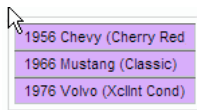
Bu kaplamaları düzenleyerek List öğesinin hücrelerinin görünümünü değiştirebilirsiniz. Aşağıdaki prosedür, normal devre dışı durumunda List öğesinin görünümünü değiştirmek için Up kaplamasının rengini değiştirir.

- 1 Yeni bir Flash dosyası (ActionScript 3.0) belgesi oluşturun.
- 2 List bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **aList** örnek adını verin.
- 3 Kaplamalar paletini açmak için List öğesini çift tıklayın.
- 4 Cell Renderer kaplamalarının paletini açmak için Cell Renderer kaplamasını çift tıklayın.
- 5 Up_Skin kaplamasını düzenlemek üzere açmak için çift tıklayın.
- 6 Kaplamanın dolgu alanını tıklatarak seçin. Özellik denetçisinde, kaplamanın geçerli dolgu rengiyle bir Dolgu rengi seçici görüntülenir.
- 7 Up_Skin kaplamasının dolgusuna uygulamak için Dolgu rengi seçicisini kullanarak #CC66FF rengini seçin.
- 8 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 9 List öğesine veri eklemek için Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
aList.setStyle("contentPadding", 5);  
aList.setSize(145, 200);  
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});  
aList.addItem({label:"1966 Mustang (Classic)", data:27000});  
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});  
aList.rowCount = aList.length;
```

- 10 Kontrol Et > Filmi Test Et'i seçin.

List öğesi, aşağıdaki gibi görüntülenir:



özel Up_Skin renkli List hücreleri

contentPadding stilinin ayarlanması sonucunda kareler oluşturulur.

NumericStepper bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında NumericStepper bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini veya `width`, `height`, `scaleX` ve `scaleY` gibi NumericStepper sınıfının ilgili özelliklerini ve yöntemlerini kullanın.

NumericStepper bileşeni yeniden boyutlandırıldığında, aşağı ve yukarı ok düğmelerinin genişliği değişmez. Adımlayıcı, varsayılan yükseklikten daha büyük bir değere yeniden boyutlandırılırsa, varsayılan davranış ok düğmelerini bileşenin üst ve alt kısmına sabitler. Aksi takdirde, 9 dilimli ölçekleme düğmelerin nasıl çizileceğini belirler. Ok düğmeleri her zaman metin kutusunun sağında görüntülenir.

Stiller ve NumericStepper bileşeni

NumericStepper bileşeninin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. Stiller, bileşen çizildiğinde bileşenin kaplamaları, dolgusu ve metin biçimi için değerleri belirtir. `textFormat` stili, NumericStepper'ın boyut ve görünüm değerini değiştirmenizi sağlar. Çeşitli kaplama stilleri, kaplama için kullanılmak üzere farklı sınıflar belirtmenizi sağlar. Kaplama stillerini kullanma hakkında daha fazla bilgi için bkz. "[Kaplamlar Hakkında](#)" sayfa 101.

Bu prosedür, NumericStepper öğesinin görüntülediği değer görünümünü değiştirmek için `textFormat` stilini kullanır.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 NumericStepper bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **myNs** örnek adını verin.
- 3 Ana Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin:

```
var tf:TextFormat = new TextFormat();  
myNs.setSize(100, 50);  
tf.color = 0x0000CC;  
tf.size = 24;  
tf.font = "Arial";  
tf.align = "center";  
myNs.setStyle("textFormat", tf);
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

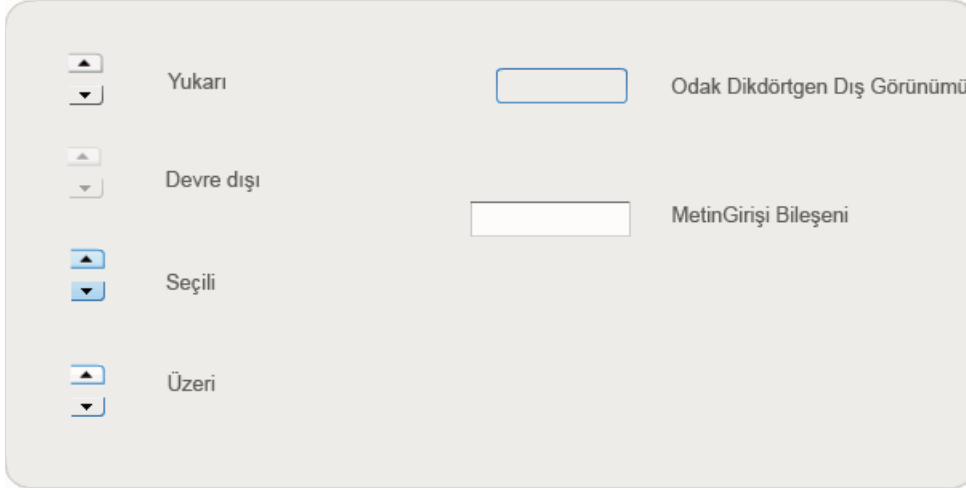
Kaplamlar ve NumericStepper bileşeni

NumericStepper bileşeni, düğmelerinin basılmamış, basılmış, devre dışı ve seçili durumlarını temsil eden kaplamalara sahiptir.

Adımlayıcı etkinleştirilirse, işaretçi üzerine geldiğinde basılmış ve basılmamış düğmeleri üzerinde duruma görüntülenir. Üzerine basıldığında düğmeler basılmış duruma görüntülenir. Fare serbest bırakıldığında, düğmeler basılmamış duruma geri döner. Fare basılı durumdayken işaretçi düğmeden ayrılırsa, düğmeler orijinal duruma geri döner.

Adımlayıcı devre dışı bırakılırsa, kullanıcı müdahalesine bakılmaksızın adımlayıcı devre dışı duruma görüntülenir.

NumericStepper bileşeni şu kaplamaları içerir:



NumericStepper kaplamaları

- 1 Yeni bir FLA dosyası oluşturun.
- 2 NumericStepper bileşenini Sahne Alanı'na sürükleyin.
- 3 Düzenlemek üzere görüntüyü büyötmek için Yakınlaştırma denetimini %400 değerine ayarlayın.
- 4 Grup düzeyine ininceye ve Özellik denetçisindeki Dolgu rengi seçicisinde arka plan rengi görüntüleninceye kadar kaplamalar panelinde TextInput kaplamasının arka planını çift tıklatın.
- 5 Özellik denetçisinde Dolgu rengi seçicisini kullanarak #9999FF rengini TextInput kaplamasının arka planına uygulamak için seçin.
- 6 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 7 Kaplamalar panelini yeniden açmak için NumericStepper ögesini tekrar çift tıklatın.
- 8 Arka plan seçilinceye ve arka planın rengi Özellik denetçisindeki Dolgu rengi seçicisinde görüntüleninceye kadar Up grubunda yukarı ok düğmesinin arka planını çift tıklatın.
- 9 Yukarı ok düğmesinin arka planına uygulamak için #9966FF rengini seçin.
- 10 Up grubunda aşağı ok için 8 ve 9 adımlarını yineleyin.
- 11 Kontrol Et > Filmi Test Et'i seçin.

NumericStepper örneğinin aşağıda gösterildiği gibi görünmesi gerekir:



ProgressBar bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında ProgressBar bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında setSize() yöntemini veya height, width, scaleX ve scaleY gibi ProgressBar sınıfının ilgili özelliklerini kullanın.

ProgressBar üç kaplama içerir: yol kaplaması, çubuk kaplaması ve belirsiz kaplaması. Bu öge, varlıkları ölçeklemek için 9 dilimli ölçekleme kullanır.

Stiller ve ProgressBar bileşeni

ProgressBar örneğinin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. ProgressBar ögesinin stilleri, bileşen çizildiğinde bileşenin kaplaması ve dolgusu için değerleri belirtir. Şu örnek, ProgressBar örneğinin boyutunu artırır ve barPadding stilini ayarlar.

- 1 Yeni bir FLA dosyası oluşturun.
- 2 ProgressBar bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve bu bileşene **myPb** örnek adını verin.
- 3 Ana Zaman Çizelgesi'nin Kare 1'inde, Eylemler Paneli'ne şu kodu girin:

```
myPb.width = 300;  
myPb.height = 30;  
  
myPb.setStyle("barPadding", 3);
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

Kaplama stilleri ayarlama hakkında bilgi için bkz. "[Kaplamlar Hakkında](#)" sayfa 101.

Kaplamlar ve ProgressBar bileşeni

Aşağıda gösterildiği gibi ProgressBar bileşeni, ilerleme çubuğu yolunu, tamamlandı çubuğunu ve belirsiz çubuğunu temsil etmek için kaplamalar kullanır.



ProgressBar kaplamaları

Çubuk, konumlandırmasını belirlemek için barPadding ögesi kullanılarak yol kaplamasının üzerine yerleştirilir. Varlıklar, 9 dilimli ölçekleme kullanılarak ölçeklenir.

ProgressBar örneğinin indeterminate özelliği true olarak ayarlandığında belirsiz çubuğu kullanılır. Kaplama, ProgressBar ögesinin boyutuna sığacak şekilde dikey olarak ve yatay olarak yeniden boyutlandırılır.

ProgressBar ögesinin görünümünü değiştirmek için bu kaplamaları düzenleyebilirsiniz. Örneğin, şu örnek belirsiz çubuğunun rengini değiştirir.

- 1 Yeni bir FLA dosyası oluşturun.
- 2 Bir ProgressBar bileşenini sahne alanına sürükleyin ve kaplama simgeleri panelini açmak için bileşeni çift tıklayın.
- 3 Belirsiz çubuk kaplamasını çift tıklayın.
- 4 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %400 değerine ayarlayın.
- 5 Diagonal çubuklardan birini çift tıklayın ve sonra Shift tuşunu basılı tutup diğerlerini teker teker tıklayın. Geçerli renk, Özellik denetçisindeki Dolgu rengi seçicisinde görüntülenir.

- 6 Özellik denetçisindeki Dolgu rengi seçicisini açmak için tıklatın ve #00CC00 rengini seçili diyagonal çubuklara uygulamak için seçin.
- 7 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 8 Kontrol Et > Filmi Test Et'i seçin.
ProgressBar öğesinin aşağıda gösterildiği gibi görünmesi gerekir.



RadioButton bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında RadioButton bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini kullanın.

RadioButton bileşeninin sınırlama kutusu görünür hale gelir ve bileşenin vuruş alanını atar. Bileşenin boyutunu artırırsanız, vuruş alanının boyutunu da artırırsınız.

Bileşenin sınırlama kutusu bileşen etiketi için çok küçükse, etiket sınırlama kutusuna sığacak şekilde kırpılır.

RadioButton bileşeniyle stilleri kullanma

RadioButton örneğinin görünümünü değiştirmek için stil özelliklerini ayarlayabilirsiniz. RadioButton bileşeninin stil özellikleri, bileşen çizildiğinde bu bileşenin kaplamaları, simgeleri, metin biçimlendirmesi ve dolgusu için değerleri belirtir. RadioButton bileşeninin stilleri, bileşen çizildiğinde bu bileşenin mizanpajının kaplamaları ve dolgusu için değerleri belirtir.

Şu örnek, bir CheckBox bileşeninden `textFormat` stilini alır ve etiketlerinin stilini aynı hale getirmek için bu stili RadioButton öğesine uygular.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 Bir CheckBox bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene **myCh** örnek adını verin.
- 3 Bir RadioButton bileşenini Sahne Alanı'na sürükleyin ve Özellik denetçisinde bileşene **myRb** örnek adını verin.
- 4 Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin.

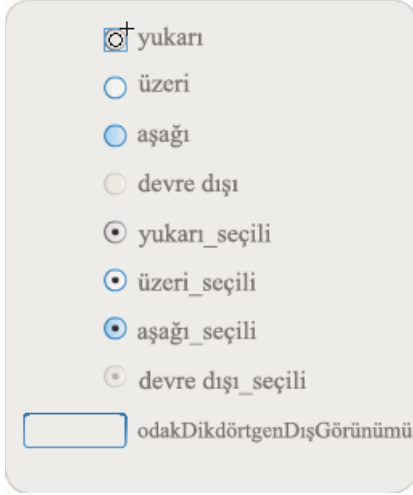
```
var tf:TextFormat = new TextFormat();  
tf.color = 0x00FF00;  
tf.font = "Georgia";  
tf.size = 18;  
myCh.setStyle("textFormat", tf);  
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Bu kod, CheckBox için `textFormat` stilini ayarlar ve CheckBox öğesinde `getStyle()` yöntemini çağırarak bu stili RadioButton öğesine uygular.

- 5 Kontrol Et > Filmi Test Et'i seçin.

Kaplamalar ve RadioButton bileşeni

RadioButton bileşeni, görünümünü değiştirmek için düzenleyebileceğiniz şu kaplamaları içerir:



RadioButton kaplamaları

Bir RadioButton etkinleştirilmiş ve seçilmemişse, kullanıcı işaretçiyi üzerine getirdiğinde, over kaplamasını görüntüler. Kullanıcı bir RadioButton öğesini tıklattığında, girdi odağını alır ve selected_down kaplamasını görüntüler. Kullanıcı fareyi serbest bıraktığında, RadioButton öğesi selected_up kaplamasını görüntüler. Kullanıcı fare düğmesine basarken işaretçiyi RadioButton öğesinin vuruş alanından kaldırır, RadioButton yeniden up kaplamasını görüntüler.

RadioButton devre dışı bırakılırsa, kullanıcı müdahalesine bakılmaksızın devre dışı durumda görüntülenir.

Şu örnek, seçili durumunu belirten selected_up kaplamasının yerine geçer.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 RadioButton bileşenini Sahne Alanı'na sürükleyin ve bileşenin kaplamalar paletini açmak için üzerini çift tıklayın.
- 3 Düzenlemek üzere simgeyi büyütmek için yakınlaştırma denetimini %800 değerine ayarlayın.
- 4 selected_up kaplamasını seçmek için çift tıklayın ve silmek için Delete tuşuna basın.
- 5 Araçlar panelinde Dikdörtgen aracını seçin.
- 6 Özellik denetçisinde, çizgi rengini kırmızı (#FF0000) ve Dolgu rengini siyah (#000000) olarak ayarlayın.
- 7 Dikdörtgen çizmek için, sembolün kayıt noktasını (ayrıca *başlangıç noktası* veya *sıfır noktası*) işaretleyen artı işaretinden başlayarak işaretçiyi tıklatıp sürükleyin.
- 8 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 9 Kontrol Et > Filmi Test Et'i seçin.
- 10 RadioButton öğesini seçmek için tıklayın.

Seçili durumundaki RadioButton öğesinin aşağıda gösterilene benzer şekilde görünmesi gerekir.



ScrollPane bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında ScrollPane bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini veya `height`, `width`, `scaleX` ve `scaleY` gibi ScrollPane sınıfının ilgili özelliklerini veya yöntemlerini kullanın.

ScrollPane bileşeni, şu grafiksel özellikleri içerir:

- İçeriğinin kayıt noktası (ayrıca *başlangıç noktası* veya *sıfır noktası*), bölmenin sol üst köşesinde bulunur.
- Yatay kaydırma çubuğu devre dışı bırakıldığında, kaydırma bölmesinin sağ tarafında üstten alta doğru dikey kaydırma çubuğu görüntülenir. Dikey kaydırma çubuğu devre dışı bırakıldığında, kaydırma bölmesinin alt kısmında soldan sağa doğru yatay kaydırma çubuğu görüntülenir. Her iki kaydırma çubuğunu da devre dışı bırakabilirsiniz.
- Kaydırma bölmesi çok küçükse, içerik doğru şekilde görüntülenmeyebilir.
- Kaydırma bölmesi yeniden boyutlandırıldığında, kaydırma yolu ve kaydırma kutusu (kaydırma çubuğu) genişler veya küçülür ve vuruş alanları yeniden boyutlandırılır. Düğmeler aynı boyutta kalır.

ScrollPane bileşeniyle stilleri kullanma

ScrollPane bileşeninin stil özellikleri, bileşen çizildiğinde bileşenin mizanpajının kaplamaları ve dolgusu için değerleri belirtir. Çeşitli kaplama stilleri, bileşenin kaplamaları için kullanılmak üzere farklı sınıflar belirtmenizi sağlar. Kaplama stillerini kullanma hakkında daha fazla bilgi için bkz. "[Kaplama Hakkında](#)" sayfa 101.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 Bir ScrollPane bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **mySp** örnek adını verin.
- 3 Özellik denetçisinde Parametreler sekmesini tıklatın ve `source` parametresi için şu değeri girin:
<http://www.helpexamples.com/flash/images/image1.jpg>

- 4 Ana Zaman Çizelgesi'nin Kare 1'inde, Eylemler paneline şu kodu ekleyin:

```
mySp.setStyle("contentPadding", 5);
```

Kaydırma çubuklarının dışında, bileşenin kenarlığı ile içeriği arasına dolgu uygulandığını unutmayın.

- 5 Kontrol Et > Filmi Test Et'i seçin.

Kaplama ve ScrollPane

ScrollPane bileşeni, kaydırma varlıkları için bir kenarlık ve kaydırma çubukları kullanır. Kaydırma çubukları için kaplama oluşturma hakkında bilgi almak için bkz. "[UIScrollBar bileşeniyle kaplamaları kullanma](#)" sayfa 133.

Slider bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında Slider bileşenini yatay olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında, `setSize()` yöntemini veya `width` ve `scaleX` özellikleri gibi Slider sınıfının ilgili özelliklerini kullanın.

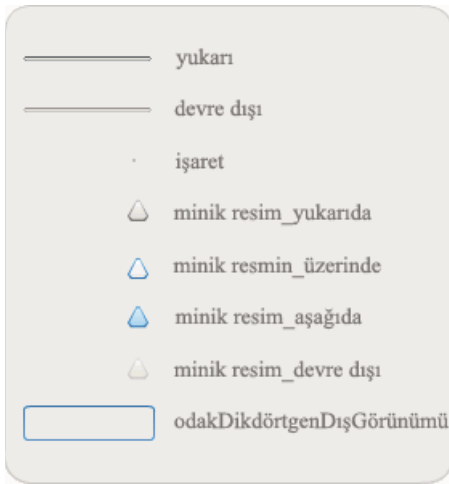
Kaydırıcıyı yalnızca daha uzun hale getirebilirsiniz. Yüksekliğini artıramazsınız. Flash, `setSize()` yönteminin `height` özelliğini ve `height` parametresini yoksayar. Ancak dikey olarak daha uzun hale getirmek için dikey bir kaydırıcı oluşturabilirsiniz.

Stiller ve Slider bileşeni

Slider bileşeninin stilleri yalnızca bileşenin sınırlama kutusu ile dış sınırı arasındaki dolgu için kullanılacak piksel sayısını belirten `FocusRectPadding` değerini ve kaplamalarının sınıflarını belirtir. Kaplama stillerini kullanma hakkında daha fazla bilgi için bkz. “[Kaplamlar Hakkında](#)” sayfa 101.

Kaplamlar ve Slider bileşeni

Slider bileşeni, görünümünü değiştirmek için düzenleyebileceğiniz şu kaplamaları kullanır.



Slider kaplamaları

Şu örnek, up yolunun rengini mavi olarak değiştirmek için up yolunu düzenler.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 Slider bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin.
- 3 Slider bileşeninin kaplamalar panelini açmak için bileşeni çift tıklayın.
- 4 Sembol düzenleme modunda açmak için, kayıt noktasında up yolunu çift tıklayın.
- 5 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %800 değerine ayarlayın. Slider yolunun üç çubuktan oluştuğuna dikkat edin.
- 6 Üst çubuğu seçmek için tıklayın. Seçildiği zaman, üst çubuğun rengi Özellik denetçisindeki Dolgu rengi seçicisinde görüntülenir.
- 7 Özellik denetçisinde Dolgu rengi seçicisini kullanarak #000066 rengini Slider yolunun üst çubuğuna uygulamak için seçin.
- 8 Slider yolunun orta çubuğunu seçmek için tıklayın. Seçildiği zaman, orta çubuğun rengi Özellik denetçisindeki Dolgu rengi seçicisinde görüntülenir.
- 9 Özellik denetçisinde Dolgu rengi seçicisini kullanarak #0066FF rengini Slider yolunun orta çubuğuna uygulamak için seçin.

- 10 Slider yolunun alt çubuğunu seçmek için tıklattın. Seçildiği zaman, alt çubuğun rengi Özellik denetçisindeki Dolgu rengi seçicisinde görüntülenir.
- 11 Özellik denetçisinde Dolgu rengi seçicisini kullanarak #00CCFF rengini Slider yolunun alt çubuğuna uygulamak için seçin.
- 12 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklattın.
- 13 Kontrol Et > Filmi Test Et'i seçin.
Slider ögesi, aşağıdaki gibi görüntülenir:



TextArea bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında TextArea bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında, `setSize()` yöntemini veya `height`, `width`, `scaleX` ve `scaleY` gibi, TextArea sınıfının ilgili özelliklerini kullanın.

Bir TextArea bileşeni yeniden boyutlandırıldığında, kenarlık yeni sınırlama kutusuna yeniden boyutlandırılır. Gerekiyorsa, kaydırma çubukları alt ve sağ kenara yerleştirilir. Daha sonra kalan alan içinde metin alanı yeniden boyutlandırılır; TextArea bileşeninde sabit boyutlu bir öğe yoktur. TextArea bileşeninin genişliği, metnin boyutunu görüntüleyemeyecek kadar küçükse, metin kırpılır.

Stiller ve TextArea bileşeni

TextArea bileşeninin stilleri, bileşen çizildiğinde bileşenin kaplamaları, dolgusu ve metin biçimi için değerleri belirtir. `textFormat` ve `disabledTextFormat` stilleri, TextArea ögesinin görüntülediği metnin stilini belirler. Kaplama stili özellikleri hakkında bilgi almak için bkz. "[TextArea bileşeniyle kaplamaları kullanma](#)" sayfa 128.

Şu örnek, TextArea devre dışı bırakıldığında metnin görünümünü değiştirmek için `disabledTextFormat` stilini ayarlar ancak etkinleştirilmiş bir TextArea için `textFormat` stilinin ayarlanmasında da aynı işlem geçerlidir.

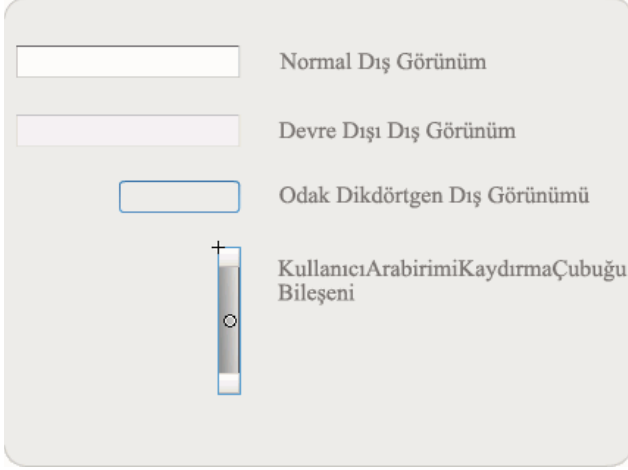
- 1 Yeni bir Flash dosyası oluşturun.
- 2 Bir TextArea bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **myTa** örnek adını verin.
- 3 Ana Zaman Çizelgesi'nin Kare 1 ögesindeki Eylemler paneline şu kodu ekleyin.

```
var tf:TextFormat = new TextFormat();
tf.color = 0xCC99FF;
tf.font = "Arial Narrow";
tf.size = 24;
myTa.setStyle("disabledTextFormat", tf);
myTa.text = "Hello World";
myTa.setSize(120, 50);
myTa.move(200, 50);
myTa.enabled = false;
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

TextArea bileşeniyle kaplamaları kullanma

TextArea bileşeni, görünümünü değiştirmek için düzenleyebileceğiniz şu kaplamaları kullanır.



TextArea kaplamaları

Not: Bir bileşendeki ScrollBar kaplaması değiştirildiğinde, ScrollBar ögesini kullanan diğer tüm bileşenler için de ScrollBar kaplaması değiştirilir.

Aşağıdaki prosedür, TextArea ögesinde odak ve Normal kaplama olduğunda görüntülenen Focus Rect Kaplamasının kenarlık renklerini değiştirir.

- 1 Yeni bir Flash dosyası oluşturun.
- 2 Bir TextArea bileşenini sahne alanına sürükleyin ve kaplama simgeleri panelini açmak için bileşeni çift tıklayın.
- 3 Focus Rect Kaplamasını çift tıklayın.
- 4 Focus Rect Kaplamasının kenarlığını seçmek için tıklayın. Seçildiği zaman, geçerli rengi Özellik denetçisindeki Dolgu rengi seçicisinde görüntülenir.
- 5 Özellik denetçisindeki Dolgu rengi seçicisini açmak için tıklayın ve #CC0000 rengini kenarlığa uygulamak için seçin.
- 6 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 7 TextArea bileşeninin kaplama simgeleri panelini açmak için bileşeni çift tıklayın.
- 8 Normal kaplamayı çift tıklayın.
- 9 Normal kaplamanın kenarlığının her kenarını teker teker seçin ve rengini #990099 olarak ayarlayın.
- 10 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 11 Kontrol Et > Filmi Test Et'i seçin.

Metin girmeye başlamak için TextArea ögesini seçtiğinizde, bu ögenin kenarlığının aşağıda gösterildiği gibi görünmesi gerekir:



Dış kenarlık Focus Rect kaplaması ve iç kenarlık da Normal kaplamanın kenarlığıdır.

UIScrollBar kaplamasının düzenlenmesiyle ilgili bilgi için bkz. “UIScrollBar bileşenini özelleştirme” sayfa 132.

TextInput bileşenini özelleştirme

Geliştirme sırasında veya çalışma zamanında TextInput örneğinin boyutunu değiştirebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında setSize() yöntemini veya height, width, scaleX ve scaleY gibi TextInput sınıfının ilgili özelliklerini kullanın.

Bir TextInput bileşeni yeniden boyutlandırıldığında, kenarlık yeni sınırlama kutusuna yeniden boyutlandırılır. TextInput bileşeni kaydırma çubuklarını kullanmaz ancak kullanıcı metinle etkileşim kurdukça ekleme noktası otomatik olarak kaydırılır. Daha sonra kalan alan içinde metin alanı yeniden boyutlandırılır; TextInput bileşeninde sabit boyutlu bir öğe yoktur. TextInput bileşeni, metni görüntüleyemeyecek kadar küçükse, metin kırılır.

Stiller ve TextInput bileşeni

TextInput bileşeninin stilleri, bileşen çizildiğinde bileşenin kaplamaları, dolgusu ve metin biçimlendirmesi için değerleri belirtir. textFormat ve disabledTextFormat stilleri, bileşende görüntülenen metnin stilini belirler. Kaplama stili özellikleri hakkında daha fazla bilgi almak için bkz. “Kaplamlar ve TextInput bileşeni” sayfa 129.

Şu örnek, TextInput bileşeninde görüntülenen metnin fontunu, boyutunu ve rengini ayarlamak için textFormat stilini ayarlar. Aynı işlem, bileşen devre dışı bırakıldığında uygulanan disabledTextFormat stilinin ayarlanmasında da geçerlidir.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 Bir TextInput bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **myTi** örnek adını verin.
- 3 Ana Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin.

```
var tf:TextFormat = new TextFormat();
tf.color = 0x0000FF;
tf.font = "Verdana";
tf.size = 30;
tf.align = "center";
tf.italic = true;
myTi.setStyle("textFormat", tf);
myTi.text = "Enter your text here";
myTi.setSize(350, 50);
myTi.move(100, 50);
```

- 4 Kontrol Et > Filmi Test Et'i seçin.

Kaplamlar ve TextInput bileşeni

TextInput bileşeni, görünümünü değiştirmek için düzenleyebileceğiniz şu kaplamaları kullanır:



TextInput resim yazısı

Aşağıdaki prosedür, TextInput bileşeninin kenarlığını ve arka plan renklerini değiştirir:

- 1 Yeni bir Flash dosyası oluşturun.
- 2 Bir TextInput bileşenini Sahne Alanı'na sürükleyin ve kaplamalar panelini açmak için bileşeni çift tıklayın.
- 3 Normal kaplamayı çift tıklayın.
- 4 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %800 değerine ayarlayın.
- 5 Normal kaplamanın kenarlığının her kenarını teker teker seçin ve uygulanacak rengini #993399 olarak ayarlayın.
- 6 Özellik denetçisindeki Dolgu rengi seçicisinde rengi görüntüleninceye kadar arka planı çift tıklayın. Arka plana uygulamak için #99CCCC rengini seçin.
- 7 Belge düzenleme moduna geri dönmek için, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 8 Kontrol Et > Filmi Test Et'i seçin.

TextInput bileşeninin aşağıda gösterildiği gibi görünmesi gerekir:



TileList bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında TileList bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında, `setSize()` yöntemini veya `width`, `height`, `columnCount`, `rowCount`, `scaleX` ve `scaleY` gibi ilgili özellikleri kullanın. TileList bileşenin içerisinde bulunan ScrollBar, liste kutusu ile birlikte ölçeklenir.

Stiller ve TileList bileşeni

TileList bileşeninin stilleri, bileşen çizildiğinde bileşenin kaplamaları, dolgusu ve metin biçimlendirmesi için değerleri belirtir. `textFormat` ve `disabledTextFormat` stilleri, bileşende görüntülenen metnin stilini belirler. Kaplama stilleri hakkında daha fazla bilgi almak için bkz. "[TileList bileşeniyle kaplamaları kullanma](#)" sayfa 131.

Şu örnek, TileList örneğinde görüntülenen etiketlerin font, boyut, renk ve metin özelliklerini ayarlamak için `textFormat` stilini kullanarak `setRendererStyle()` yöntemini çağırır. Aynı işlem, `enabled` özelliği `false` değerine ayarlandığında uygulanan `disabledTextFormat` stilinin ayarlanmasında da geçerlidir.

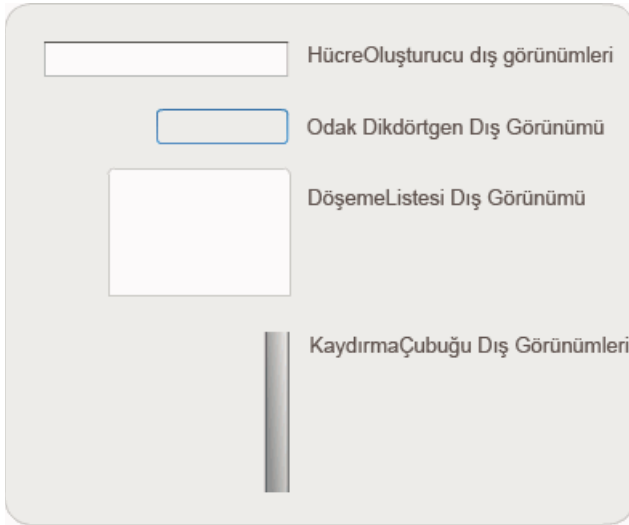
- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 TileList bileşenini Sahne Alanı'na sürükleyin ve bu bileşene `myTI` örnek adını verin.

3 Zaman Çizelgesi'nin Kare 1 öğesindeki Eylemler paneline şu kodu ekleyin.

```
myTl.setSize(100, 100);  
myTl.addItem({label:"#1"});  
myTl.addItem({label:"#2"});  
myTl.addItem({label:"#3"});  
myTl.addItem({label:"#4"});  
var tf:TextFormat = new TextFormat();  
tf.font = "Arial";  
tf.color = 0x00FF00;  
tf.size = 16;  
tf.italic = true;  
tf.bold = true;  
tf.underline = true;  
tf.align = "center";  
myTl.setRendererStyle("textFormat", tf);
```

TileList bileşeniyle kaplamaları kullanma

TileList bileşeni, bir TileList kaplaması, bir CellRenderer kaplaması ve bir ScrollBar kaplaması içerir. TileList öğesinin görünümünü değiştirmek için bu kaplamaları düzenleyebilirsiniz:



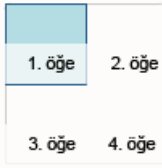
TileList kaplamaları

Not: Bir bileşendeki ScrollBar kaplaması değiştirildiğinde, ScrollBar öğesini kullanan diğer tüm bileşenler için de ScrollBar kaplaması değiştirilir.

Aşağıdaki prosedür, TileList öğesinin CellRenderer Selected_Up kaplamasının rengini değiştirir.

- 1 Bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 TileList bileşenini Sahne Alanı'na sürükleyin ve kaplamalar panelini açmak için bileşeni çift tıklayın.
- 3 CellRenderer kaplamasını çift tıklayın, ardından Selected_Up kaplamasını çift tıklayın ve dikdörtgen arka planını tıklayın.
- 4 Özellik denetçisindeki Dolgu rengi seçicisini kullanarak #99FFFF rengini Selected_Up kaplamasına uygulamak için seçin.

- 5 Belge düzenleme moduna geri dönünceye kadar, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklatın.
- 6 Değerler iletişim kutusunu açmak için, Özellik denetçisinin Parametreler sekmesinde, dataProvider satırının ikinci sütununu çift tıklatın. Şu etiketlerle öğeleri ekleyin: 1. öğe, 2. öğe, 3. öğe, 4. öğe.
- 7 Kontrol Et > Filmi Test Et'i seçin.
- 8 TileList öğesindeki hücrelerden birini seçmek için tıklatın ve sonra fareyi seçili hücreden kaldırın. Seçili hücrenin aşağıdaki gibi görüntülenmesi gerekir:



Değiştirilmiş Selected_Up kaplama rengi ile TileList bileşeni

UILoader bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında UILoader bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında, `setSize()` yöntemini veya `width`, `height`, `scaleX` ve `scaleY` gibi ilgili özellikleri kullanın.

UILoader bileşeninin boyutlandırma davranışı, `scaleContent` özelliği tarafından denetlenir. `scaleContent` öğesi `true` olduğunda, yükleyicinin sınırları içine sığacak şekilde içerik ölçeklenir (ve `setSize()` çağrıldığında içerik yeniden ölçeklenir). `scaleContent` öğesi `false` olduğunda, bileşenin boyutu içeriğin boyutuna sabitlenir ve `setSize()` ve boyutlandırma özelliklerinin herhangi bir etkisi olmaz.

UILoader bileşeninin, stil veya kaplama uygulayabileceğiniz herhangi bir kullanıcı arabirimi öğesi yoktur.

UIScrollBar bileşenini özelleştirme

Geliştirme sırasında ve çalışma zamanında UIScrollBar bileşenini yatay olarak ve dikey olarak dönüştürebilirsiniz. Ancak dikey bir UIScrollBar öğesi genişliği değiştirmenize ve yatay bir UIScrollBar öğesi de yüksekliği değiştirmenize izin vermez. Geliştirme sırasında, Sahne Alanı'nda bileşeni seçin ve Serbest Dönüştürme aracını veya Değiştir > Dönüştür komutlarından herhangi birini kullanın. Çalışma zamanında `setSize()` yöntemini veya `width`, `height`, `scaleX` ve `scaleY` gibi UIScrollBar sınıfının ilgili özelliklerini kullanın.

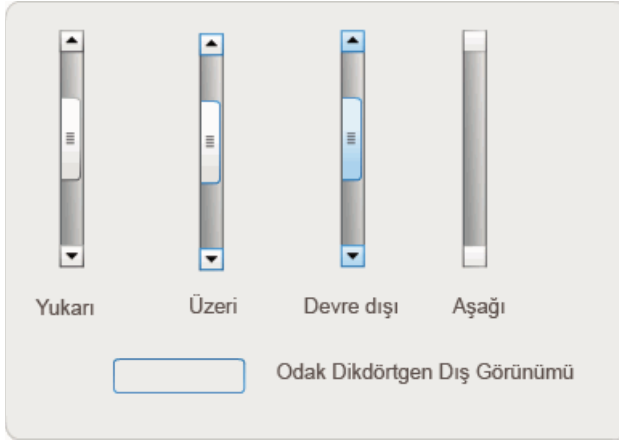
Not: `setSize()` yöntemini kullanırsanız, yatay bir kaydırma çubuğunun yalnızca genişliğini veya dikey bir kaydırma çubuğunun yalnızca yüksekliğini değiştirebilirsiniz. Geliştirme sırasında, yatay bir kaydırma çubuğunun yüksekliğini veya dikey bir kaydırma çubuğunun genişliğini ayarlayabilirsiniz, ancak film yayınlandığında değerler sıfırlanır. Kaydırma çubuğunun yalnızca uzunluğuna karşılık gelen boyutu değiştirilebilir.

UIScrollBar bileşeniyle stilleri kullanma

UIScrollBar bileşeninin stilleri yalnızca bileşenin sınırlama kutusu ile dış sınırı arasındaki dolgu için kullanılacak piksel sayısını belirten FocusRectPadding değerini ve kaplamalarının sınıflarını belirtir. Kaplama stillerini kullanma hakkında daha fazla bilgi için bkz. “[Kaplamlar Hakkında](#)” sayfa 101.

UIScrollBar bileşeniyle kaplamaları kullanma

UIScrollBar bileşeni şu kaplamaları kullanır.



UIScrollBar kaplamaları

Yatay ve dikey kaydırma çubukları aynı kaplamaları kullanır; yatay bir kaydırma çubuğu görüntülenirken, UIScrollBar bileşeni gerektiğinde kaplamalarını döndürür.

Not: Bir bileşendeki ScrollBar kaplaması değiştirildiğinde, ScrollBar ögesini kullanan diğer tüm bileşenler için de ScrollBar kaplaması değiştirilir.

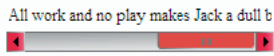
Şu örnek, UIScrollBar kaydırma kutusunun ve ok düğmelerinin renginin nasıl değiştirildiğini gösterir.

- 1 Yeni bir Flash belgesi (ActionScript 3.0) oluşturun.
- 2 UIScrollBar bileşenini Sahne Alanı'na sürükleyin ve bu bileşene **mySb** örnek adını verin. Parametreler sekmesinde, yönü yatay olarak ayarlayın.
- 3 Kaplamalar panelini açmak için kaydırma çubuğunu çift tıklayın.
- 4 Up kaplamasını seçmek için tıklayın.
- 5 Düzenlemek üzere simgeyi büyütme için yakınlaştırma denetimini %400 değerine ayarlayın.
- 6 Arka plan seçilinceye ve arka planın rengi, Özellik denetçisindeki Dolgu rengi seçicisinde görüntüleninceye kadar sağ okun (veya dikey kaydırma çubuğu için yukarı okun) arka planını çift tıklayın.
- 7 Düğme arka planına uygulamak için #CC0033 rengini seçin.
- 8 Belge düzenleme moduna geri dönünceye kadar, Sahne Alanı'nın yukarısındaki düzenleme çubuğunun sol tarafındaki Geri düğmesini tıklayın.
- 9 Kaydırma kutusu ve sol ok (veya dikey kaydırma çubuğu için aşağı ok) için 6, 7 ve 8 adımlarını yineleyin.
- 10 Kaydırma çubuğunu bir TextField ögesine eklemek için Zaman Çizelgesi'nin Kare 1'indeki Eylemler paneline şu kodu ekleyin.

```
var tf:TextField = new TextField();  
addChild(tf);  
tf.x = 150;  
tf.y = 100;  
mySb.width = tf.width = 200;  
tf.height = 22;  
tf.text = "All work and no play makes Jack a dull boy. All work and no play makes Jack a  
dull boy. All . . .";  
mySb.y = tf.y + tf.height;  
mySb.x = tf.x + tf.width;x  
mySb.scrollTarget = tf;
```

11 Kontrol Et > Filmi Test Et'i seçin.

UIScrollBar bileşeninin aşağıdaki gibi görüntülenmesi gerekir:



Kaydırma kutusu ile sol ve sağ okları kırmızı renkte olan dikey ScrollBar

Bölüm 6: FLVPlayback Bileşenini kullanma

FLVPlayback bileşeni, HTTP üzerinden aşamalı olarak indirilen video dosyalarını veya Adobe Macromedia Flash Media Server'daki ya da Flash Video Streaming Service'deki (FVSS) video akış dosyalarını oynatmak için Adobe Flash CS5 Professional uygulamanıza kolayca bir video oynatıcı dahil etmenize olanak tanır.

Adobe Flash Player 9 Update 3'ün (sürüm 9.0.115.0 veya sonrası) piyasaya sürülmesiyle, Flash Player uygulamasında video oynatmaya yönelik önemli gelişmeler elde edilmiştir. Bu güncelleme, daha iyi video oynatma performansı sağlamak için son kullanıcı sisteminin video donanımından yararlanan FLVPlayback bileşenindeki değişiklikleri içerir. FLVPlayback bileşenindeki değişiklikler arasında, tam ekran modunda görüntülenen video dosyalarının aslına uygunluğunun artması da yer alır.

Ayrıca, Flash Player 9 Update 3, endüstri standardı H.264 kodlamasını kullanan yüksek tanımlamalı MPEG-4 video biçimleri için destek ekleyerek FLVPlayback bileşeninin işlevlerini artırır. Bu biçimler arasında MP4, M4A, MOV, MP4V, 3GP ve 3G2 yer alır.

Not: Korunmuş MP4 dosyaları—örn. iTunes®'dan indirilenler veya FairPlay® tarafından dijital olarak şifrelenenler—desteklenmez.

Kullanımı kolay FLVPlayback bileşeni şu özellik ve avantajlara sahiptir:

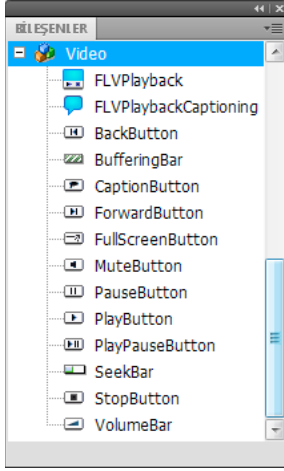
- Sahne Alanı'na sürüklenip hızlı ve başarılı şekilde uygulanabilir
- Tam ekran boyutunu destekler
- Oynatma denetimlerinin görünümünü özelleştirmenizi sağlayan önceden tasarlanmış bir *kaplamalar* koleksiyonu sunar
- Önceden tasarlanmış kaplamalar için renk ve alfa değerleri seçmenize olanak sağlar
- İleri düzey kullanıcıların kendi kaplamalarını oluşturmasına olanak sağlar
- Geliştirme sırasında canlı önizleme sağlar
- Yeniden boyutlandırma sırasında video merkezinin ortalanması için mizanpaj özellikleri sağlar
- Aşamalı olarak indirilen video dosyasının yeterli bölümü indirildiğinde oynatma işlemini başlatır
- Videonuzu metin, grafik ve animasyonla senkronize etmenize olanak veren işaret noktaları sağlar
- SWF dosyasının makul ölçüdeki boyutunu korur.

FLVPlayback bileşenini kullanma

FLVPlayback bileşeninin kullanımı, bileşenin Sahne Alanı'na koyulması ve bileşenin oynatması için bir video dosyasının belirtilmesinden ibarettir. Ayrıca, davranışını belirlemek ve video dosyasını tanımlamak için çeşitli parametreler de belirleyebilirsiniz.

FLVPlayback bileşeni, ActionScript uygulama programlama arabirimi de (API) içerir. API, *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu* bölümünde eksiksiz tanımlanan şu sınıfları kapsar: CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState ve bir çok olay sınıfı - AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent ve VideoProgressEvent.

FLVPlayback bileşeni, FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerini içerir. FLVPlayback bileşeni, video dosyasını ve bu dosyayı çalıştırmayı sağlayan denetimleri görüntülediğiniz bir görüntüleme alanı ile video oynatıcı birleşimidir. FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri, video dosyasını oynatmak, durdurmak, duraklatmak ve denetlemek için kullanabildiğiniz denetim düğmeleri ve mekanizmalar sağlar. Bu denetimler arasında, BackButton, BufferingBar, CaptionButton (FLVPlaybackCaptioning için), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton ve VolumeBar yer alır. FLVPlayback bileşeni ve FLV Oynatma Özel Kullanıcı Arabirimi denetimleri, aşağıda gösterildiği gibi Bileşenler panelinde görüntülenir:



Bileşenler panelindeki FLVPlayback bileşenleri

FLVPlayback bileşenine oynatma denetimleri ekleme işlemine *kaplama oluşturma* denir. FLVPlayback bileşeni, oynatma, durdurma, geri, ileri, arama çubuğu, sessiz, ses düzeyi, tam ekran ve resim yazısı ekleme denetimlerini sağlayan bir varsayılan başlangıç kaplamasına sahiptir. Bu kaplamayı değiştirmek için şu seçenekler bulunur:

- Önceden tasarlanmış kaplamalar koleksiyonundan seçim yapma
- Özel bir kaplama oluşturup önceden tasarlanmış kaplamalar koleksiyonuna ekleme
- FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerinden tek tek denetimleri seçip özelleştirme

Önceden tasarlanmış bir kaplamayı seçtiğinizde, geliştirme sırasında veya çalışma zamanında kaplama rengini ve alfa değerlerini ayrı ayrı seçebilirsiniz. Daha fazla bilgi için bkz. “[Önceden tasarlanmış bir kaplama seçme](#)” sayfa 153.

Farklı bir kaplama seçildikten sonra, seçilen kaplama yeni varsayılan kaplama olur.

FLVPlayback bileşeni için bir dış görünüm seçme ya da oluşturma hakkında daha fazla bilgi için bkz. “[FLVPlayback bileşenini özelleştirme](#)” sayfa 153.

FLVPlayback bileşeniyle uygulama oluşturma

Şu yolları kullanarak FLVPlayback bileşenini uygulamanıza dahil edebilirsiniz:

- FLVPlayback bileşenini Bileşenler panelinden Sahne Alanı'na sürükleyin ve `source` parametresi için bir değer belirtin.
- Sahne Alanı'nda bileşen oluşturmak için Video İçe Aktarma sihirbazını kullanın ve bir kaplama seçerek bileşeni özelleştirin.
- Bir FLVPlayback örneğini Sahne Alanı'nda dinamik olarak oluşturmak için `FLVPlayback()` yapıcısını kullanın (bileşenin kütüphanede olduğunu varsayılır).

Not: ActionScript ile bir FLVPlayback örneği oluşturursanız, ActionScript ile `skin` özelliğini ayarlayarak bu örneğe bir kaplama da atamanız gerekir. Kaplamayı bu şekilde uyguladığınızda, kaplama SWF dosyasıyla otomatik olarak yayınlanmaz. Hem uygulama SWF dosyasını hem de kaplama SWF dosyasını uygulama sunucunuza kopyalamanız gerekir, aksi takdirde, uygulamayı çalıştırdığınızda kaplama SWF dosyası kullanılamaz.

FLVPlayback bileşenini Bileşenler panelinden sürükleme

- 1 Bileşenler panelinde, video girişini açmak için Artı (+) düğmesini tıklatın.
- 2 FLVPlayback bileşenini Sahne Alanı'na sürükleyin.
- 3 Sahne Alanı'nda seçilen FLVPlayback bileşeniyle, Bileşen denetçisinin Parametreler sekmesinde `source` parametresinin Değer hücrelerini bulun ve aşağıdakilerden birini belirten bir dize girin:
 - Video dosyasının yerel yolu
 - Video dosyasının URL'si
 - Video dosyasının nasıl oynatılacağını açıklayan, senkronize edilmiş Multimedia Integration Language (SMIL) dosyasının URL'siBir veya daha fazla FLV dosyası açıklamak için SMIL dosyasının nasıl oluşturulacağı hakkında bilgi almak için bkz. "[SMIL dosyasını kullanma](#)" sayfa 164.
- 4 Bileşen denetçisinin Parametreler sekmesinde, Sahne Alanı'nda FLVPlayback bileşeni seçili durumdayken, `skin` parametresi için Değer hücrelerini tıklatın.
- 5 Kaplama Seç iletişim kutusunu açmak için büyüteç simgesini tıklatın.
- 6 Şu seçeneklerden birini belirleyin:
 - Bir oynatma denetimleri kümesini bileşene eklemek için, Kaplama açılır listesindeki önceden tasarlanmış kaplamalardan birini seçin.
 - Özel bir kaplama oluşturduysanız, açılır menüden Özel Kaplama URL'si seçeneğini belirleyin ve URL kutusuna, kaplamayı içeren SWF dosyasının URL'sini girin.
 - Yok seçeneğini belirleyin ve oynatma denetimleri eklemek için tek tek FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerini Sahne Alanı'na sürükleyin.

Not: İlk iki durumda, açılır menünün yukarıdaki görüntüleme bölümünde kaplamanın önizlemesi görüntülenir. Kaplamanın rengini değiştirmek için Renk seçicisini kullanabilirsiniz.

Özel Kullanıcı Arabirimi denetiminin rengini değiştirmek için, denetimi özelleştirmeniz gerekir. Özel Kullanıcı Arabirimi denetimlerini kullanma hakkında daha fazla bilgi için, bkz. "[FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri için ayrı ayrı kaplama oluşturma](#)" sayfa 154.
- 7 Kaplama Seç iletişim kutusunu kapatmak için Tamam'ı tıklatın.
- 8 SWF dosyasını çalıştırıp videoyu başlatmak için Kontrol Et > Filmi Test Et seçeneklerini belirleyin. Aşağıdaki prosedür, FLVPlayback bileşeni eklemek için Video İçer Aktarma sihirbazını kullanır:

Video İçer Aktarma sihirbazını kullanma:

- 1 Dosya > İçer Aktar > Video İçer Aktar seçeneklerini belirleyin.
- 2 Aşağıdaki seçeneklerden birini belirleyerek video dosyasının konumunu belirtin:
 - Yerel bilgisayarında
 - Bir web sunucusuna, Flash Video Streaming Service'e veya Flash Media Server'a zaten konuşlandırılmış
- 3 Tercihinize bağlı olarak, video dosyasının konumunu belirten yolu veya URL'yi girin, ardından İleri'yi tıklatın.

4 Bir dosya yolu seçtiyseniz, daha sonra, videonuzu nasıl konuşlandırmak istediğinizi belirtmek için listelenen seçeneklerden birini belirleyebileceğiniz Konuşlandırma iletişim kutusunu görürsünüz:

- Standart bir web sunucusundan aşamalı indirme
- Flash Video Streaming Service'ten akış
- Flash Media Server'dan akış
- Videoyu SWF dosyasına gömme ve Zaman Çizelgesi'nde oynatma

Önemli: Video Göm seçeneğini belirlemeyin. FLVPlayback bileşeni yalnızca harici akış videosunda oynatılır. Bu seçenek, Sahne Alanı'na bir FLVPlayback bileşeni yerleştirmez.

5 İleri'yi tıklatın.

6 Şu seçeneklerden birini belirleyin:

- Bir oynatma denetimleri kümesini bileşene eklemek için, Kaplama açılır listesindeki önceden tasarlanmış kaplamalardan birini seçin.
- Bileşen için özel bir kaplama oluşturduysanız, açılır menüden Özel Kaplama URL'si seçeneğini belirleyin ve URL kutusuna kaplamayı içeren SWF dosyasının URL'sini girin.
- Yok seçeneğini belirleyin ve oynatma denetimleri eklemek için tek tek FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerini Sahne Alanı'na sürükleyin.

Not: İlk iki durumda, açılır menünün yukarısındaki görüntüleme bölümünde kaplamanın önizlemesi görüntülenir.

7 Kaplama Seç iletişim kutusunu kapatmak için Tamam'ı tıklatın.

8 Sonra ne olacağını not etmek için Video İçer Aktarmayı Bitir iletişim kutusunu okuyun ve Bitir'i tıklatın.

9 FLA dosyanızı kaydetmediyseniz, Farklı Kaydet iletişim kutusu görüntülenir.

10 SWF dosyasını çalıştırıp videoyu başlatmak için Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

Aşağıdaki prosedür, ActionScript kullanarak FLVPlayback bileşeni ekler.

ActionScript kullanarak dinamik olarak bir örnek oluşturma:

1 FLVPlayback bileşenini Bileşenler panelinden Kütüphane paneline (Pencere > Kütüphane) sürükleyin.

2 Zaman Çizelgesi'nin Kare 1 ögesindeki Eylemler paneline şu kodu ekleyin. *install_drive* ögesini, Flash uygulamasını yüklediğiniz sürücüyle değiştirin ve yüklemeniz için Kaplamalar klasörünün konumunu yansıtmak üzere yolu değiştirin.

Windows yüklü bilgisayarda:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///install_drive|/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

Macintosh bilgisayarda:

```
import fl.video.*;
var my_FLVPlayback = new FLVPlayback();
my_FLVPlayback.x = 100;
my_FLVPlayback.y = 100;
addChild(my_FLVPlayback);
my_FLVPlayback.skin = "file:///Macintosh HD:Applications:Adobe Flash
CS5:Configuration:FLVPlayback Skins:ActionScript 3.0SkinOverPlaySeekMute.swf"
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/water.flv";
```

Not: *source ve skin özelliklerini ayarlamazsanız, oluşturulan film klibi boş görüntülenir.*

3 SWF dosyasını çalıştırıp video dosyasını başlatmak için Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

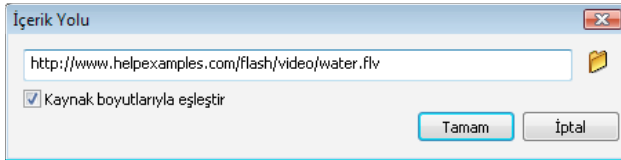
FLVPlayback bileşeni parametreleri

Her FLVPlayback bileşeni örneği için, Bileşen denetçisinde veya Özellik denetçisinde şu parametreleri ayarlayabilirsiniz: align, autoPlay, cuePoints, preview, scaleMode, skin, skinAutoHide, skinBackgroundAlpha, skinBackgroundColor, source ve volume. Bu parametrelerin her biri, aynı adda, karşılık gelen ActionScript özelliğine sahiptir. Bu parametrelere bir değer atadığınızda, uygulamada özelliğin başlangıç durumunu ayarlarsınız. Özellik ActionScript'te ayarlandığında, parametrede ayarladığınız değer geçersiz kalır. Bu parametrelerin olası değerleriyle ilgili bilgi için [Flash Professional için ActionScript 3.0 Başvurusu](#) bölümündeki FLVPlayback sınıfına bakın.

FLVPlayback source parametresini belirtme

source parametresi, Flash uygulamasına dosyanın nasıl oynatılacağını bildiren, video dosyasının adını ve konumunu belirtmenizi sağlar.

Bileşen denetçisindeki source parametresinin Değer hücrelerini çift tıklatarak İçerik Yolu iletişim kutusunu açın.



FLVPlayback Bileşen Yolu iletişim kutusu

İçerik Yolu iletişim kutusu, Sahne Alanı'nda FLVPlayback örneğinin kaynak video dosyasının boyutlarıyla eşleşip eşleşmeyeceğini belirten, Kaynak FLV Boyutlarını Eşletir onay kutusunu sağlar. Kaynak video dosyası, oynatma için tercih edilen yükseklik ve genişlik boyutlarını içerir. Bu seçeneği belirlerseniz, FLVPlayback örneğinin boyutları, bu tercih edilen boyutlarla eşleşecek şekilde yeniden boyutlandırılır.

Kaynak

Video dosyasının nasıl oynatılacağını açıklayan video dosyası veya XML dosyasının URL'sini ya da yerel yolunu girin. Video dosyasının tam konumunu bilmiyorsanız, doğru konumu bulmanıza yardımcı olması için Tarayıcı iletişim kutusunu açmak üzere klasör simgesini tıklatın. Bir video dosyasına göz atarken, hedef SWF dosyasının olduğu konumda ya da altındaysa, dosyayı bir web sunucusundan sunabilmeniz için Flash yolu otomatik olarak o konuma göre yapar. Aksi takdirde yol mutlak bir Windows veya Macintosh yoludur. Yerel XML dosyasının adını girmek için yolu ve adı girin.

Bir HTTP URL'si belirtirseniz, video dosyası aşamalı indirme olarak oynatılır. RTMP URL olan bir URL belirtirseniz, video dosyası Flash Media Server'dan veya FVSS'den akışa alınır. XML dosyasının bir URL'si de Flash Media Server veya FVSS'den akış video dosyası olabilir.

Önemli:

Çoklu bant genişlikleri için çoklu video dosyası akışının nasıl oynatılacağını açıklayan bir SMIL dosyasının konumunu da belirtebilirsiniz. Bu dosya, FLV dosyalarını açıklamak için Synchronized Multimedia Integration Language (SMIL) kullanır. SMIL dosyasının açıklaması için bkz. “[SMIL dosyasını kullanma](#)” sayfa 164.

ActionScript FLVPlayback ögesini kullanarak video dosyasının adını ve konumunu da belirtebilirsiniz. `source` özelliği ve `FLVPlayback.play()` ve `FLVPlayback.load()` yöntemleri. Bu üç alternatif, Bileşen denetçisinde `source` parametresine göre öncelik elde eder. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde FLVPlayback sınıfı için olan `FLVPlayback.source`, `FLVPlayback.play()` ve `FLVPlayback.load()` girişlerine bakın.

Tam ekran desteği

FLVPlayback bileşeninin ActionScript 3.0 sürümü, Flash Player 9.0.28.0 veya sonrasını gerektiren tam ekran modunu destekler ve HTML de tam ekran görüntüleme için doğru şekilde ayarlanmıştır. Önceden tasarlanmış kaplamaların bazıları tam ekran modunu etkinleştirme ve devre dışı bırakma için bir aç/kapa düğmesi içerir. Aşağıdaki resimde, denetim çubuğunun sağ tarafında `FullScreenButton` simgesi görüntülenmektedir.



Denetim çubuğunda tam ekran simgesi

Yalnızca `fullScreenTakeOver` özelliği `true` değerine ayarlanırsa (varsayılan değer budur) tam ekran desteği gerçekleşir.

Tam ekran desteği, donanım ivmesi desteği ile veya bu destek olmadan da gerçekleşebilir. Donanım hızlandırma desteği hakkında bilgi için bkz. “[Donanım hızlandırma](#)” sayfa 143.

FLVPlayback ögesine yönelik tam ekran desteği uygulamak için:

- 1 FLVPlayback bileşenini uygulamanıza ekleyin ve bu bileşene bir video dosyası atayın.
- 2 FLVPlayback bileşeni için, tam ekran düğmesi içeren bir kaplama (örn. `SkinUnderPlaySeekFullscreen.swf`) seçin veya `FullScreenButton` kullanıcı arabirimi bileşenini Bileşenler panelindeki Video bölümünden FLVPlayback bileşenine ekleyin.
- 3 Dosya > Yayınlama Ayarları seçeneklerini belirleyin.
- 4 Yayınlama Ayarları iletişim kutusunda HTML sekmesini tıklayın.
- 5 HTML sekmesinde, Şablon açılır menüsünden Tam Ekran Destekli Flash seçeneğini belirleyin.
- 6 Ayrıca HTML sekmesinde, kullandığınız Flash Player sürümüne bağlı olarak Flash Sürümünü Algıla onay kutusunu seçin ve 9.0.28 veya sonrası bir sürümü belirtin.
- 7 Biçimler sekmesini seçin ve hem Flash (.swf) hem de HTML (.html) seçeneklerinin belirlenmiş olduğundan emin olun. Varsayılan dosya adlarını değiştirebilirsiniz.
- 8 Yayınla'yı ve ardından Tamam'ı tıklayın.

Adım 7'ye alternatif olarak, dış aktarılan HTML dosyasını varsayılan tarayıcınızda otomatik olarak açmak için Tamam'ı tıklayıp sonra Dosya > Yayın Önizlemesi > Varsayılan - (HTML) seçeneklerini belirleyebilirsiniz. Aksi takdirde, tam ekran seçeneğini test etmek için dış aktarılan HTML dosyasını tarayıcınızla açın.

Tam ekran destekli FLVPlayback bileşenini web sayfanıza eklemek için, dış aktarılan HTML dosyasını açın ve SWF dosyasını gömen kodu web sayfanızın HTML dosyasına kopyalayın. Bu kodun şu örneğe benzer şekilde görünmesi gerekir:


```
//from the <head> section

<script language="javascript"> AC_FL_RunContent = 0; </script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
<!--
// -----
// Globals
// Major version of Flash required
var requiredMajorVersion = 9;
// Minor version of Flash required
var requiredMinorVersion = 0;
// Revision of Flash required
var requiredRevision = 28;
// -----
// -->
</script>

//and from the <body> section

<script language="JavaScript" type="text/javascript">
<!--
if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
    alert("This page requires AC_RunActiveContent.js.");
} else {
    var hasRightVersion = DetectFlashVer(requiredMajorVersion,
        requiredMinorVersion, requiredRevision);
    if(hasRightVersion) { // if we've detected an acceptable version
        // embed the Flash movie
        AC_FL_RunContent(
            &apos;codebase&apos;, &apos;http://download.macromedia.com/pub/
            shockwave/cabs/flash/swflash.cab#version=9,0,28,0&apos;,
            &apos;width&apos;, &apos;550&apos;,
            &apos;height&apos;, &apos;400&apos;,
            &apos;src&apos;, &apos;fullscreen&apos;,
            &apos;quality&apos;, &apos;high&apos;,
            &apos;pluginspage&apos;, &apos;http://www.macromedia.com/go/
            getflashplayer&apos;,
            &apos;align&apos;, &apos;middle&apos;,
            &apos;play&apos;, &apos>true&apos;,
            &apos;loop&apos;, &apos>true&apos;,
            &apos;scale&apos;, &apos;showall&apos;,
            &apos;wmode&apos;, &apos;window&apos;,
            &apos;devicefont&apos;, &apos>false&apos;,
            &apos;id&apos;, &apos;fullscreen&apos;,
            &apos;bgcolor&apos;, &apos;#ffffff&apos;,
            &apos;name&apos;, &apos;fullscreen&apos;,
            &apos;menu&apos;, &apos>true&apos;,
            &apos;allowScriptAccess&apos;, &apos;sameDomain&apos;,
```

```
        &apos;allowFullScreen&apos;;, &apos>true&apos;;,  
        &apos;movie&apos;;, &apos;fullscreen&apos;;,  
        &apos;salign&apos;;, &apos;&apos;; ); //end AC code  
    } else { // Flash is too old or we can&apos;t detect the plug-in.  
        var alternateContent = &apos;Alternative HTML content should be placed  
            here.&apos;;  
        + &apos;This content requires Adobe Flash Player.&apos;;  
        + &apos;<a href=http://www.macromedia.com/go/getflash/>Get Flash</a>  
            &apos;;;  
        document.write(alternateContent); // Insert non-Flash content.  
    }  
}  
// -->  
</script>  
<noscript>  
    // Provide alternative content for browsers that do not support scripting  
    // or for those that have scripting disabled.  
    Alternative HTML content should be placed here. This content requires Adobe Flash Player.  
    <a href="http://www.macromedia.com/go/getflash/">Get Flash</a>  
</noscript>
```

Alternatif olarak, web sayfanızın şablonu olarak dışa aktarılan HTML dosyasını kullanabilir ve bu dosyaya diğer içeriğinizi ekleyebilirsiniz. Ancak bunu yaparsanız, daha sonra Flash uygulamasından FLVPlayback HTML dosyasını tekrar dışa aktardığınızda kazara üzerine yazmayacağınız şekilde HTML dosyasının adını değiştirin.

Herhangi bir durumda, HTML dosyasıyla aynı klasöre dışa aktarılan AC_RunActiveContent.js dosyasını web sunucunuza yüklemeniz gerekir.

Tam ekran modu için ActionScript desteği, `fullScreenBackgroundColor`, `fullScreenSkinDelay`, ve `fullScreenTakeOver` özelliklerini ve `enterFullScreenDisplayState()` yöntemini içerir. Bu ActionScript öğeleriyle ilgili bilgi için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

enterFullScreenDisplayState() ögesini kullanma

Şu örnekte gösterildiği gibi, `enterFullScreenDisplayState()` ActionScript yöntemini çağırarak da tam ekran modunu çağırabilirsiniz.

```
function handleClick(e:MouseEvent):void {  
    myFLVPlybk.enterFullScreenDisplayState();  
}  
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

Bu örnekte, tam ekran modu FLVPlayback kaplamasındaki tam ekran aç/kapa düğmesi tıklanarak *değil*, bunun yerine web sayfası oluşturucusunun tam ekran modunu çağırmak için dahil ettiği bir düğme (MyButton) tıklanarak çağırılır. Bu düğme tıklatıldığında `handleClick` olay işleyicisi çağırılarak `enterFullScreenDisplayState()` yöntemi çağırılır.

`enterFullScreenDisplayState()` yöntemi, `Stage.displayState` özelliğini `StageDisplayState.FULL_SCREEN` olarak ayarlar ve böylece `displayState` özelliğiyle aynı kısıtlamaları taşır. `enterFullScreenDisplayState()` yöntemi ve `Stage.displayState` özelliği için bkz. [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#).

Donanım hızlandırma

Flash Player 9.0.115.0 ve sonraki sürümler, FLVPlayback ögesinin tam ekran modunda oynattığı FLV dosyalarının performans ve aslına uygunluğunu geliştirmek için kullanılabilir video donanımından yararlanan kod içerir. Ön koşullar karşılanır ve `fullScreenTakeOver` özelliği `true` değerine ayarlanırsa, Flash Player yazılım üzerinden ölçeklemek yerine video dosyasını ölçeklemek için donanım ivmesini kullanır. FLVPlayback bileşeni önceki bir Flash Player sürümünde çalışıyorsa veya donanım ivmesi ön koşulu yoksa, Flash Player önceden yaptığı gibi video dosyasını kendisi ölçekler.

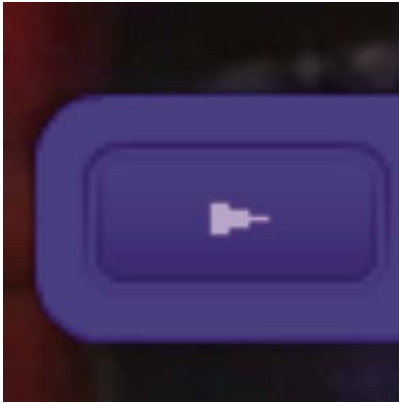
Tam ekran desteğine yönelik donanım ivmesinden yararlanmak için, bilgisayarınızda DirectX 7 uyumlu 4 MB veya daha yüksek VRAM'e (video RAM) sahip ekran kartı bulunmalıdır. Bu donanım desteği, Windows 2000 veya Mac OS X 10.2 işletim sistemlerinde ve bunların sonraki sürümlerinde mevcuttur. Direct X®, diğerlerinin arasında üç boyutlu ve iki boyutlu grafiklere ivme vermek için yazılım ile video donanımı arasında bir arabirim oluşturan API'ler sağlar.

Donanım ivmesi modundan yararlanmak için aşağıdaki yollardan birini kullanarak tam ekran modunu da çağırmanız gerekir:

- FLVPlayback kaplamasında tam ekran aç/kapa düğmesini kullanma
- `FullScreenButton` video denetimini kullanma
- ActionScript `enterFullScreenDisplayState()` yöntemini kullanma. Daha fazla bilgi için bkz. "[enterFullScreenDisplayState\(\) ögesini kullanma](#)" sayfa 142.

`Stage.displayState` özelliğini `StageDisplayState.FULLSCREEN` olarak ayarlayıp tam ekran modunu çağırırsanız, video donanımı ve belleği kullanılabilir olsa da FLVPlayback donanım ivmesini kullanmaz.

Tam ekran desteği için donanım ivmesi kullanılmasının sonucunda, FLVPlayback kaplamaları video oynatıcı ve video dosyasıyla birlikte ölçeklenir. Aşağıdaki görüntüde, FLVPlayback kaplamasında donanım ivmesi bulunurken tam ekran modunun oluşturduğu etki tam çözünürlükte ayrıntılı şekilde gösterilmektedir.



1600 x 1200 çözünürlüğe sahip bir monitörde 320x240 piksel video ile tam ekran modu

Bu görüntü, 320 genişliğe ve 240 yüksekliğe (bunlar varsayılan FLVPlayback boyutlarıdır) sahip bir video dosyasıyla 1600 x 1200 monitörde tam ekran modu kullanımının sonucunu göstermektedir. Kaplamadaki deformasyon etkisi, daha küçük boyutlardaki FLV dosyalarında veya daha büyük monitörlerde daha belirgin olur. Bunun tersine, deformasyon etkisi daha büyük FLV dosyalarında veya daha küçük monitörlerde daha az belirgin olur. Örneğin, 640 x 480 değerinden 1600 x 1200 değerine değiştirildiğinde, kaplamanın boyutu artar ancak kaplama daha az deforme edilmiş şekilde görünür.

FLVPlayback kaplamasının ölçeklemesini sınırlamak için `skinScaleMaximum` özelliğini ayarlayabilirsiniz. Varsayılan değer %4,0 veya 400'dür. Ancak kaplamanın ölçeklemesinin sınırlandırılması için, FLV'yi ölçeklemek üzere donanım ve yazılımın birleşimi gerekir ve bu da yüksek bit oranında kodlanmış büyük boyutlarla FLV'lerin performansını olumsuz şekilde etkileyebilir. Video büyükse (örneğin, 640 piksel veya daha yüksek genişlikte ve 480 piksel veya daha yüksek uzunlukta), `skinScaleMaximum` ögesini küçük bir değere ayarlamamanız gerekir, aksi takdirde bu büyük monitörlerde dikkate değer performans sorunlarına yol açabilir. `skinScaleMaximum` özelliği, performans ve kalite ile büyük kaplama görünümü arasında bir denge elde etmenize olanak sağlar.

Tam ekran modundan çıkma

Tam ekran modundan çıkmak için, tekrar tam ekran düğmesini tıklatın veya Esc tuşuna basın.

Şu özelliklerin ayarlanması ve şu yöntemlerin çağrılması, FLVPlayback bileşeninin tam ekran modundan çıkmasına neden olan mizanpaj değişikliklerine neden olabilir: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` veya `setSize()`.

`align` veya `scaleMode` özelliklerini ayarlarsanız, tam ekran modundan çıkılıncaya kadar FLVPlayback bunları `center` ve `maintainAspectRatio` değerine ayarlar.

Tam ekran kullanılırken `fullScreenTakeOver` özelliği `true` değerinden `false` değerine değiştirildiğinde, donanım ivmesi modu Flash uygulamasının da tam ekran modundan çıkmasına neden olur.

Çoklu video dosyalarını oynatmak için mizanpaj düzenlemesi

ActionScript 3.0 FLVPlayback ögesi, video dosyasının yeniden boyutlandırıldığında veya bileşenin üstüne, altına, soluna ya da sağına konumlandırıldığında ortalanıp ortalanmayacağını belirten bir `align` özelliğine sahiptir. Bileşenin `x`, `y`, `width` ve `height` özelliklerine ek olarak, ActionScript 3.0 bileşeni ayrıca `registrationX`, `registrationY`, `registrationWidth` ve `registrationHeight` özelliklerine de sahiptir. Başlangıçta bunlar `x`, `y`, `width` ve `height` özellikleriyle eşleşir. Sonraki video dosyaları yüklenirken, otomatik yeniden düzenleme bunları değiştirmez, böylece yeni video dosyası aynı yerde ortalanabilir. `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO` olursa, bileşenin genişlik ve yüksekliğinin değişmesine neden olmak yerine sonraki FLV dosyaları bileşenin orijinal boyutlarına sığabilir.

Aşamalı olarak indirilen video dosyalarının otomatik oynatılması

Aşamalı olarak indirilen video dosyası yüklenirken FLVPlayback yalnızca video dosyasının yeterli miktarı indirildiğinde dosyayı oynatmaya başlar, böylece video dosyası baştan sona oynatılabilir.

Yeterli miktarı indirilmeden video dosyasını oynatmak isterseniz, herhangi bir parametre olmadan `play()` yöntemini çağırın.

İndirilecek video dosyasının yeterli miktarı için bekleme durumuna geri dönmek isterseniz, önce `pause()` yöntemini, sonra da `playWhenEnoughDownloaded()` yöntemini çağırın.

İşaret noktalarını kullanma

İşaret noktası, bir video dosyası oynatılırken video oynatıcının bir `cuePoint` olayı gönderdiği noktadır. Web sayfasında başka bir öge için bir eylemin oluşmasını istediğiniz zamanlarda bir FLV dosyasına işaret noktaları ekleyebilirsiniz. Örneğin, metin veya grafik görüntülemek, bir Flash animasyonu senkronize etmek veya FLV dosyasını duraklatarak, videoda farklı bir noktayı arayarak ya da farklı bir FLV dosyasına geçerek FLV dosyasının oynatımını etkilemek isteyebilirsiniz. İşaret noktaları, FLV dosyanızdaki noktaları web sayfasındaki diğer eylemlerle senkronize etmek için ActionScript kodunuzda denetim elde etmenize olanak sağlar.

Üç tür işaret noktası vardır: gezinme, olay ve ActionScript. Gezinme ve olay işaret noktaları FLV dosya akışında ve FLV dosyasının meta veri paketinde gömülü olduğundan *gömülü* işaret noktaları olarak da bilinir.

Gezinme işaret noktası, FLV dosyası içinde belirttiğiniz zamana olabildiğince yakın bir anahtar kare oluşturduğundan, FLV dosyası içinde belirli bir kare aramanıza olanak sağlar. *Anahtar kare*, FLV veri akışında görüntü kareleri arasında gerçekleşen bir veri parçasıdır. Bir gezinme işaret noktası aradığınızda, bileşen anahtar kareyi arar ve `cuePoint` olayını başlatır.

Olay işaret noktası, FLV dosyası içindeki zamanda yer alan bir noktayı web sayfasındaki harici bir olayla senkronize etmenize olanak sağlar. `cuePoint` olayı tam olarak belirtilen zamanda gerçekleşir. Video İçerik Aktarma Sihirbazını veya Flash Video Kodlayıcı'yı kullanarak gezinmeyi ve olay işaret noktalarını bir FLV dosyasına gömebilirsiniz. Video İçerik Aktarma Sihirbazı ve Flash Video kodlayıcı hakkında daha fazla bilgi için bkz. Bölüm 16, "Video ile Çalışma," *Flash Uygulamasını Kullanma*.

ActionScript işaret noktası, bileşenin Flash Video İşaret Noktaları iletişim kutusu üzerinden veya `FLVPlayback.addASCuePoint()` yöntemiyle ekleyebileceğiniz harici bir işaret noktasıdır. Bileşen, FLV dosyasından ayrı olarak ActionScript işaret noktalarını saklar ve izler, böylece doğruluk oranı gömülü işaret noktalarından daha düşüktür. ActionScript işaret noktalarının doğruluğu saniyenin onda biri kadardır. Oynatma kafası güncellendiğinde, bileşen ActionScript işaret noktaları için `cuePoint` olayını oluşturduğundan, `playheadUpdateInterval` özelliğinin değerini düşürerek ActionScript işaret noktalarının doğruluğunu artırabilirsiniz. Daha fazla bilgi için, [Flash Professional için ActionScript 3.0 Başvurusu](#) bölümündeki `FLVPlayback.playheadUpdateInterval` özelliğine bakın.

ActionScript'te ve FLV dosyasının meta verisinde işaret noktası, şu özellikleri içeren bir nesne olarak temsil edilir: `name`, `time`, `type` ve `parameters`. `name` özelliği, işaret noktasının atanmış adını içeren bir dizedir. `time` özelliği, işaret noktasının gerçekleştiği zamanı (SS:DD:SS.mmm) olarak temsil eden bir sayıdır. `type` özelliği, oluşturduğunuz işaret noktası türüne bağlı olarak değeri "navigation", "event" veya "actionscript" olan bir dizedir. `parameters` özelliği, belirtilen ad ve değer çiftlerinin bir dizisidir.

`cuePoint` olayı gerçekleştiğinde, işaret noktası nesnesi, `info` özelliği üzerinden olay nesnesinde kullanılabilir.

Flash Video İşaret Noktaları iletişim kutusunu kullanma

Bileşen denetçisinde `cuePoints` parametresinin Değer hücrelerini çift tıklatarak Flash Video İşaret Noktaları iletişim kutusunu açın. İletişim kutusu aşağıdaki şekilde görünür:

İşaret Noktaları

Ad	Süre	Tür
n1	00:00:15.000	ActionScript
Yeni İşaret Noktası	00:00:25.000	ActionScript
Yeni İşaret Noktası	00:00:40.000	ActionScript
Yeni İşaret Noktası	00:00:50.000	ActionScript
Yeni İşaret Noktası	00:01:04.000	ActionScript

"n1" parametreleri:

Ad	Değer
CC	hello 1

Not: Olay ve Gezinme işaret noktaları düzenlenemez.

Tamam İptal

İşaret Noktaları iletişim noktası

İletişim kutusu, gömülü işaret noktalarını ve ActionScript işaret noktalarını görüntüler. cue point parametrelerinin yanı sıra ActionScript işaret noktaları eklemek ve silmek için bu iletişim kutusunu kullanabilirsiniz. Ayrıca gömülü işaret noktalarını etkinleştirebilir veya devre dışı bırakabilirsiniz. Ancak gömülü işaret noktalarını ekleyemez, değiştiremez veya silemezsiniz.

ActionScript işaret noktası ekleme:

- 1 Flash İşaret Noktaları iletişim kutusunu açmak için, Bileşen denetçisindeki `cuePoints` parametresinin değer hücreğini çift tıklatın.
- 2 Varsayılan bir ActionScript işaret noktası girişi eklemek için, sol üst köşede, işaret noktaları listesinin yukarısındaki artı (+) işaretini tıklatın.
- 3 Ad sütununda Yeni İşaret Noktası metnini tıklatın ve metni işaret noktasının adı olacak şekilde düzenleyin.
- 4 00:00:00:000 Zaman değerini düzenlemek için tıklatın ve işaret noktasının gerçekleşmesi için bir zaman atayın. Zamanı saat, dakika, saniye ve milisaniye (SS:DD:SS.mmm) olarak belirtebilirsiniz.
Birden çok işaret noktası varsa, iletişim kutusu yeni işaret noktasını listedeki kronolojik konumuna taşır.
- 5 Seçilen işaret noktasına bir parametre eklemek için, Parametreler bölümünün yukarısındaki artı (+) işaretini tıklatın ve Ad ve Değer sütunlarına değerleri girin. Her parametre için bu adımı yineleyin.
- 6 Daha fazla ActionScript işaret noktası eklemek için, her biri için 2 ile 5 arasındaki adımları yineleyin.
- 7 Değişikliklerinizi kaydetmek için Tamam'ı tıklatın.

ActionScript işaret noktasını silme:

- 1 Flash İşaret Noktaları iletişim kutusunu açmak için, Bileşen denetçisindeki `cuePoints` parametresinin değer hücreğini çift tıklatın.
- 2 Silmek istediğiniz işaret noktasını seçin.
- 3 Sol üst köşede, işaret noktaları listesinin yukarısındaki eksi (-) işaretini silmek için tıklatın.
- 4 Silmek istediğiniz her işaret noktası için 2 ile 3 adımlarını yineleyin.
- 5 Değişikliklerinizi kaydetmek için Tamam'ı tıklatın.

Gömülü bir FLV dosyası işaret noktasını etkinleştirmek veya devre dışı bırakmak için:

- 1 Flash İşaret Noktaları iletişim kutusunu açmak için, Bileşen denetçisindeki `cuePoints` parametresinin değer hücreğini çift tıklatın.
- 2 Etkinleştirmek veya devre dışı bırakmak istediğiniz işaret noktasını seçin.
- 3 Açılır menüyü tetiklemek için Tür sütunundaki değeri tıklatın veya aşağı oku tıklatın.
- 4 İşaret noktası türünün adını (örneğin, Olay veya Gezinme) etkinleştirmek için tıklatın. Devre dışı bırakmak için Devre Dışı öğesini tıklatın.
- 5 Değişikliklerinizi kaydetmek için Tamam'ı tıklatın.

ActionScript ile işaret noktalarını kullanma

ActionScript işaret noktalarını eklemek, `cuePoint` olaylarını dinlemek, herhangi türde veya belirli bir türde bir işaret noktasını bulmak, gezinme işaret noktasını aramak, işaret noktasını etkinleştirmek veya devre dışı bırakmak, bir işaret noktasının etkinleştirilip etkinleştirilmediğini kontrol etmek ve işaret noktasını kaldırmak için ActionScript'i kullanabilirsiniz.

Bu bölümdeki örnekler, şu üç işaret noktasını içeren `cuepoints.flv` adında bir FLV dosyasını kullanır:

Ad	Zaman	Tür
point1	00:00:00.418	Gezinme
point2	00:00:07.748	Gezinme
point3	00:00:16.020	Gezinme

ActionScript işaret noktaları ekleme

`addASCuePoint()` yöntemini kullanarak FLV dosyasına ActionScript işaret noktaları ekleyebilirsiniz. Şu örnek, FLV dosyası oynatılmaya hazır olduğunda FLV dosyasına iki ActionScript işaret noktası ekler. Bir işaret noktası nesnesi kullanarak, özelliklerinde işaret noktasının zamanını, adını ve türünü belirten birinci işaret noktasını ekler. İkinci çağrı, yöntemin `time` ve `name` parametrelerini kullanarak zamanı ve adı belirtir.

```
// Requires an FLVPlayback instance called my_FLVPlybk on Stage
import fl.video.*;
import fl.video.MetadataEvent;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var cuePt:Object = new Object(); //create cue point object
cuePt.time = 2.02;
cuePt.name = "ASpt1";
cuePt.type = "actionscript";
my_FLVPlybk.addASCuePoint(cuePt); //add AS cue point
// add 2nd AS cue point using time and name parameters
my_FLVPlybk.addASCuePoint(5, "ASpt2");
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.addASCuePoint()` yöntemine bakın.

cuePoint olaylarını dinleme

`cuePoint` olayı, bir `cuePoint` olayı gerçekleştiğinde ActionScript kodunuzda denetim elde etmenizi sağlar. Şu örnekte işaret noktaları gerçekleştiğinde, `cuePoint` dinleyicisi, `playheadTime` özelliğinin değerini ve işaret noktasının adını ve türünü görüntüleyen bir olay işleyici işlevini çağırır. Sonuçları görmek için, bir önceki ActionScript işaret noktaları ekleme bölümündeki örnekle birlikte bu örneği kullanın.

```
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Elapsed time in seconds: " + my_FLVPlybk.playheadTime);
    trace("Cue point name is: " + eventObject.info.name);
    trace("Cue point type is: " + eventObject.info.type);
}
```

`cuePoint` olayıyla ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.cuePoint` olayına bakın.

İşaret noktalarını bulma

ActionScript'i kullanarak herhangi türde bir işaret noktası bulabilir, bir zamana en yakın işaret noktasını bulabilir veya belirli bir ada sahip sonraki işaret noktasını bulabilirsiniz.

Şu örnekteki `ready_listener()` olay işleyici, `ASpt1` işaret noktasını bulmak için `findCuePoint()` yöntemini çağırır ve sonra `ASpt1` işaret noktasının zamanına en yakın gezinme işaret noktasını bulmak için `findNearestCuePoint()` yöntemini çağırır:

```
import fl.video.FLVPlayback;
import fl.video.CuePointType;
import fl.video.VideoEvent;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlayback.addASCuePoint(2.02, "ASpt1");//add AS cue point
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlayback.findCuePoint("ASpt1", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlayback.findNearestCuePoint(rtn_obj.time, CuePointType.NAVIGATION);
    traceit(rtn_obj);
}
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}
```

Şu örnekte, `ready_listener()` olay işleyici `ASpt` işaret noktasını bulur ve aynı ada sahip sonraki işaret noktasını bulmak için `findNextCuePointWithName()` yöntemini çağırır:

```
import fl.video.*;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlayback.addASCuePoint(2.02, "ASpt");//add AS cue point
my_FLVPlayback.addASCuePoint(3.4, "ASpt");//add 2nd ASpt
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlayback.findCuePoint("ASpt", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlayback.findNextCuePointWithName(rtn_obj);
    traceit(rtn_obj);
}
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}
```

İşaret noktalarını bulmayla ilgili daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` ve `FLVPlayback.findNextCuePointWithName()` yöntemlerine bakın.

Gezinme işaret noktalarını arama

Bir gezinme işaret noktasını arayabilir, belirtilen bir zamandan sonraki gezinme işaret noktasını arayabilir ve belirtilen bir zamandan önceki gezinme işaret noktasını arayabilirsiniz. Şu örnek, `ready` olayı gerçekleştiğinde `cuepoints.flv` FLV dosyasını oynatır ve 7,748 değerindeki işaret noktasını arar. `cuePoint` olayı gerçekleştiğinde örnek, birinci işaret noktasını aramak için `seekToPrevNavCuePoint()` yöntemini çağırır. Bu `cuePoint` olayı gerçekleştiğinde örnek, geçerli işaret noktasının zamanı olan `eventObject.info.time` ögesine 10 saniye daha ekleyerek son işaret noktasını aramak için `seekToNextNavCuePoint()` yöntemini çağırır.


```
import fl.video.*;

my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:Object):void {
    my_FLVPlybk.seekToNavCuePoint("point2");
}
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace(eventObject.info.time);
    if(eventObject.info.time == 7.748)
        my_FLVPlybk.seekToPrevNavCuePoint(eventObject.info.time - .005);
    else
        my_FLVPlybk.seekToNextNavCuePoint(eventObject.info.time + 10);
}
my_FLVPlybk.source = "http://helpexamples.com/flash/video/cuepoints.flv";
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` ve `FLVPlayback.seekToPrevNavCuePoint()` yöntemlerine bakın.

Gömülü FLV dosyası işaret noktalarını etkinleştirme ve devre dışı bırakma

`setFLVCuePointEnabled()` yöntemini kullanarak gömülü FLV dosyası işaret noktalarını etkinleştirebilir ve devre dışı bırakabilirsiniz. Devre dışı bırakılan işaret noktaları, `cuePoint` olaylarını tetiklemez ve `seekToCuePoint()`, `seekToNextNavCuePoint()` veya `seekToPrevNavCuePoint()` yöntemleriyle çalışmaz. Ancak devre dışı bırakılan işaret noktalarını `findCuePoint()`, `findNearestCuePoint()` ve `findNextCuePointWithName()` yöntemleriyle bulabilirsiniz.

Gömülü bir FLV dosyası işaret noktasının etkinleştirilip etkinleştirilmediğini `isFLVCuePointEnabled()` yöntemini kullanarak test edebilirsiniz. Şu örnek, video oynatılmaya hazır olduğunda `point2` ve `point3` gömülü işaret noktalarını devre dışı bırakır. Ancak birinci `cuePoint` olayı gerçekleştiğinde, olay işleyici `point3` işaret noktasının devre dışı bırakılıp bırakılmadığını test eder ve devre dışı bırakılmışsa etkinleştirir.

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv";
my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    my_FLVPlybk.setFLVCuePointEnabled(false, "point2");
    my_FLVPlybk.setFLVCuePointEnabled(false, "point3");
}
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point time is: " + eventObject.info.time);
    trace("Cue point name is: " + eventObject.info.name);
    trace("Cue point type is: " + eventObject.info.type);
    if (my_FLVPlybk.isFLVCuePointEnabled("point2") == false) {
        my_FLVPlybk.setFLVCuePointEnabled(true, "point2");
    }
}
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.isFLVCuePointEnabled()` ve `FLVPlayback.setFLVCuePointEnabled()` yöntemlerine bakın.

ActionScript işaret noktasını kaldırma

`removeASCuePoint()` yöntemini kullanarak ActionScript işaret noktasını kaldırabilirsiniz. Şu örnek, `ASpt1` işaret noktası gerçekleştiğinde `ASpt2` işaret noktasını kaldırır:

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt2");//add 2nd Aspt
my_FLVPlybk.addEventListener(MouseEvent.CLICK, cp_listener);
function cp_listener(eventObject:MouseEvent):void {
    trace("Cue point name is: " + eventObject.info.name);
    if (eventObject.info.name == "ASpt1") {
        my_FLVPlybk.removeASCuePoint("ASpt2");
        trace("Removed cue point ASpt2");
    }
}
```

Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki FLVPlayback.removeASCuePoint() yöntemine bakın.

Çoklu video dosyalarını oynatma

Önceki video dosyası oynatma işlemini bitirdiğinde `source` özelliğine yeni bir URL yükleyerek sonraki video dosyalarını FLVPlayback örneğinde oynatabilirsiniz. Örneğin, şu ActionScript kodu, bir video dosyası oynatma işlemini bitirdiğinde gerçekleşen `complete` olayını dinler. Bu olay gerçekleştiğinde, kod `source` özelliğinde yeni bir video dosyasının adını ve konumunu ayarlar ve yeni videoyu oynatmak için `play()` yöntemini çağırır.

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/clouds.flv";
my_FLVPlybk.addEventListener(VideoEvent.COMPLETE, complete_listener);
// listen for complete event; play new FLV
function complete_listener(eventObject:VideoEvent):void {
    if (my_FLVPlybk.source == "http://www.helpexamples.com/flash/video/clouds.flv") {
        my_FLVPlybk.play("http://www.helpexamples.com/flash/video/water.flv");
    }
};
```

Çoklu video oynatıcılarını kullanma

Çoklu videolar oynatmak ve oynatılırken videolar arasında geçiş yapmak için FLVPlayback bileşeninin tek bir örneğinde birden çok video oynatıcı açabilirsiniz.

FLVPlayback bileşenini Sahne Alanı'na sürüklediğinizde başlangıç video oynatıcısını oluşturursunuz. Bileşen otomatik olarak başlangıç video oynatıcısına 0 numarasını atar ve bunu varsayılan oynatıcı yapar. Ek bir video oynatıcı oluşturmak için `activeVideoPlayerIndex` özelliğini yeni bir numaraya ayarlayın. `activeVideoPlayerIndex` özelliği ayarlandığında belirtilen video oynatıcı *etkin* video oynatıcı olur ve FLVPlayback sınıfının özellikleri ve yöntemleri tarafından etkilenir. Ancak `activeVideoPlayerIndex` özelliği ayarlandığında video oynatıcı görünür olmaz. Video oynatıcısı görünür hale getirmek için `visibleVideoPlayerIndex` özelliğini video oynatıcısının numarasına ayarlayın. Bu özelliklerin FLVPlayback sınıfının yöntemleri ve özellikleri ile nasıl etkileşimde oldukları hakkında daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki FLVPlayback.activeVideoPlayerIndex ve FLVPlayback.visibleVideoPlayerIndex özelliklerine bakın.

Şu ActionScript kodu, varsayılan video oynatıcıda video dosyasını oynatmak ve bu video dosyasına bir işaret noktası eklemek için `source` özelliğini yükler. `ready` olayı gerçekleştiğinde, olay işleyici `activeVideoPlayerIndex` özelliğini 1 numarasına ayarlayarak ikinci bir video oynatıcı açar. İkinci video oynatıcı için bir FLV dosyası ve işaret noktası belirtir ve sonra varsayılan oynatıcısı (0) tekrar etkin video oynatıcı haline getirir.

```
/**
    Requires:
    - FLVPlayback component on the Stage with an instance name of my_FLVPlybk
*/
// add a cue point to the default player
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/clouds.flv";
my_FLVPlybk.addASCuePoint(3, "1st_switch");
my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    // add a second video player and create a cue point for it
    my_FLVPlybk.activeVideoPlayerIndex = 1;
    my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
    my_FLVPlybk.addASCuePoint(3, "2nd_switch");
    my_FLVPlybk.activeVideoPlayerIndex = 0;
};
```

Bir FLV dosyası oynatılırken başka bir FLV dosyasına geçiş yapmak için ActionScript kodunuzda geçiş yapmanız gerekir. İşaret noktaları, bir cuePoint olayı kullanarak FLV dosyasında belirli noktalarda müdahale etmenize olanak sağlar. Şu kod, cuePoint olayı için bir dinleyici oluşturur ve etkin video oynatıcısını (0) duraklatan bir işleyici işlevi çağırır, ikinci oynatıcıya (1) geçiş yapar ve ikinci oynatıcının FLV dosyasını oynatır:

```
import fl.video.*;
// add listener for a cuePoint event
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
// add the handler function for the cuePoint event
function cp_listener(eventObject:MetadataEvent):void {
    // display the no. of the video player causing the event
    trace("Hit cuePoint event for player: " + eventObject.vp);
    // test for the video player and switch FLV files accordingly
    if (eventObject.vp == 0) {
        my_FLVPlybk.pause(); //pause the first FLV file
        my_FLVPlybk.activeVideoPlayerIndex = 1; // make the 2nd player active
        my_FLVPlybk.visibleVideoPlayerIndex = 1; // make the 2nd player visible
        my_FLVPlybk.play(); // begin playing the new player/FLV
    } else if (eventObject.vp == 1) {
        my_FLVPlybk.pause(); // pause the 2nd FLV
        my_FLVPlybk.activeVideoPlayerIndex = 0; // make the 1st player active
        my_FLVPlybk.visibleVideoPlayerIndex = 0; // make the 1st player visible
        my_FLVPlybk.play(); // begin playing the 1st player
    }
}
my_FLVPlybk.addEventListener(VideoEvent.COMPLETE, complete_listener);
function complete_listener(eventObject:VideoEvent):void {
    trace("Hit complete event for player: " + eventObject.vp);
    if (eventObject.vp == 0) {
        my_FLVPlybk.activeVideoPlayerIndex = 1;
        my_FLVPlybk.visibleVideoPlayerIndex = 1;
        my_FLVPlybk.play();
    } else {
        my_FLVPlybk.closeVideoPlayer(1);
    }
};
```

Yeni bir video oynatıcı oluşturduğunuzda, FLVPlayback örneği, özelliklerini varsayılan video oynatıcının `source`, `totalTime`, ve `isLive` özellikleri dışındaki değerine ayarlar, FLVPlayback örneği bu istisna özellikleri her zaman şu varsayılan değerlere ayarlar: sırayla boş dize, 0 ve `false`. Varsayılan video oynatıcı için `true` olan `autoPlay` özelliğini `false` değerine ayarlar. `cuePoints` özelliğinin, varsayılan video oynatıcısına yapılan sonraki yüklemeler de dahil herhangi bir etkisi yoktur.

Ses düzeyini, konumlandırmayı, boyutları, görünürlüğü ve kullanıcı arabirimi denetimlerini denetleyen yöntemler ve özellikler her zaman geneldir ve bunların davranışı `activeVideoPlayerIndex` özelliğinin ayarlanmasıyla etkilenmez. Bu yöntemler ve özellikler ve `activeVideoPlayerIndex` özelliğini ayarlamanın etkisi hakkında daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.activeVideoPlayerIndex` özelliğine bakın. Kalan özellikler ve yöntemler, `activeVideoPlayerIndex` özelliğinin değeriyle tanımlanan video oynatıcısını hedefler.

Ancak boyutları denetleyen özellikler ve yöntemler `visibleVideoPlayerIndex` özelliğiyle *etkileşimkurar*. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.addASCuePoint()` yöntemine bakın.

Flash Media Server'dan akış FLV dosyaları

Flash Media Server'dan FLV dosyalarını akışa alma gereksinimleri, Flash Video Streaming Service sağlayıcınızda yerel bant genişliği saptamasının kullanılabilir olup olmamasına bağlı olarak farklılık gösterir. Yerel bant genişliği saptama özelliği, bant genişliği saptaması özelliğinin akış sunucusunda yerleşik olduğunu ve daha iyi performans sağladığını belirtir. Yerel bant genişliği saptama özelliğinin kullanılabilir olup olmadığını sağlayıcınızla kontrol edin.

Flash Media Server da FLV dosyalarınıza erişmek için `rtmp://my_servername/my_application/stream.flv` gibi bir URL kullanın.

Flash Media Server ile canlı akış oynatırken `FLVPlayback.isLive` özelliğini `true` değerine ayarlamanız gerekir. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümündeki `FLVPlayback.isLive` özelliğine bakın.

Canlı akış ayarlama gibi, Flash Media Server yönetimi hakkında daha fazla bilgi için bkz. Flash Media Server belgeleri, www.adobe.com/support/documentation/en/flashmediaserver/.

Yerel bant genişliği saptaması özelliğinin olması veya olmaması durumu için

`NCManagerNative` sınıfı, bazı Flash Video Streaming Service sağlayıcılarının destekleyebileceği yerel bant genişliği saptamasını destekleyen `NCManager` ögesinin bir alt sınıfıdır. `NCManagerNative` ögesini kullandığınızda Flash Media Server da herhangi bir özel dosya gerekmez. Bant genişliği saptaması gerekmiyorsa, `NCManagerNative`, `main.asc` dosyası olmadan tüm Flash Media Server'ın sürümlerine bağlantı olanağı sağlar.

Varsayılan `NCManager` sınıfı yerine `NCManagerNative` ögesini kullanmak için, şu kod satırlarını FLA dosyanızın birinci karesine ekleyin:

```
import fl.video*;  
VideoPlayer.INCManagerClass = fl.video.NCManagerNative;
```

Yerel olmayan bant genişliği saptaması için

Flash Video Streaming Service sağlayıcınız yerel bant genişliği saptaması özelliği sunmuyorsa ancak bu özelliğe ihtiyacınız varsa, `main.asc` dosyasını Flash Media Server FLV uygulamanıza eklemeniz gerekir. `main.asc` dosyasını www.adobe.com/go/learn_fl_samples_tr adresinde çevrimiçi olarak bulabilirsiniz. `Samples\ComponentsAS2\FLVPlayback` dizininde `Samples.zip` dosyasında bulunur.

Akış FLV dosyalarına yönelik Flash Media Server ayarlamak için:

- 1 Flash Media Server uygulama klasörünüzde bir klasör oluşturun ve bu klasöre **my_application** gibi bir ad verin.
- 2 main.asc dosyasını my_application klasörüne kopyalayın.
- 3 my_application klasöründe **streams** adında bir klasör oluşturun.
- 4 streams klasöründe **_definst_** adında bir klasör oluşturun.
- 5 FLV dosyalarınızı **_definst_** klasörüne yerleştirin.

FLVPlayback bileşenini özelleştirme

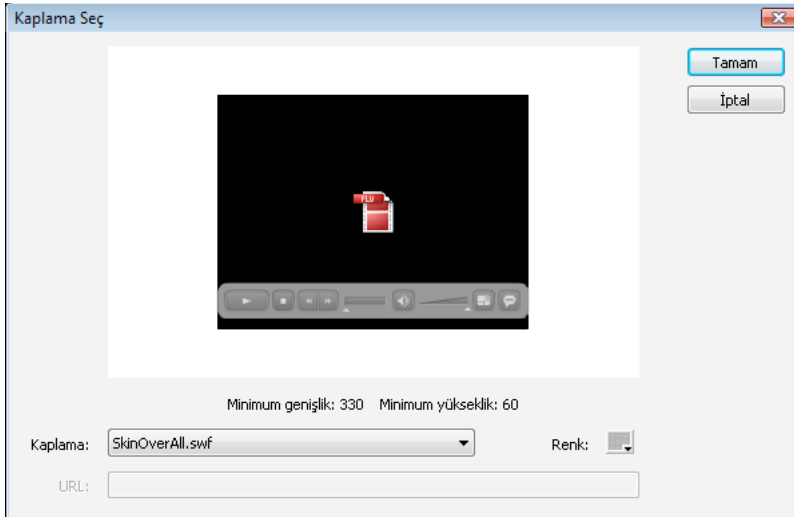
Bu bölümde, FLVPlayback bileşeninin nasıl özelleştirildiği açıklanmaktadır. Ancak diğer bileşenleri özelleştirmek için kullanılan yöntemlerin çoğu FLVPlayback bileşeniyle çalışmaz. FLVPlayback bileşenini özelleştirmek için yalnızca bu bölümde açıklanan teknikleri kullanın.

FLVPlayback bileşenini özelleştirmek için şu seçenekleriniz vardır: önceden tasarlanmış bir kaplama seçme, FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri için ayrı ayrı kaplama oluşturma veya yeni bir kaplama oluşturma. Kaplamanın davranışını değiştirmek için de FLVPlayback özelliklerini kullanabilirsiniz.

Not: *Kaplamanın FLVPlayback bileşeninizle çalışması için, kaplama SWF dosyanızı uygulama SWF dosyanızla birlikte web sunucusuna yüklemeniz gerekir.*

Önceden tasarlanmış bir kaplama seçme

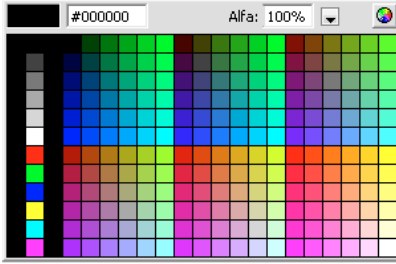
Özellik denetçisinde `skin` parametresinin `value` hücreliğini tıklatarak FLVPlayback bileşeni için bir kaplama seçebilirsiniz. Bir kaplama seçebileceğiniz veya kaplama SWF dosyasının konumunu belirten bir URL sağlayabileceğiniz aşağıdaki Kaplama Seç iletişim kutusunu açmak için büyüteç simgesini tıklatın.



FLVPlayback Kaplama Seç iletişim kutusu

Kaplama açılır menüsünde listelenen kaplamalar, /Flash Configuration/FLVPlayback Skins/ActionScript 3.0 Flash uygulama klasöründe bulunur. Kaplamalar oluşturularak ve klasöre SWF dosyasını yerleştirilerek bu iletişim kutusu için yeni kaplamaları kullanılabilir duruma getirebilirsiniz. Açılır menüde .swf uzantısıyla ad görüntülenir. Kaplama kümesi oluşturma hakkında daha fazla bilgi için bkz. “Yeni bir kaplama oluşturma” sayfa 159.

Geliştirme sırasında veya çalışma zamanında ActionScript ile skin parametresini ayarlayıp `skin` özelliğini ayarlayarak atadığınız kaplamalar için, kaplama seçiminden bağımsız olarak renk ve alfa (saydamlık) değerleri atayabilirsiniz. Geliştirme sırasında renk ve alfa değerleri atamak için, burada gösterildiği gibi, Kaplama Seç iletişim kutusunda Renk seçicisini açın.



Kaplama Seç iletişim kutusunda Renk seçici

Rengi seçmek için paneldeki bir renk örneğini tıklattın veya metin kutusuna rengin sayısal değerini girin. Alfa değerini seçmek için kaydırıcıyı kullanın veya Alfa metin kutusuna bir yüzde değeri yazın.

Çalışma zamanında renk ve alfa değerleri atamak için, `skinBackgroundColor` ve `skinBackgroundAlpha` özelliklerini ayarlayın. `skinBackgroundColor` özelliğini bir `0xRRGGBB` (kırmızı, yeşil, mavi) değerine ayarlayın. `skinBackgroundAlpha` özelliğini 0,0 ile 1,0 arasında bir sayıya ayarlayın. Şu örnek, `skinBackgroundColor` ögesini `0xFF0000` (kırmızı) değerine ve `skinBackgroundAlpha` ögesini 0,5 değerine ayarlar.

```
my_FLVPlayback.skinBackgroundColor = 0xFF0000;  
my_FLVPlayback.skinBackgroundAlpha = .5;
```

Varsayılan değerler, kullanıcı tarafınsan seçilen son değerlerdir.

FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerini kullanarak FLVPlayback bileşeni için kaplama oluşturmak isterseniz, açılır menüden Yok seçeneğini belirleyin.

FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri için ayrı ayrı kaplama oluşturma

FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri, FLA dosyası içindeki FLVPlayback denetimlerinin görünümünü özelleştirmenize olanak verir ve web sayfanızı özitlediğinizde sonuçları görmenizi sağlar. Ancak bu bileşenler ölçeklenecek şekilde tasarlanmamıştır. Film klipini ve içeriklerini belirli bir boyuta düzenlemeniz gerekir. Bu nedenle, genellikle `scaleMode` ögesini `exactFit` değerine ayarlayarak Sahne Alanı'ndaki FLVPlayback bileşeninin istediğiniz boyutta olmasını sağlamak en iyisidir.

Başlamak için, istediğiniz FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerini Bileşenler panelinden sürükleyin ve Sahne Alanı'nda bulunmasını istediğiniz yere yerleştirip bunlara örnek adları verin.

Bu bileşenler herhangi bir ActionScript olmadan çalışabilir. Bunları FLVPlayback bileşeniyle aynı zaman çizelgesine ve kareye koyarsanız ve bileşende herhangi bir kaplama kümesi yer almıyorsa, FLVPlayback bileşeni bunlara otomatik olarak bağlanır. Sahne Alanı'nda birden çok FLVPlayback bileşeni bulunuyorsa veya özel denetim ve FLVPlayback örneği aynı Zaman Çizelgesi'nde bulunmuyorsa, Eylem gerekir.

Bileşenleriniz Sahne Alanı'ndayken bunları herhangi bir sembol gibi düzenleyebilirsiniz. Bileşenleri açtıktan sonra, her birinin diğerlerinden biraz daha farklı şekilde ayarlandığını görebilirsiniz.

Button bileşenleri

Button bileşenleri de benzer bir yapıya sahiptir. Bu düğmeler arasında BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton ve StopButton yer alır. Bunların çoğu, placeholder_mc örnek adıyla Kare 1'de tek bir film klipi içerir. Bu genellikle düğme için normal bir durum örneğidir ancak böyle olmak zorunda değildir. Kare 2'de, her görüntüleme durumu için Sahne Alanı'nda dört film klipi vardır: normal, üzerinde, basılmış ve devre dışı. (Çalışma zamanında bileşen asla gerçekten Kare 2'ye gitmez; bu film klipleri düzenleme işlemi daha kolay hale getirmek ve Sembol Özellikleri iletişim kutusunda Birinci karede Dışa Aktar onay kutusu seçilmeden SWF dosyasına yüklenmeye zorlamak için buraya yerleştirilir. Ancak ActionScript için Dışa Aktar seçeneğini seçmeniz gerekir.)

Düğme için kaplama oluşturmak üzere bu film kliplerinin her birini düzenlemeniz yeterlidir. Bunların görünülerinin yanı sıra boyutlarını da değiştirebilirsiniz.

Bazı ActionScript'ler genellikle Kare 1'de görüntülenir. Bu komut dosyasını değiştirmeniz gerekir. Bu, Kare 1'de oynatma kafasını durdurur ve hangi durum için hangi film klipinin kullanılacağını belirtir.

PlayPauseButton, MuteButton, FullScreenButton ve CaptionButton düğmeleri

PlayPauseButton, MuteButton, FullScreenButton ve CaptionButton düğmeleri diğer düğmelerden farklı şekilde ayarlanmıştır; bunların iki katmana sahip tek bir karesi vardır ve herhangi bir komut dosyası yoktur. Bu karede birbirinin üzerinde bulunan iki düğme vardır—PlayPauseButton durumunda Oynat/Çal ve Duraklat düğmesi; MuteButton durumunda tam ekran açık ve tam ekran kapalı düğmesi; CaptionButton durumda ise resim yazısı etkin ve resim yazısı devre dışı düğmesi. Bu düğmeler için kaplama oluşturmak üzere, [“FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri için ayrı ayrı kaplama oluşturma”](#) sayfa 154 bölümünde açıklandığı gibi bu iki dahili düğmenin her biri için kaplama oluşturun; bunun dışında bir eylem gerekmez.

CaptionButton ögesi FLVPlaybackCaptioning bileşeni için olup FLVPlayback bileşenine değil, bu bileşene eklenmelidir.

BackButton ve ForwardButton düğmeleri

BackButton ve ForwardButton düğmeleri de diğer düğmelerden farklı şekilde ayarlanmıştır. Kare 2'de, düğmelerden biri veya her ikisinin etrafında kare olarak kullanabileceğiniz fazladan film klipleri bulunur. Bu film klipleri gerekli olmayıp özel bir yeteneğe sahip değildir; yalnızca kolaylık sunmak üzere sağlanmıştır. Bunları kullanmak için Kütüphane panelinizden Sahne Alanı'na sürükleyin ve istediğiniz yere yerleştirin. Bunları istemiyorsanız kullanmayın veya Kütüphane panelinizden silin.

Tüm düğmelerin görünümünü aynı anda değiştirebilmeniz için, sağlanan düğmelerin çoğu ortak bir film klipi kümesini esas almıştır. Bu yeteneği kullanabilir veya bu ortak klipleri değiştirip her düğmeye farklı bir görünüm verebilirsiniz.

BufferingBar bileşeni

Buffering bar bileşeni basittir: Bileşen arabelleğe alma durumuna girdiğinde görünür hale gelen bir animasyon içerir ve konfigüre edilmesi için özel bir ActionScript gerektirmez. Varsayılan olarak bu, “şeritli direk” etkisi vermek için üzerindeki dikdörtgen maskeyle soldan sağa taşınan çizgili bir çubuktur ancak bu konfigürasyonla ilgili özel bir şey yoktur.

Çalışma zamanında ölçeklenmesi gerektiğinden kaplama SWF dosyalarındaki arabelleğe alma çubukları 9 dilimli ölçekleme kullansa da, BufferingBar FLV Özel Kullanıcı Arabirimi Bileşeni yuvalanmış film klipleri içerdiğinden 9 dilimli ölçeklemeyi kullanmaz ve *kullanamaz*. BufferingBar ögesini daha geniş veya uzun hale getirmek istiyorsanız, ölçeklemek yerine bu ögenin içeriklerini değiştirebilirsiniz.

SeekBar ve VolumeBar bileşenleri

SeekBar ve VolumeBar bileşenleri farklı işlevlere sahip olsa da, birbirine benzer. Her birinin tutamacı vardır, her biri aynı tutamaç izleme mekanizmalarını kullanır ve ilerleme ve tamlık düzeyini izlemek üzere içinde yuvalanmış klipler için destek içerir.

FLVPlayback bileşenindeki ActionScript kodunun, SeekBar veya VolumeBar bileşeninizin kayıt noktasının (ayrıca *origin point* veya *zero point*) içeriğin sol üst köşesinde olduğunu varsaydığı birçok yer vardır, bu nedenle bu kuralın korunması önemlidir. Aksi takdirde, tutamaçlarla ve ilerleme ve tamlık film klipleriyle ilgili sorun yaşayabilirsiniz.

Çalışma zamanında ölçeklenmesi gerektiğinden kaplama SWF dosyalarındaki arama çubukları 9 dilimli ölçekleme kullansa da, SeekBar FLV Özel Kullanıcı Arabirimi bileşeni yuvalanmış film klipleri içerdiğinden 9 dilimli ölçeklemeyi kullanmaz ve *kullanamaz*. SeekBar öğesini daha geniş veya uzun hale getirmek istiyorsanız, ölçeklemek yerine bu öğenin içeriklerini değiştirebilirsiniz.

Tutamaç

Tutamaç film klbinin bir örneği, Kare 2'dedir. (BackButton ve ForwardButton bileşenlerinde olduğu gibi, bileşen asla gerçekten Kare 2'ye gitmez; bu film klipleri düzenleme işlemini daha kolay hale getirmek ve Sembol Özellikleri iletişim kutusunda Birinci Karede Dışa Aktar onay kutusu seçilmeden SWF dosyasına yüklenmeye zorlamak için buraya yerleştirilir. Ancak ActionScript için Dışa Aktar seçeneğini seçmeniz gerekir.

Tutamaç film klbinin, alfa değeri 0 olarak ayarlı şekilde arka planında bir dikdörtgen bulunduğunu fark edebilirsiniz. Bu dikdörtgen, tutamacın vuruş alanının boyutunu artırarak vurma durumuna benzer şekilde düşmenin görünümü değiştirilmeden yakalanmasını kolaylaştırır. Tutamaç çalışma zamanında dinamik olarak oluşturulduğundan bunun bir düğme değil, film klbi olması gerekir. Alfa değeri 0 olarak ayarlı olan bu dikdörtgen başka bir nedenle gerekli değildir, genellikle tutamacın içine istediğiniz görüntüyü yerleştirebilirsiniz. Ancak bu en çok kayıt noktasını tutamaç film klbinin ortasında yatay olarak ortalanmış şekilde tutmak için işe yarar.

Şu ActionScript kodu, tutamacı yönetmek için SeekBar bileşeninin Kare 1'inde bulunur:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

Kare 2'deki içerik nedeniyle `stop()` işlevine çağrı gerekir.

İkinci satır, tutamaç olarak hangi sembolün kullanılacağını belirtir ve Kare 2'deki tutamaç film klbi örneğini düzenlerseniz bunu değiştirmeniz gerekmez. Çalışma zamanında FLVPlayback bileşeni, Bar bileşeni örneğinin eş ögesi olarak Sahne Alanı'nda belirtilen film klbinin bir örneğini oluşturur, başka bir deyişle bunlar aynı üst film klbine sahiptir. Bu nedenle, çubuğunuz kök düzeyindeyse tutamacınızın da kök düzeyinde olması gerekir.

`handleLeftMargin` değişkeni, tutamacın orijinal konumunu (%0) belirler ve `handleRightMargin` değişkeni de tutamacın sonunda nerede olduğunu (%100) belirtir. Sayılar, çubuk denetiminin sol ve sağ uçlarından uzaklık sağlar; burada pozitif sayılar, çubuk içindeki sınırları ifade ederken, negatif sayılar da çubuğun dışındaki sınırları ifade eder. Bu uzaklıklar, kayıt noktasını esas alarak tutamacın gidebileceği yeri belirtir. Kayıt noktasını tutamacın ortasına koyarsanız, tutamacın en sol ve sağ kenarları, kenar boşluklarını geçer Arama çubuğu film klbinin düzgün çalışması için içeriğinin sol üst köşesi olarak kayıt noktasının bulunması gerekir.

`handleY` değişkeni, çubuk örneğine göre tutamacın `y` konumunu belirler. Bu, her film klbinin kayıt noktalarını esas alır. Örnek tutamacındaki kayıt noktası, görünmez vurma durumu dikdörtgenini dikkate almadan görünür bölüme göre üçgeni yerleştirmek için üçgenin en üst kısmında bulunur. Ayrıca, çubuk film klbinin düzgün çalışması için içeriğinin sol üst köşesi olarak kayıt noktasını tutması gerekir.

Bu nedenle, örneğin bu sınırlarla bir çubuk denetimi (100, 100) değerine ayarlanırsa ve 100 piksel genişliğinde olursa, tutamaç yatay olarak 102 ile 198 arasında değişiklik gösterebilir ve dikey olarak 111 değerinde kalır.

`handleLeftMargin` ve `handleRightMargin` öğelerini -2 değerine ve `handleY` öğesini -11 değerine getirirseniz, tutamaç yatay olarak 98 ile 202 arasında değişiklik gösterebilir ve dikey olarak 89 değerinde kalır.

İlerleme ve tamlık film klipleri

SeekBar bileşeni bir *ilerleme* film kliline, VolumeBar bileşeni de bir *tamlık* film kliline sahiptir, ancak uygulamada herhangi bir SeekBar veya VolumeBar öğesi bu film kliplerinden herhangi birine veya her ikisine sahip olabilir veya hiçbirini içermeyebilir. Bunlar yapısal olarak aynıdır ve benzer şekilde davranır ancak farklı değerleri izler. İlerleme film klipi, FLV dosyası indirildikçe dolar (FMS'den akışa alınırken her zaman dolu olduğundan yalnızca HTTP indirilmesi için kullanışlıdır) ve tamlık film klipi de tutamaç soldan sağa hareket ettikçe dolar.

FLVPlayback bileşeni, belirli bir örnek adını arayarak bu film klipi örneklerini bulur, bu nedenle ilerleme film klipi örneğinizin üst öğesi olarak çubuk film klibinize ve `progress_mc` örnek adına sahip olması gerekir. Tamlık film klipi örneğinizin `fullness_mc` örnek adına sahip olması gerekir.

İçeride `fill_mc` film klipi yuvalanmış veya yuvalanmamış şekilde ilerleme ve tamlık film kliplerini ayarlayabilirsiniz. VolumeBar `fullness_mc` film klipi, `fill_mc` ile yöntemi gösterir ve SeekBar `progress_mc` film klipi de `fill_mc` film klipi *olmadan* yöntemi gösterir.

İçeride `fill_mc` film klibinin yuvalanmış olduğu yöntem, görünümü deforme etmeden ölçeklenemeyen bir dolgu istediğinizde kullanışlıdır.

VolumeBar `fullness_mc` film klisinde, yuvalanmış `fill_mc` film klipi örneği maskelenir. Film klibini oluşturduğunuzda maskeleyebilirsiniz veya çalışma zamanında dinamik olarak bir maske oluşturulur. Film klibiyle maskelerseniz, örneğe **mask_mc** adını verin ve `fill_mc` öğesi %100 olduğundaki gibi görüntülenecek şekilde örneği ayarlayın. `fill_mc` öğesini maskeleyemezseniz, dinamik olarak oluşturulan maske dikdörtgen olur ve %100 durumdaki `fill_mc` ile aynı boyutta olur.

`fill_mc.slideReveal` değerinin `true` veya `false` olmasına bağlı olarak `fill_mc` film klipi şu iki yoldan biri kullanılarak maskeyle gösterilir.

`fill_mc.slideReveal` öğesi `true` değerinde olursa, `fill_mc` öğesi bunu maske üzerinden göstermek için soldan sağa taşınır. %0 değerindeyken bu en soldadır, böylece hiçbir bölümü maske üzerinden görünmez. Yüzde değeri arttıkça, bu öğe de %100 değerine kadar sağa hareket eder, Sahne Alanı'nda oluşturulduğu konuma geri döner.

`fill_mc.slideReveal` öğesi `false` değerinde veya tanımsız (varsayılan davranış) olursa, `fill_mc` öğesinin daha çok kısmını göstermek için maske soldan sağa yeniden boyutlandırılır. %0 değerindeyken, maske yatay olarak 05 değerine ölçeklenir ve yüzde değeri arttıkça `scaleX` değeri de %100'e kadar artarak `fill_mc` öğesinin tümünü gösterir. `mask_mc` öğesi oluşturulduğunda ölçeklenmiş olabileceğinden, mutlaka `scaleX = 100` olmak zorunda değildir.

`fill_mc` olmadan gerçekleşen yöntem, `fill_mc` ile gerçekleşen yöntemden daha basittir ancak dolguyu yatay olarak deforme eder. Deformasyon olmasını istemiyorsanız `fill_mc` öğesini kullanmanız gerekir. SeekBar `progress_mc` bu yöntemi göstermektedir.

İlerleme veya tamlık film klipi, yüzdeyi esas alarak yatay olarak ölçeklenir. %0 değerinde, örneğin `scaleX` değeri sıfır olarak ayarlanarak örnek görünür hale getirilir. Yüzde değeri arttıkça, `scaleX` değeri de %100'e kadar ayarlanır, bu noktada klip, oluşturulduğu zaman Sahne Alanı'ndaki boyutuyla aynı boyuttadır. Klip örneği de oluşturulduğunda ölçeklenmiş olabileceğinden, mutlaka `scaleX = 100` olmak zorunda değildir.

FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerinizi bağlama

Özel Kullanıcı Arabirimi bileşenlerini FLVPlayback bileşeniyle aynı Zaman Çizelgesi'ne ve kareye koyarsanız ve `skin` özelliğini ayarlamadıysanız, FLVPlayback herhangi bir ActionScript gerekmeden bunlara otomatik olarak bağlanır.

Sahne Alanı'nda birden çok FLVPlayback bileşeni varsa veya özel denetim ve FLVPlayback aynı Zaman Çizelgesi'nde değilse, Özel Kullanıcı Arabirimi bileşenlerinizi FLVPlayback bileşeninizin örneğine bağlamak için ActionScript kodu yazmanız gerekir. Öncelikle FLVPlayback örneğine bir ad atayın ve sonra FLV Oynatma Özel Kullanıcı Arabirimi bileşeni örneklerini karşılık gelen FLVPlayback özelliklerine atamak için ActionScript kullanın. Şu örnekte, FLVPlayback örneği `my_FLVPlybk` olur, FLVPlayback özellik adları noktalardan (.) sonra gelir ve FLV Oynatma Özel Kullanıcı Arabirimi denetimi örnekleri de eşittir (=) işaretlerinin sağındadır:

```
//FLVPlayback instance = my_FLVPlybk
my_FLVPlybk.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlybk.pauseButton = pausebtn;
my_FLVPlybk.playPauseButton = playpausebtn;
my_FLVPlybk.stopButton = stopbtn;
my_FLVPlybk.muteButton = mutebtn;
my_FLVPlybk.backButton = backbtn;
my_FLVPlybk.forwardButton = forbtn;
my_FLVPlybk.volumeBar = volbar;
my_FLVPlybk.seekBar = seekbar;
my_FLVPlybk.bufferingBar = bufbar;
```

Şu adımlar uygulandığında StopButton, PlayPauseButton, MuteButton ve SeekBar denetimleri oluşturulur:

- 1 FLVPlayback bileşenini Sahne Alanı'na sürükleyin ve bu bileşene `my_FLVPlybk` örnek adını verin.
- 2 Bileşen denetçisi üzerinden `source` parametresini <http://www.helpexamples.com/flash/video/cuepoints.flv> olarak ayarlayın.
- 3 Skin parametresini None olarak ayarlayın.
- 4 StopButton, PlayPauseButton ve MuteButton öğesini Sahne Alanı'na sürükleyip FLVPlayback örneğinin üzerine yerleştirin, ardından sola dikey olarak yığın. Özellik denetçisinde düğmelerin her birine bir örnek adı verin (örn. `my_stopbtt`, `my_plypausbtt` ve `my_mutebtt`).
- 5 Kütüphane panelinde FLVPlayback Kaplamaları klasörünü ve sonra bu klasörün aşağısındaki SquareButton klasörünü açın.
- 6 SquareBgDown film klibini seçin ve Sahne Alanı'nda açmak için çift tıklayın.
- 7 Sağ tıklayın (Windows) veya Control tuşunu basılı tutarak tıklayın (Macintosh), menüden Tümünü Seç öğesini seçin ve sembolü silin.
- 8 Oval aracını seçin, aynı konuma bir oval çizin ve dolguyu mavi (`#0033FF`) olarak ayarlayın.
- 9 Özellik denetçisinde genişliği (W:) **40** değerine ve yüksekliği de (H:) **20** değerine ayarlayın. x koordinatını (X:) **0,0** değerine ve y koordinatını (Y:) **0,0** değerine ayarlayın.
- 10 SquareBgNormal için 6 ile 8 arasındaki adımları yineleyin, ancak dolguyu sarı (`#FFFF00`) olarak ayarlayın.
- 11 SquareBgOver için 6 ile 8 arasındaki adımları yineleyin, ancak dolguyu yeşil (`#006600`) olarak ayarlayın.
- 12 Düğmeler içindeki çeşitli sembol simgelerinin film kliplerini düzenleyin (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon ve StopIcon). Bu film kliplerini FLV Playback Skins/*Etiket* Button/Assets konumundaki Kütüphane panelinde bulabilirsiniz; burada *Etiket* düğmenin adıdır, örn. Oynat/Çal, Duraklat, vb. Her biri için şu adımları uygulayın:
 - a Tümünü Seç seçeneğini belirleyin.
 - b Rengi kırmızı (`#FF0000`) olarak ayarlayın.
 - c %300 ölçekleyin.
 - d Her düğme durumunda simgenin yatay yerleşimini değiştirmek için içeriğin X: konumunu **7.0** değerine getirin.

***Not:** Konumu bu şekilde değiştirerek her düğme durumunun açılmasını ve simge film klipi örneğinin taşınmasını önlersiniz.*

- 13 Sahne 1, Kare 1'e geri dönmek için Zaman Çizelgesi'nin yukarıdaki mavi Geri okunu tıklatın.
- 14 SeekBar bileşenini Sahne Alanı'na sürükleyip FLVPlayback örneğinin sağ alt köşesine yerleştirin.
- 15 Kütüphane panelinde, SeekBar öğesini Sahne Alanı'nda açmak için çift tıklatın.
- 16 %400 ölçekleyin.
- 17 Anahattı seçin ve rengi kırmızı (#FF0000) olarak ayarlayın.
- 18 FLVPlayback Skins/Seek Bar klasöründe SeekBarProgress öğesini çift tıklatın ve rengi sarı (#FFFF00) olarak ayarlayın.
- 19 FLVPlayback Skins/Seek Bar klasöründe SeekBarHandle öğesini çift tıklatın ve rengi kırmızı (#FF0000) olarak ayarlayın.
- 20 Sahne 1, Kare 1'e geri dönmek için Zaman Çizelgesi'nin yukarıdaki mavi Geri okunu tıklatın.
- 21 Sahne Alanı'nda SeekBar örneğini seçin ve bu örneğe **my_seekbar** örnek adını verin.
- 22 Zaman Çizelgesi'nin Kare 1'indeki Eylemler panelinde, video sınıfları için bir içe aktarma ifadesi ekleyin ve aşağıdaki örnekte gösterildiği gibi, karşılık gelen FLVPlayback özelliklerine düğme ve arama çubuğu adlarını atayın:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbtttn;
my_FLVPlybk.playPauseButton = my_plypausbtttn;
my_FLVPlybk.muteButton = my_mutebtttn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Filmi test etmek için Control+Enter tuşlarına basın

Yeni bir kaplama oluşturma

Kaplama SWF dosyası oluşturmanın en iyi yolu, Flash uygulamasıyla birlikte gelen kaplama dosyalarından birini kopyalayıp başlangıç noktası olarak kullanmaktır. Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/ yolundaki Flash uygulama klasöründe bu kaplamaların FLA dosyalarını bulabilirsiniz. Biten kaplama SWF dosyanızı Kaplama Seç iletişim kutusunda bir seçenek olarak kullanılabilir duruma getirmek için, kaplama dosyasını Configuration/FLVPlayback Skins/ActionScript 3.0 klasöründeki Flash uygulama klasörüne veya kullanıcının yerel Configuration/FLVPlayback Skins/ActionScript 3.0 klasörüne koyun.

Bir kaplamanın rengini kaplama seçiminden bağımsız olarak ayarlayabildiğinizden, rengi değiştirmek için FLA dosyasını düzenlemeniz gerekmez. Belirli renge sahip bir kaplama oluşturursanız ve bu kaplamanın Kaplama Seç iletişim kutusunda düzenlenebilir olmasını istemiyorsanız, kaplama FLA dosyasının ActionScript kodunda `this.border_mc.colorMe = false;` ayarını yapın. Kaplamanın rengini ayarlama hakkında bilgi almak için bkz. “Önceden tasarlanmış bir kaplama seçme” sayfa 153.

Yüklenen kaplama FLA dosyalarına bakıldığında Sahne Alanı'ndaki belirli öğeler gereksizmiş gibi görünebilir ancak bunların çoğu kılavuz katmanlarına koyulur. 9 ölçeğini kullanan canlı önizleme sayesinde, SWF dosyasında gerçekten ne görüntüleneceğini hızlı şekilde görebilirsiniz.

Şu bölümler, SeekBar, BufferingBar ve VolumeBar film kliplerinin daha karmaşık özelleştirmelerini ve değişikliklerini ele alır.

Kaplama mizanpajını kullanma

Bir Flash kaplama FLA dosyasını açtığınızda, kaplamanın film kliplerinin ana Zaman Çizelgesi'nde düzenlendiğini görürsünüz. Aynı karede bulunan bu klipler ve ActionScript kodu, denetimlerin çalışma zamanında nasıl düzenleneceğini tanımlar.

Mizanpaj katmanı, kaplamanın çalışma zamanında görüldüğü haline çok yakından benzese de, bu katmanın içerikleri çalışma zamanında görünür değildir. Bu içerikler yalnızca denetimlerin yerleştirileceği yeri hesaplamak için kullanılır. Sahne Alanı'ndaki diğer denetimler çalışma zamanında kullanılır.

Mizanpaj katmanının içinde, video_mc adındaki FLVPlayback bileşeninin yer tutucusu bulunur. Diğer tüm denetimler video_mc ögesine göre düzenlenir. Flash FLA dosyalarından biriyle işe başlar ve denetimlerin boyutunu değiştirirseniz, bu yer tutucu klipleri taşıyarak mizanpajı sabitleyebilirsiniz.

Yer tutucu kliplerinin her biri belirli bir örnek adına sahiptir. Yer tutucu kliplerinin adları şunlardır: playpause_mc, play_mc, pause_mc, stop_mc, captionToggle_mc, fullScreenToggle_mc, back_mc, bufferingBar_mc, bufferingBarFill_mc, seekBar_mc, seekBarHandle_mc, seekBarProgress_mc, volumeMute_mc, volumeBar_mc ve volumeBarHandle_mc. Bir kaplama rengi seçtiğinizde yeniden renklendirilen parça border_mc olarak adlandırılır.

Denetim için hangi klabin kullanıldığı önemli değildir. Genelde düğmeler için normal durum klipi kullanılır. Diğer denetimler için bu denetimin klipi kullanılır ancak bu yalnızca kolaylık sağlamak içindir. Tüm önemli olan, x (yatay) ve y (dikey) konumu ve yer tutucusunun yüksekliği ve genişliğidir.

Standart denetimlerin yanı sıra istediğiniz kadar ek klabiniz de olabilir. Bu kliplere yönelik tek gereksinim, Bağlantı iletişim kutusunda kütüphane sembollerinin ActionScript için Dış Aktar seçeneğinin işaretlenmiş olmasıdır. Mizanpaj katmanındaki özel klipler, yukarıda listelenen ayrılmış örnek adları dışında herhangi bir örnek adına sahip olabilir. Örnek adı yalnızca mizanpajı belirlemek için kliplerde ActionScript ayarlamak için gerekir.

border_mc klipi özeldir. FlvPlayback.skinAutoHide özelliğini true değerine ayarlarsanız, fare border_mc klibinin üzerindeyken kaplama gösterilir. Video oynatıcı sınırları dışında görüntülenen kaplamalar için bu önemlidir. skinAutoHide özelliği hakkında bilgi almak için bkz. "[Kaplama davranışını değiştirme](#)" sayfa 163.

Flash FLA dosyalarında krom için ve İleri ve Geri düğmelerinin etrafındaki kenarlık için border_mc kullanılır.

border_mc ayrıca skinBackgroundAlpha ve skinBackgroundColor özellikleri tarafından alfası ve rengi değiştirilen kaplama bölümüdür. Özelleştirilebilir renk ve alfaya olanak sağlamak için, kaplama FLA dosyasındaki ActionScript şunları içermelidir:

```
border_mc.colorMe = true;
```

ActionScript ve kaplama mizanpajı

Şu ActionScript kodu genellikle tüm denetimler için geçerlidir. Bazı denetimler, ek davranış belirten belirli bir ActionScript içerir ve bu söz konusu denetimle ilgili bölümde açıklanmıştır.

Başlangıç ActionScript'i her bileşenin durumlarının birer birer sınıf adlarını belirten büyük bir bölümdür. Bu sınıf adlarının tümünü SkinOverAll fla dosyasında görebilirsiniz. Duraklat ve Oynat/Çal düğmeleri için kod şöyle görünür, örneğin:

```
this.pauseButtonDisabledState = "fl.video.skin.PauseButtonDisabled";  
this.pauseButtonDownState = "fl.video.skin.PauseButtonDown";  
this.pauseButtonNormalState = "fl.video.skin.PauseButtonNormal";  
this.pauseButtonOverState = "fl.video.skin.PauseButtonOver";  
this.playButtonDisabledState = "fl.video.skin.PlayButtonDisabled";  
this.playButtonDownState = "fl.video.skin.PlayButtonDown";  
this.playButtonNormalState = "fl.video.skin.PlayButtonNormal";  
this.playButtonOverState = "fl.video.skin.PlayButtonOver";
```

Sınıf adlarının gerçek harici sınıf dosyaları yoktur; kütüphanedeki tüm film klipleri için Bağlantı iletişim kutusunda belirtilirler.

ActionScript 2.0 bileşeninde, Sahne Alanı'nda gerçekten çalışma zamanında kullanılan film klipleri vardı. ActionScript 3.0 bileşeninde bu film klipleri yine FLA dosyasındadır ancak burada amaç yalnızca düzenleme işlemini kolaylaştırmaktır. Şimdi bunların tümü kılavuz katmanlarında olup dışa aktarılamaz. Kütüphanedeki tüm kaplama varlıkları, birinci karede dışa aktarılacak şekilde ayarlanır ve örneğin bunun gibi bir kodla dinamik olarak oluşturulur:

```
new fl.video.skin.PauseButtonDisabled();
```

Bu bölümden sonra, kaplamanın minimum genişliğini ve yüksekliğini tanımlayan ActionScript kodu yer alır. Kaplama Seç iletişim kutusu bu değerleri gösterir ve bunlar kaplamanın minimum boyutun altında bir değere ölçeklenmesini önlemek için çalışma zamanında kullanılır. Minimum bir boyut belirtmek istemezseniz, bunu tanımsız olarak veya sıfır değerine ya da sıfırdan düşük bir değere ayarlı şekilde bırakın.

```
// minimum width and height of video recommended to use this skin,  
// leave as undefined or <= 0 if there is no minimum  
this.minWidth = 270;  
this.minHeight = 60;
```

Her yer tutucu, kendisine uygulanmış şu özelliklere sahiptir:

Özellik	Açıklama
anchorLeft	Boolean. Denetimi, FLVPlayback örneğinin sol tarafına göre konumlandırır. anchorRight açıkça true değerine ayarlanmazsa varsayılan olarak true değerini alır ve sonra varsayılan olarak false değerine geçer.
anchorRight	Boolean. Denetimi, FLVPlayback örneğinin sağ tarafına göre konumlandırır. Varsayılan olarak false değerini alır.
anchorBottom	Boolean. Denetimi, FLVPlayback örneğinin alt tarafına göre konumlandırır. anchorTop ögesi açıkça true değerine ayarlanmadığı sürece varsayılan olarak true değerini alır ve sonra varsayılan olarak false değerine geçer.
anchorTop	Boolean. Denetimi, FLVPlayback örneğinin üst tarafına göre konumlandırır. Varsayılan olarak false değerini alır.

Hem anchorLeft hem de anchorRight özellikleri true değerinde olursa, çalışma zamanında denetim yatay olarak ölçeklenir. Hem anchorTop hem de anchorBottom özellikleri true değerinde olursa, çalışma zamanında denetim dikey olarak ölçeklenir.

Bu özelliklerin etkisini görmek için, bunların Flash kaplamalarında nasıl kullanıldığına bakın. Ölçeklenenler yalnızca BufferingBar ve SeekBar denetimleridir, bunlar birbirinin üzerine yerleştirilir ve her ikisinin de anchorLeft ve anchorRight özellikleri true değerine ayarlanmıştır. BufferingBar ve SeekBar öğelerinin solundaki tüm denetimlerin anchorLeft ögesi true değerine ve sağlarındaki tüm denetimlerin anchorRight ögesi de true değerine ayarlanmıştır. Tüm denetimlerin anchorBottom ögesi true değerine ayarlanmıştır.

Denetimlerin altta değil de üstte bulunduğu bir kaplama oluşturmak için Mizanpaj katmanında film kliplerini düzenlemeyi deneyebilirsiniz. Tek yapmanız gereken, video_mc ögesine göre denetimleri üste taşımak ve tüm denetimler için anchorTop ögesini true değerine eşit olacak şekilde ayarlamaktır.

Arabelleğe alma çubuğu

Arabelleğe alma çubuğu iki film kliline sahiptir: bufferingBar_mc ve bufferingBarFill_mc. Göreceli konumlandırma korunduğundan, her klbin Sahne Alanı'nda diğer klbe göre konumu önemlidir. Bileşen bufferingBarFill_mc ögesini değil, bufferingBar_mc ögesini ölçeklediğinden, arabelleğe alma çubuğu iki ayrı klip kullanır.

bufferingBar_mc klibine 9 dilimli ölçekleme uygulanmıştır, bu nedenle ölçeklendiğinde kenarlıklar deforme edilmez. bufferingBarFill_mc klipi çok geniştir, bu nedenle ölçeklemeye gerek kalmayacak şekilde yeterince geniş olur. Yalnızca uzatılan bufferingBar_mc öğesinin yukarıdaki kısmı göstermek için çalışma zamanında otomatik olarak maskelenir. Varsayılan olarak, maskenin gerçek boyutları bufferingBar_mc ve bufferingBarFill_mc öğelerinin x (yatay) konumları arasındaki fark esas alınarak bufferingBar_mc içinde sol ve sağda eşit kenar boşluğunu korur. ActionScript koduyla konumlandırmayı özelleştirebilirsiniz.

Arabelleğe alma çubuğunun ölçeklenmesi gerekmiyorsa veya 9 dilimli ölçekleme kullanmıyorsa, bunu FLV Oynatma Özel Kullanıcı Arabirimi BufferingBar bileşeni olarak ayarlayabilirsiniz. Daha fazla bilgi için bkz. “[BufferingBar bileşeni](#)” sayfa 155.

Arabelleğe alma çubuğu şu ek özelliğe sahiptir:

Özellik	Açıklama
fill_mc:MovieClip	Arabelleğe alma çubuğu dolgusunun örnek adını belirtir. Varsayılan olarak bufferingBarFill_mc ayarındadır.

Arama çubuğu ve ses düzeyi çubuğu

Arama çubuğu ayrıca şu iki film klibine sahiptir: seekBar_mc ve seekBarProgress_mc. Bu göreceli konumlandırma korunduğundan, her klipin Mizanpaj katmanında diğer klibe göre konumu önemlidir. Her iki klip ölçeklense de, seekBar_mc öğesi, yuvalanmış film klipleriyle düzgün çalışmayan 9 dilimli ölçeklemeyi kullandığından, seekBar_mc içinde seekBarProgress_mc öğesi yuvalanamaz.

seekBar_mc klibine 9 dilimli ölçekleme uygulanmıştır, bu nedenle ölçeklendiğinde kenarlıklar deforme edilmez. seekBarProgress_mc klipi de ölçeklenir ancak deforme edilmez. Bir dolgu olduğu için deforme edildiğinde düzgün görüldüğünden, 9 dilimli ölçekleme kullanmaz.

progress_mc klipinin FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerinde çalıştığı gibi, seekBarProgress_mc klipi de fill_mc olmadan çalışır. Başka bir deyişle, bu maskelenmez ve yatay olarak ölçeklenir. seekBarProgress_mc öğesinin %100 değerinde gerçek boyutları, seekBarProgress_mc klipinin içindeki sol ve sağ kenar boşluklarıyla tanımlanır. Bu boyutlar varsayılan olarak eşit olup seekBar_mc ve seekBarProgress_mc öğelerinin x (yatay) konumlarını esas alır. Şu örnekte gösterildiği gibi, arama çubuğu film klibinde ActionScript ile boyutları özelleştirebilirsiniz:

```
this.seekBar_mc.progressLeftMargin = 2;  
this.seekBar_mc.progressRightMargin = 2;  
this.seekBar_mc.progressY = 11;  
this.seekBar_mc.fullnessLeftMargin = 2;  
this.seekBar_mc.fullnessRightMargin = 2;  
this.seekBar_mc.fullnessY = 11;
```

Bu kodu SeekBar film klipi Zaman Çizelgesi'ne koyabilir veya ana Zaman Çizelgesi'nde başka bir ActionScript koduyla birlikte koyabilirsiniz. Mizanpajı değiştirmek yerine bu kodla özelleştirirseniz, dolgunun Sahne Alanı'nda olması gerekmez. Yalnızca kütüphanede bulunması gerekiyorsa, doğru sınıf adıyla Kare 1'de ActionScript için dışa aktar öğesini ayarlayın.

FLV Oynatma Özel Kullanıcı Arabirimi SeekBar bileşeninde olduğu gibi, arama çubuğu için tamlık film klipi oluşturulması mümkündür. Arama çubuğunuzun ölçeklenmesi gerekmiyorsa veya arama çubuğunuz ölçeklenir ancak 9 dilimli ölçeklemeyi kullanmazsa, FLV Oynatma Özel Kullanıcı Arabirimi bileşenleri için kullanılan herhangi bir yöntemi kullanarak progress_mc veya fullness_mc öğenizi ayarlayabilirsiniz. Daha fazla bilgi için bkz. .

Flash kaplamalarındaki ses düzeyi çubuğu ölçeklenmediğinden, VolumeBar FLV Oynatma Özel Kullanıcı Arabirimi bileşeniyle aynı şekilde oluşturulur. Daha fazla bilgi için bkz. “[SeekBar ve VolumeBar bileşenleri](#)” sayfa 156. Tek istisna, tutamacın farklı şekilde uygulanmasıdır.

Seekbar ve VolumeBar tutamaçları

SeekBar ve VolumeBar tutamaçları, çubuğun yanındaki Mizanpaj katmanına yerleştirilir. Varsayılan olarak, tutamacın sol kenar boşluğu, sağ kenar boşluğu ve y ekseni değerleri, çubuk film kliline göre konumuyla belirlenir. Sol kenar boşluğu, tutamacın x (yatay) konumu ile çubuğun x (yatay) konumu arasındaki farkla belirlenir ve sağ kenar boşluğu sol kenar boşluğuna eşittir. Bu değerleri, SeekBar veya VolumeBar film klisinde ActionScript yoluyla özelleştirebilirsiniz. Şu örnek, FLV Oynatma Özel Kullanıcı Arabirimi bileşenleriyle kullanılan ActionScript koduyla aynıdır:

```
this.seekBar_mc.handleLeftMargin = 2;  
this.seekBar_mc.handleRightMargin = 2;  
this.seekBar_mc.handleY = 11;
```

Bu kodu SeekBar film klisi Zaman Çizelgesi'ne koyabilir veya ana Zaman Çizelgesi'nde başka bir ActionScript koduyla birlikte koyabilirsiniz. Mizanpajı değiştirmek yerine bu kodla özelleştirirseniz, tutamacın Sahne Alanı'nda olması gerekmez. Yalnızca kütüphanede bulunması gerekiyorsa, doğru sınıf adıyla Kare 1'de ActionScript için dışa aktar öğesini ayarlayın.

Bu özelliklerin dışında, tutamaçlar basit film klipleridir ve FLV Oynatma Özel Kullanıcı Arabirimi bileşenleriyle aynı şekilde ayarlanır. Her ikisi de `alpha` özelliği 0 değerine ayarlanmış dikdörtgen arka plana sahiptir. Bunlar yalnızca vuruş bölgesini artırmak için sağlanmış olup zorunlu değildir.

Arka plan ve ön plan klipleri

`chrome_mc` ve `forwardBackBorder_mc` film klipleri, arka plan klipleri olarak uygulanır.

Sahne Alanı'ndaki `ForwardBackBorder`, `ForwardBorder` ve `BackBorder` film klipleri ve yer tutucu İleri ve Geri düğmeleri arasında kılavuz katmanında *olmayan* yalnızca `ForwardBackBorder` öğesidir. Bu yalnızca İleri ve Geri düğmelerini gerçekten kullanan kaplamalarda bulunur.

Bu kliplerin tek gereksinimi, kütüphanede Kare 1'de ActionScript için dışa aktarılması gerekesidir.

Kaplama davranışını değiştirme

`bufferingBarHidesAndDisablesOthers` özelliği ve `skinAutoHide` özelliği, FLVPlayback kaplamasının davranışını özelleştirmenize olanak sağlar.

`bufferingBarHidesAndDisablesOthers` özelliği `true` değerine ayarlandığında, FLVPlayback bileşeninin SeekBar öğesini ve tutamacını gizlemesi ve bileşen arabelleğe alma durumuna girdiğinde Oynat/Çal ve Duraklat düğmelerini devre dışı bırakması sağlanır. Bu, `bufferTime` özelliği için (örneğin, 10) yüksek ayarlı yavaş bir bağlantı üzerinden bir FLV dosyası FMS'den akışa alındığında kullanışlıdır. Bu durumda sabırsız bir kullanıcı Oynat/Çal ve Duraklat düğmelerini tıklatarak aramayı başlatabilir ve bu da dosyanın oynatılmasını daha da uzun süre geciktirebilir. `bufferingBarHidesAndDisablesOthers` öğesini `true` değerine ayarlayıp bileşen arabelleğe alma durumundayken SeekBar öğesini ve Duraklat ve Oynat/Çal düğmelerini devre dışı bırakarak bu etkinliği önleyebilirsiniz.

`skinAutoHide` özelliği, FLV Oynatma Özel Kullanıcı Arabirimi bileşenlerinden oluşturulan denetimleri değil, yalnızca önceden tasarlanmış kaplama SWF dosyalarını etkiler. `true` değerine ayarlanırsa, fare görüntüleme alanının üzerinde olmadığında FLVPlayback bileşeni kaplamayı gizler. Bu özelliğin varsayılan değeri `false` olur.

SMIL dosyasını kullanma

VideoPlayer sınıfı, çoklu bant genişlikleri için birden çok akışı işlemek üzere SMIL'nin bir alt kümesini destekleyen yardımcı sınıfını kullanır. Video akışının konumunu, FLV dosyasının mizanpajını (genişlik ve yükseklik), farklı bant genişliklerine karşılık gelen kaynak FLV dosyalarını tanımlamak için SMIL kullanılır. SMIL, FLV dosyasının bit hızını ve süresini belirtmek için de kullanılabilir.

SMIL dosyasının konumunu belirlemek için `source` parametresini ya da `FLVPlayback.source` özelliğini (ActionScript) kullanın. Daha fazla bilgi için, [Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu](#) bölümünde ve `FLVPlayback.source` özelliğine bakın.

Aşağıdaki örnek, RTMP kullanarak FMS'den çoklu bant genişliği FLV dosyalarını akışa alan bir SMIL dosyasını gösterir:

```
<smil>
<head>
<meta base="rtmp://myserver/myapp/" />
<layout>
<root-layout width="240" height="180" />
</layout>
</head>
<body>
    <switch>
        <ref src="myvideo_cable.flv" dur="3:00.1"/>
        <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
        <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
    </switch>
</body>
</smil>
```

`<head>` etiketi `<meta>` ve `<layout>` etiketlerini içerebilir. `<meta>` etiketi yalnızca akış videosunun URL'sini (FMS'den RTMP) belirtmek için kullanılan `base` niteliğini destekler.

`<layout>` etiketi yalnızca `height` ve `width` niteliklerini ayarlamak için kullanılan ve böylece FLV dosyasının oluşturulduğu pencerenin boyutunu belirleyen `root-layout` öğesini destekler. Bu nitelikler yüzdeleri değil yalnızca piksel değerlerini kabul eder.

SMIL dosyasının gövdesinde, FLV kaynak dosyasına tek bir bağ dahil edebilirsiniz veya FMS'den çoklu bant genişlikleri için birden çok dosya akışa alıyorsanız (önceki örnekte olduğu gibi), kaynak dosyaları listelemek için `<switch>` etiketini kullanabilirsiniz.

`<switch>` etiketindeki `video` ve `ref` etiketleri eşanlıdır—her ikisi de FLV dosyalarını belirtmek için `src` niteliğini kullanabilir. Ayrıca her biri bölgeyi, gerekli minimum bant genişliğini ve FLV dosyasının süresini belirtmek için `region`, `system-bitrate` ve `dur` niteliklerini kullanabilir.

`<body>` etiketi içinde `<video>`, `<src>` veya `<switch>` etiketlerinin yalnızca birinin kullanılmasına izin verilir.

Aşağıdaki örnek, bant genişliği saptaması kullanmayan tek bir FLV dosyası için aşamalı indirmeyi gösterir:


```
<smil>
  <head>
    <layout>
      <root-layout width="240" height="180" />
    </layout>
  </head>
  <body>
    <video src="myvideo.flv" />
  </body>
</smil>
```

<smil>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<smil>
...
child tags
...
</smil>
```

Nitelikler

Yok.

Alt öge etiketleri

<head>, <body>

Üst öge etiketi

Yok.

Açıklama

SMIL dosyasını tanımlayan üst düzey etiket.

Örnek

Aşağıdaki örnek, üç FLV dosyası belirten bir SMIL dosyasını gösterir:

```
<smil>
<head>
<meta base="rtmp://myserver/myapp/" />
<layout>
<root-layout width="240" height="180" />
</layout>
</head>
<body>
<switch>
      <ref src="myvideo_cable.flv" dur="3:00.1"/>
      <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
      <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
</switch>
</body>
</smil>
```

<head>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<head>
...
child tags
...
</head>
```

Nitelikler

Yok.

Alt öge etiketleri

<meta>, <layout>

Üst öge etiketi

<smil>

Açıklama

<meta> ve <layout> etiketlerini destekleyerek kaynak FLV dosyalarının konumunu ve varsayılan mizanpajını (yükseklik ve genişlik) belirtir.

Örnek

Şu örnek başlangıç mizanpajını 240 x 180 piksel olarak ayarlar:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

<meta>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<meta/>
```

Nitelikler

base

Alt öge etiketleri

```
<layout>
```

Üst öge etiketi

Yok.

Açıklama

Kaynak FLV dosyalarının konumunu (RTMP URL) belirten *base* niteliğini içerir.

Örnek

Aşağıdaki örnek, *myserver* üzerinde taban konum için bir meta etiketi gösterir:

```
<meta base="rtmp://myserver/myapp/" />
```

<layout>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<layout>  
...  
child tags  
...  
</layout>
```

Nitelikler

Yok.

Alt öge etiketleri

```
<root-layout>
```

Üst öge etiketi

```
<meta>
```

Açıklama

FLV dosyasının genişliğini ve yüksekliğini belirtir.

Örnek

Aşağıdaki örnek, 240 piksel x 180 piksel mizanpajı belirtir:

```
<layout>  
  <root-layout width="240" height="180" />  
</layout>
```

<root-layout>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<root-layout...attributes.../>
```

Nitelikler

Genişlik, yükseklik

Alt öge etiketleri

Yok.

Üst öge etiketi

```
<layout>
```

Açıklama

FLV dosyasının genişliğini ve yüksekliğini belirtir.

Örnek

Aşağıdaki örnek, 240 piksel x 180 piksel mizanpajı belirtir:

```
<root-layout width="240" height="180" />
```

<body>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<body>  
  ...  
  child tags  
  ...  
</body>
```

Nitelikler

Yok.

Alt öge etiketleri

<video>, <ref>, <switch>

Üst öge etiketi

<smil>

Açıklama

Kaynak FLV dosyasının adını, minimum bant genişliğini ve FLV dosyasının süresini belirten <video>, <ref> ve <switch> etiketlerini içerir. system-bitrate niteliği yalnızca <switch> etiketi kullanıldığında desteklenir. <body> etiketi içinde <switch>, <video> veya <ref> etiketlerinin yalnızca birinin kullanılmasına izin verilir.

Örnek

Aşağıdaki örnek, ikisi video etiketini ve biri ref etiketini kullanan üç FLV dosyasını belirtir:

```
<body>
  <switch>
    <ref src="myvideo_cable.flv" dur="3:00.1"/>
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
  </switch>
</body>
```

<video>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<video...attributes.../>
```

Nitelikler

src, system-bitrate, dur

Alt öge etiketleri

Yok.

Üst öge etiketi

<body>

Açıklama

<ref> etiketiyle eşanımlıdır. Kaynak FLV dosyasının adını ve süresini belirten src ve dur niteliklerini destekler. dur niteliği, tam (00:03:00:01) ve kısmi (03:00:01) saat biçimlerini destekler.

Örnek

Aşağıdaki örnek, bir videonun kaynağını ve süresini ayarlar:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

<ref>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<ref...attributes.../>
```

Nitelikler

src, system-bitrate, dur

Alt öge etiketleri

Yok.

Üst öge etiketi

```
<body>
```

Açıklama

<video> etiketiyle eşanlamlıdır. Kaynak FLV dosyasının adını ve süresini belirten src ve dur niteliklerini destekler. dur niteliği, tam (00:03:00:01) ve kısmi (03:00:01) saat biçimlerini destekler.

Örnek

Aşağıdaki örnek, bir videonun kaynağını ve süresini ayarlar:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

<switch>

Kullanılabilirlik

Flash Professional 8.

Kullanım

```
<switch>  
...  
child tags  
...  
</switch/>
```

Nitelikler

Yok.

Alt öge etiketleri

<video>, <ref>

Üst öge etiketi

```
<body>
```

Açıklama

Çoklu bant genişliği video akışı için FLV dosyalarını listelemek üzere <video> veya <ref> alt öge etiketleriyle kullanılır. <switch> etiketi, src ve dur niteliklerinin yanı sıra minimum bant genişliğini belirten system-bitrate niteliğini destekler.

Örnek

Aşağıdaki örnek, ikisi video etiketini ve biri ref etiketini kullanan üç FLV dosyasını belirtir:

```
<switch>
  <ref src="myvideo_cable.flv" dur="3:00.1"/>
  <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
  <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1" />
</switch>
```

Bölüm 7: FLVPlayback Captioning Bileşenini Kullanma

FLVPlayback bileşeni, indirilen Adobe Flash Video (FLV veya F4V) dosyalarını ve akış FLV veya F4V dosyalarını oynatmak için Adobe Flash CS5 Professional uygulamanıza bir video oynatıcısı dahil etmenize olanak sağlar. FLVPlayback hakkında daha fazla bilgi için, bkz. “[FLVPlayback Bileşenini kullanma](#)” sayfa 135.

FLVPlaybackCaptioning bileşeni, videonuz için kapalı resim yazıcı desteği dahil etmenize olanak sağlar. Resim yazısı bileşeni W3C standardı XML Timed Text formatını destekler ve şu özellikleri içerir:

Gömülü Olay İşaret Noktalarıyla Resim Yazısı Timed Text XML dosyası kullanmak yerine resim yazısı sağlamak için bir FLV dosyasındaki gömülü olay işaret noktalarını XML ile ilişkilendirin.

Birden Çok FLVPlayback Resim Yazısı Birden çok FLVPlayback örneği için birden çok FLVPlayback resim yazısı örneği oluşturun.

Aç/Kapa Düğmesi Denetimi Resim yazısı aç/kapa düğmesi üzerinden resim yazısıyla kullanıcı etkileşimi sağlar.

FLVPlaybackCaptioning bileşenini kullanma

Bir veya birkaç FLVPlayback bileşeniyle FLVPlaybackCaptioning bileşenini kullanırsınız. En basit şekilde, sahne alanında bir FLVPlayback bileşenini sürüklersiniz, aynı sahne alanında bir FLVPlaybackCaptioning bileşenini de sürüklersiniz, resim yazınızın URL'sini tanımlarsınız ve görüntülenecek resim yazılarını belirlersiniz. Ayrıca FLVPlayback resim yazınızı özelleştirmek için çeşitli parametreler de ayarlayabilirsiniz.

FLVPlayback bileşenine resim yazısı ekleme

Herhangi bir FLVPlayback bileşenine FLVPlaybackCaptioning bileşenini ekleyebilirsiniz. Uygulamanıza FLVPlayback bileşenleri ekleme hakkında bilgi almak için, bkz. “[FLVPlayback bileşeniyle uygulama oluşturma](#)” sayfa 136.

Bileşenler panelinden FLVPlaybackCaptioning bileşenini ekleme:

- 1 Bileşenler panelinde Video klasörünü açın.
- 2 FLVPlaybackCaptioning bileşenini sürükleyin (veya çift tıklayın) ve resim yazısını eklemek istediğiniz FLVPlayback bileşeniyle aynı sahne alanına ekleyin.


Not: Adobe, FLVPlaybackCaptioning bileşenini öğrenmenize yardımcı olmak için iki dosya sağlar: *caption_video.flv* (bir FLVPlayback örneği) ve *caption_video.xml* (bir resim yazısı örneği). Bu dosyalara şu adresten erişebilirsiniz: www.helpexamples.com/flash/video/caption_video.flv and www.helpexamples.com/flash/video/caption_video.xml.

- 3 (İsteğe bağlı) CaptionButton bileşenini FLVPlayback ve FLVPlaybackCaptioning bileşenleriyle aynı sahne alanına sürükleyin. CaptionButton bileşeni, kullanıcının resim yazısını etkinleştirmesine ve devre dışı bırakmasına olanak sağlar.

Not: CaptionButton bileşenini etkinleştirmek için, bu bileşeni FLVPlayback ve FLVPlaybackCaptioning bileşenleriyle aynı sahne alanına sürüklemeniz gerekir.

4 Sahne Alanı'nda FLVPlaybackCaptioning bileşeni seçili durumdayken, Özellik denetçisinin Parametreler sekmesinde şu zorunlu bilgileri belirtin:

- showCaptions ögesini true değerine ayarlayın.
- İndirilecek Timed Text XML dosyasının source ögesini belirtin.

 Resim yazılarınızı test etmek için Flash uygulamasında çalışırken, showCaptions özelliğini true değerine ayarlamanız gerekir. Ancak kullanıcıların resim yazısını etkinleştirmesine ve devre dışı bırakmasına olanak sağlamak için CaptionButton bileşenini dahil ederseniz, showCaptions özelliğini false değerine ayarlamanız gerekir.

Diğer parametreler, FLVPlaybackCaptioning bileşenini özelleştirmenize yardımcı olmak için sunulmuştur. Daha fazla bilgi için bkz. "[FLVPlaybackCaptioning bileşenini özelleştirme](#)" sayfa 182 ve *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*.

5 Videoyu başlatmak için, Kontrol Et > Filmi Test Et seçeneklerini belirleyin.

ActionScript kullanarak dinamik olarak bir örnek oluşturma:

- 1 FLVPlayback bileşenini Bileşenler panelinden Kütüphane paneline (Pencere > Kütüphane) sürükleyin.
- 2 FLVPlaybackCaptioning bileşenini Bileşenler panelinden Kütüphane paneline sürükleyin.
- 3 Zaman Çizelgesi'nin Kare 1 ögesindeki Eylemler paneline şu kodu ekleyin:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "install_drive:/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinUnderPlaySeekCaption.swf";
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/caption_video.flv";
var my_FLVPlybkcap = new FLVPlaybackCaptioning();
addChild(my_FLVPlybkcap);
my_FLVPlybkcap.source = "http://www.helpexamples.com/flash/video/caption_video.xml";
my_FLVPlybkcap.showCaptions = true;
```

- 4 install_drive ögesini, Flash uygulamasını yüklediğiniz sürücüyle değiştirin ve yüklemeniz için Kaplamalar klasörünün konumunu yansıtmak üzere yolu değiştirin.

Not: ActionScript ile bir FLVPlayback örneği oluşturursanız, ActionScript ile skin özelliğini ayarlayarak bu örneğe bir kaplama da atamanız gerekir. ActionScript ile kaplamayı uyguladığınızda, kaplama SWF dosyasıyla otomatik olarak yayınlanmaz. Kaplama SWF dosyasını ve uygulama SWF dosyasını sunucunuza kopyalayın, aksi takdirde SWF dosyası kullanıcı tarafından çalıştırıldığında kullanılabilir durumda olmaz.

FLVPlaybackCaptioning bileşeni parametrelerini ayarlama

FLVPlaybackCaptioning bileşeninin her bir örneği için, Özellik denetçisinde veya Bileşen denetçisinde şu parametreleri ayarlayarak bileşeni daha fazla özelleştirebilirsiniz. Aşağıdaki listede özellikler tanımlanmakta ve özelliklerin kısa bir açıklaması sağlanmaktadır:

autoLayout FLVPlaybackCaptioning bileşeninin resim yazısı alanının boyutunu denetleyip denetlemediğini belirler. Varsayılan değer true şeklindedir.

captionTargetName Resim yazılarını içeren TextField veya MovieClip örnek adını tanımlar. Varsayılan değer auto şeklindedir.

flvPlaybackName Resim yazısı eklemek istediğiniz FLVPlayback örnek adını tanımlar. Varsayılan değer `auto` şeklindedir.

simpleFormatting `true` değerine ayarlandığında Timed Text XML dosyasının formatlama talimatlarını sınırlar. Varsayılan değer `false` şeklindedir.

showCaptions Resim yazılarının görüntülenip görüntülenmeyeceğini belirler. Varsayılan değer `true` şeklindedir.

source Timed Text XML dosyasının konumunu tanımlar.

Tüm FLVPlaybackCaptioning parametreleri hakkında daha fazla bilgi almak için bkz. *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*.

Kaynak parametreyi belirtme

Filminizin resim yazılarını içeren Timed Text XML dosyasının adını ve konumunu belirtmek için `source` parametresini kullanın. URL yolunu doğrudan Bileşen denetçisindeki `source` hücresine girin.

Resim yazılarını görüntüleme

Resim yazısını görüntülemek için, `showCaptions` parametresini `true` değerine ayarlayın.

Tüm FLVPlaybackCaptioning bileşen parametreleri hakkında daha fazla bilgi almak için bkz. *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*.

Önceki örneklerde, resim yazılarını görüntülemek için FLVPlaybackCaptioning bileşeninin nasıl oluşturulduğunu ve etkinleştirildiğini öğrendiniz. Resim yazılarınız için kullanabileceğiniz iki kaynak vardır: (1) resim yazılarınızı içeren bir Timed Text XML dosyası veya (2) gömülü olay işaret noktalarıyla ilişkilendirdiğiniz resim yazısı metnini içeren bir XML dosyası.

Timed Text resim yazılarını kullanma

FLVPlaybackCaptioning bileşeni, bir Timed Text (TT) XML dosyası indirerek ilişkilendirilmiş FLVPlayback bileşeni için resim yazısı eklemenize olanak sağlar. Timed Text formatı hakkında daha fazla bilgi için, www.w3.org adresinde AudioVideo Timed Text bilgilerini inceleyin.

Bu bölümde, desteklenen Timed Text etiketlerine, zorunlu resim yazısı dosyası etiketlerine genel bakış ve bir Timed Text XML dosyası örneği sağlanmaktadır. Tüm desteklenen Timed Text etiketleri hakkında ayrıntılı bilgi için, bkz. “[Timed Text etiketleri](#)” sayfa 176.

FLVPlaybackCaptioning bileşeni, şu Timed Text etiketlerini destekler:

Kategori	Görev
Paragraf formatlama desteği	Paragrafın sağa, sola veya ortaya hizalanmasını sağlar
Metin formatlama desteği	<ul style="list-style-type: none">• Mutlak piksel boyutlarıyla veya delta stiliyle (örneğin, +2, -4) metnin boyutunun ayarlanmasını sağlar.• Metin renginin ve fontunun ayarlanmasını sağlar• Metnin kalın ve italik yapılmasını sağlar• Metin yaslamasının ayarlanmasını sağlar
Diğer formatlama destekleri	<ul style="list-style-type: none">• Resim yazıları için TextField ögesinin arka plan renginin ayarlanmasını sağlar• Resim yazıları için TextField ögesinin arka plan renginin saydam (alpha 0) olarak ayarlanmasını sağlar• Resim yazılarının TextField ögesi için kelime kaydırma işlevinin ayarlanmasını sağlar (etkin veya devre dışı)

FLVPlaybackCaptioning bileşeni, FLV dosyasının zaman koduyla eşleşir. Her bir resim yazısının başlangıcını belirten bir `begin` niteliği olması gerekir. Resim yazısı `dur` veya `end` niteliğine sahip değilse, bir sonraki resim yazısı görüntülendiğinde veya FLV dosyası sona erdiğinde söz konusu resim yazısı kaybolur.

Aşağıda bir Timed Text XML dosyası örneği yer almaktadır. Bu dosya (`caption_video.xml`), `caption_video.flv` dosyası için resim yazısı sağlar. Bu dosyalara şu adreslerden erişebilirsiniz:

www.helpexamples.com/flash/video/caption_video.flv ve www.helpexamples.com/flash/video/caption_video.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
  <tt xml:lang="en"
xmlns="http://www.w3.org/2006/04/ttaf1"xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling">
<head>
  <styling>
<style id="1" tts:textAlign="right"/>
<style id="2" tts:color="transparent"/>
<style id="3" style="2" tts:backgroundColor="white"/>
<style id="4" style="2 3" tts:fontSize="20"/>
  </styling>
</head>
<body>
  <div xml:lang="en">
<p begin="00:00:00.00" dur="00:00:03.07">I had just joined <span
tts:fontFamily="monospaceSansSerif,proportionalSerif,TheOther"tts:fontSize="+2">Macromedia</span> in 1996,</p>
<p begin="00:00:03.07" dur="00:00:03.35">and we were trying to figure out what to do about the internet.</p>
<p begin="00:00:06.42" dur="00:00:03.15">And the company was in dire straights at the time.</p>
<p begin="00:00:09.57" dur="00:00:01.45">We were a CD-ROM authoring company,</p>
<p begin="00:00:11.42" dur="00:00:02.00">and the CD-ROM business was going away.</p>
<p begin="00:00:13.57" dur="00:00:02.50">One of the technologies I remember seeing was Flash.</p>
<p begin="00:00:16.47" dur="00:00:02.00">At the time, it was called <span
tts:fontWeight="bold" tts:color="#ccc333">FutureSplash</span>.</p>
<p begin="00:00:18.50" dur="00:00:01.20">So this is where Flash got its start.</p>
<p begin="00:00:20.10" dur="00:00:03.00">This is smart sketch running on the <span
tts:fontStyle="italic">EU-pin computer</span>,</p>
<p begin="00:00:23.52" dur="00:00:02.00">which was the first product that FutureWave did.</p>
<p begin="00:00:25.52" dur="00:00:02.00">So our vision for this product was to</p>
<p begin="00:00:27.52" dur="00:00:01.10">make drawing on the computer</p>
<p begin="00:00:29.02" dur="00:00:01.30" style="1">as <span tts:color="#ccc333">easy</span>
as drawing on paper.</p>
</div>
</body>
</tt>
```

Timed Text etiketleri

FLVPlaybackCaptioning bileşeni, resim yazısı XML dosyaları için Timed Text etiketlerini destekler. Ses Video Timed Text etiketleri hakkında daha fazla bilgi almak için, www.w3.org adresindeki bilgileri inceleyin. Aşağıdaki tabloda, desteklenen ve desteklenmeyen etiketler listelenmektedir.

İşlev	Etiket/Değer	Kullanım/Açıklama	Örnek
Yoksayılan etiketler	meta veri	Yoksayılır / belgenin herhangi bir düzeyinde izin verilir	
	set	Yoksayılır / belgenin herhangi bir düzeyinde izin verilir	
	xml:lang	Yoksayılır	
	xml:space	Yoksayılır / Davranış şununla geçersiz kılınır: xml:space="default"	

İşlev	Etiket/Değer	Kullanım/Açıklama	Örnek
	layout	Yoksayılr / mizanpaj etiketi bölümündeki bölge etiketleri dahil	
	br etiketi	Tüm nitelikler ve içerikler yoksayılr.	
Resim Yazıları için Ortam Zamanlama	begin nitelikleri	Yalnızca p etiketlerinde izin verilir. Resim yazılarının ortam zamanı konuşlandırması için zorunludur.	<p begin="3s">
	dur nitelikleri	Yalnızca p etiketlerinde izin verilir. Önerilir. Dahil edilmezse, resim yazısı FLV dosyasıyla veya başka bir resim yazısı başladığında sonlandırılır.	
	end nitelikleri	Yalnızca p etiketlerinde izin verilir. Önerilir. Dahil edilmezse, resim yazısı FLV dosyasıyla veya başka bir resim yazısı başladığında sonlandırılır.	
Resim Yazıları için Saat Zamanlama	00:03:00.1	Tam saat formatı	
	03:00.1	Kısmi saat formatı	
	10	Birimler olmadan öteleme süreleri. Öteleme, saniyeleri temsil eder.	
	00:03:00:05 00:03:00:05.1 30f 30t	Desteklenmez. Kareleri veya onay işaretlerini içeren saat formatları desteklenmez.	
Body etiketi	body	Zorunludur / Yalnızca tek body etiketini destekler.	<body><div>...</div></body>
İçerik etiketi	div etiketi	Sıfır değerine veya daha fazlasına izin verilir. Birinci etiket kullanılır.	
	p etiketi	Sıfır değerine veya daha fazlasına izin verilir.	
	span etiketi	Metin içerik birimleri sırası için mantıksal bir konteynerdir. Yuvalanmış yayılmaları desteklemez. Nitelik stili etiketlerini destekler.	
	br etiketi	Açık satır kesmesini belirtir.	
Stil Uygulama Etiketleri (Tüm stil etiketleri p etiketi içinde kullanılır)	style	Bir veya daha fazla stil ögesine başvurur. Etiket veya nitelik olarak kullanılabilir. Etiket olarak, kimlik niteliği zorunludur (stil belgede yeniden kullanılabilir). Stil etiketi içinde bir veya daha fazla stil etiketini destekler.	
	tts:background Color	Bir bölgenin arka plan rengini tanımlayan stil özelliğini belirtir. Arka planı saydam hale getirmek için, sıfır değerine ayarlanmadığı sürece alpha değeri yoksayılr. Renk formatı #RRGGBBAA şeklindedir.	

İşlev	Etiket/Değer	Kullanım/Açıklama	Örnek
	tts:color	Ön plan rengini tanımlayan bir stil özelliğini belirtir. Herhangi bir renk için alpha değeri desteklenmez. transparent değeri, siyaha çevrilir.	<pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia"("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua"("cyan")=#00FFFFFF</pre>
	tts:fontFamily	Font ailesini tanımlayan bir stil özelliğini belirtir.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Font boyutunu tanımlayan bir stil özelliğini belirtir. İki değer varsa, yalnızca birinci (dikey) değer kullanılır. Yüzde değerleri ve birimleri yoksayılr. Mutlak piksel (örneğin, 12) ve göreceli stil (örneğin, +2) boyutlarını destekler.	
	tts:fontStyle	Font stilini tanımlayan bir stil özelliğini belirtir.	<pre>"normal" "italic" "inherit"* * Varsayılan davranış; çevreleyen etiketten stili miras alır.</pre>

İşlev	Etiket/Değer	Kullanım/Açıklama	Örnek
	tts:fontWeight	Font ağırlığını tanımlayan bir stil özelliğini belirtir.	"normal" "bold" "inherit"* * Varsayılan davranış; çevreleyen etiketten stili miras alır.
	tts:textAlign	Kapsayan bir blok alanı içinde satır içi alanların nasıl hizalandığını tanımlayan bir stil özelliğini belirtir.	"left" "right" "center" "start" ("left") "end" ("right") "inherit"* *Çevreleyen etiketten stili miras alır. Herhangi bir textAlign etiketi ayarlanmazsa, varsayılan değer "left" olur.
	tts:wrapOption	Etkilenen öğenin bağlamında otomatik satır sarma (kesme) özelliğinin uygulanıp uygulanmayacağını tanımlayan bir stil özelliğini belirtir. Bu ayar, resim yazısı ögesindeki tüm paragrafları etkiler.	"wrap" "noWrap" "inherit"* *Çevreleyen etiketten stili miras alır. Herhangi bir wrapOption etiketi ayarlanmazsa, varsayılan değer "wrap" olur.
Desteklenmeyen Nitelikler	tts:direction tts:display tts:displayAlign tts:dynamicFlow tts:extent tts:lineHeight tts:opacity tts:origin tts:overflow tts:padding tts:showBackground tts:textOutline tts:unicodeBidi tts:visibility tts:writingMode tts:zIndex		

Resim yazısı ile işaret noktalarını kullanma

İşaret noktaları, bir video ile etkileşim kurmanıza olanak sağlar; örneğin, bir FLV dosyasının oynatımını etkileyebilir veya videoda belirli zamanlarda metin görüntüleyebilirsiniz. FLV dosyasıyla kullanılacak bir Timed Text XML dosyanız olmazsa, olay işaret noktalarını bir FLV dosyasına gömebilir ve sonra bu işaret noktalarını metinle ilişkilendirebilirsiniz. Bu bölümde, FLVPlaybackCaptioning bileşeninin işaret noktaları standartları hakkında bilginin yanı sıra resim yazısı eklemek üzere bu işaret noktalarının metinle nasıl ilişkilendirildiğine yönelik kısa bir genel bakış sağlanmaktadır. Olay işaret noktalarının Video İçerme Sihirbazıyla veya Flash Video kodlayıcısıyla nasıl gömüldüğü hakkında daha fazla bilgi almak için bkz. Bölüm 16, "Video ile Çalışma", *Flash Uygulamasını Kullanma*

FLVPlaybackCaptioning işaret noktası standartlarını anlama

FLV dosyasının meta verisinde işaret noktası, şu özellikleri içeren bir nesne olarak temsil edilir: `name`, `time`, `type` ve `parameters`. FLVPlaybackCaptioning ActionScript işaret noktaları şu niteliklere sahiptir:

name `name` özelliği, işaret noktasının atanmış adını içeren bir dizedir. `name` özelliğinin *fl.video.caption.2.0*. önekiyle başlayıp bir dizeyle öneki izlemesi gerekir. Dize, her adın benzersiz olmasını sağlamak için her seferinde artan pozitif tam sayılar serisidir. Önek ayrıca FLVPlayback sürüm numarasıyla da eşleşen sürüm numarasını içerir. Adobe Flash CS4 ve sonrası için sürüm numarasını 2.0 değerine ayarlamanız gerekir.

time `time` özelliği, resim yazısının görüntülenmesi gereken zamandır.

type `type` özelliği, değeri "event" olan bir dizedir.

parameters `parameters` özelliği, şu ad ve değer çiftlerini destekleyen bir dizidir:

- **text:String** Resim yazısı için HTML formatında metin. Bu metin doğrudan `TextField.htmlText` özelliğine iletilir. FLVPlaybackCaptioning bileşeni, çoklu dil yolunun kullanılmasını destekleyen isteğe bağlı bir `text:n` özelliğini destekler. Daha fazla bilgi için bkz. "Gömülü işaret noktalarıyla çoklu dil yolunu destekleme" sayfa 182.
- **endTime:Number** Resim yazısının kaybolması gereken zaman. Bu özelliği belirtmezseniz, FLVPlaybackCaptioning bileşeni bunun bir sayı olmadığını (NaN) varsayar ve böylece FLV dosyası tamamlanıncaya kadar (FLVPlayback örneği `VideoEvent.COMPLETE` olayını gönderinceye kadar) resim yazısı görüntülenir. `endTime:Number` özelliğini saniye cinsinden belirtin.
- **backgroundColor:uint** Bu parametre, `TextField.backgroundColor` ögesini ayarlar. Bu özellik isteğe bağlıdır.
- **backgroundColorAlpha:Boolean** `backgroundColor` ögesi %0 alfa değerine sahipse, bu durumda parametre `TextField.background = !backgroundColor` ayarını yapar. Bu özellik isteğe bağlıdır.
- **wrapOption:Boolean** Bu parametre `TextField.wordWrap` ayarını yapar. Bu özellik isteğe bağlıdır.

Gömülü olay işaret noktaları için resim yazısını anlama

FLV dosyanızın resim yazılarını içeren bir Timed Text XML dosyanız yoksa, resim yazılarını içeren bir XML dosyasıyla ilişkilendirerek resim yazısı oluşturabilirsiniz. XML örneği, videonuzda gömülü olay işaret noktaları oluşturmak için aşağıdaki adımları uyguladığınızı varsayar:

- Olay işaret noktalarını ekleyin (FLVPlaybackCaptioning standartlarını izleyerek) ve videoyu kodlayın.
- Flash uygulamasında bir FLVPlayback bileşenini ve bir FLVPlaybackCaptioning bileşenini Sahne Alanı'na sürükleyin.
- FLVPlayback ve FLVPlaybackCaptioning bileşenlerinin source özelliklerini (FLV dosyanızın konumu ve XML dosyanızın konumu) ayarlayın.
- Yayınlayın.

Aşağıdaki örnek, XML ögesini kodlayıcıya içe aktarır:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<FLVCoreCuePoints>

  <CuePoint>
    <Time>9136</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index1</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the first cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

  <CuePoint>
    <Time>19327</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index2</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the second cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

  <CuePoint>
    <Time>24247</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index3</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the third cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

  <CuePoint>
    <Time>36546</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index4</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the fourth cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

</FLVCoreCuePoints>
```

FLVPlaybackCaptioning bileşeni de gömülü işaret noktasında çoklu dil yolunu destekler. Daha fazla bilgi için bkz. [“Gömülü işaret noktalarıyla çoklu dil yolunu destekleme”](#) sayfa 182.

Gömülü işaret noktalarıyla çoklu dil yolunu destekleme

Timed Text XML dosyası FLVPlaybackCaptioning işaret noktası standartlarını izlediği sürece, FLVPlaybackCaptioning `track` özelliği, gömülü işaret noktalarıyla çoklu dil yolunu destekler. (Daha fazla bilgi için, bkz. “[FLVPlaybackCaptioning işaret noktası standartlarını anlama](#)” sayfa 180.) Ancak, FLVPlaybackCaptioning bileşeni, ayrı XML dosyalarında çoklu dil yollarını desteklemez. `track` özelliğini kullanmak için, bu özelliği 0 dışında bir değere ayarlayın. Örneğin, `track` özelliğini 1 değerine ayarlarsanız (`track == 1`), FLVPlaybackCaptioning bileşeni, işaret noktası parametrelerini arar. Eşleşmez bulunmazsa, işaret noktası parametrelerindeki `track` özelliği kullanılır. Daha fazla bilgi için *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*'ndaki `track` özelliğine bakın.

Resim yazısı ile birden çok FLV dosyası oynatma

Çoklu videolar oynatmak ve oynatılırken videolar arasında geçiş yapmak için FLVPlayback bileşeninin tek bir örneğinde birden çok video oynatıcı açabilirsiniz. Ayrıca FLVPlayback bileşeni içinde her video oynatıcısıyla resim yazısı ilişkilendirebilirsiniz. Birden çok video oynatıcısı açma hakkında daha fazla bilgi için bkz. “[Çoklu video oynatıcılarını kullanma](#)” sayfa 150. Birden çok video oynatıcısında resim yazısı kullanmak için, her bir `VideoPlayer` ögesi için bir FLVPlaybackCaptioning bileşeni örneği oluşturun ve FLVPlaybackCaptioning `videoPlayerIndex` ögesini karşılık gelen dizine ayarlayın. Yalnızca bir `VideoPlayer` bulunduğunda, `VideoPlayer` dizini varsayılan olarak 0 olur.

Aşağıda, benzersiz videolara benzersiz resim yazıları atayan bir kod örneği yer almaktadır. Bu örneğin uygulanması için, örnekteki kurgusal URL'lerin çalışan URL'ler ile değiştirilmesi gerekir.

```
captioner0.videoPlayerIndex = 0;  
captioner0.source = "http://www. [yourDomain] .com/mytimedtext0.xml";  
flvPlayback.play ("http://www. [yourDomain] .com/myvideo0.flv");  
captioner1.videoPlayerIndex = 1;  
captioner1.source = "http://www. [yourDomain] .com/mytimedtext1.xml";  
flvPlayback.activeVideoIndex = 1;  
flvPlayback.play ("http://www. [yourDomain] .com/myvideo1.flv");
```

FLVPlaybackCaptioning bileşenini özelleştirme

FLVPlaybackCaptioning bileşenini hızlı şekilde kullanmaya başlamak için, doğrudan FLVPlayback bileşeni üzerinden resim yazısını yerleştiren FLVPlaybackCaptioning varsayılanlarını kullanmayı seçebilirsiniz. Resim yazısını videodan uzaklaştırmak için FLVPlaybackCaptioning bileşenini özelleştirmek isteyebilirsiniz.

Şu kod, resim yazısı aç/kapa düğmesi içeren bir FLVPlayback nesnesinin dinamik olarak nasıl oluşturulacağını gösterir:

- 1 FLVPlayback bileşenini sahne alanında 0,0 konumuna yerleştirin ve **player** örnek adını sağlayın.
- 2 FLVPlayback bileşenini sahne alanında 0,0 konumuna yerleştirin ve **captioning** örnek adını sağlayın.
- 3 `CaptionButton` bileşenini sahne alanına yerleştirin.
- 4 Aşağıdaki kod örneğinde, `testVideoPath:String` değişkenini bir FLV dosyasına ayarlayın (mutlak veya göreceli bir yol kullanarak).

Not: Kod örneği, `testVideoPath` değişkenini Flash video örneğine (`caption_video.flv`) ayarlar. Bu değişkeni, resim yazısı `Button` bileşeni eklediğiniz resim yazısı video bileşeninin yoluyla değiştirin.

- 5 Aşağıdaki kod örneğinde, `testCaptioningPath:String` değişkenini uygun bir Timed Text XML dosyasına ayarlayın (mutlak veya göreceli bir yol kullanarak).

Not: Bu kod, `testCaptioningPath` değişkenini Timed Text XML dosyasına (`caption_video.xml`) ayarlar. Bu değişkeni, videonuzun resim yazılarını içeren Timed Text XML dosyasının yoluyla değiştirin.

- 6 Bu kodu FLA dosyanızla aynı dizine `FLVPlaybackCaptioningExample.as` olarak kaydedin.
7 FLA dosyasındaki `DocumentClass` ögesini `FLVPlaybackCaptioningExample` olarak ayarlayın.

```
package
{
    import flash.display.Sprite;
    import flash.text.TextField;
    import fl.video.FLVPlayback;
    import fl.video.FLVPlaybackCaptioning;

    public class FLVPlaybackCaptioningExample extends Sprite {

        private var testVideoPath:String =
"http://www.helpexamples.com/flash/video/caption_video.flv";
        private var testCaptioningPath:String =
"http://www.helpexamples.com/flash/video/caption_video.xml";

        public function FLVPlaybackCaptioningExample() {
            player.source = testVideoPath;
            player.skin = "SkinOverAllNoCaption.swf";
            player.skinBackgroundColor = 0x666666;
            player.skinBackgroundAlpha = 0.5;

            captioning.flvPlayback = player;
            captioning.source = testCaptioningPath;
            captioning.autoLayout = false;
            captioning.addEventListener("captionChange", onCaptionChange);
        }
        private function onCaptionChange(e:*) :void {
            var tf:* = e.target.captionTarget;
            var player:FLVPlayback = e.target.flvPlayback;

            // move the caption below the video
            tf.y = 210;
        }
    }
}
```

Tüm `FLVPlaybackCaptioning` parametreleri hakkında daha fazla bilgi almak için bkz. *Adobe® Flash® Professional CS5 için ActionScript® 3.0 Başvurusu*.